Integrated safety technology

User's manual

Version: **1.141 (April 2019)**
Model no.: **MASAFETY-ENG**

Translation of the original documentation

1 Integrated safety technology

1.1 General information

Information:

Please note that the "Integrated Safety Technology user's manual" itself is NOT a part of the certification process and that the information it contains is solely for informational purposes. The data sheets for the individual products only are included in the certification process.

B&R keeps user's manuals as current as possible. From a safety standpoint however, the current certified version of the data sheet must be used. Especially if contradictions arise, the specifications in the data sheets are applicable.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

1.1.1 Manual history

Version	Date	Comment
1.141	April 2019	• Revised general chapters. • Added new modules. • Updated existing data sheets. • Added "DATA_to_SafeDATA_SF". • Added "LightCurtain_SF". • Updated SafeDESIGNER libraries.
1.120	June 2017	• Chapter "Safety response time": "Data transmission time on the bus": Added reACTION. • Added "Math_Utilities_SF". • Updated "Table_SF". • Updated "Utilities_SF".
1.110	April 2017	• Revised general chapters. • Added new data sheets. • Updated existing data sheets. • Updated connection examples. • Added SafeDESIGNER libraries.
1.90	February 2015	New edition • Revised general chapters. • Added new data sheets. • Updated existing data sheets.
1.01	August 2008	• Added MTTFd values to tables for safety characteristics. • Updated the descriptions of channels in Automation Studio.
1.00	May 2008	First edition

Table 1: Manual history

1.1.2 Release information

A manual version always describes the respective range of functions for a given product set release. The following table shows the relationship between manual versions and releases.

<table>
<tr><th>Manual version</th><th>Valid for</th></tr>
<tr><td>V1.141
V1.140
V1.131
V1.130
V1.123
V1.122
V1.121
V1.120
V1.111
V1.110
V1.103
V1.102
V1.101
V1.100
V1.92
V1.91
V1.90
V1.80
V1.71
V1.70
V1.64
V1.63.2
V1.63.1
V1.63
V1.62
V1.61
V1.60
V1.52.1
V1.52
V1.51
V1.50.1
V1.50
V1.42
V1.41
V1.40
V1.20
V1.10</td><td>
<table>
<tr><th>Version</th><th>Starting with</th><th>Up to</th></tr>
<tr><td>Product set</td><td>Release 1.2</td><td>Release 1.10</td></tr>
<tr><td>SafeDESIGNER</td><td>2.70</td><td>4.9</td></tr>
<tr><td>Firmware</td><td>270</td><td>399</td></tr>
<tr><td>Upgrades</td><td>1.2.0.0</td><td>1.10.999.999</td></tr>
</table>
</td></tr>
<tr><td>V1.02
V1.01
V1.00</td><td>
<table>
<tr><th>Version</th><th>Starting with</th><th>Up to</th></tr>
<tr><td>Product set</td><td>Release 1.0</td><td>Release 1.1</td></tr>
<tr><td>SafeDESIGNER</td><td>2.58</td><td>2.69</td></tr>
<tr><td>Firmware</td><td>256</td><td>269</td></tr>
<tr><td>Upgrades</td><td>1.0.0.0</td><td>1.1.999.999</td></tr>
</table>
</td></tr>
</table>

Table 2: Release information

1.1.3 Organization of safety notices

Safety notices in this manual are organized as follows:

Safety notice	Description
Danger!	Failure to observe these safety guidelines and notices can result in severe injury, death or substantial damage to property.
Information:	These instructions are important for avoiding malfunctions.

Table 3: Organization of safety notices

1.1.4 EC declaration of conformity

This document was originally written in the German language. The German edition therefore represents the original documentation in accordance with the 2006/42/EC Machinery Directive. Documents in other languages are to be interpreted as translations of the original documentation.

Product manufacturer:

B&R Industrial Automation GmbH

B&R Strasse 1

5142 Eggelsberg

Austria

Telephone: +43 7748 6586-0

Fax: +43 7748 6586-26

office@br-automation.com

The place of jurisdiction, in accordance with article 17 of the European Convention on Courts of Jurisdiction and Enforcement, is A-4910

Ried im Innkreis, Austria, commercial register court: Ried im Innkreis, Austria

Commercial register number: FN 111651 v.

The place of fulfillment in accordance with article 5 of the European Convention on Courts of Jurisdiction and Enforcement is A-5142 Eggelsberg, Austria

VATIN: ATU62367156

The EC declarations of conformity for B&R products can be downloaded from the B&R website www.br-automation.com.

1.2 Intended use

> **Danger!**
>
> **Danger from incorrect use of safety-related products/functions**
>
> **Proper functionality is only ensured if the products/functions are used in accordance with their intended use by qualified personnel and the provided safety information is taken into account. The aforementioned conditions must be observed or covered by supplementary measures on your own responsibility in order to ensure the specified protective functions.**

1.2.1 Qualified personnel

Use of safety-related products is restricted to the following persons:

- Qualified personnel who are familiar with relevant safety concepts for automation technology as well as applicable standards and regulations
- Qualified personnel who plan, develop, install and commission safety equipment in machines and systems

Qualified personnel in the context of this manual's safety guidelines are those who, because of their training, experience and instruction combined with their knowledge of relevant standards, regulations, accident prevention guidelines and operating conditions, are qualified to carry out essential tasks and recognize and avoid potentially dangerous situations.
In this regard, sufficient language skills are also required in order to be able to properly understand this manual.

1.2.2 Application range

The safety-related B&R control components described in this manual were designed, developed and manufactured for special applications for machine and personnel protection. They are not suitable for any use involving serious risks or hazards that could lead to the injury or death of several people or serious environmental impact without the implementation of exceptionally stringent safety precautions. In particular, this includes the use of these devices to monitor nuclear reactions in nuclear power plants, flight control systems, air traffic control, the control of mass transport vehicles, medical life support systems and the control of weapon systems.

When using safety-oriented control components, the safety precautions applying to industrial control systems (e.g. the provision of safety devices such as emergency stop circuits, etc.) must be observed in accordance with applicable national and international regulations. The same applies for all other devices connected to the system, e.g. drives or light curtains.

The safety guidelines, information about connection conditions (nameplate and documentation) and limit values specified in the technical data must be read carefully before installation and commissioning and must be strictly observed.

1.2.3 Security concept

B&R products communicate via a network interface and were developed for integration into a secure network. The network and B&R products are affected by the following hazards (not a complete list):

- Unauthorized access
- Digital intrusion
- Data leakage
- Data theft
- A variety of other types of IT security breaches

It is the responsibility of the operator to provide and maintain a secure connection between B&R products and the internal network as well as other networks, such as the Internet, if necessary. The following measures and security solutions are suitable for this purpose:

- Segmentation of the network (e.g. separation of the IT and OT networks)
- Firewalls for the secure connection of network segments
- Implementation of a security-optimized user account and password concept
- Intrusion prevention and authentication systems
- Endpoint security solutions with modules for anti-malware, data leakage prevention, etc.
- Data encryption

It is the responsibility of the operator to take appropriate measures and to implement effective security solutions.

B&R Industrial Automation GmbH and its subsidiaries are not liable for damages and/or losses resulting from, for example, IT security breaches, unauthorized access, digital intrusion, data leakage and/or data theft.

Before B&R releases products or updates, they are subjected to appropriate functional testing. Independently of this, the development of customized test processes is recommended in order to be able to check the effects of changes in advance. Such changes include, for example:

- Installation of product updates
- Notable system modifications such as configuration changes
- Import of updates or patches for third-party software (non-B&R software)
- Hardware replacement

These tests should ensure that implemented security measures remain effective and that systems behave as expected.

1.2.4 Safety technology disclaimer

The proper use of all B&R products must be guaranteed by the customer through the implementation of suitable training, instruction and documentation measures. The guidelines set forth in system user's manuals must be taken into consideration here as well. B&R has no obligation to provide verification or warnings with regard to the customer's purpose of using the delivered product.

Changes to the devices are not permitted when using safety-related components. Only certified products are permitted to be used. Currently valid product versions in each case are listed in the corresponding certificates. Current certificates are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product. The use of non-certified products or product versions is not permitted.

All relevant information regarding these safety products must be read in the latest version of the related data sheet and the corresponding safety notices observed before the safety products are permitted to be operated. Certified data sheets are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product.

B&R and its employees are not liable for any damages or loss resulting from the incorrect use of these products. The same applies to misuse that may result from specifications or statements made by B&R in connection with sales, support or application activities. It is the sole responsibility of the user to check all specifications and statements made by B&R for proper application as it pertains to safety-related applications. In addition, the user assumes sole responsibility for the proper design of the safety function as it pertains to safety-related applications.

1.2.5 X20 / X67 system characteristics

Because all X20 and X67 safety products are seamlessly integrated into the B&R base system, the same system characteristics and user notices from the X20 and X67 system user's manuals also apply to X20 and X67 safety products.

Warning!

Possible failure of safety function

Malfunction of module due to unspecified operating conditions
The notes for installation and operation of the modules provided in the applicable documents must be observed.

In this regard, this means the content and user notices in the following applicable documentation must be observed for X20 and X67 safety products:

- X20 system user's manual
- X67 System user's manual
- Installation / EMC guide

1.2.6 Installation notes for X20 modules

Products must be protected against impermissible dirt and contaminants. Products are protected from dirt and contaminants up to pollution degree II as specified in the IEC 60664 standard.

Pollution degree II can usually be achieved in an enclosure with IP54 protection, but uncoated modules are NOT permitted to be operated in condensing relative humidity and temperatures under 0°C.

The operation of coated modules is allowed in condensing relative humidity.

Danger!

Pollution levels higher than specified by pollution degree II in standard IEC 60664 can result in dangerous failures. It is extremely important that you ensure a proper operating environment.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

The power supply of X20 potential groups must generally be protected using a fuse with a maximum of 10 A.
For more information, see chapter "Mechanical and electrical configuration" of the X20 or X67 user's manual.

1.2.7 Installation notes for X67 modules

Danger!

The following must be taken into consideration to ensure IP67 protection:

- **The union nuts on female/male connectors must be tightly secured with the specified tightening torque. The tightening torque value can be found in the X67 system user's manual.**
- **Female/Male connectors that are not being used must be closed with threaded caps!**
    - **M8 threaded caps, 50 pcs.: X67AC0M08**
    - **M12 threaded caps, 50 pcs.: X67AC0M12**

Danger!

The shock and vibration resistance values (see the X67 system user's manual: chapter "International and national certifications") apply if cables are laid solidly.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

Danger!

Unprotected female connectors must be covered with threaded caps (X67AC0M08 or X67AC0M12 accessory). Otherwise, hazardous conditions may arise if the module fails to function properly.

1.2.8 Safe state

If an error is detected by the module (internal or wiring error), the modules enable the safe state. The safe state is structurally designed as a low state or cutoff and cannot be modified.

Danger!

Applications in which the safe state must actively switch on an actuator cannot be implemented with this module. In these cases, other measures must be taken to meet this safety-related requirement (e.g. mechanical brakes for hanging load that engage on power failure).

1.2.9 Mission time

All safety modules are designed to be maintenance-free. Repairs are not permitted to be carried out on safety modules.

All safety modules have a maximum mission time of 20 years.

This means that all safety modules must be taken out of service one week (at the latest) before the expiration of this 20-year time span (starting from B&R's delivery date).

Danger!

Operating safety modules beyond the specified mission time is not permitted! The user must ensure that all safety modules are replaced by new safety modules or removed from operation before their mission time expires.

1.3 System features

1.3.1 X20 and X67 system

The system characteristics for all safety modules remain the same here because of their seamless integration into the X20 and X67 system.
Only the specific characteristics of the safety module are described below.

1.3.2 Integrated safety technology

Seamless integration of safety technology in the standard application is a reality with B&R's safety technology products. This allows fixed wiring to be replaced by safe data transfer via the existing machine bus system. Flexibly configured or programmed safety behavior can be adapted optimally to various safety situations. Complete diagnostic information about safety components accessible via the machine bus system provides detailed data about the state of the machine.

Insufficient protection from manipulation and the inadequacy of existing safety solutions provide some justification for dangerous behavior when it comes to operating the machine. The possibilities provided by the latest safety technology allow considerable potential for improvement here. In the pursuit of improving machine safety, safety technology guidelines are continually updated according to the current technological situation. Consequently, improvements become mandatory. B&R's integrated safety technology is state-of-the-art and meets both current and expected future demands on safety components.

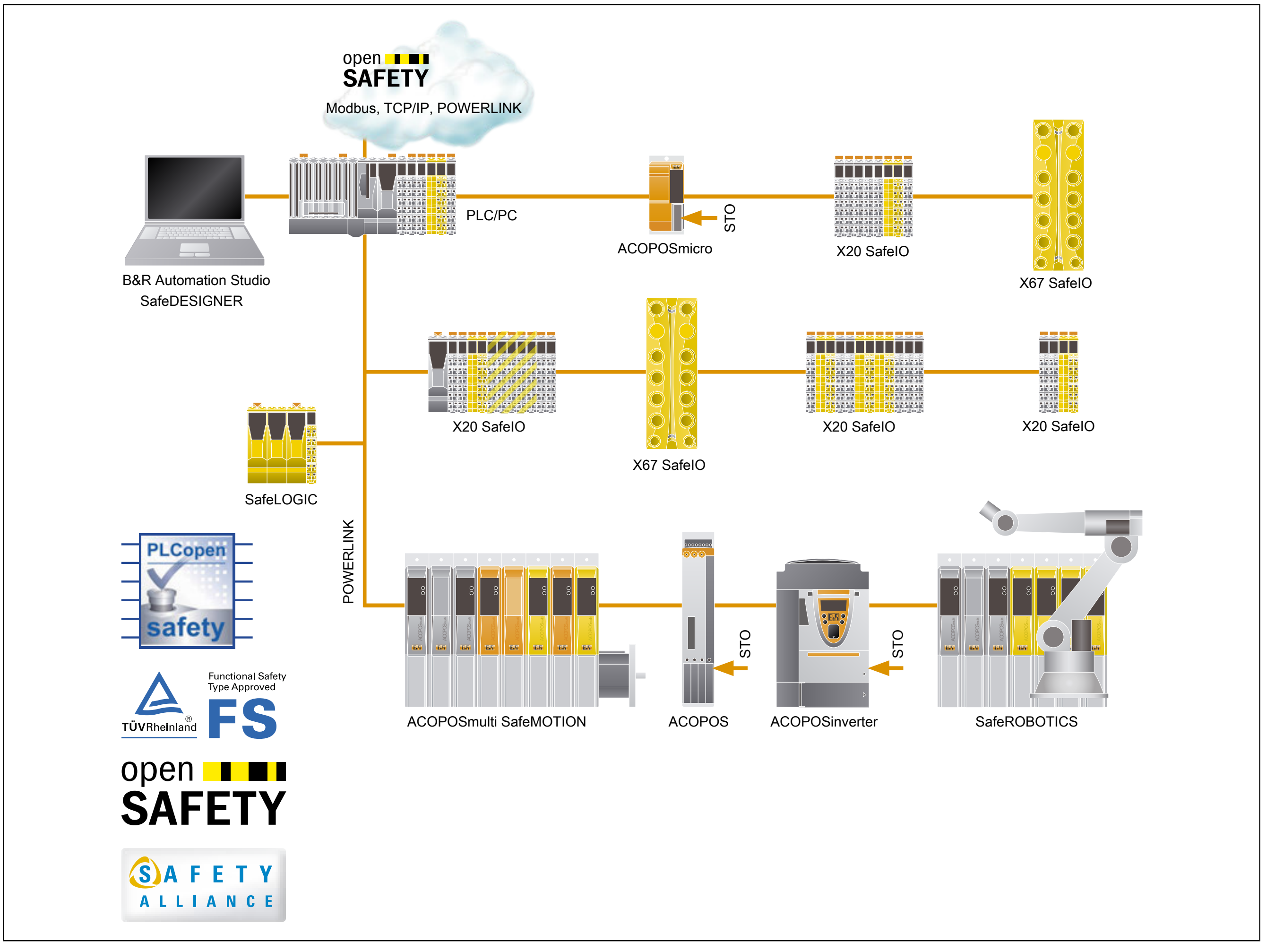
Safety shutdowns do not always have to involve a full machine stop. When opening a protective cover, it is often sufficient to reduce the speed. Smart, safe reactions to various situations provide safety without having to stop the production process. The machine does not need to be emptied and set up again, and manipulation is not necessary. This results in real advantages for the user that can be implemented with programmable safety behavior.

Integrated safety technology products are certified for use in safety applications up to:

- EN ISO 13849, PLe
- IEC 62061, SIL 3
- IEC 61508, SIL 3
- IEC 61511, SIL 3

A cycle time of 200 µs for SIL 3 safety applications is a new feature for safety communication. Response times decrease by a factor of 10, and the advantages of hard-wired solutions are combined with the possibilities of advanced, integrated and intelligent safety bus technology. POWERLINK and openSAFETY accomplish all this using only standard Ethernet mechanisms. As a result, these protocols can be combined with all conventional and, more important, newer Ethernet profiles. openSAFETY is the fastest and most flexible real-time safety bus system on the market.

The reduction to a single cable allows safety-related data to be transferred using the existing infrastructure. Additional cabling of a safe line is not necessary. Transparent and non-reactive access of secure data is an integral part of functional machine control. Complicated communication mechanisms between the safe application and standard application are also no longer necessary. Smart, safe reactions instead of hard machine stops provide advantages for processes, help to avoid manipulation and therefore increase the value of machines.



SafeIO modules can be used anywhere within the X20 and X67 infrastructure. The only requirement is that the underlying transfer protocol is X2X or POWERLINK (i.e. SafeIO modules are not supported on a CAN, DeviceNet, Ethernet/IP, Modbus/TCP or PROFIBUS DP bus controller).

1.3.3 System requirements

Integrated safety technology requires the use of the following software and hardware:

- POWERLINK V2
- Automation Studio V3.0.80 or higher
- Automation Runtime V3.00 or higher
- SG4 CPUs

1.3.4 Safety response time

The safety response time is the time between the arrival of the signal on the input channel and the output of the cutoff signal on the output.

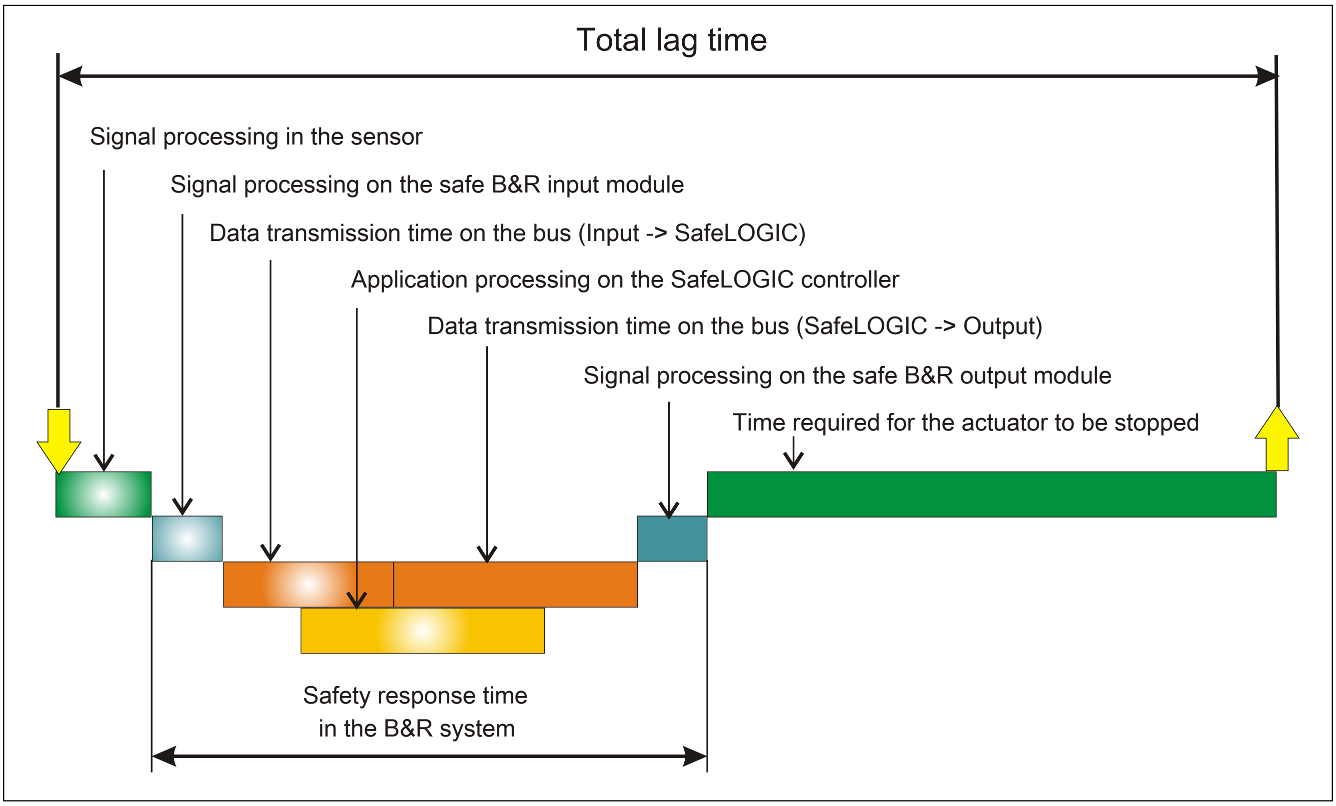


Figure 1: Total lag time

As illustrated in the figure, the safety response time in the B&R system is composed of the following partial response times:

- Signal processing on the safe B&R input module
- Data transmission time on the bus (Input -> SafeLOGIC)
- Data transmission time on the bus (SafeLOGIC -> Output)
- Signal processing on the safe B&R output module

Danger!

The following sections are dedicated exclusively to the safety response time in the B&R system. When assessing the complete safety response time, the user must include signal processing in the sensor as well as the time until the actuator is stopped.

Be sure to validate the total lag time on the system!

Information:

The safety response time in B&R products already contains all delays caused by sampling input data (sampling theorem).

1.3.4.1 Signal processing on the safe B&R input module

The maximum I/O update time in the "I/O update time" chapter for the respective module must be taken into account when processing signals in the safe B&R input module.

1.3.4.2 Data transmission time on the bus

The following relationship must be taken into consideration for the data transmission time on the bus:

- The time needed to transfer data from the input to the SafeLOGIC controller or to the output depends on the sum of the cycle times and CPU copy times in effect on the transfer line.
- POWERLINK MN (managing node, standard CPU) settings are important for the actual timing on the bus, but they cannot be used from a safety point of view since the values can be changed at any time in the course of modifications made outside of the safety application.
- In the SafeLOGIC controller, data transmission times are monitored on the bus using openSAFETY services. The time needed to process the application on the SafeLOGIC controller is taken into account in this test (system-dependent). Monitoring is defined in SafeDESIGNER using the parameters in parameter group "Safety Response Time".

Information:

The safety components located in this network segment could be cut off by the SafeLOGIC controller if modified parameters on the POWERLINK MN alter the data transmission times on the bus so that they lie outside of the SafeDESIGNER parameters defined in parameter group "Safety Response Time".

Information:

The safety components located in this network segment could be cut off by the SafeLOGIC controller if EMC disturbances cause data failures that fall outside of the SafeDESIGNER parameters defined in parameter group "Safety Response Time".

Calculating the maximum data transmission time - up to Release 1.9:

- The total max. data transmission time on the bus is calculated by adding parameter "Worst_Case_Response_Time_us" for the safe input module and parameter "Worst_Case_Response_Time_us" for the safe output module. When doing this, be sure to check parameter "Manual_Configuration". If parameter "Manual_Configuration" is set to "No", the value specified for parameter "Default_Worst_Case_Response_Time_us" is used.
- **Special case: Local inputs on the X20SLX module:**
  The total max. data transmission time on the bus is calculated by adding parameter "Cycle_Time_max_us" + 2000 µs and parameter "Worst_Case_Response_Time_us" for the safe output module. When doing this, be sure to check parameter "Manual_Configuration". If parameter "Manual_Configuration" is set to "No", the value specified for parameter "Default_Worst_Case_Response_Time_us" is used.

Calculating the maximum data transmission time - Release 1.10 and later:

The following parameters are relevant for calculating the data transmission time between the safe input module and safe output module; parameter "Manual Configuration" deserves special attention.

- Relevant parameters for "Manual Configuration = No":
  - "PacketLoss1": Parameter "Default Additional Tolerated Packet Loss" of group "Safety Response Time Defaults" of the SafeLOGIC controller
  - "DataDuration1": Parameter "Default Safe Data Duration" of group "Safety Response Time Defaults" of the SafeLOGIC controller
  - "NetworkSyncCompensation1": 12 ms
  - "PacketLoss2": Same as "PacketLoss1"
  - "DataDuration2": Same as "DataDuration1"
  - "NetworkSyncCompensation2": Same as "NetworkSyncCompensation1"
- Relevant parameters for "Manual Configuration = Yes":
  - "PacketLoss1": Parameter "Additional Tolerated Packet Loss" of group "Safety Response Time" of the safe input module
  - "DataDuration1": Parameter "Safe Data Duration" of group "Safety Response Time" of the safe input module
  - "NetworkSyncCompensation1": 12 ms
  - "PacketLoss2": Parameter "Additional Tolerated Packet Loss" of group "Safety Response Time" of the safe output module
  - "DataDuration2": Parameter "Safe Data Duration" of group "Safety Response Time" of the safe output module
  - "NetworkSyncCompensation2": Same as "NetworkSyncCompensation1"
- **Special case: Local inputs on the X20SLX module:**
  - "PacketLoss1": 0
  - "DataDuration1": Parameter "Cycle Time max" of group "Module Configuration" of the X20SLX + 2000 µs
  - "NetworkSyncCompensation1": 0 ms
- **Special case: Local outputs on the X20SLX module:**
  - "PacketLoss2": 0
  - "DataDuration2": Parameter "Cycle Time max" of group "Module Configuration" of the X20SLX + 2000 µs
  - "NetworkSyncCompensation2": 0 ms
- **Special case: Linking local inputs with local outputs on the X20SRT module:**
  - "PacketLoss1": 0
  - "PacketLoss2": 0
  - "DataDuration1": Parameter "Cycle time" of group "General"
  - "DataDuration2": Parameter "Cycle time" of group "General"
  - "NetworkSyncCompensation1": 0 ms
  - "NetworkSyncCompensation2": 0 ms

The following equation is used to calculate the maximum data transmission time between the safe input module and safe output module:

Maximum data transmission time = (PacketLoss1+1)* DataDuration1 + NetworkSyncCompensation1 + (PacketLoss2+1)* DataDuration2 + NetworkSyncCompensation2

Information:

In addition to the data transmission time on the bus, the time for signal processing in the safe B&R input and output module must be taken into account (see section 1.3.4 "Safety response time").

Information:

For more information about the actual data transmission time, see Automation Help, section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime. The cycle time of the safety application must also be added.

1.3.4.3 Signal processing on the safe B&R output module

The maximum I/O update time in the "I/O update time" chapter for the respective module must be taken into account when processing signals in the safe B&R output module.

1.3.4.4 Minimum signal lengths

The parameters in group "Safety Response Time" in SafeDESIGNER influence the maximum number of data packets that are permitted to fail without triggering a safety response. These parameters therefore act like a switch-off filter. If several data packets are lost within the tolerated amount, safety signals may not be detected if their low phase is shorter than the determined data transmission time.

Danger!

Lost signals can result in serious safety errors. Check all signals to determine the smallest possible pulse length and make sure that it is larger than the determined data transmission time.

Suggested solution:

- The switch-on filter can be used to extend the low phase of a signal on the input module.
- Low phases of signals from the SafeLOGIC controller can be lengthened with restart interlock functions or timer function blocks.

1.4 Mechanical and electrical configuration

Because of the X20 safety modules' seamless integration into the X20 system, all of the electrical and mechanical configuration specifications are also valid here.

Only the specific characteristics of the X20 safety modules are described below.

1.4.1 X20 safety module dimensions

1.4.1.1 SafeLOGIC

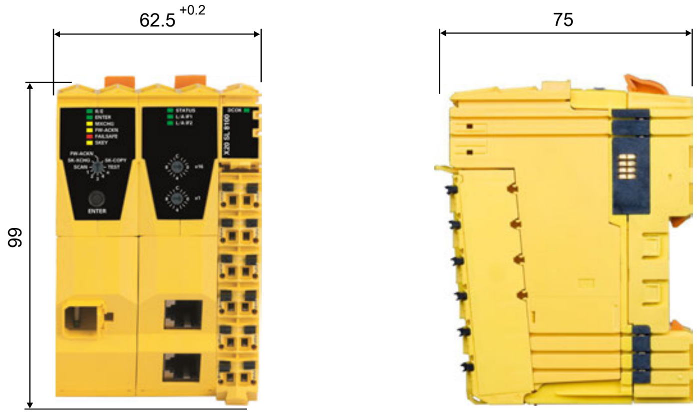


Figure 2: SafeLOGIC controller dimensions

1.4.1.2 SafeIO modules

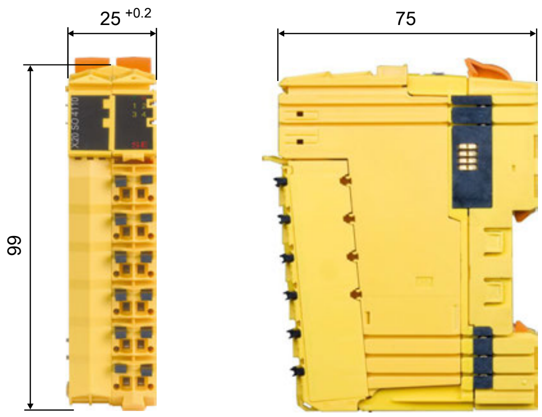


Figure 3: SafeIO module dimensions

1.5 Problems and solutions

1.5.1 SafeLOGIC and SafeIO

This section lists some problems which may occur while you are working with the X20 system. For each problem, the measures to be taken and the required user response are described.

Problem	Solution
SafeIO	
The module is not functioning properly. • "r" LED is not lit constantly or • "e" LED is not off or • "ModulOK" signal is FALSE	Error in module functionality: • Module not booted by CPU • Problems with X2X Link • Firmware update (takes up to 10 minutes) • Defective module Extensive error analysis using LED codes and entries in the log file.
The module is not functioning properly. • "r" LED is lit constantly and • "e" LED is lit constantly and • "SE" LEDs are off and • All channel LEDs are red	Error in the 24 V I/O supply
The module is not functioning properly. • "r" LED is lit constantly and • "e" LED is off and • "SE" LEDs - one or both are on • "ModulOK" signal is TRUE	Defective module. The module must be replaced.
The module is not functioning properly. • "r" LED is lit constantly and • "e" LED is off and • "SE" LEDs are single-flashing • All channel LEDs are red • "ModulOK" signal is TRUE	Module is in the PREOP state. Possible causes: • SafeLOGIC controller not active • User action for this module has not yet been acknowledged on the SafeLOGIC controller (UDID mismatch, firmware update) • Network problems between SafeLOGIC and SafeIO • Incompatible firmware Extensive error analysis using entries in the log file.
The module is not functioning properly. • "r" LED is lit constantly and • "e" LED is off and • "SE" LEDs are double-flashing • All channel LEDs are red • "ModulOK" signal is TRUE	Module in OP state, safe communication channel showing errors. Possible causes: • Incompatible configuration of X2X / POWERLINK / copy task cycle time in the standard application for the "Worst case response time" parameters in SafeDESIGNER. • There are more network disruptions than allowed for by the "Worst case response time" parameters in SafeDESIGNER. Extensive error analysis using entries in the log file.
Individual channels not functioning properly. • Affected channel LEDs are red	Wiring error on the respective channel. Extensive error analysis using entries in the log file.
Safe digital input module	
Individual channels not functioning properly. • "OO" or "OC" LEDs are red	Check the switching behavior of the two-channel sensor or check the "Discrepancy_Time_us" parameter
Individual channels not functioning properly.	Check the channel-specific parameters (filter, "Pulse Mode", "Discrepancy Time", parameters for safety response time)
Safe digital output module	
Output channel not being set. • Affected channel LED is off	Check the signal relevant to setting an output channel ("DigitalOutput", "SafeDigitalOutput", "ReleaseOutput", restart inhibit). Extensive error analysis using the "FBK_Status_1" signal and the entries in the log file.
Individual channels not functioning properly. • Affected channel LEDs are red	Check the max. switching frequency against the parameters in the I/O configuration. Extensive error analysis using entries in the log file.
SafeLOGIC	
The module is not functioning properly. • "S/E" LED on the POWERLINK interface is not lit constantly.	Error in SafeLOGIC controller functionality: • SafeLOGIC controller not booted by CPU • Problems with POWERLINK • Firmware update • Defective module Extensive error analysis using LED codes and entries in the log file.
The module is not functioning properly. • "S/E" LED on the POWERLINK interface is not lit constantly. • "FAIL", "FI" or "AL" LED is constantly lit on the safety processor • "ModulOK" signal is TRUE	Defective module. The module must be replaced.

Table 4: X20 problems and solutions

Problem	Solution
The module is not functioning properly. • "R/E" LED on the safety processor is blinking green	Application exists, but CPU stopped • "Automatic start" not selected in the download dialog box in SafeDESIGNER. • Boot phase, i.e. not all necessary safe modules on the network have been correctly configured yet.
The module is not functioning properly. • "R/E" LED on the safety processor showing rapid orange blinking sequence	No application exists on the SafeKEY.
The module is not functioning properly. • "R/E" LED on the safety processor showing orange blinking sequence	SafeDESIGNER in debug mode, application stopped
The module is not functioning properly. • "L" LED on the safety processor constantly lit red	The SafeLOGIC controller is in the PREOP state. Possible causes: • User action for this module has not yet been acknowledged on the SafeLOGIC controller (UDID mismatch, firmware update) • Incompatible firmware Extensive error analysis using entries in the log file.

Table 4: X20 problems and solutions

1.5.2 SafeDESIGNER

This section lists some problems which may occur while you are working with the SafeDESIGNER. For each problem, the measures to be taken and the required user response are described.

The descriptions are divided into several categories according to the part of the programming system reporting the problem:

- General (affects the whole programming system)
- Code editor and variable editor
- Project tree
- Device configuration editor
- Compiler
- Online communication between the programming system and the safety controller
- Messages from the safety controller

Problem	Solution
General	
When starting up the safe programming system, the installation check routine detects a corrupt system file. A corresponding message window is displayed.	Uninstall the safe programming system and start the setup program from the installation CD to reinstall the software.
The operating system verification routine determined that you wanted to start the safe programming system on an unsupported operating system.	Install an operating system which is supported by the safe programming system or ask our technical support whether a newer version of the programming system is available that supports the operating system you are currently using.
An error has occurred (indicated by a corresponding message) that cannot be remedied by the corrective measures listed here.	Please contact our technical support department.
The safe programming system or one of its functionalities does not behave as described in the user's manual or help system.	Please contact our technical support department.
Use of Symantec Endpoint anti-virus program on the PC	Add an exception for handling files with extension ".sto" in the anti-virus program.
Code editor and variable editor	
You tried to open a worksheet, but it could not be loaded due to a checksum error. The editor displays a message window.	The affected POU is corrupt and has to be deleted. In case of the "Main" POU (which cannot be deleted), the entire project can no longer be used. Revert back to the last safety copy of the project.
After entering data into a worksheet, the editor reacts unexpectedly. Example: You entered a coil, but a contact is displayed instead. This may result from faulty user operation, a sporadic error or a systematic error.	Undo the last command (<Ctrl>+<Z>) and retry the input. If the result is still faulty, please contact our technical support.
While you are editing, a message window appears in which the editor reports a corrupt file, sporadic error or systematic error.	The worksheet closes automatically. You do not have the chance to save any changes that were made.
You tried to open the global variable worksheet, but the variable worksheet could not be opened due a checksum error. The editor displays a corresponding message box.	The project can no longer be used since the variable worksheet for global declarations cannot be deleted. Revert back to the last safety copy of the project.
Project tree	
After copying a POU to the project tree (using the clipboard), the POU does not contain the expected code or variable worksheet.	Delete the copy of the faulty POU and try to copy it again. If the problem persists, please contact our technical support department.
Compiler	
A worksheet cannot be read by the compiler due to a checksum error. A corresponding error message is displayed in the message window.	The affected POU is corrupt and has to be deleted. In case of the "Main" POU (which cannot be deleted), the entire project can no longer be used. Revert back to the last safety copy of the project.
The second compiler detected an error in the project structure and shows a corresponding error message in the message window.	Please contact our technical support department.
A compiler detects a syntax or semantic error in the user program and displays a corresponding error message in the message window.	Open the affected worksheet and correct the error. Worksheets containing an error can be accessed directly from the message window by double-clicking on the error message. The error position (object/item) is automatically highlighted in the worksheet.
A compiler outputs an error message into the message window, the cause of which you cannot correct (e.g. "Internal error").	Try compiling the project again. If the error is still reported, please contact our technical support department.
Online communication between the programming system and the safety controller	
An error occurred while downloading the project to the safety controller, or it is not possible to establish a connection to the safety controller at all. A message window will open in both of these cases.	Check the connection settings and cabling and then try it again. If the problem persists: Try to establish a connection to another safety controller that is currently unused (if available). If this works, replace the faulty safety controller. If the problem persists: Try to establish a connection to the safety controller from another PC. If you are successful, the interface or network adapter on the computer may not be working properly. If the problem persists, please contact our technical support department.
After successfully transferring the project, the programming system detects that the checksum of the project on the safety controller does not match the checksum of the project on the PC. A corresponding message window is displayed.	Try transferring the project again. If the problem persists: Try to download the project to another safety controller that is currently unused (if available). If this works, replace the safety controller. If the problem persists, please contact our technical support department.
Messages from the safety controller	
The safety controller reports an error while compiling the program from the 1st compiler (the program from the 2nd compiler has already been completely compiled on the PC).	Please contact our technical support department.
The safety controller is reporting an internal error.	Please contact our technical support department.

Table 5: SafeDESIGNER problems and solutions

2 X20 system

2.1 General information

The following points listed in this chapter are excerpts from the X20 System user's manual (V3.35):

- "General information"
- "System features"
- "Mechanical and electrical configuration"
- "Accessories"
- "Mechanical handling"
- "International and national certifications"

2.1.1 Transport and storage

During transport and storage, devices must be protected against undue stress (mechanical loads, temperature, moisture, corrosive atmospheres, etc.).

Devices contain components sensitive to electrostatic charges that can be damaged by improper handling. It is therefore necessary to provide the required protective measures against electrostatic discharge when installing or removing these devices (see "Protection against electrostatic discharges" on page 38).

2.1.2 Mounting orientation

- Installation must be performed according to this documentation using suitable equipment and tools.
- Devices are only permitted to be installed by qualified personnel without voltage applied.
- General safety guidelines and national accident prevention regulations must be observed.
- Electrical installation must be carried out in accordance with applicable guidelines (e.g. line cross sections, fuses, protective ground connections).
- Take the necessary steps to protect against electrostatic discharges (see "Protection against electrostatic discharges" on page 38).

2.1.2.1 Inserting and removing I/O modules while the controller is running

I/O modules may be connected and disconnected while the controller is running under the following conditions:

- Connectors are not allowed to carry voltages and must be removed.
- Replacing a module during operation must be supported by the software; otherwise, disconnecting a module will cause an emergency stop of the controller.

2.1.2.2 Connecting/Disconnecting IF modules while the controller is running

Unlike I/O modules, IF module CANNOT be connected or disconnected while the controller is running.

> **Warning!**
>
> **IF modules that are connected or disconnected while the controller is running are not recognized by the CPU or bus controller and will cause the application to behave incorrectly.**

2.1.3 Operation

2.1.3.1 Protection against touching electrical parts

Danger!

In order to operate programmable logic controllers, operating and monitoring devices and the uninterruptible power supply, it is necessary for certain components to carry dangerous voltages. Touching one of these components can result in a life-threatening electric shock. There is a risk of death, serious injury or damage to property.

Before switching on the programmable logic controllers, operating and monitoring devices and uninterruptible power supply, it must be ensured that the housing is properly connected to ground potential (PE rail). The ground connection must also be made if the operating and monitoring device and uninterruptible power supply are only connected for testing purposes or only operated for a short time!

Before switching on the device, all voltage-carrying components must be securely covered. During operation, all covers must remain closed.

2.1.4 Environmentally friendly disposal

All B&R control components are designed to inflict as little harm on the environment as possible.

2.1.4.1 Separation of materials

It is necessary to separate different materials so the device can undergo an environmentally friendly recycling process.

Component	Disposal
X20 modules Cables	Electronic recycling
Cardboard/paper packaging	Paper/Cardboard recycling

Table 6: Environmentally friendly separation of materials

Disposal must comply with applicable legal regulations.

2.2 Safety guidelines

2.2.1 Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 7: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 8: Organization of general notices

2.2.2 Protection against electrostatic discharges

Electrical components that can be damaged by ESD (**E**lectro**S**tatic **D**ischarges) must be handled properly.

2.2.2.1 Packaging

- Electrical components with a housing
  ...do not require special ESD packaging but must be handled properly
  (see "Electrical components with a housing" on page 39).
- Electrical components without a housing
  ...are protected by ESD-suitable packaging.

2.2.2.2 Guidelines for proper ESD handling

Electrical components with a housing

- Do not touch the male connector contacts on the device (bus data contacts).
- Do not touch the male connector contacts on connected cables
- Do not touch the contact tips on circuit boards

Electrical components without a housing

The following points apply in addition to the points listed under "Electrical components with a housing":

- Any persons handling electrical components or devices with installed electrical components must be grounded.
- Components are only permitted to be touched on their narrow sides or front plate.
- Components must always be placed on or stored in a suitable medium (ESD packaging, conductive foam, etc.).

> **Information:**
>
> **Metallic surfaces are not suitable storage surfaces.**

- Components must not be subjected to electrostatic discharge (e.g. caused by charged plastics).
- Observe a minimum distance of 10 cm from monitors and television sets.
- Measuring instruments and equipment must be grounded.
- Probe tips of galvanically isolated measuring instruments must be temporarily discharged on suitably grounded surfaces before taking measurements.

Individual components

- ESD protective measures for individual components are thoroughly implemented at B&R (conductive floors, footwear, arm bands, etc.).
- Increased ESD protective measures for individual components are not required for handling B&R products at customer locations.

2.3 System features

2.3.1 Setting the standards in automation

There are many different I/O slice systems. With the X20 system, B&R continues to set standards according to its motto "Perfection in Automation". Born from experience gained from applications all over the world, numerous conversations with customers and with the aim for easier, more economical and secure usage, the X20 system is a universal solution for any automated task in machine and system manufacturing.

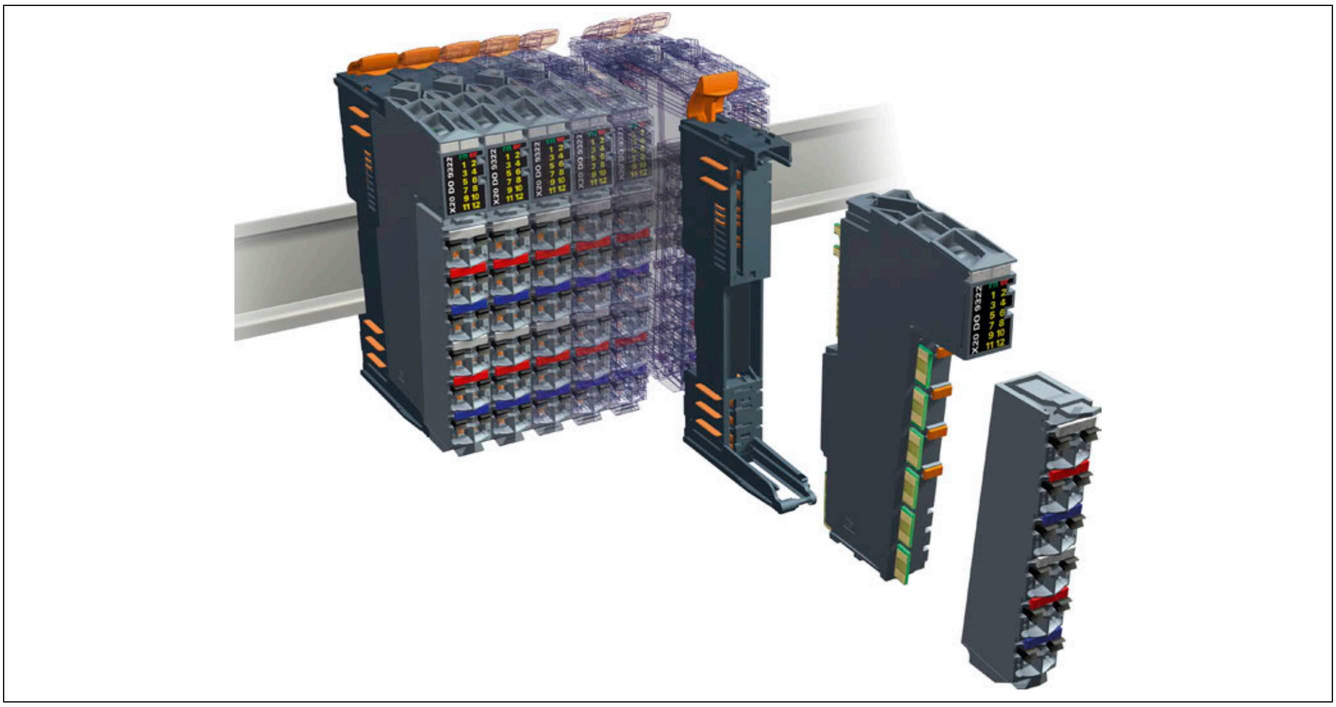


Figure 4: Each module consists of 3 subcomponents – terminal block, electronics module and bus module.

2.3.1.1 More than just I/O

With its well thought-out details and a sophisticated ergonomic design, the X20 system is more than a remote I/O system – it's a complete control solution. The X20 system family makes it possible to combine the exact components needed to meet any application requirements.

- The X20 system is the ideal addition to a standard fieldbus and expands the possibilities of conventional control systems. Simply connect it, configure it and you're done.
- Teamed up with other B&R components, the X20 system achieves its full potential and allows the implementation of applications with unimagined performance and flexibility. This type of seamless integration is a major advantage.

2.3.1.2 3 x 1 = 1

Three basic elements make up one module: Terminal block – Electronic module – Bus module

This modularity results in a system that combines the advantages of both rack and I/O slice systems:

- Prewiring without the module
- Hot pluggable electronics
- Extra bus slots for added options

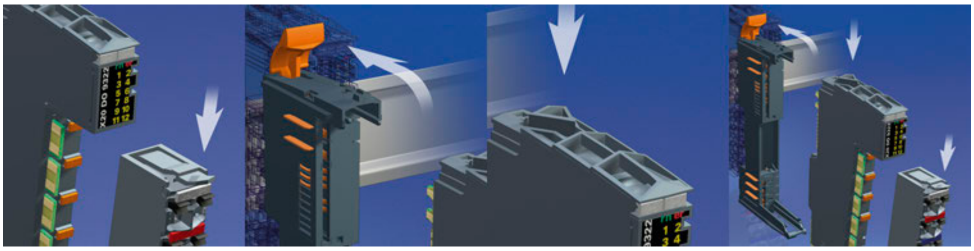


Figure 5: X20 modules are divided into three parts to guarantee the simplest usability

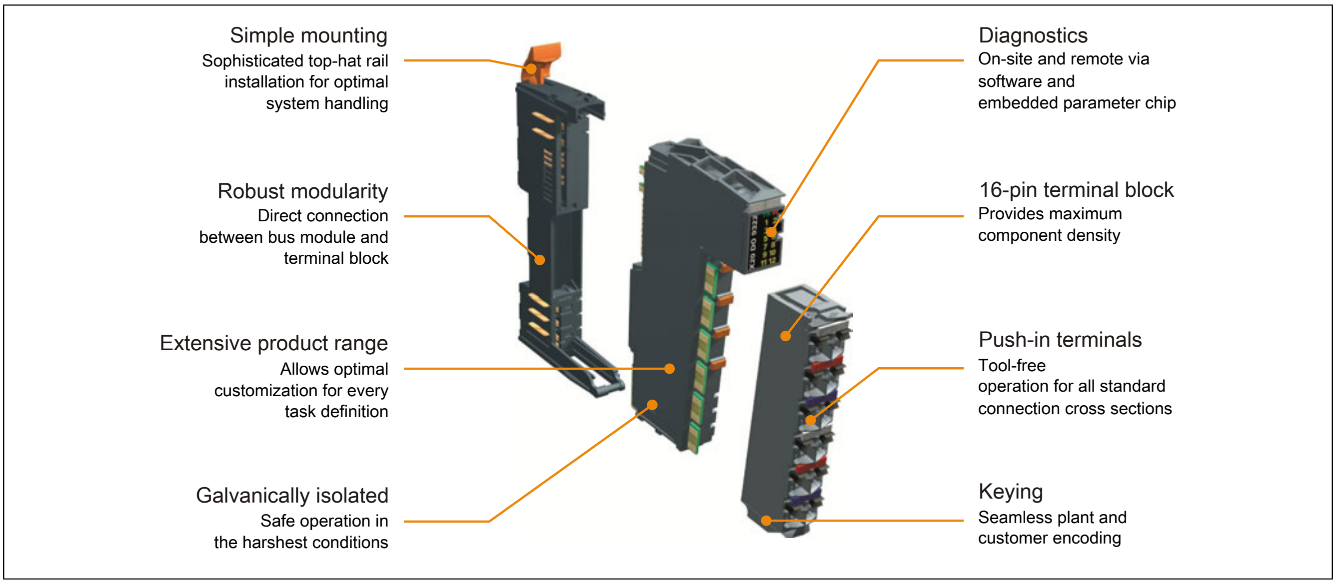
The X20 system delivers 50% more component density, perfected connection technology and optimal granularity.

- **Added value**
  12 channels with a width of 12.5 mm allow a component density never before achieved with optimal terminal ergonomics. As a result, the X20 system offers 50% more channels than conventional slice systems. And this without sacrificing terminal connections.
- **Uniformity**
  Consistent implementation of 1-, 2- or 3-wire connections – no additional jumper terminals needed.
- **Granularity**
  1-channel and 2-channel modules: Maximum flexibility so you only have to pay for what you really need.

2.3.2 Optimized design

X20 modules consist of three submodules to provide maximum ease of use throughout their entire life cycle. This division into bus module, electronics module and terminal block has several advantages.

- **Preconfigured for different machine types**
  The X20 system bus modules are the basic platform for many machine variations. The design of the machine determines which electronics modules are used. The software recognizes this design automatically and makes sure that the right functions are provided where they are needed. Handling a range of different machine variants couldn't be easier.
- **Industrial control cabinet construction**
  X20 system terminal blocks are separate from the electronics module and make it possible to pre-wire the entire control cabinet. This is especially ideal for series-produced machines.
- **Easy maintenance**
  X20 modules can be easily exchanged to simplify troubleshooting. The electronic modules can be exchanged without interrupting operation. The wiring remains exactly the same thanks to the separate terminal blocks. Being able to exchange the automation components quickly reduces downtime.



2.3.3 Remote backplane

The main idea: Remote backplane for a rack system – in other words, the cable is the backplane. All modules are connected to the uniform backplane (X2X Link). Directly connected X20, X67 or XV modules can each be placed at a distance of up to 100 m outside the confines of the control cabinet. X2X Link guarantees the highest possible level of resistance to disturbances based on twisted copper cables.

This not only provides a universal remote backplane which handles the communication between bus modules and via the X2X Link cable, but makes it possible without converters or any loss in performance. A unique feature of the X20 system is the possibility to later integrate machine options on bus modules that are not yet being used without having to change the software addressing.

Note:

A 100 m X2X Link cable is available from B&R for custom assembly (model number: X67CA0X99.1000).

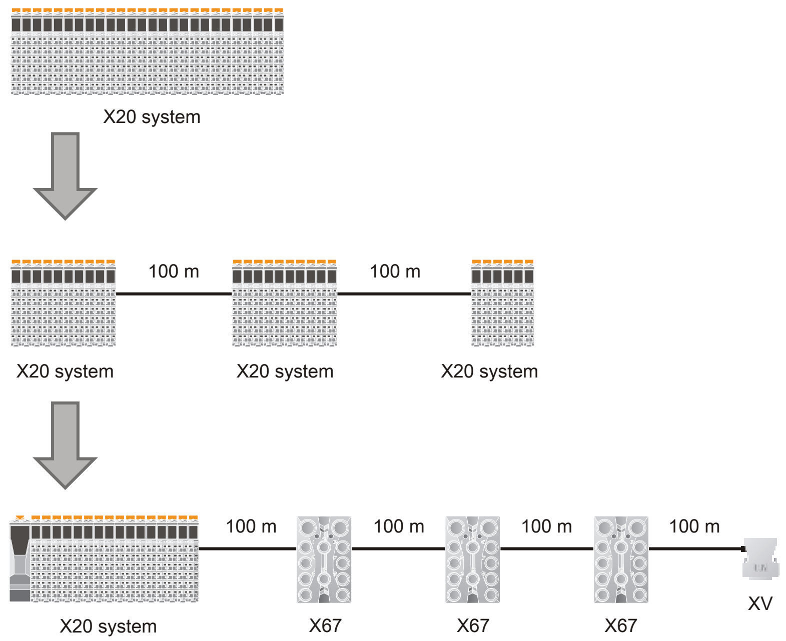
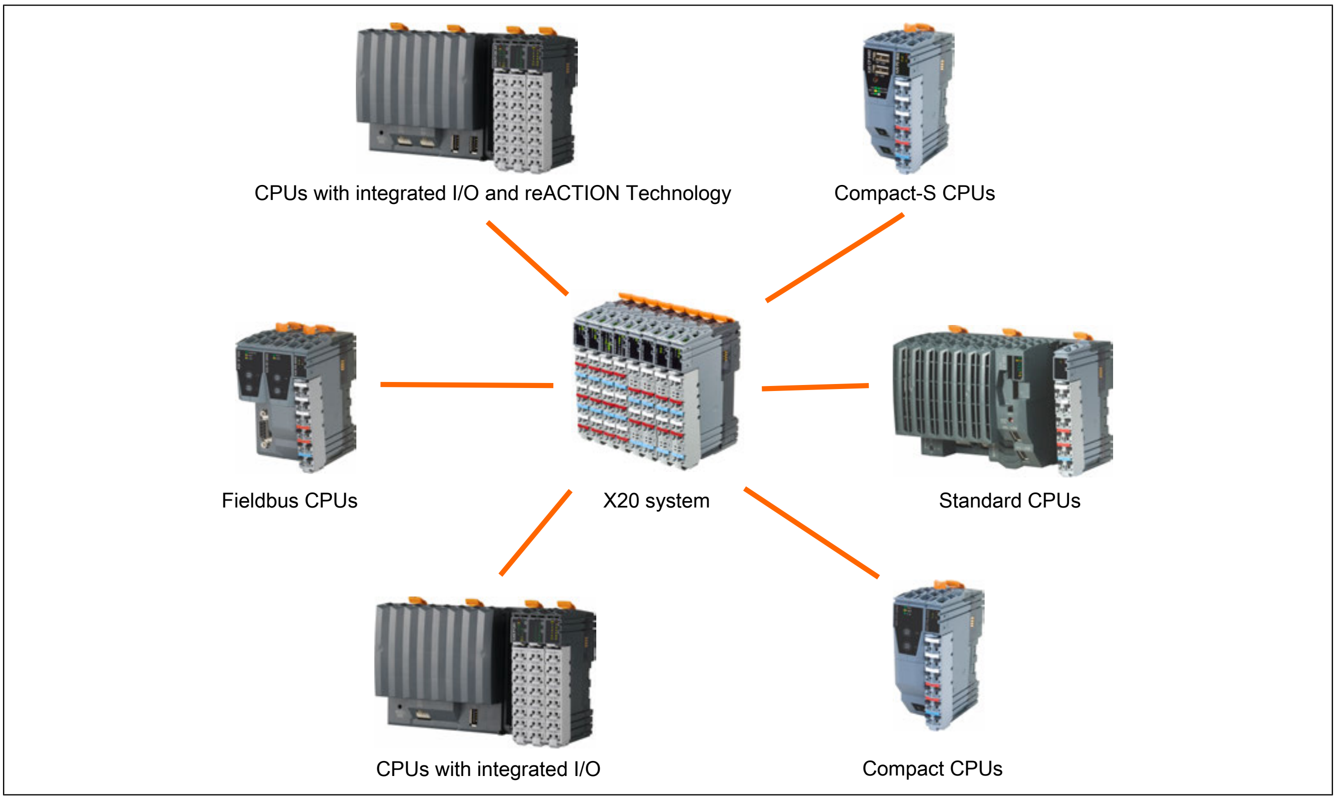


Figure 6: X2X Link - universal backplane based on twisted copper cables

2.3.4 X20 CPUs

The series of CPUs in the X20 system landscape cover a wide range of requirements. Areas of use include simple applications where cycle times in the millisecond range are sufficient to applications that place the highest demands on performance. In these, even cycle times of 100 µs can be used effectively.

The design of the CPUs is in line with the X20 system landscape. X20 I/O modules are connected directly to the CPU. Attached seamlessly to the CPU, they allow the entire space-saving system to fit inside the control cabinet. The supply for the CPU, X2X Link network and I/O modules is part of the CPU. No additional power supply modules are required.



CPUs are divided into the following 6 categories:

- Standard CPUs
- Standard CPUs with integrated I/O
- Standard CPUs with integrated I/O and reACTION Technology
- Compact CPUs
- Compact-S CPUs
- Fieldbus CPUs

2.3.4.1 Features

Remote backplane

A power supply integrated in the CPU with I/O power supply terminals powers the backplane, I/O sensors and actuators, eliminating the need for additional system components. A direct I/O connection to an X20 CPU opens up all of the advantages of a remote backplane, in particular the ability to place as many I/O sections as needed within 100 m using a cable or to add modules with IP67 protection.

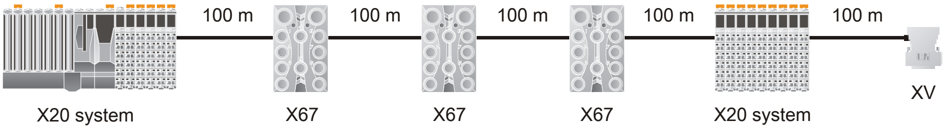


Figure 7: X20 CPUs - Direct I/O connection to X20 CPUs and the advantages of the remote backplane

Programming

B&R Automation Studio is the only programming tool needed for all platforms. All relevant IEC 61131-3 languages and C can be used to create the application software.

Depending on the CPU series, integrated HMI, NC / soft CNC functions and web server technologies complete the range of powerful features.

Industrial strength

Covering a broad performance spectrum and equipped with standard interfaces that can be expanded as needed with interface modules, system dimensions still remain extremely compact. The size and shape of the CPU matches that of the X20 modules to prevent unnecessary wasted space in the control cabinet.

None of the processors require a fan, making them virtually maintenance-free.

2.3.4.2 Many CPU series

To cover a wide range of requirements, X20 CPUs are divided into 6 different series.

Standard CPUs

Based on Intel ATOM processor technology, this CPU series covers a wide spectrum of requirements. Areas of use include standard applications to applications that place the highest demands on performance.

The basic model includes USB, Ethernet, POWERLINK V1/V2 and removable CompactFlash card. The standard Ethernet interface is capable of handling gigabit communication. For even more real-time network performance, the onboard POWERLINK interface supports poll response chaining mode (PRC).
Up to 3 more slots are available for additional interface modules to increase flexibility.

- CPU clock frequency from 100 to 1600 MHz
- Onboard Ethernet, POWERLINK V1/V2 with poll response chaining and USB
- 1 or 3 slots for modular interface expansion
- CompactFlash as removable application memory
- Up to 512 MB DDR2 SRAM according to performance requirements
- CPU redundancy possible
- Fanless

Standard CPUs with integrated I/O

This CPU series is available with processor speeds of 200 MHz and 400 MHz. Depending on the variant, up to 256 MB RAM and up to 32 kB nonvolatile onboard RAM is available. A built-in flash drive is available to store up to 2 GB of application and other data.

All CPUs come equipped with Ethernet, USB and one RS232 interface. In both performance classes, integrated POWERLINK and CAN bus interfaces are also available. If additional fieldbus connections are needed, the CPU can be upgraded with an interface module from the standard X20 product range. These CPUs do not require fans or batteries and are therefore maintenance-free. 30 digital inputs and outputs and 2 analog inputs are integrated in the devices. 1 analog input can be used for PT1000 resistance temperature measurement.

- CPU clock frequency from 200 to 400 MHz
- Onboard Ethernet, POWERLINK with poll response chaining and USB
- 1 slot for modular interface expansion
- 30 digital inputs/outputs and 2 analog inputs integrated in the device
- 1/2 GB onboard flash drive
- 128/256 MB DDR3 SDRAM
- Fanless with no batteries
- Battery-backed real-time clock

Standard CPUs with integrated I/O and reACTION Technology

This CPU series is available with processor speeds of 200 MHz and 400 MHz. Depending on the variant, up to 256 MB RAM and up to 32 kB nonvolatile onboard RAM is available. A built-in flash drive is available to store up to 2 GB of application and other data.

These CPUs come equipped with POWERLINK, Ethernet, CAN bus, 2x USB and one RS232 interface. If additional fieldbus connections are needed, the CPU can be upgraded with an interface module from the standard X20 product range. These CPUs do not require fans or batteries and are therefore maintenance-free. 30 digital inputs and outputs and 2 analog inputs are integrated in the devices. 1 analog input can be used for PT1000 resistance temperature measurement.

The CPUs are equipped with the ultrafast reACTION Technology. All integrated I/O channels are reACTION-capable and can be operated by the reACTION program. These I/O channels are controlled with a response time up to 1 µs. All of the commands that can be used for reACTION programs are provided as function blocks in special libraries (e.g. AsIORTI). Programming in compliance with IEC 61131-3 requirements takes place in the Function Block Diagram editor in Automation Studio.

- CPU clock frequency from 200 to 400 MHz
- Onboard Ethernet, POWERLINK with poll response chaining and USB
- 1 slot for modular interface expansion
- Onboard reACTION Technology
- 30 digital inputs/outputs and 2 analog inputs integrated in the device
- 1/2 GB onboard flash drive
- 128/256 MB DDR3 SDRAM
- Fanless with no batteries
- Battery-backed real-time clock

reACTION TECHNOLOGY

Compact CPUs

Compact CPUs are ideal for situations where cycle times in the millisecond range are sufficient and a cost-benefit analysis plays a decisive role. A range of models with CAN and Ethernet can be adapted optimally to meet all requirements.

- Embedded µP 16 / µP 25 with additional I/O processor
- 100/750 kB User SRAM
- 1/3 MB User FlashPROM
- X20CP0291 and X20CP0292: Onboard Ethernet
- No battery
- Only 37.5 mm wide

Compact-S CPUs

The CPUs in the X20 Compact-S family are available in 5 different variants. This way, customers get the product that best meets the requirements of the machine – technically and economically.

The processor performance of the compact CPUs ranges from 166 MHz (compatible) to 667 MHz. The most economical variant comes equipped with 128 MB RAM, 8 kB nonvolatile RAM and 256 MB flash drive. The most powerful version of the Compact-S CPUs achieves cycle times down to 400 µs and has 64 kB nonvolatile RAM as well as 2 GB internal flash drive.

With POWERLINK, Ethernet, USB and RS232, the CPUs offer a wide range of communication options. An optional CAN interface is also available. If the application requires additional interfaces, the CPU can be modularly expanded by one or two X20 interface slots. This allows the entire product range of X20 fieldbus interfaces to be used.

The fanless, battery-free design of Compact-S CPUs means they are completely maintenance-free.

- CPU clock frequency from 166 to 667 MHz
- Depending on the variant: POWERLINK with poll-response chaining
- 2x onboard USB
- Up to 2 slots for modular interface expansions
- 128 to 256 MB DDR3 SDRAM
- 256 MB to 2 GB onboard flash drive
- Fanless with no batteries
- Width
    - Without fieldbus slot: 37.5
    - 1 fieldbus slot: 62.5 mm
    - 2 fieldbus slots: 87.5 mm

Fieldbus CPUs

Fieldbus CPUs are variants of Compact CPUs. In addition to these features, fieldbus modules can be connected to the left side. These CPUs make applications possible in which data preprocessing must take place remotely on the I/O bus interface.

- Embedded µP 16 / µP 25 with additional I/O processor
- 100/750 kB User SRAM
- 1/3 MB User FlashPROM
- X20XC0292: Onboard Ethernet
- Up to 2 slots for fieldbus modules
- No battery
- Width
    - 1 fieldbus slot: 62.5 mm
    - 2 fieldbus slots: 87.5 mm

2.3.5 For all fieldbuses, integration through standardization

The X20 system is ideally suited for expanding existing control systems using standard fieldbus technology.

For example, a bus controller allows the X20 system to be used as a powerful I/O expansion unit. Standardized EDS or GSD description files allow X20 system components to be integrated, configured, and programmed in the programming environment of a non-B&R system.

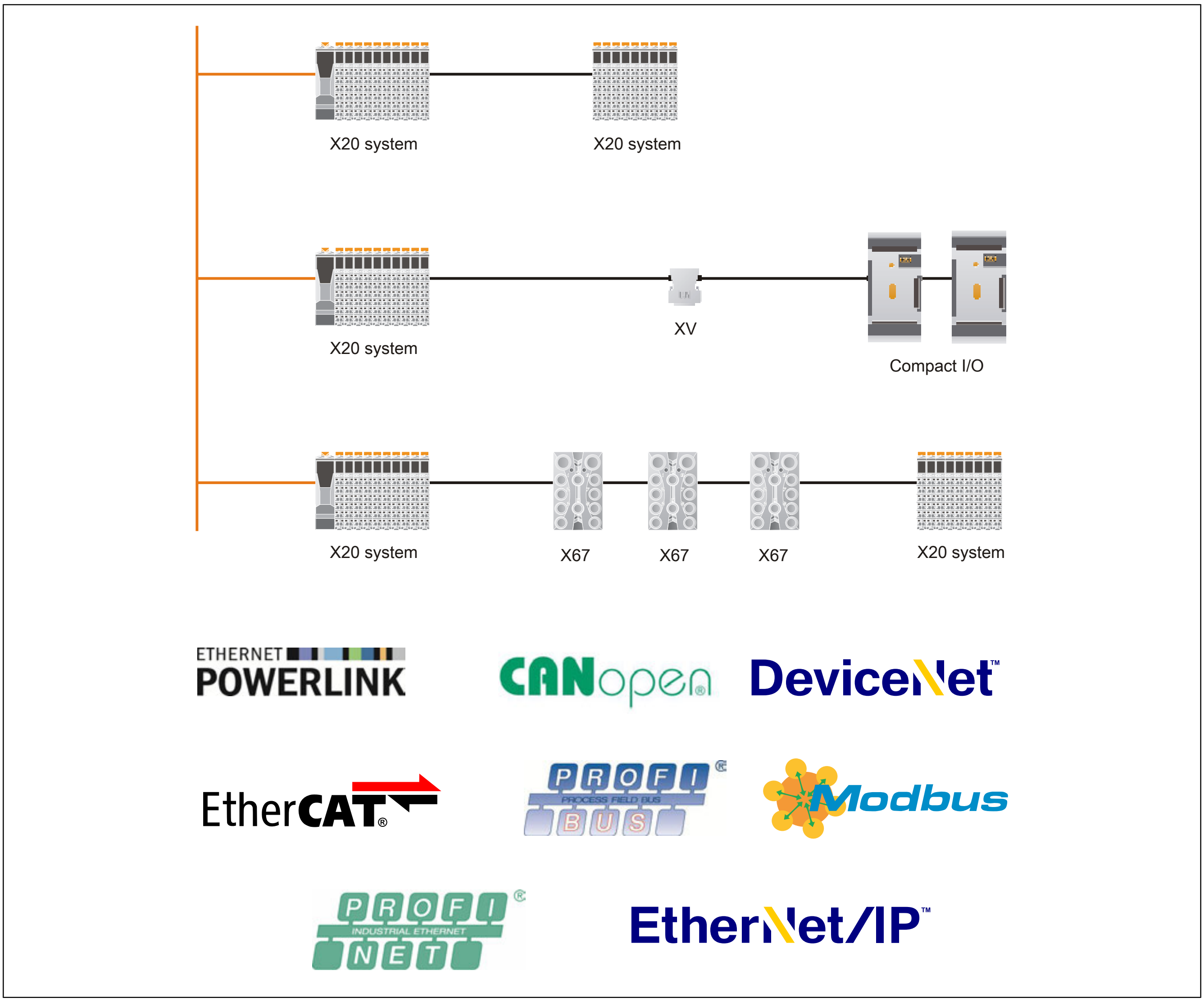


Figure 8: Expansion of existing control systems using standard fieldbuses and the X20 system

2.3.6 Complete system

2.3.6.1 IP67 - then X67

The X67 is the robust version of the X20 for use outside the control cabinet. The same basic technology, with an extremely robust housing and 4 to 32 channel modules, guarantees economical solutions in the roughest conditions.

2.3.6.2 Integrated valve terminal control

The development of the XV system allows for the first time direct and manufacturer-independent control of valve terminals. A complete digital output module in a size and form comparable with a normal DSUB connector. XV allows any valve terminal manufacturer to be selected because it is connected directly to the standardized multiple pin connector on the valve terminal.

Fully integrated in the remote backplane, it rounds off the X20 and X67 for complete automation solutions. One system, several variations - advantages that pay off. You select your automation components and distribute them as needed inside and outside the control cabinet.

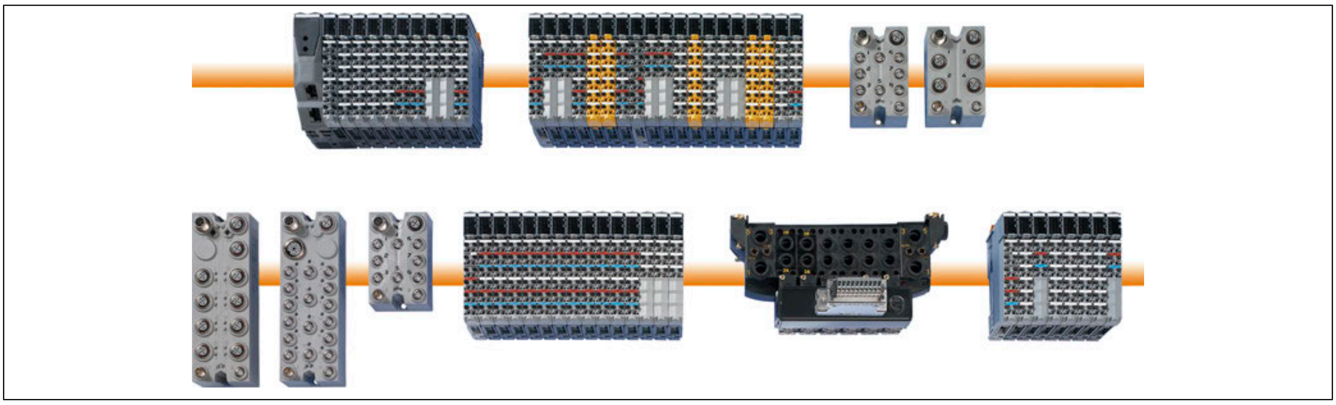


Figure 9: X20, X67, XV - variations of a single system

2.3.7 Easy wiring

Industrial control cabinet construction streamlines production cycles. Prefabricated cable trees enable faster and easier assembly directly on the machine or system. The X20 system supports efficient prewiring of the entire control cabinet using separate terminal blocks. The complete X20 system configuration is mounted in the control cabinet and connected to the prewired cable trees.

The supply of the X20 modules and the supply of the sensors and actuators do not add any requirements for energy distribution. The X20 system reduces manual wiring to a minimum.

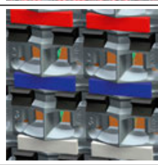
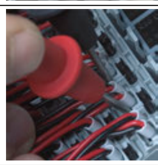
2.3.7.1 Install the wires, plug it in, and it's ready to go

Simple, tool-free wiring for fast installation. The X20 system terminal blocks use a fully integrated and proven push-in connector system. Each terminal can also handle double wire sleeves up to a diameter of 2x 0.75 mm². The user saves time wiring the system multiple times and distributing the signals.

The wire connections can be removed with a screwdriver. Each terminal also has an access point for a measurement probe.

Important!

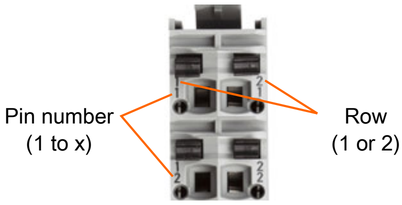
To avoid damaging the terminals, the X20AC0SD1 B&R screwdriver should be used.

	**Detached** The terminals can be prewired apart from the actual I/O module. This provides many advantages for control cabinet construction. Separate manufacturing, just-in-time logistics and the installation of preassembled systems during start-up become reality.		**Tool-free** Simple, tool-free wiring for fast installation. The X20 system terminals use a fully integrated and proven push-in connector system. Available with 6-pin and extremely compact 12-pin terminals.
	**Coded in the system** Factory coding prevents dangerous mix-ups. Coding guarantees that only parts which are permitted to be combined can be combined. Intuitively and without additional work.		**Ergonomic** Component density does not have to negatively affect ergonomics. With terminal spacing of more than 5 mm, this was handled optimally on the X20 system. Experience gained in the field - used in the field.
	**Coded in the application** Incorrectly inserting terminals does not necessarily damage the electronics, but always causes faulty functioning of the system. Application coding prevents this problem.		**Unmistakable** Distinct forms intuitively define various functions, such as clearly assigned latching and unlatching functions for terminals. This prevents errors from the very beginning.
	**Labeling** Each terminal is clearly labeled, directly in the plastic. Additional label tags are available as system accessories including a printer with ECAD connection.		**Easy servicing** A system's strengths can be seen in its details: In addition to the terminal connector and unlocking mechanism, each terminal has an access point for a test probe. You can easily measure the terminal potential without disconnecting the wire.

2.3.7.2 Unique terminal numbering

Each terminal connection is unique and can be identified by the numbers in the plastic. In this way, terminal assignments can be clearly assigned in the planning stage without any danger of mix-up.

- Upper number: Row number 1 or 2
- Lower number: Pin numbers 1 to 3 (6-pin terminal block), 1 to 6 (12-pin terminal block), 1 to 8 (16-pin terminal block)



2.3.8 Sophisticated mechanics

The name B&R stands for many years of experience in developing and manufacturing industrial electronics. But it's also the mechanics of the X20 system that have been thought through to the last detail. Its robust design, long guides and strengthened housing guarantee the stability it needs in industrial environments. These features allow the X20 system to be mounted on a top-hat rail with the same ease as a rack system. They also make it just as simple to remove it from the rail.

The sophisticated mechanics of the X20 are needed not just to provide this type of handling, but also to be able to quickly and easily remove I/O slices from the entire system.

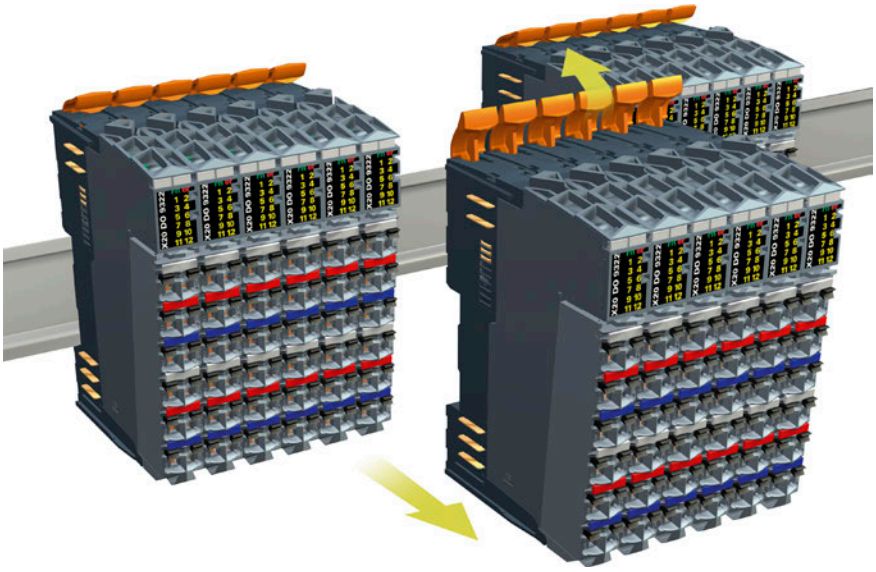
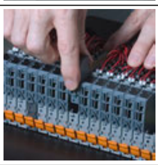


Figure 10: Easy mounting on and removal from the top-hat rail

	**Unlocking mechanism with two positions** Closed for secure fit on the top-hat rail.		**Defined open position makes the difference** Open to remove a module or the entire system.
	**Removing a single module from the system** Remove or reconnect vertically.		**Mount the entire system as a whole** Or just as easily removing the entire system.

2.3.9 Diagnostics

Outstanding diagnostic options are needed for errors to be found quickly. The X20 system offers several levels of diagnostics:

- Direct on the module using visual LED displays. Bus status, I/O status and channel states are displayed in direct relationship to the channels or the function. The different states are displayed in different ways, e.g. green for OK, red for error.
  For details, see "Diagnostic LEDs" on page 53.
- Via software in the cyclic data image. With the X20 system, status data does not result in an additional communication load, which would result in considerable differences between theoretically possible bus speeds and real requirements during operation. All necessary status data is always transferred cyclically, with no exceptions.
- Expanded diagnostic data in acyclic data traffic without loss in performance. If a problem occurs, detailed diagnostic data can be requested from the application by the respective module using an asynchronous channel. This does not result in additional communication load and cycle times remain unchanged.

Figure 11: Visual diagnostics directly on the module using LED indicators

2.3.9.1 Diagnostic LEDs

LEDs for diagnostics are located at the top of most X20 I/O modules. The following LEDs are available depending on the module to indicate the operating state:

- LEDs "r" (green) and "e" (red)
- LED "s" (red/green)

Additional LEDs are module-specific and usually indicate the state of I/O channels. Green LEDs are usually used for inputs, while orange LEDs are usually used for outputs. These I/O LED status indicators are only operational in mode RUN on some modules.

Operating states and error states

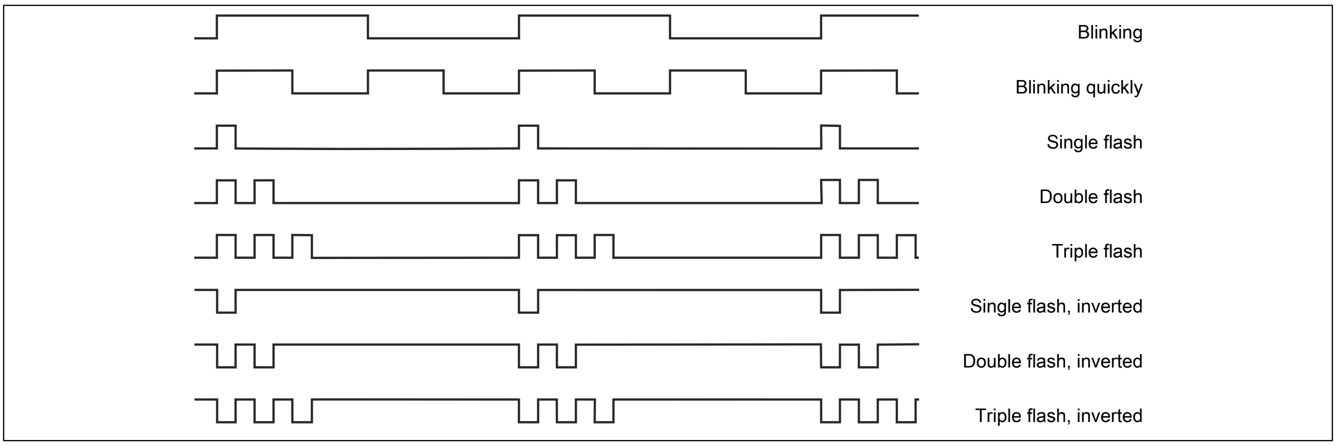
The following table provides a complete description of all operating states and error states for X20 I/O modules. The operating state and error state actually indicated by the I/O module depends on the type of module as well as how it is being used.

	LED	Description	Note
	All LEDs off	No power to module	The module does not have power.
	Module status: Green LED		
	Single flash (red LED = Off)	Mode RESET	No connection to the X2X Link master, or the X2X Link master is not yet running. Some modules remain in single flash mode during a firmware update.
		Not configured	The module was connected behind bus receiver X20BR7300, but it is not configured.[2]
	Single flash (red LED = On)	Invalid firmware	Invalid firmware: Occurs when a firmware update has been interrupted. The firmware is reloaded as soon as the X2X Link master is active again. It is only loaded if the module is also entered in the configuration, however.
	Double flash	BOOT mode (mode RESET with communication)	Firmware update. A firmware update usually only takes place once after the module has been replaced or if new firmware has been loaded to the master CPU during a project update. **Depending on the configuration, a firmware update can take several minutes.**
		Not configured	The module was connected behind bus receiver X20BR7300, but it is not configured.[2]
	Blinking	Mode PREOPERATIONAL	Modules whose slot is configured for a different module (or none at all) remain in mode PREOPERATIONAL. Possible errors: • Incorrect module connected or slot not configured • Incorrect slot number for bus modules with node number switches
	Blinking quickly	Mode SYNC	Module synchronizing with X2X Link network
	On	Mode RUN	No error
	Error status: Red LED (green LED = On)		
	Off		Everything OK
	On	Fatal error	It is not possible for the module to continue functioning correctly. Possible errors: • Power supply outside warning range • Operating temperature outside permissible range **Monitoring for fatal errors is not integrated into all modules.**
	Single flash or blinking	I/O channel error	An error or warning is present on one or more I/O channels. Which channel error on the module is being indicated depends on the module and can be determined with the respective module description.
	Double flash	System errors	A system error occurred in the module. The cause of error depends on the module and can be determined with the respective module description.
	Triple flash	I/O error and system error	An I/O error and system error occurred at the same time.
	Single flash, inverted[1]	Fatal error and I/O error	A fatal error and I/O error occurred at the same time.
	Double flash, inverted[1]	Fatal error and system error	A fatal error and system error occurred at the same time.
	Triple flash, inverted[1]	Fatal error, I/O error and system error	A fatal error, I/O error and system error occurred at the same time.

1) Only on modules that monitor for fatal errors.
2) The blinking behavior (single or double flash) depends on the X20 module being used.

LED status indicators - Blinking patterns

The blinking patterns shown in this image specify only the principle ratio between the switch-on and switch-off time for the LED. The actual ratio of blink times to each other can vary depending on the module.



2.3.10 Embedded parameter chip

Information such as module type, serial number, functionality and version number is contained in the embedded parameter chip of the X20 module. This information is automatically evaluated by the programming environment (Automation Studio) and by the application program. This prevents errors during both commissioning and service. In addition, the system configuration is automated and flexible variations are made possible.

For validated systems, it is becoming increasingly important to have module serial numbers that are unique worldwide – as required by the FDA, for example.

Information:

All modules that require 0.01 W of power on the X2X Link network must be supplied via the internal I/O power supply. If the I/O power supply fails, the module shuts down and communication is lost.
In this case, ModuleOk returns the value "False" and data can no longer be read from the "embedded parameter chip".

2.3.11 Space for options

The X20 system makes it possible to combine the exact components necessary depending on the user's demands and individual application requirements. This allows machine options to be implemented easily and flexibly. Bus modules provide the base, and are more or less a rack replacement. Depending on the option, the necessary electronics modules are then inserted in the predefined slots.

Addresses are assigned implicitly via the slot. Software that has been developed once is valid for all versions and does not need to be changed. This is even possible for later machine expansion. The I/O modules are simply inserted in the defined bus modules, and assigned to the corresponding potential groups and E-stop groups.

To prevent unwanted expansion, each module can be identified and then enabled using the application software.

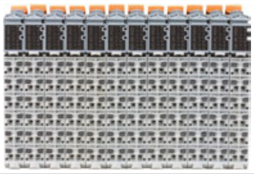
2.3.12 Flexibility for options

The implementation of different machine variations using free bus modules is only one of the many features that the X20 system offers. With the support of Automation Studio, there is an optimized solution using I/O mapping.

In the process, each I/O configuration is created optimally according to the actual requirements. However, the application software is designed to handle all potential options. Only the I/O channels that are actually available are mapped to the application program. If an expansion is required, then the additional hardware needed can be easily connected and the I/O mapping changed. This is possible without having to compile the application software.

It doesn't matter where the I/O mapping list is created:

- Manually in B&R Automation Studio
- With tools, e.g. with a database or a table calculation program
- Directly from an ERP system, just like the parts list for the machine
- Automatically in the application software, depending on the hardware used

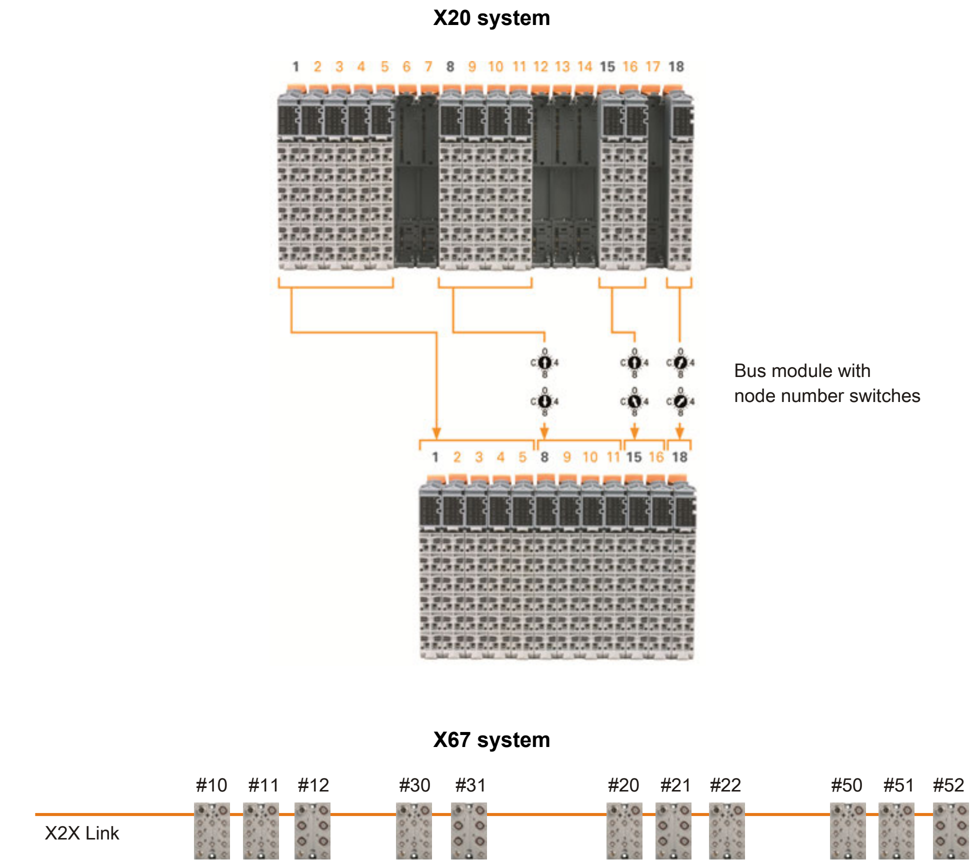
	**Machine variation A** The possibilities of the X20 system can be best explained using examples. This is a machine constellation with two variations, A and B. All of the necessary electronics modules for machine variation A are shown in the picture to the left. The bus modules needed for variation B are also present, but without electronic modules.
	**Machine variation B** Variation B shows the necessary electronic modules but the modules necessary for variation A are missing. The distribution of the free bus modules for the variations is also clear: The variable I/O modules can be very easily connected to the required electrically isolated groups and don't need to be attached in the back. The extensive process of taking apart the configuration to expand existing electrically isolated groups is also eliminated. Simply insert the electronic module and attach the terminal block.
	**Machine variation A - optimized** The features included in Automation Studio can also be used to achieve completely optimized hardware configurations without losing the advantage of comprehensive application software for all variations. As described earlier, simply mapping physical I/O points to the application program makes it extremely easy to optimize the hardware variations without even requiring compilation.

2.3.13 Configurable X2X Link address

The remote X2X Link backplane, which connects the individual I/O modules with each other, is set up to be self-addressing. Because of this, it is not necessary to set the node numbers. The module address is assigned according to its position in the X2X Link line.

In certain cases, e.g. when configurations of modular machines change, it is necessary to define specific module groups at a fixed address, regardless of the preceding modules in the line.

For this purpose, there are modules in both the X20 system and the X67 system with node number switches that allow you to set the X2X Link address. All subsequent modules refer to this offset and use it automatically for addressing purposes.

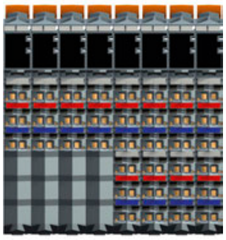
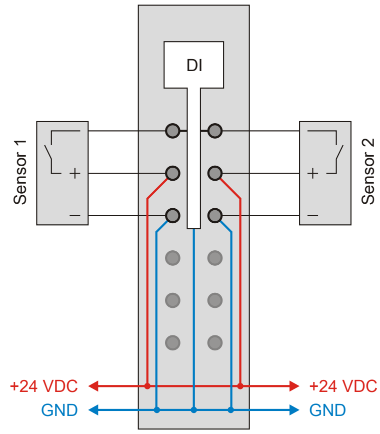
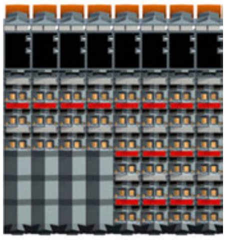
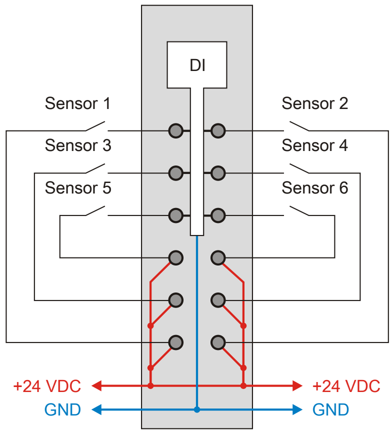
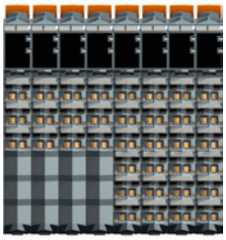
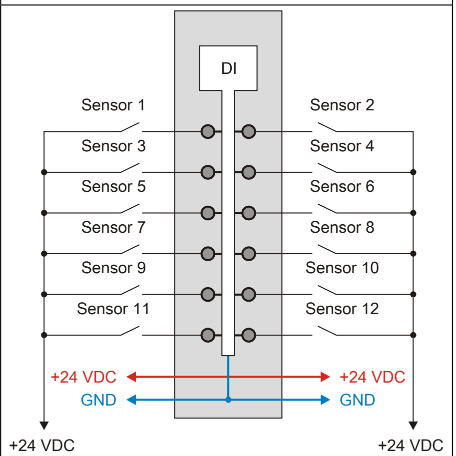


2.3.13.1 Bus modules with node number switches

Symbols are printed on the locking lever of bus modules with node number switches. This provides a way to see from outside that the X20 system mounted in this slot is using node number switches.

2.3.14 Universal 1, 2, 3-wire connections

Consistent connection types for all requirements – no additional jumper terminals are needed. All connection types can also be mixed and matched.

System-wide 3-wire connections Integrated supply and ground for sensors and actuators.		
System-wide 2-wire connections No extra terminals needed.		
System-wide 1-wire connections 12 channels - unequaled component density		

2.3.15 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

Differences between coated and uncoated modules

- Suitable for operation in adverse atmospheric environments
- Suitable for operation in 100% humidity, condensing
- Coated modules have a different Module ID than the corresponding uncoated variants

2.3.16 Redundancy

The X20 system provides the following forms of redundancy:

- Controller
- Network
- Power supply modules for X20 standalone devices and expandable POWERLINK bus controllers
- X2X Link supply

The first 3 areas are covered in user's manual "Redundancy for control systems". This user's manual is available in the Downloads section under www.br-automation.com.

For a description of the redundant X2X Link power supply, see section "X2X Link supply" on page 82.

2.3.17 reACTION Technology

This module is equipped with ultrafast reACTION Technology. This allows the I/O channels integrated in the reACTION module to be controlled with cycle times down to 100 µs. In particular, this new technology allows time-critical subprocesses to be managed using standard hardware, which lowers hardware costs by reducing the load on the controller and allowing it to be scaled down accordingly.

All commands that can be used for reACTION programs are available as function blocks in special libraries (e.g. AsIORTI). Programming in compliance with IEC 61131-3 requirements takes place in the Function Block Diagram editor in Automation Studio.

reACTION TECHNOLOGY

2.3.18 X20 system configuration

The X20 system is designed so that it can be connected to standard fieldbuses (with a bus controller) or the remote X2X Link backplane (with a bus receiver). The connection to the next station is made with a bus transmitter. Power supply modules and I/O modules are placed between the bus receiver or bus controller and the bus transmitter as needed.

The power supply concept used by the X20 system is described in section 2.4.8 "Supply concept".

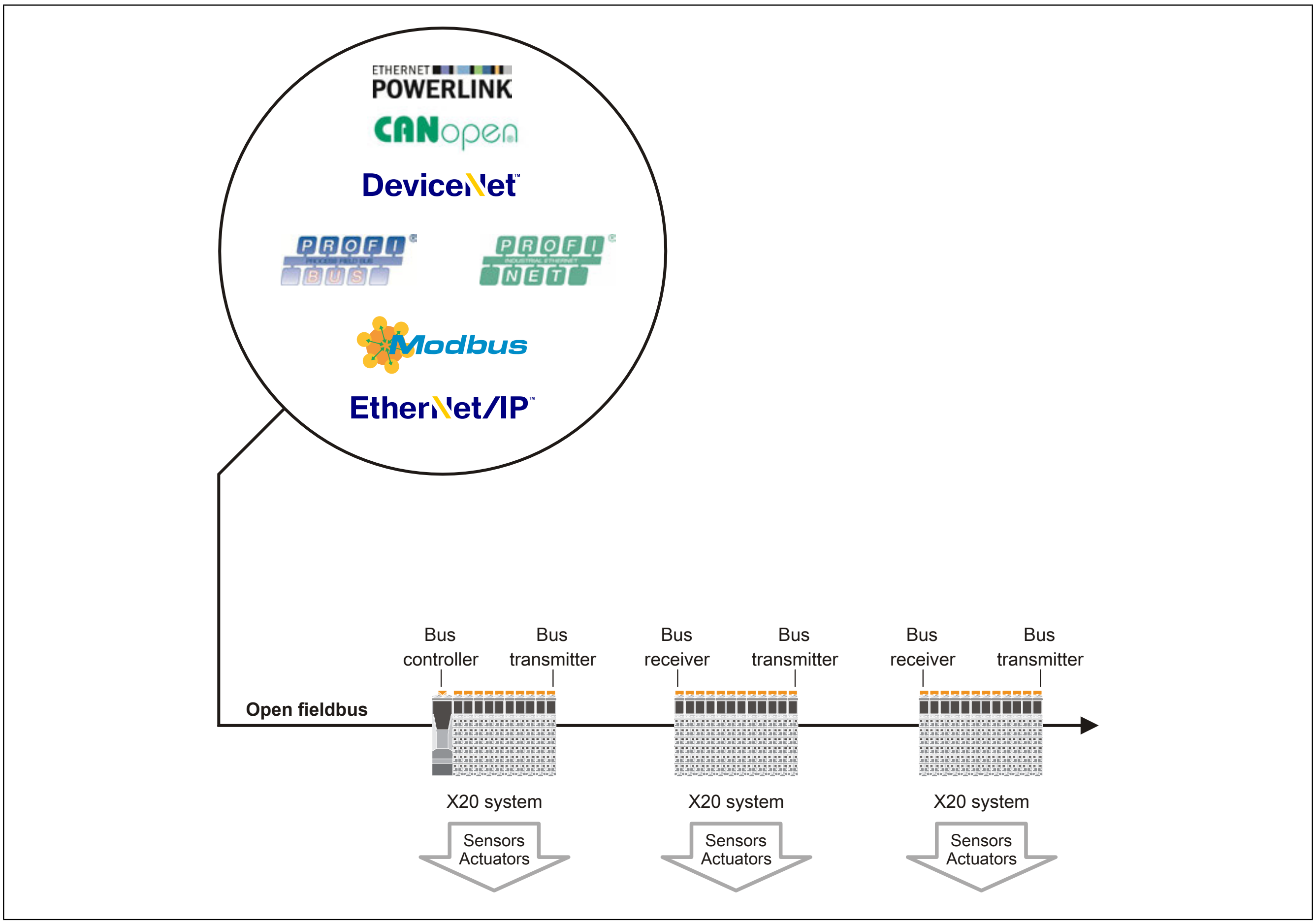


Figure 12: X20 system configuration

2.3.18.1 Fieldbus connection

Several bus controllers for standard fieldbus technologies like POWERLINK, DeviceNet, PROFIBUS, CANopen, ModbusTCP or EtherNet/IP are available to connect X20 modules to existing control systems. Fieldbus configurators transparently integrate the X20 system into the 3rd-party development environment.

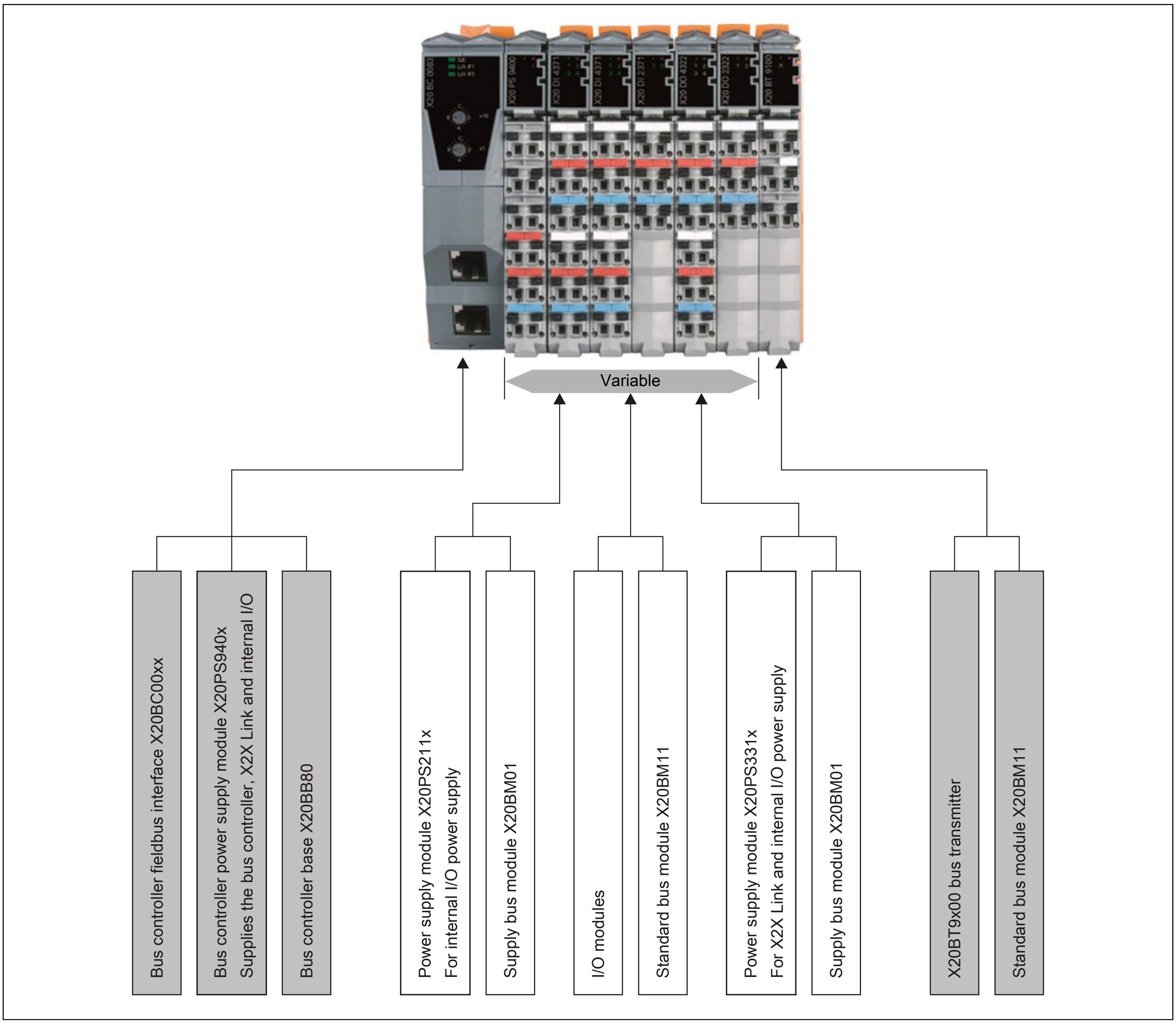


Figure 13: X20 system configurator for fieldbus connection

2.3.18.2 Connection to X2X Link backplane

The bus receiver X20BR9300 is used to connect the X20 system directly to the remote X2X Link backplane.

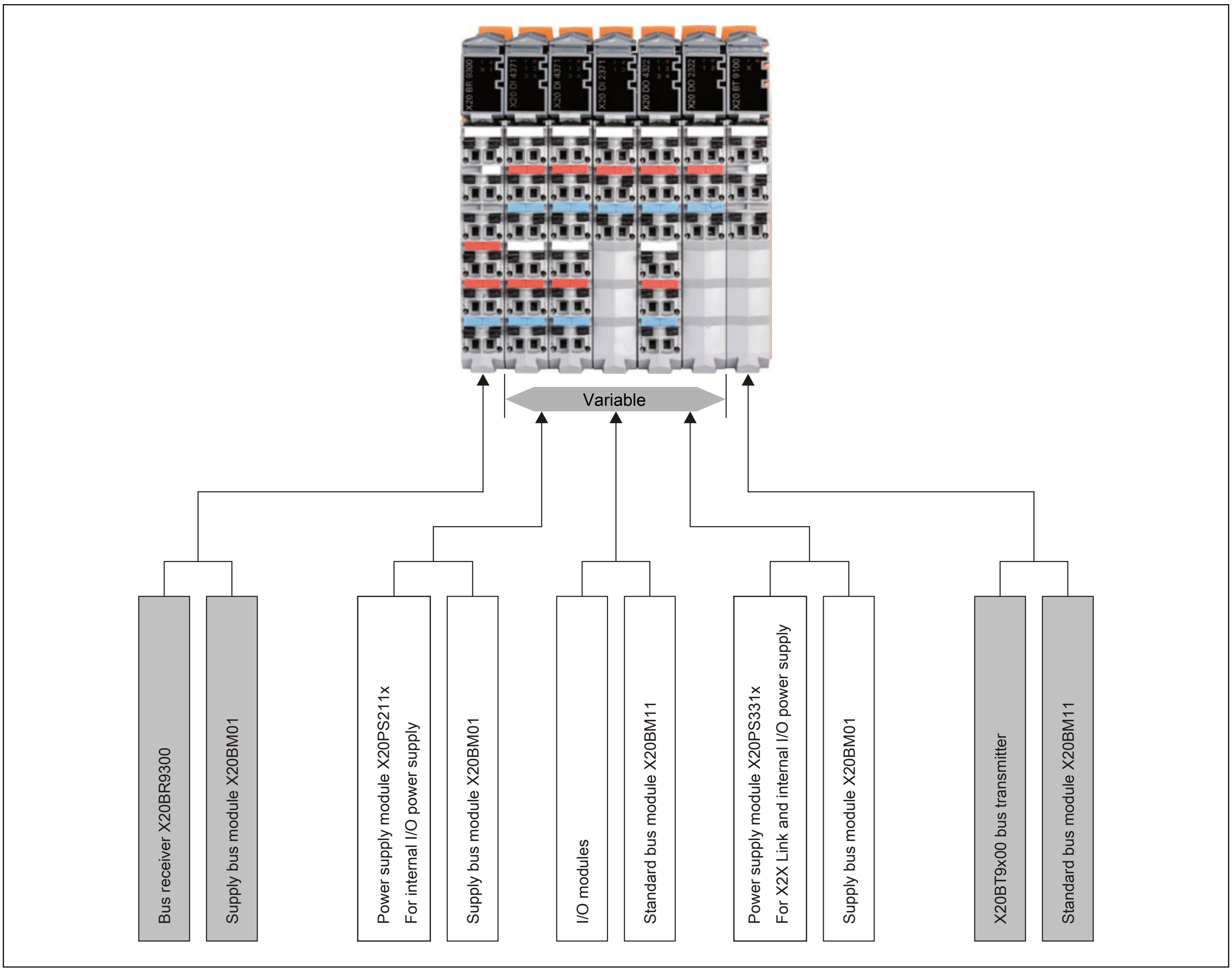


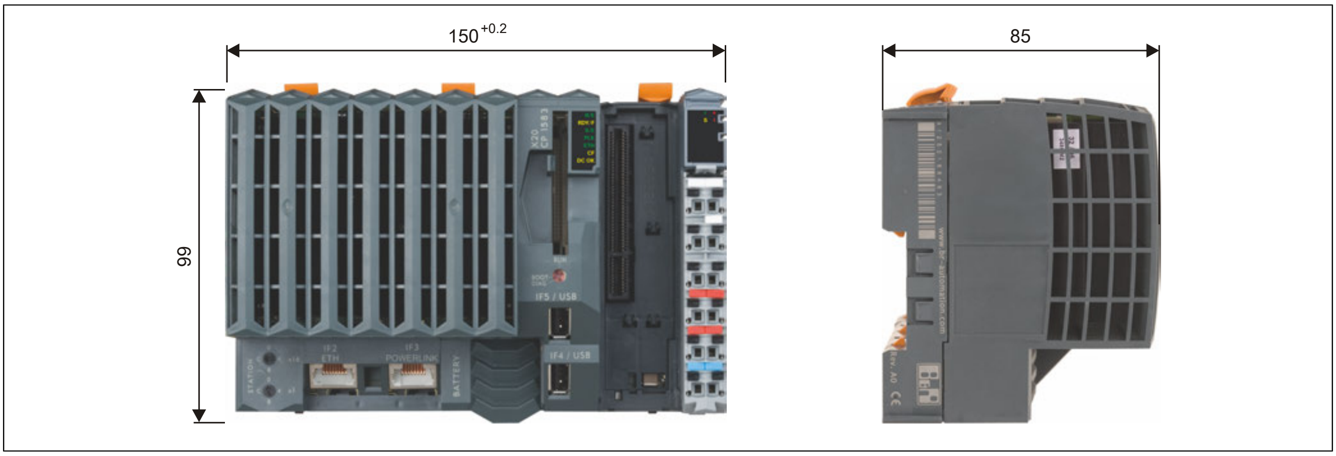
Figure 14: X20 system configurator for connection to X2X Link backplane

2.4 Mechanical and electrical configuration

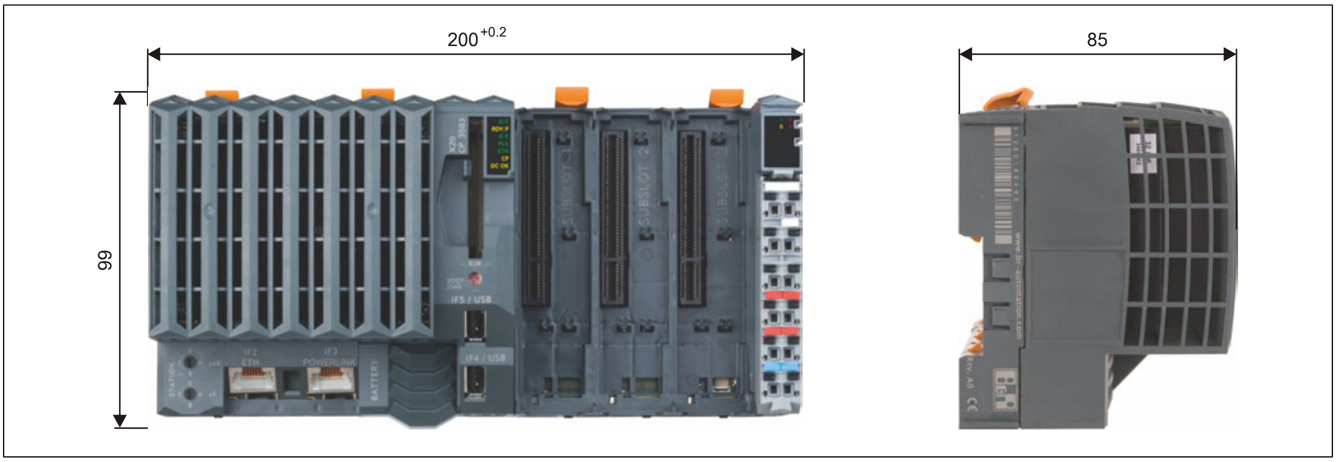
2.4.1 Dimensions

2.4.1.1 X20 CPUs

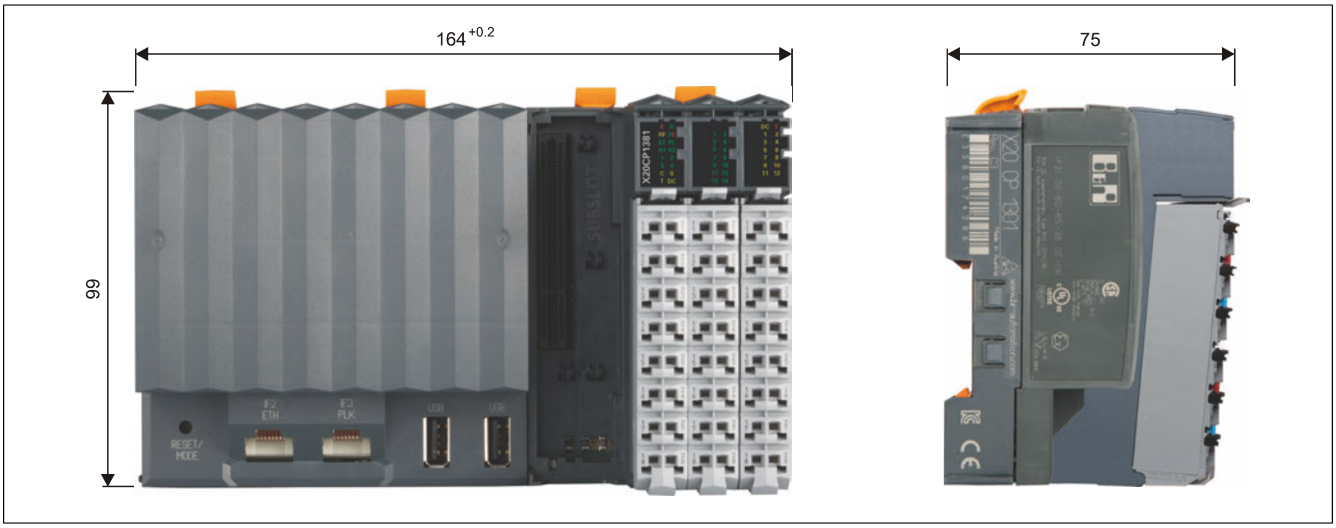
CPUs with 1 slot for interface modules



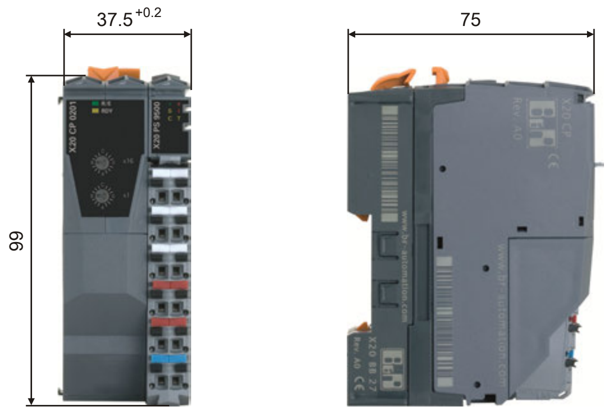
CPUs with 3 slots for interface modules



2.4.1.2 X20 CPUs with integrated I/O

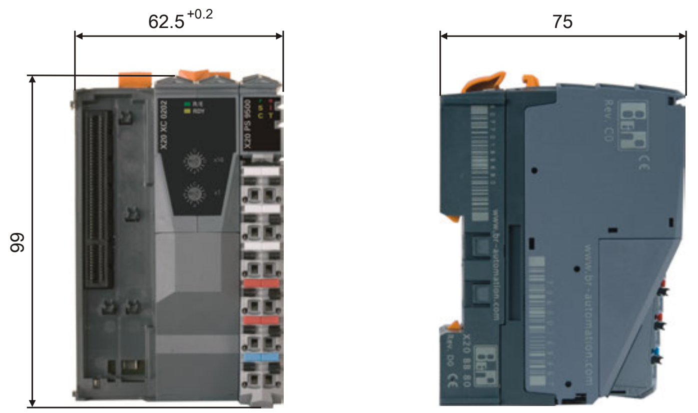


2.4.1.3 Compact/Compact-S CPUs and bus controllers

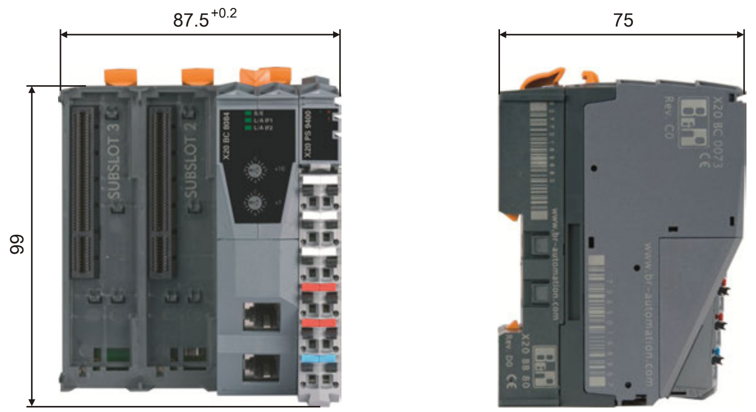


2.4.1.4 Fieldbus CPUs and expandable bus controller

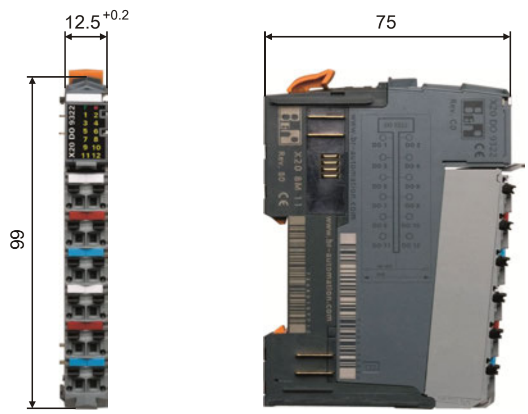
With 1 additional slot



With 2 additional slots



2.4.1.5 I/O modules



2.4.1.6 End cover plates

In addition to the dimensions for CPUs and modules specified in this section, it is possible to add end cover plates on the left and right sides of each module block. The following space must be provided for this:

- **Right side:** 5 mm
- **Left side:** 3.5 mm

2.4.2 Design support

2.4.2.1 CAD support

To ensure CAD support, the dimensions are included in the ECAD macros in 2D. STEP data is available to allow 3D viewing.

The STEP data can be downloaded from the B&R website (www.br-automation.com) in the Downloads section for the respective module.

2.4.2.2 Macros for ECAD systems

The electronics in a machine must be designed in a way that optimizes use of available space and materials. Graphic ECAD systems have proven themselves as the right tool for this job.

Every module in the X20 system is delivered with pre-designed electronic descriptions of the mechanical dimensions, electrical signals and module functions. These macros can be loaded directly to proven ECAD systems. The wiring plans are automatically applied by the configuration and programming system, Automation Studio. Design and changes are immediately reflected at all levels of development. This saves time for the more important tasks and prevents errors right from the start. The accelerated development, programming, maintenance and documentation involved with the X20 system mean lower costs, enhanced quality and increased sales by earlier entry into the market.

2.4.2.3 Printing support

System printers and standard identification labels are supported by the appropriate printer software. Printing can be done manually from table calculations or directly from ECAD software (all methods are supported). The software and printer systems correspond with the Weidmüller standard.

2.4.3 Installation

A top-hat rail conforming to the EN 60715 standard (TH35-7.5) is required to mount the PLC. The conductive top-hat rail is fastened to the back wall of the control cabinet.

The complete system including all individual modules is hung in the desired location on the top-hat rail with the unlocking mechanisms open and locked in place by closing the unlocking mechanisms. Finally, the modules are equipped with the prewired terminal blocks.

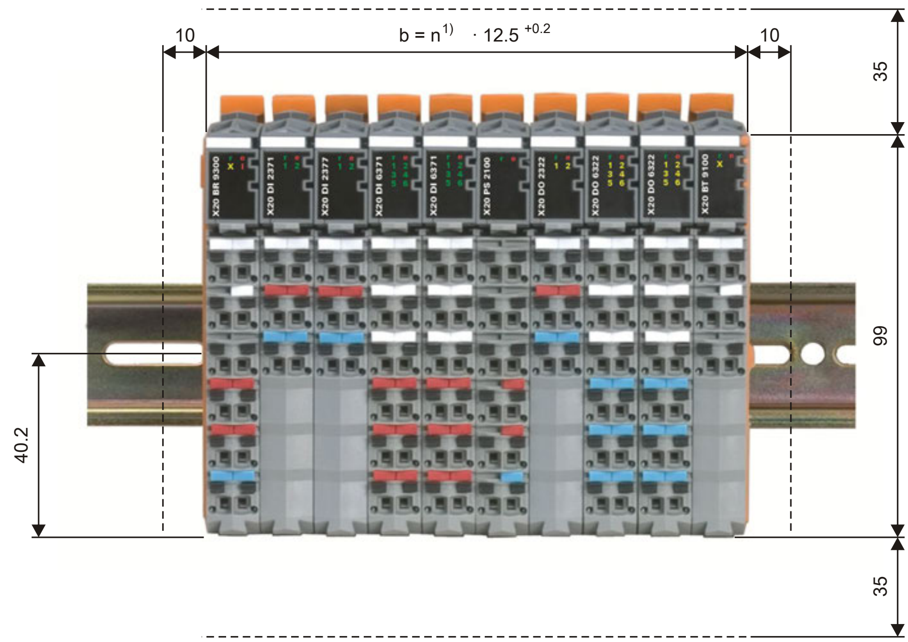
Installation methods

- Vertical installation
- Horizontal installation
- Oblique installation
- Face-up installation

Information:

Other installation methods are not permitted.

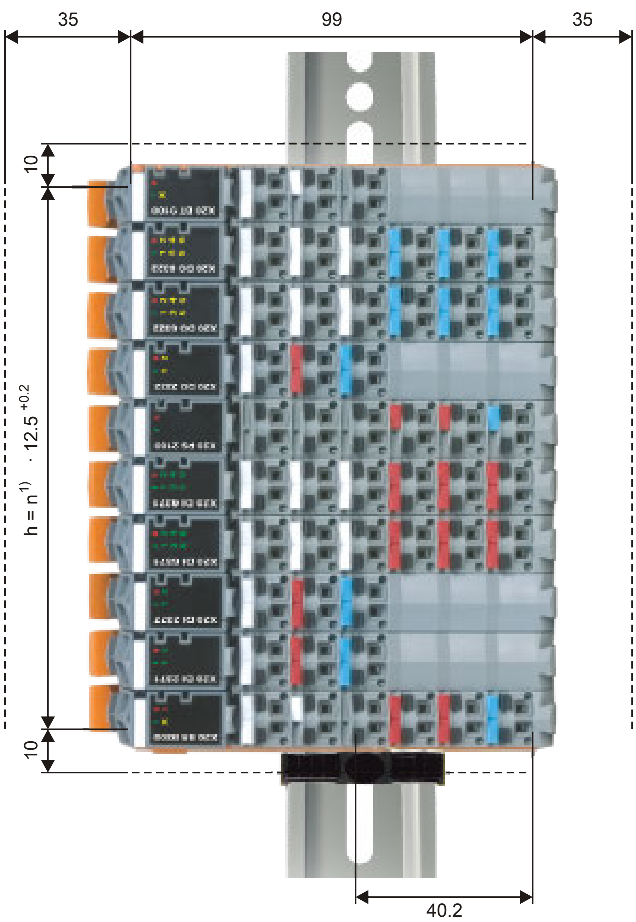
2.4.3.1 Horizontal installation



1) n ... Number of modules

For optimal cooling and air circulation, there must be at least 35 mm free space above the modules. To the left and right of the X20 system, there must be at least 10 mm of free space. Underneath the modules, 35 mm space must be left free for I/O and power supply cabling.

2.4.3.2 Vertical installation



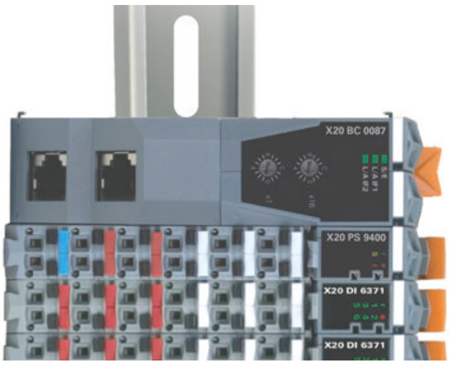
[1)] n ... Number of modules

For optimal cooling and air circulation, there must be at least 35 mm free space to the left of the modules. Above and below the X20 system, there must be at least 10 mm of free space. To the right of the modules, 35 mm space must be left free for I/O and power supply cabling.

The modules must be arranged so that the controller is on the lower end of the system. The temperature range is limited to -25 to 50°C when modules are mounted vertically.

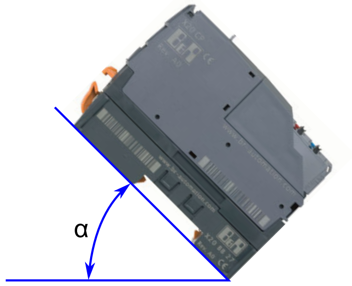
Information:

The controller must be secured against slipping with an end clamp.



When using an overhead bus controller or a CPU, an additional derating of 5°C based on the vertical installation must be observed. The additional derating applies only to the bus controller, CPU and corresponding power supply unit.

2.4.3.3 Oblique installation



For oblique installation, the derating depends on angle α.

- α < 70°: An additional derating of 15°C based on the horizontal installation must be observed (corresponds to face-up installation).
- α > 70°: No additional derating (corresponds to horizontal installation)

2.4.3.4 Face-up installation

An additional derating of 15°C based on the horizontal installation must be observed when installing with the top-hat rail at the bottom.

2.4.3.5 Installation with increased vibration requirements (4 g)

The following additional measures are necessary to fulfill increased vibration requirements regardless of whether X20 modules are installed horizontally or vertically:

1. Apply foam tape along the entire length of the module configuration under the top edge.
2. Use special end clamps to the left and right to provide additional fixation (supplement foam tape as shown in the image).
3. For CPUs with exchangeable battery, foam tape must be applied to the inner side of the battery cover to help hold the battery in place.
4. If slots are free, use dummy housings to ensure that the controller is held effectively in place.
5. Proper strain relief on all wires

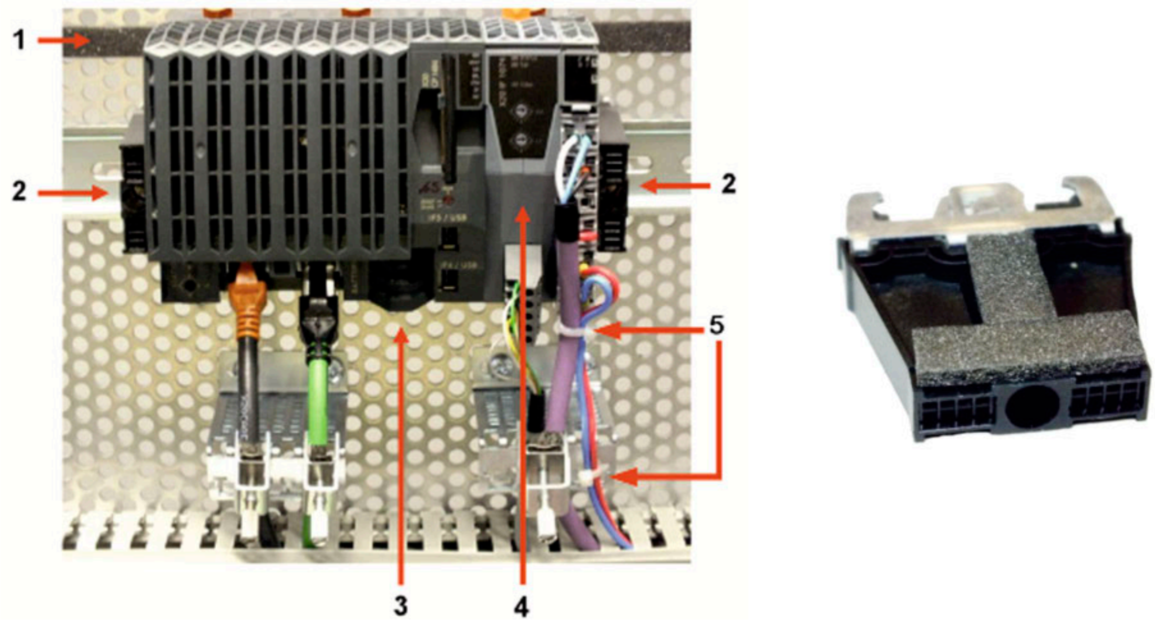


Figure 15: X20 system - Installation with increased vibration requirements (4 g) and end clamp

Important!

The attachable end cover plates for some X20 modules should be removed for "installation with increased vibration requirements (4 g)"!

Figure 16: X20 system - Removing the end cover plates

Required accessories

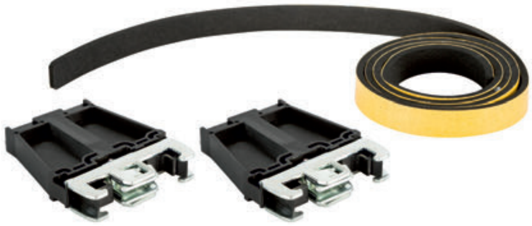
Description	Figure
1x Set X20AC0RF1 comprising 2x end clamps for top-hat rail TH35 (add foam tape) and 1x foam tape 12 x 3 x 1000 mm (height x width x length)	
Apply foam tape (L = 15 mm) to the battery cover on the CPUs.	
Use X20IF0000 dummy module in empty slots	

Table 9: Required accessories

2.4.4 Wiring

In order to achieve a secure connection in the terminal blocks, wires must be stripped accordingly.

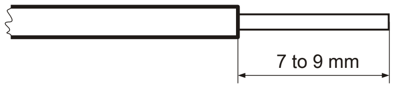


Figure 17: Wire stripping length for a secure connection

Information:

The wire stripping length is not permitted to be more or less than 7 to 9 mm.

2.4.5 Stress relief using cable ties

Figure 18: Stress relief using cable ties

The X20 system terminal blocks have slots for the cable ties. If needed, a cable tie can be fed through these slots to reduce the stress on the cable.

Cable tie dimensions:	Width ≤ 4.0 mm Thickness ≤ 1.2 mm

Figure 19: Slots through which the cable ties are fed

2.4.6 Shielding

In principle, the shield must be grounded in all shielded cables:

- Analog signals (inputs and outputs)
- Interface modules
- Counter modules
- X2X Link cables
- Fieldbus connections (PROFIBUS DP, CAN bus, etc.)

In general, the following guidelines apply for shielding:

- The X20 top-hat rail must always be mounted to a conductive backplane.
- Shielded cables must be grounded on both sides.

2.4.6.1 Direct shielding connection

The shield is twisted and connected to the bus module's ground connection using a cable lug (2.8 x 0.5 mm). The cable is additionally secured to the terminal block using a cable tie (stress relief).

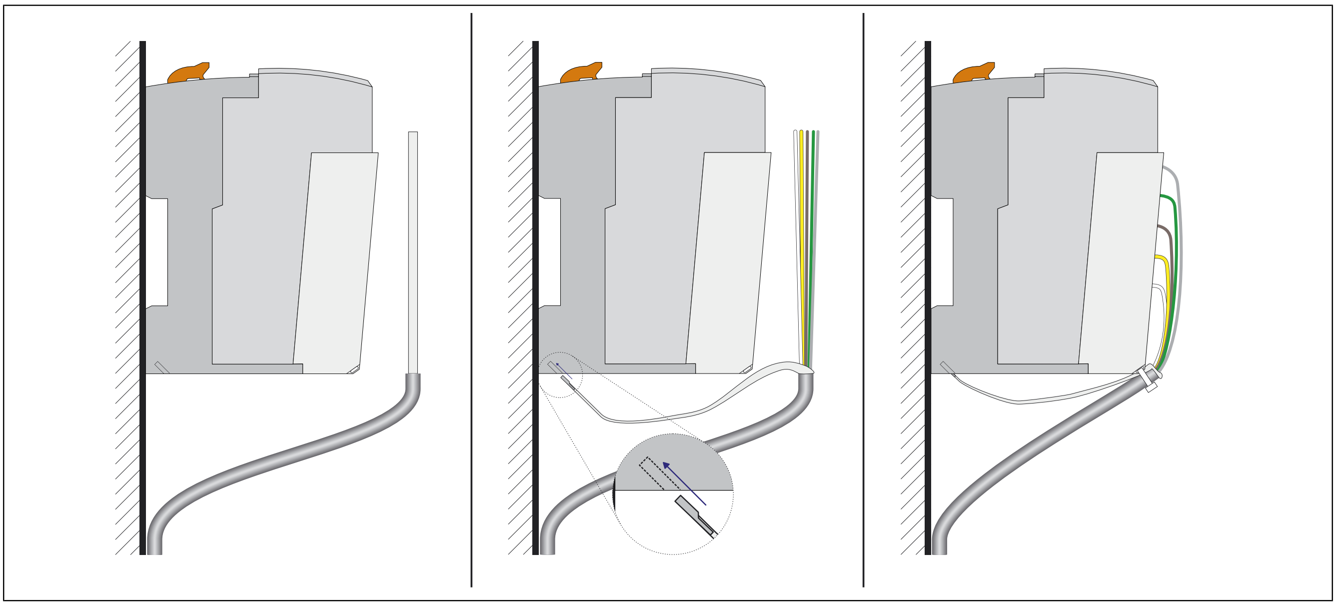


Figure 20: Direct shielding connection

Information:

The ground connection should be made as short and with as little resistance as possible.

2.4.6.2 X20 cable shield clamp

The X20 cable shield clamp (model number X20AC0SG1) is latched to the terminal block and connected to the bus module's ground connection using a cable lug. Cable ties are used to press the shield against the grounding plate.

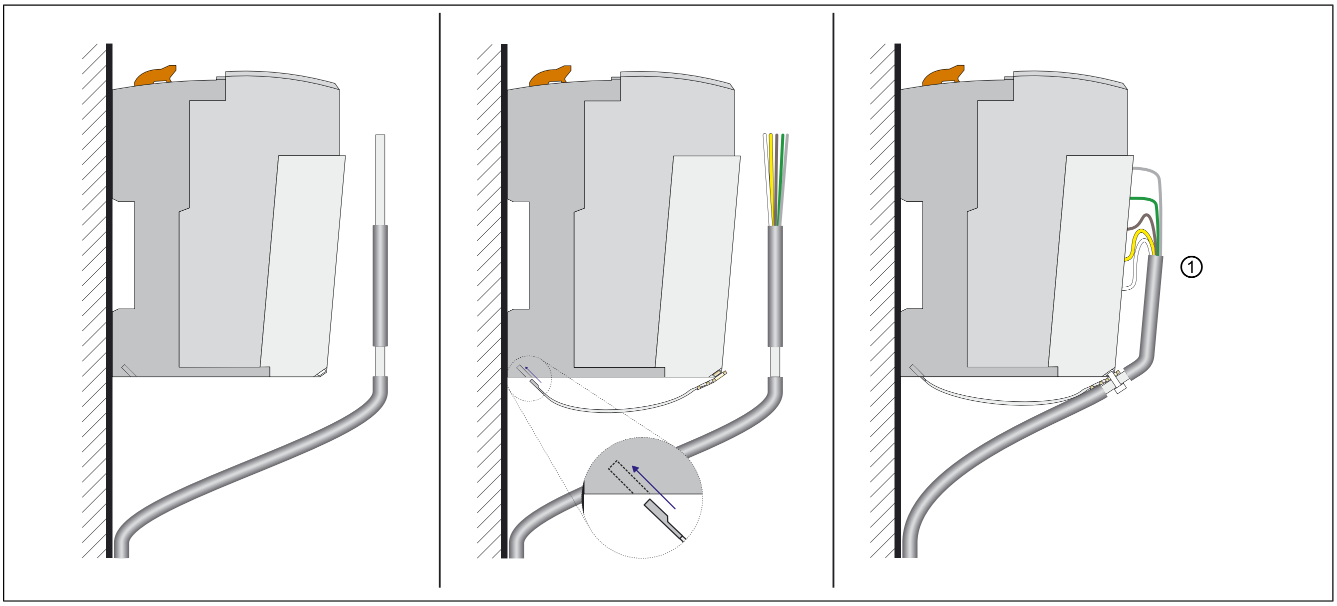


Figure 21: Shielding via X20 cable shield clamp

To reduce the EMC emissions most effectively, the cable shield must be as long as possible after the cable tie (see ① in the diagram above).

2.4.6.3 X20 shielding bracket

Order data

Model number	Short description
	Shielding bracket
X20AC0SF7.0010	X20 shielding bracket 66 mm 10 pcs.
X20AC0SF9.0010	X20 shielding bracket 88 mm 10 pcs.

Table 10: X20AC0SF7.0010, X20AC0SF9.0010 - Order data

The X20 shielding bracket is installed underneath the X20 system. The shield is pressed against the shielding bracket using ground terminals from another manufacturer (e.g. PHOENIX or WAGO) or a cable tie.

2 lengths are available depending on the application:

Model number	Length	Application
X20AC0SF7.0010	66 mm	• I/O modules • Power supply modules • Integrated I/O on X20CP13xx systems • Onboard interfaces on CPUs
X20AC0SF9.0010	88 mm	• Interface modules • Bus controller modules • CPUs constructed in the form of an interface module

Dimensions

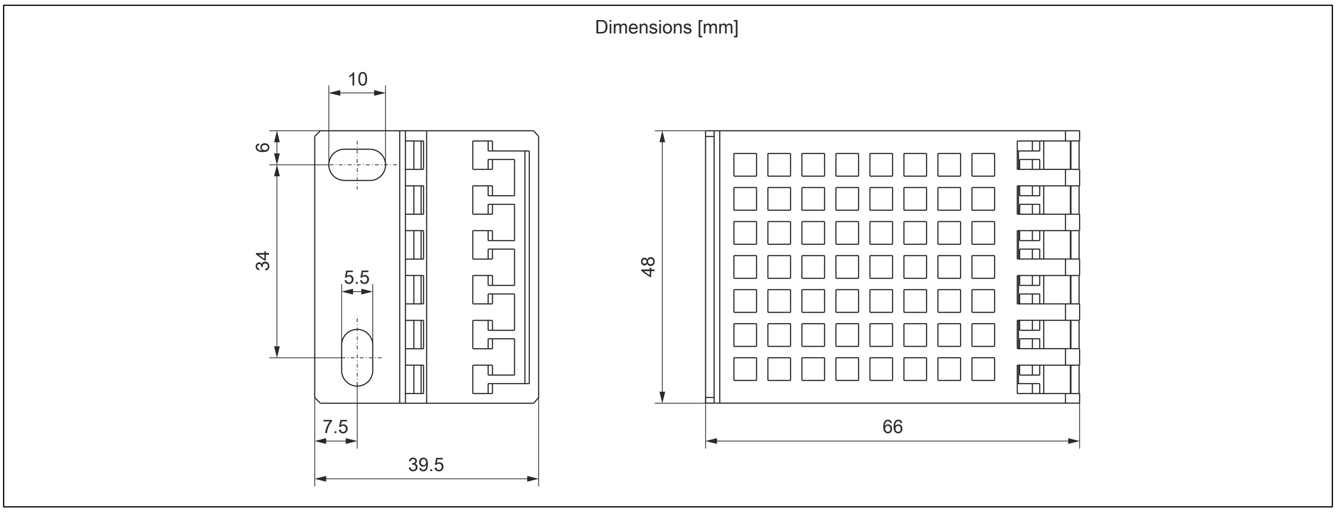


Figure 22: X20AC0SF7.0010 - Dimensions

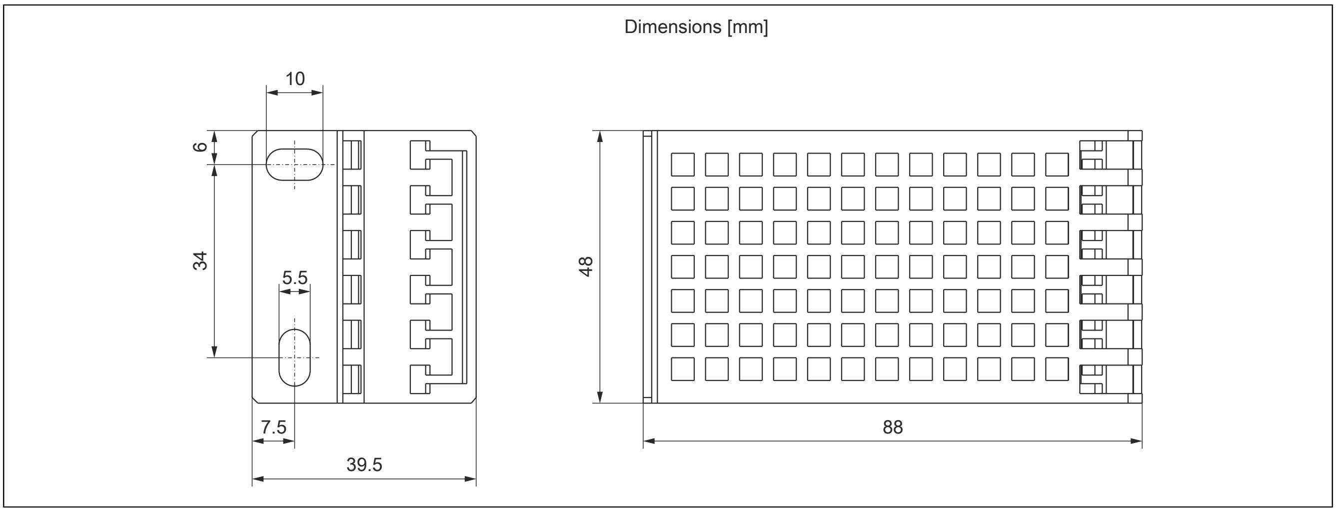


Figure 23: X20AC0SF9.0010 - Dimensions

Package contents

- 10 pcs. X20 shielding bracket
- Installation template

2.4.6.3.1 X20AC0SF7.0010 - 66 mm shielding bracket

Application example

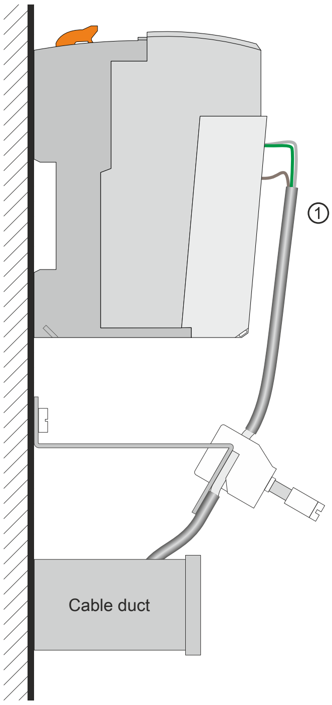
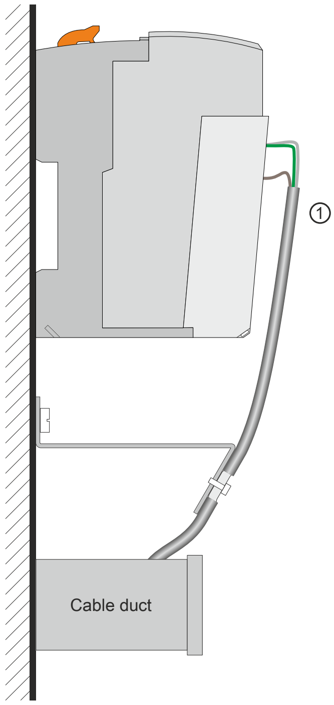
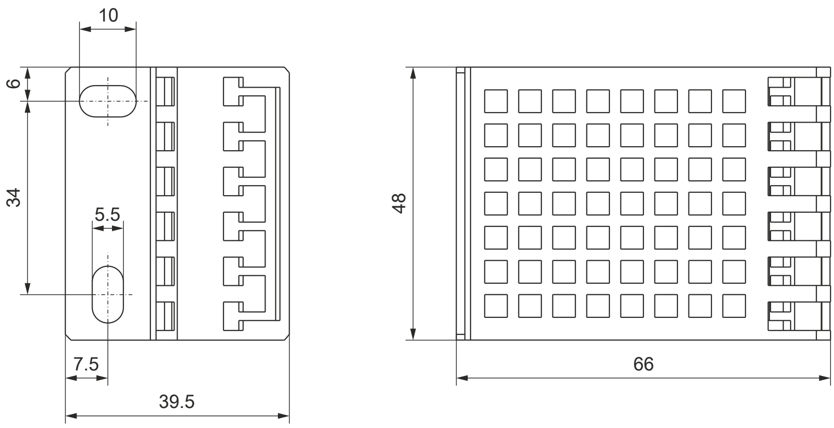
Attaching the shield with a ground terminal	Attaching the shield with a cable tie
	

Table 11: Cable shielding via X20 shielding bracket

To reduce EMC emissions as much as possible, the cable shield must reach as high as possible after attaching the cable to the shielding bracket (see ① in the figure above).

Dimensions

Dimensions [mm]



Content of delivery

- 10 X20 shielding brackets
- Installation template

2.4.6.3.2 X20AC0SF9.0010 - 88 mm shielding bracket

Application example

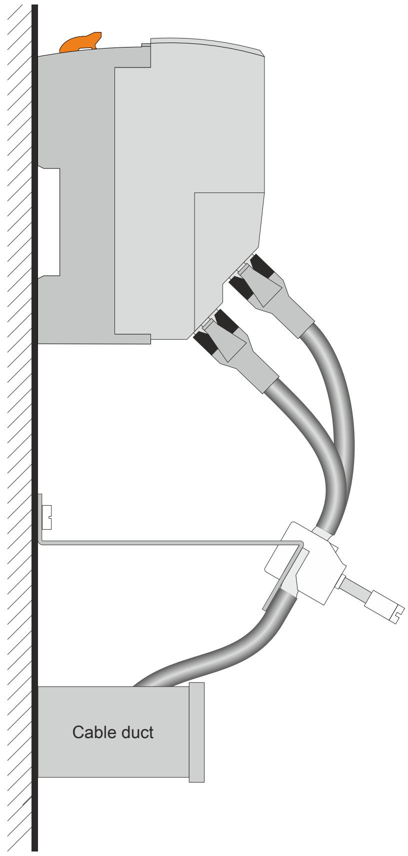
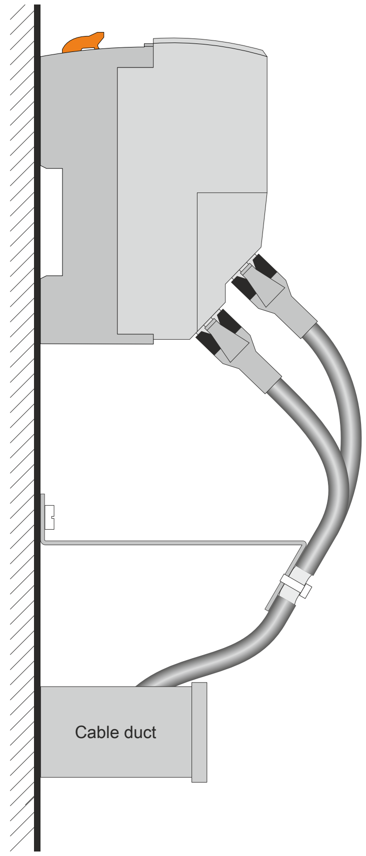
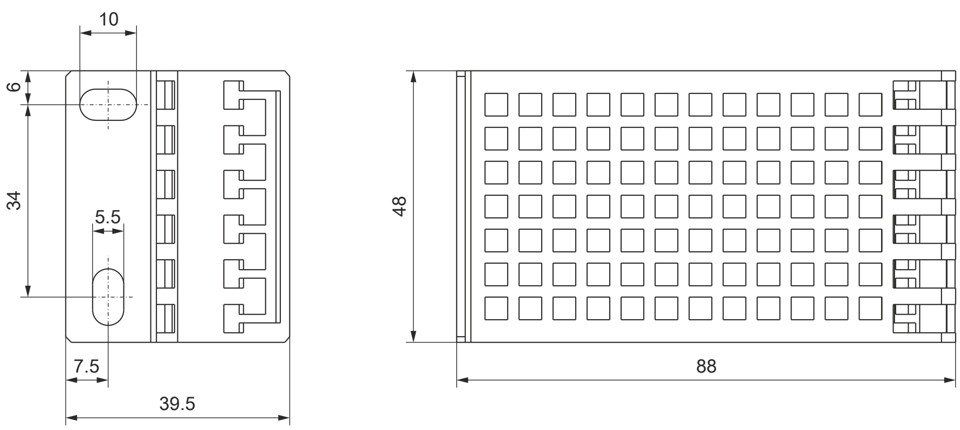
Attaching the shield with a ground terminal	Attaching the shield with a cable tie
	

Table 12: Cable shielding via X20 shielding bracket

Dimensions

Scale 3:4

Dimensions [mm]



Content of delivery

- 10 X20 shielding brackets
- Installation template

2.4.6.4 Shielding via top-hat rail or bus bar

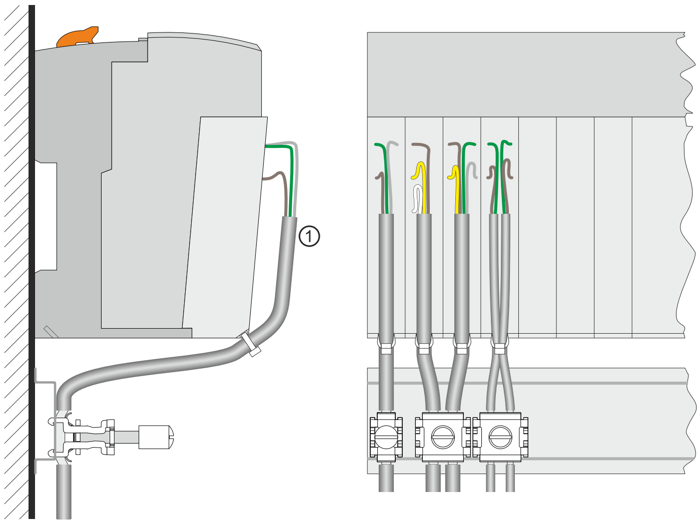


Figure 24: Shielding via top-hat rail or bus bar

Grounding terminals from other manufacturers (such as GOGATEC) can be used to achieve shielding right on the top-hat rail or on special bus bars directly below the controller.

- B&R recommends always using a grounding terminal via the top-hat rail to connect the X2X Link cable shield directly with the conductive and grounded backplane. This will generally exceed the specified EMC minimal requirements.
- The shielded cables from other modules can be grouped and clamped together. This may also be necessary due to space limitations. A different number of cables can be grounded together with a single terminal depending on the grounding terminals being used.

To reduce the EMC emissions most effectively, the cable shield must be as long as possible after the cable tie (see ① in the diagram above).

2.4.7 Cabling guidelines for X20 modules with an Ethernet cable

A number of X20 modules are based on Ethernet technology. POWERLINK cables offered by B&R can be used for the necessary wiring.

Model number	Connection type
X20CA0E61.xxxx	Connection cable - RJ45 to RJ45
X20CA3E61.xxxx	RJ45 to RJ45 connection cable, can be used in drag chains
X67CA0E41.xxxx	Attachment cable - RJ45 to M12
X67CA3E41.xxxx	RJ45 to M12 attachment cable, can be used in cable drag chains

The following cabling guidelines must be observed:

- Use CAT5 SFTP cables.
- Observe minimum cable bend radius (see data sheet for the cable).
- Secure the cable underneath the bus controller. The cable must be secured vertically under the RJ45 connector on the bus controller.

Information:

Using POWERLINK cables offered by B&R satisfies the EN 61131-2 product standard.

For any further requirements, the customer must take additional measures.

Wiring diagram

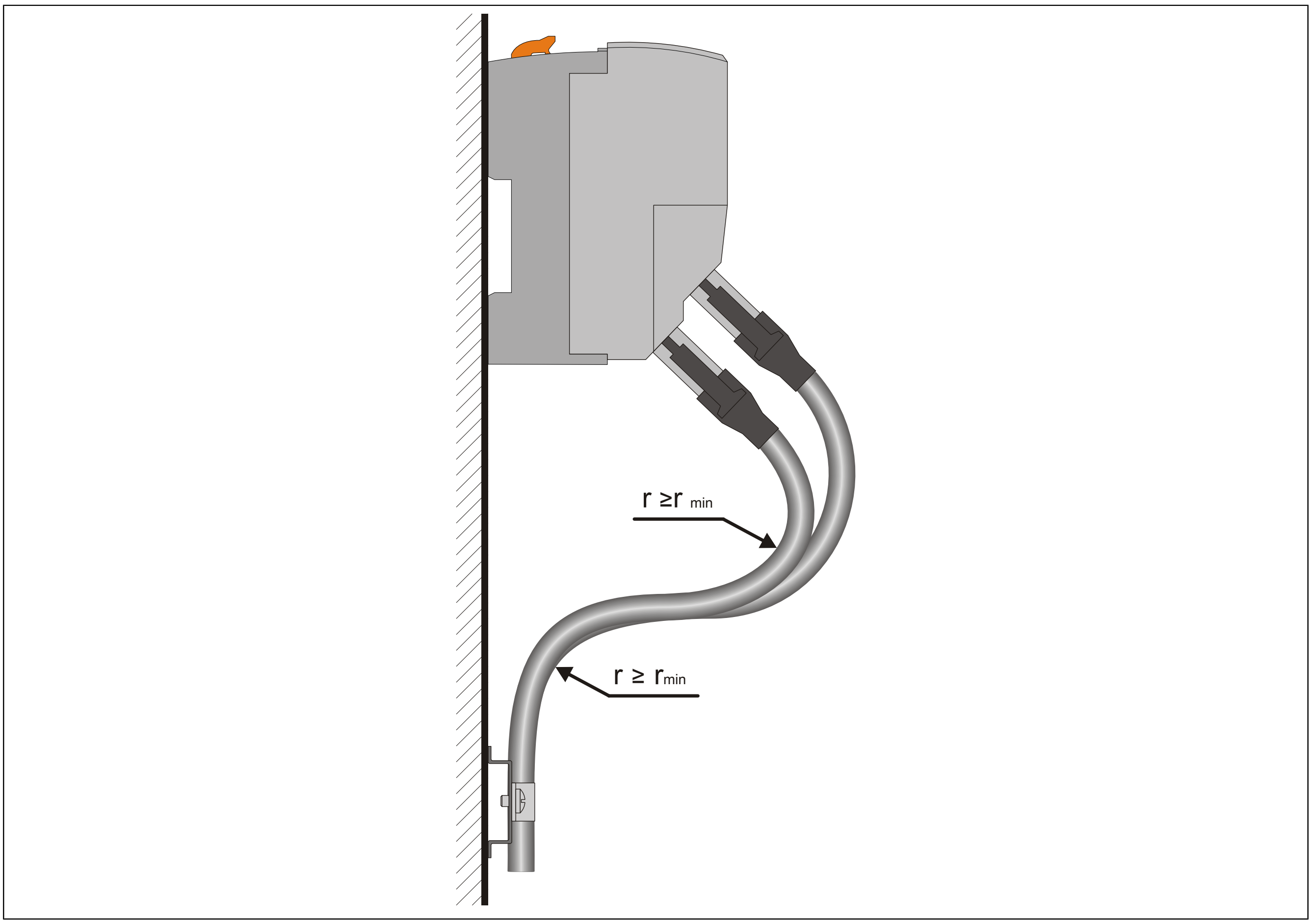


Figure 25: Wiring diagram for X20 modules with an Ethernet cable

2.4.8 Supply concept

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

2.4.8.1 Bus module rack replacement

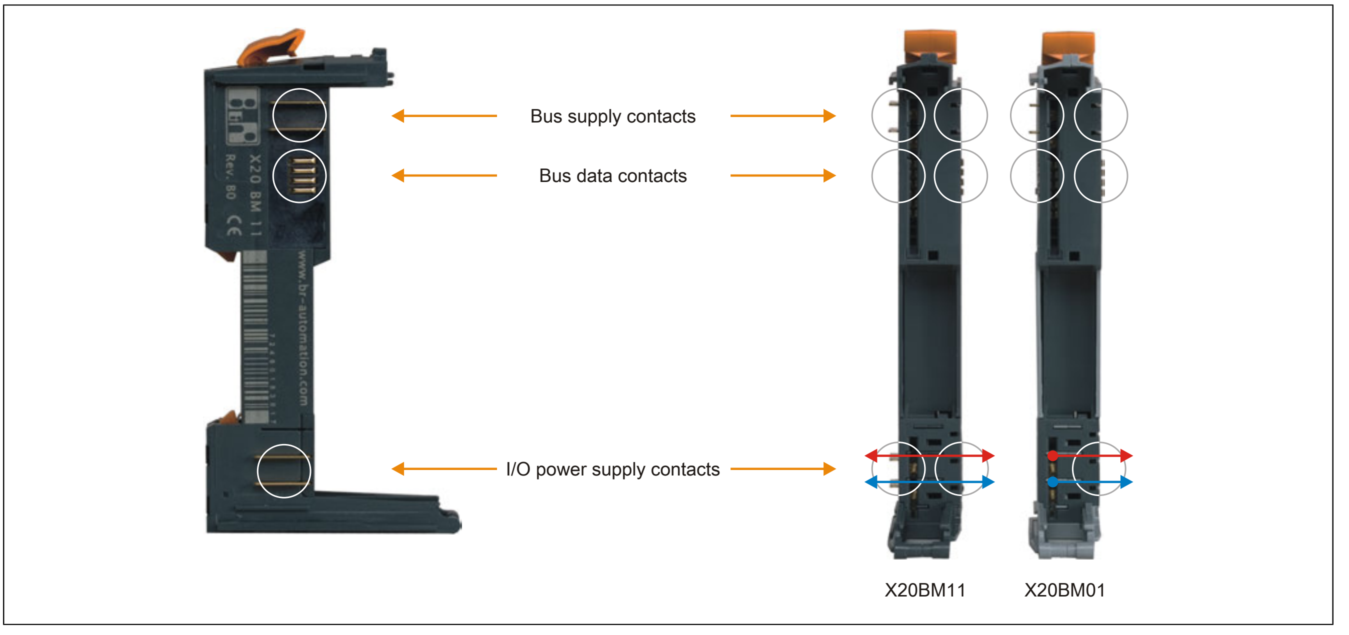


Figure 26: The bus module replaces the rack in the X20 system

The bus module is the backbone of the X20 system regarding the bus supply and bus data as well as the I/O supply for the electronics modules. Each bus module is an active bus station, even without an electronics module. There are two variations of the bus module:

- Interconnected I/O supply
- I/O supply isolated to the left (for power supply modules)

2.4.8.2 X20 system infrastructure

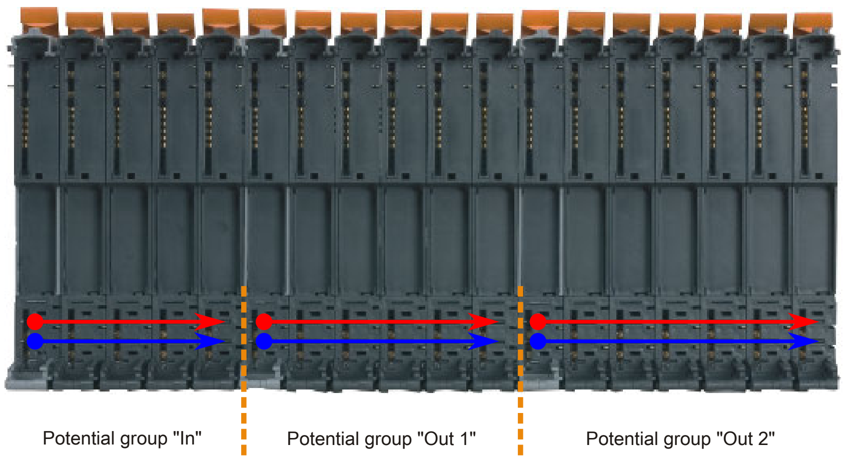


Figure 27: Simple implementation of different potential groups

Different potential groups can be implemented with the appropriate arrangement of supply bus modules, (e.g. for input groups or different emergency stop circuits on the outputs). The I/O power supply is fed by power supply modules.

2.4.8.3 Bus supply

Because the remote X2X Link backplane and I/O electronics are completely electrically isolated, the X2X Link power supply needs to be fed in at certain intervals. This is initially handled by the bus receiver. A supply module for X2X Link must be added to refresh the supply after approximately 30 modules (for an example calculation, see "Calculating the power requirements" on page 101). On the same module, a separate feed for the I/O power supply can also be connected.

2.4.8.4 Potential groups

The I/O supply is connected via the bus modules, and the supply is fed in using corresponding power supply modules. This makes it possible to implement simple potential groups (e.g. for input groups or different output groups). For isolation, the corresponding bus module is also necessary, which provides isolation of the internal I/O supply.

2.4.8.5 Output modules with supply

Generally, a power supply module is also necessary for current output modules with many channels, such as the 8 channel output module with 2 amp outputs. This is not the case with the X20 system. With this module, the supply is provided directly on the module, thereby saving power supply modules and construction width.

2.4.8.6 Bus receiver with supply

The X20BR9300 bus receiver for the X20 system is equipped with a supply for X2X Link as well as for the internal I/O supply. This eliminates the need for an additional power supply module.

2.4.8.7 Supply module for internal I/O supply

The first I/O modules in an X20 system are supplied by the bus receiver. The internal I/O supply is refreshed via the X20PS2100 power supply module.

2.4.8.8 Power supply module for internal I/O supply and bus supply

The X2X Link is fed by the X20BR9300 bus receiver. After approx. 30 modules (see section "Calculating the power requirements" on page 101 for a calculation example), the supply must be "refreshed". The X20PS3300 power supply module is used for this. This module is equipped with a feed for X2X Link as well as for the internal I/O power supply.

2.4.8.9 Bus transmitter with supply

The X20BT9100 bus transmitter has an integrated I/O supply feed. This saves a power supply module for the last potential group.

2.4.8.10 Internal I/O power supply failure (ModuleOk)

The ModuleOk status for monitoring the X20 modules is made up of different module parameters.

> ## Information:
>
> **All modules that require 0.01 W of power on the X2X Link network must be supplied via the internal I/O power supply. If the I/O power supply fails, the module shuts down and communication is lost.**
> **In this case, ModuleOk returns the value "False" and data can no longer be read from the "embedded parameter chip".**

2.4.8.11 X20 system power supply

The power supply for the X20 system is provided by B&R 24 VDC power supplies. B&R power supplies ensure that control systems are reliably supplied even when operated at the minimum mains input voltage or when maximum power is output even in the event of temporary power failures (≤10 ms).

The power to be provided by the B&R power supply must be calculated (see "Dimensioning the external 24 VDC power supply" on page 114).

2.4.8.12 X2X Link supply

The X2X Link remote backplane is supplied separately from the I/O points. This ensures that the remote backplane does not fail if there is a power failure on the I/O side, for example during an emergency stop. After approx. 30 modules, it is necessary to "refresh" with a power supply module for X2X Link.

To achieve increased supply security, it is possible to set up a redundant X2X Link power supply. To do so, the necessary X2X Link power must be determined and then covered by the corresponding quantity plus at least one additional X2X Link power supply module. This guarantees the functionality of the remote backplane even if the X2X Link power supply fails.

Please note the following for the correct calculation:

- To determine the necessary X2X Link power, calculate using 75% of the power supply module's rated power during parallel operation.

Information:

This must be done for all power supply modules at the same time for a non-redundant X2X Link power supply or when completely turning the X2X supply of an X20 module block on/off.

2.4.8.12.1 Example for extended X2X Link supply

It is possible to set up potential groups through the use of different supplies for the power supply modules.

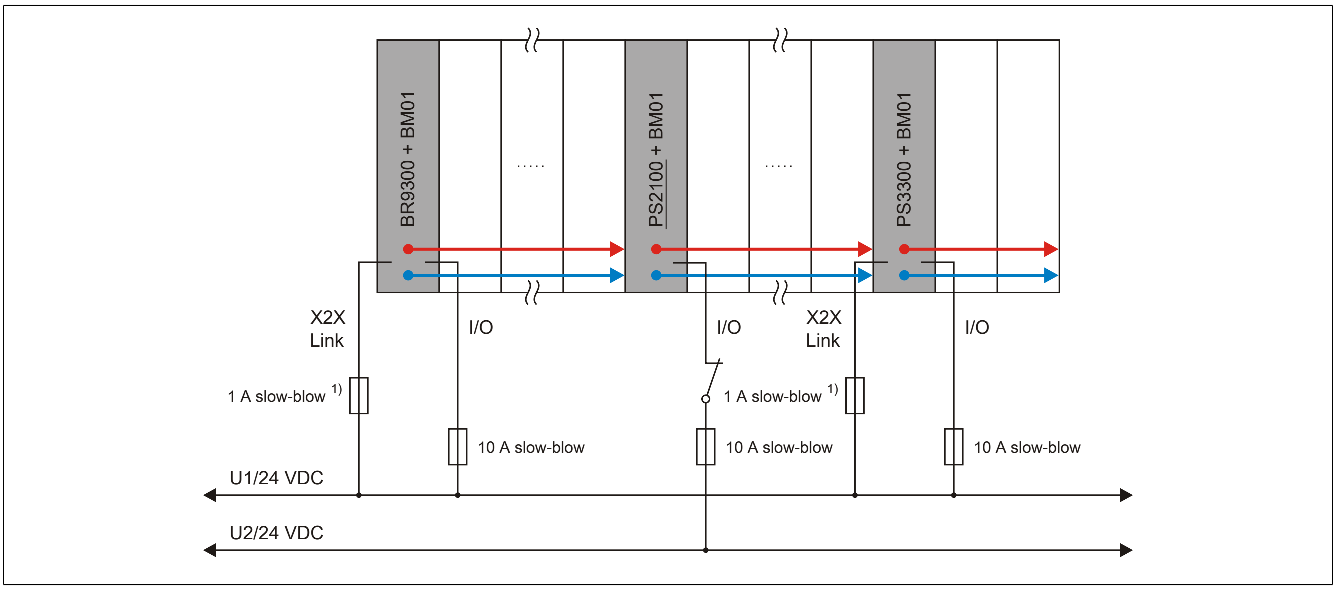


Figure 28: Example for extended X2X Link supply

1) Recommended for line protection.

The X20PS3300 power supply module supplies both the X2X Link and I/O; the X20PS2100 power supply module only supplies the I/O.

2.4.8.12.2 Example for redundant X2X Link supply

Multiple X20PS3300 power supply modules can be set up in parallel. It is possible to set up potential groups through the use of different supplies.

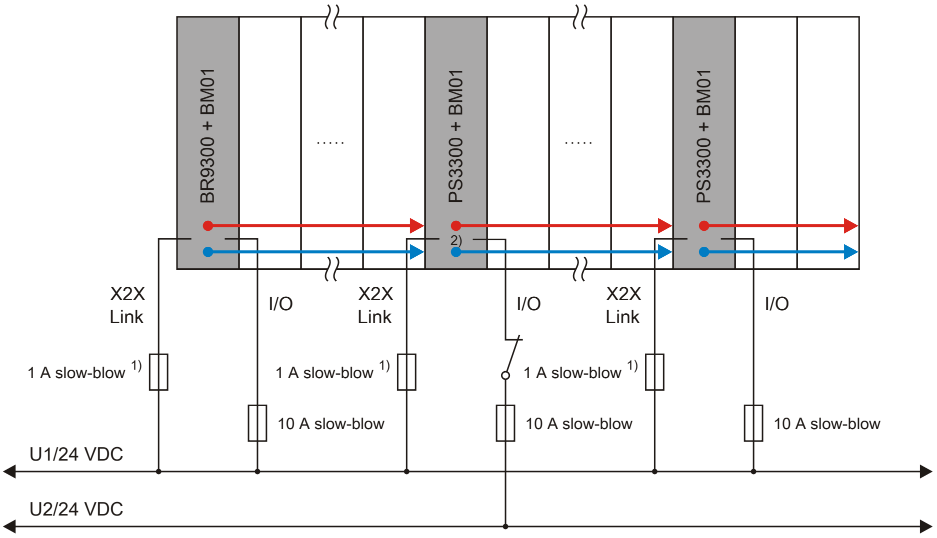


Figure 29: Example for redundant X2X Link supply

1) Recommended for line protection.
2) With separate supplies, the two reference potentials (GND_1 and GND_2) are combined via the terminal block on the PS3300.

The X20PS3300 power supply module supplies both X2X Link and the I/O.

2.4.9 X20 system protection

The protection for the X20 system depends on the supply concept.

2.4.9.1 Potential groups

Using the X20BM01 bus module and organizing the power supply bus modules accordingly allows various potential groups to be implemented (e.g. for input groups or various power circuits for the outputs).

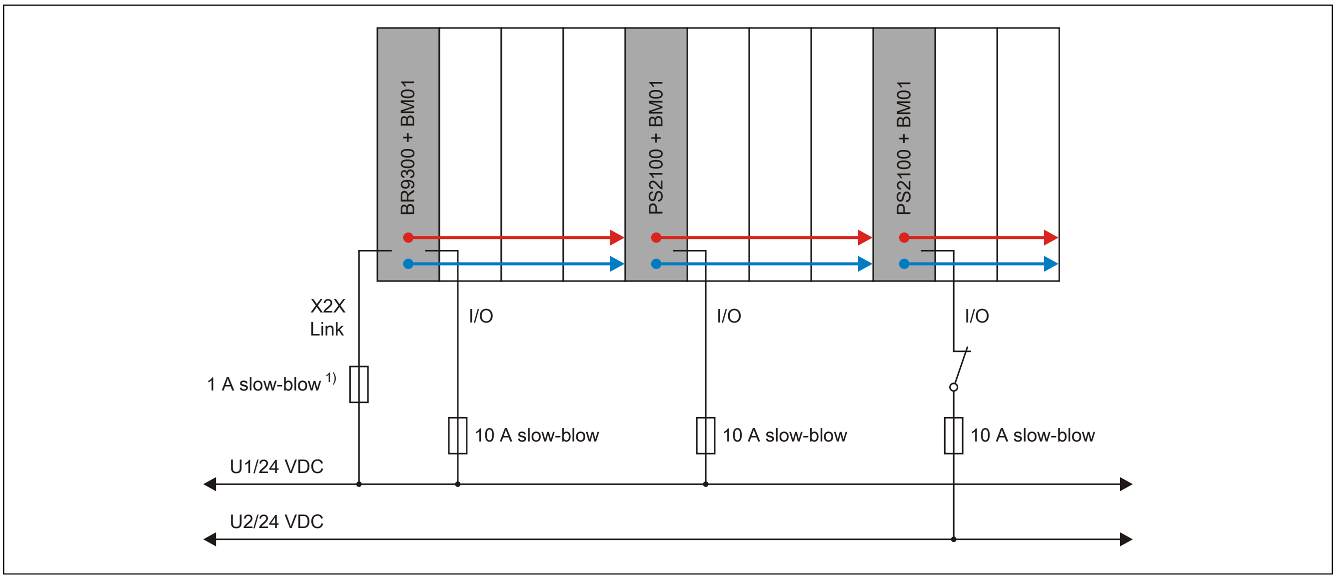


Figure 30: Protecting various potential groups

1) Recommended for line protection.

2.4.9.2 Supply via bus transmitter

The bus transmitter has an integrated internal I/O supply feed. This saves a power supply module for the last potential group.

Keep in mind: this potential group is separated from the rest of the potential groups by an I/O module with the x20(c)BM01 bus module.

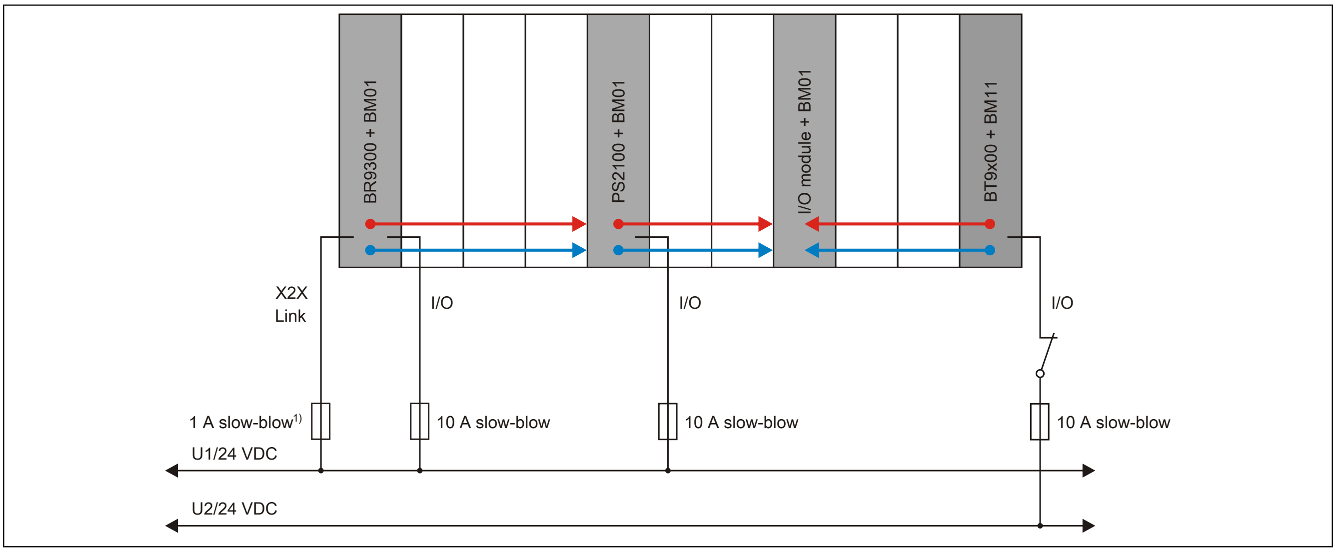


Figure 31: Protection when supplied via bus transmitter

1) Recommended for line protection.

2.4.10 Safe cutoff of a potential group

Information:

B&R keeps user's manuals as current as possible. From a safety standpoint however, the current certified version of the document must be used.

The current certified document is available for download under Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups.

2.4.10.1 Description of function

The operating principle "Safe cutoff of a potential group" enables the user to execute safety-related functions within a B&R system in combination with an external safety relay.
The safety function is limited to cutting off or interrupting the power to connected actuators.

Functionality

An external safety relay is connected to the I/O power supply for the potential group or an X20SP1130 power supply module is used. When the functional safe state is requested or state "Failsafe" occurs, then this feed cuts off the I/O power supply of the potential group. The power is then also cut off for all actuators connected to this potential group. However, module-internal energy storage devices (e.g. capacitors) remain charged and must be taken into account in the assessment of the safety function.

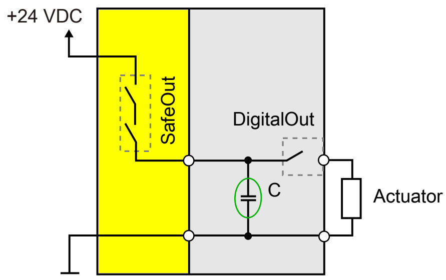


Figure 32: Functionality with internal energy storage

2.4.10.2 Scope of application / Standards referenced

The operating principle is confined to machine manufacturing applications and therefore implicitly to the following standards:

- EN ISO 13849-1:2015 / EN ISO 13849-2:2012

Requirements of other standards are not taken into consideration.

2.4.10.3 Intended use

Danger!

Danger from incorrect use of safety-related products/functions

Proper functionality is only ensured if the products/functions are used in accordance with their intended use by qualified personnel and the provided safety information is taken into account. The aforementioned conditions must be observed or covered by supplementary measures on your own responsibility in order to ensure the specified protective functions.

2.4.10.3.1 Qualified personnel

Use of safety-related products is restricted to the following persons:

- Qualified personnel who are familiar with relevant safety concepts for automation technology as well as applicable standards and regulations
- Qualified personnel who plan, develop, install and commission safety equipment in machines and systems

Qualified personnel in the context of this manual's safety guidelines are those who, because of their training, experience and instruction combined with their knowledge of relevant standards, regulations, accident prevention guidelines and operating conditions, are qualified to carry out essential tasks and recognize and avoid potentially dangerous situations.
In this regard, sufficient language skills are also required in order to be able to properly understand this manual.

2.4.10.3.2 Area of application

The safety-related B&R control components described in this manual were designed, developed and manufactured for special applications for machine and personnel protection. They are not suitable for any use involving serious risks or hazards that could lead to the injury or death of several people or serious environmental impact without the implementation of exceptionally stringent safety precautions. In particular, this includes the use of these devices to monitor nuclear reactions in nuclear power plants, flight control systems, air traffic control, the control of mass transport vehicles, medical life support systems and the control of weapon systems.

When using safety-oriented control components, the safety precautions applying to industrial control systems (e.g. the provision of safety devices such as emergency stop circuits, etc.) must be observed in accordance with applicable national and international regulations. The same applies for all other devices connected to the system, e.g. drives or light curtains.

The safety guidelines, information about connection conditions (nameplate and documentation) and limit values specified in the technical data must be read carefully before installation and commissioning and must be strictly observed.

2.4.10.3.3 Security concept

B&R products communicate via a network interface and were developed for integration into a secure network. The network and B&R products are affected by the following hazards (not a complete list):

- Unauthorized access
- Digital intrusion
- Data leakage
- Data theft
- A variety of other types of IT security breaches

It is the responsibility of the operator to provide and maintain a secure connection between B&R products and the internal network as well as other networks, such as the Internet, if necessary. The following measures and security solutions are suitable for this purpose:

- Segmentation of the network (e.g. separation of the IT and OT networks)
- Firewalls for the secure connection of network segments
- Implementation of a security-optimized user account and password concept
- Intrusion prevention and authentication systems
- Endpoint security solutions with modules for anti-malware, data leakage prevention, etc.
- Data encryption

It is the responsibility of the operator to take appropriate measures and to implement effective security solutions.

B&R Industrial Automation GmbH and its subsidiaries are not liable for damages and/or losses resulting from, for example, IT security breaches, unauthorized access, digital intrusion, data leakage and/or data theft.

Before B&R releases products or updates, they are subjected to appropriate functional testing. Independently of this, the development of customized test processes is recommended in order to be able to check the effects of changes in advance. Such changes include, for example:

- Installation of product updates
- Notable system modifications such as configuration changes
- Import of updates or patches for third-party software (non-B&R software)
- Hardware replacement

These tests should ensure that implemented security measures remain effective and that systems behave as expected.

2.4.10.3.4 Safety technology disclaimer

The proper use of all B&R products must be guaranteed by the customer through the implementation of suitable training, instruction and documentation measures. The guidelines set forth in system user's manuals must be taken into consideration here as well. B&R has no obligation to provide verification or warnings with regard to the customer's purpose of using the delivered product.

Changes to the devices are not permitted when using safety-related components. Only certified products are permitted to be used. Currently valid product versions in each case are listed in the corresponding certificates. Current certificates are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product. The use of non-certified products or product versions is not permitted.

All relevant information regarding these safety products must be read in the latest version of the related data sheet and the corresponding safety notices observed before the safety products are permitted to be operated. Certified data sheets are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product.

B&R and its employees are not liable for any damages or loss resulting from the incorrect use of these products. The same applies to misuse that may result from specifications or statements made by B&R in connection with sales, support or application activities. It is the sole responsibility of the user to check all specifications and statements made by B&R for proper application as it pertains to safety-related applications. In addition, the user assumes sole responsibility for the proper design of the safety function as it pertains to safety-related applications.

2.4.10.3.5 Installation notes for X20 safety modules

Products must be protected against impermissible dirt and contaminants. Products are protected from dirt and contaminants up to pollution degree II as specified in the IEC 60664 standard.

Pollution degree II can usually be achieved in an enclosure with IP54 protection, but uncoated modules are NOT permitted to be operated in condensing relative humidity and temperatures under 0°C.

The operation of coated modules is allowed in condensing relative humidity.

Danger!

Pollution levels higher than specified by pollution degree II in standard IEC 60664 can result in dangerous failures. It is extremely important that you ensure a proper operating environment.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

The supply of X20 potential groups must generally be protected using a fuse with a maximum of 10 A.
For more information, see chapter "Mechanical and electrical configuration" of the X20 or X67 user's manual.

2.4.10.3.6 Installation notes for X67 safety modules

Danger!

The following points must be taken into account to ensure IP67 protection:

- **The union nuts on female/male connectors must be tightly secured with the specified tightening torque. For the tightening torque, see the X67 system user's manual.**
- **Female/Male connectors that are not being used must be closed with threaded caps!**
  - **Threaded caps M8, 50 pcs.: X67AC0M08**
  - **Threaded caps M12, 50 pcs.: X67AC0M12**

Danger!

The shock and vibration resistance values (see chapter "International and national certifications" of the X67 system user's manual) apply if cables are laid solidly.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

Danger!

Unused female connectors must be covered with threaded caps (X67AC0M08 or X67AC0M12 accessory). Otherwise, hazardous conditions may arise if the module fails to function properly.

2.4.10.3.7 Safe state

If an error is detected by the module (internal or wiring error), the modules enable the safe state. The safe state is structurally designed as a low state or cutoff and cannot be modified.

Applications in which the safe state must actively switch on an actuator cannot be implemented with this module. In these cases, other measures must be taken to meet this safety-related requirement (e.g. mechanical brakes for hanging load that engage on power failure).

2.4.10.4 System-specific information

The operating principle applies to a potential group.

All potential groups are generally only permitted to be supplied by 1 power supply module. The possible further processing of the power supply on the module is not permitted to result in multiple supply instances.

In the X20 system, only modules of type X20BM01, X20BM23 and X20BM26 that ensure the interruption of the internal I/O power supply to the left are permitted to be used as bus modules for power supply modules.
On modules X20PS9400 and X20PS3300, only the I/O power supply (+24 V I/O) is permitted to be switched with the safety relay. The bus power supply (+24 V BC/X2X L.) must be isolated.

When using module X67PS1300 to supply power to the X67 potential group, only the I/O power supply (+24 V I/O) is permitted to be switched with the safety relay. The bus power supply (+24 V BC/X2X L.) must be isolated.

The operating principle is limited to the modules listed in the following certificate.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

2.4.10.5 Safety guidelines

This section provides a summary of safety notices for the user.

Danger!

Failure of the safety function due to misuse

Observe the following safety guidelines. Failure to observe any of the following notices can lead to the failure of the safety function and may result in serious injury.

- **When using the operating principle, it is the user's responsibility to adhere to the relevant standards and safety directives. In addition, the guidelines for proper use must be observed.**
- **For all potentials supplying the modules, SELV/PELV power supplies must be used.**
- **The potential groups for which the operating principle is applied are only permitted to contain modules listed from certificate "Safe cutoff of potential groups".**
- **Uncoated X20 modules in which the operating principle is used are not permitted to be operated in condensing air humidity or at temperatures below 0°C.**
- **It is not permitted to mix modules from different systems (X20, X67, 7XV) within a potential group.**
- **It is not permitted to install multiple power supplies in a potential group (particularly with regard to power supply modules that also supply the bus supply).**
- **Ensure that the upstream safety relay is wired properly.**
- **Ensure that ALL sensors and actuators connected to the potential group are wired properly.**
- **Note possible impairments of the safety function due to the internal energy storage devices. If this is sufficient to enable a connected actuator and subsequently leads to a dangerous state, the protection objective is not given and alternatives or supplementary measures must be installed.**
- **The switch-off time must be verified by a control measurement!**
- **For modules with isolated I/O potential for sensors and actuators, the upstream safety relay must shut off the supply for both the sensors and actuators.**
- **The ground connections should be used as functional ground and not as protective ground and must not be connected to the 24 V supply voltage (GND is permitted). In addition, no protective components are permitted to be used between the ground and the 24 V supply voltage.**

2.4.10.5.1 Capacitances within the potential group

The internal capacitances of the module remain charged at the time of shutdown. The total capacitance of the potential group results from the sum of the capacitances of the individual modules, upstream external safety relay and actuator.

$$C_{total} = \sum_{i=1}^{n} C_i$$

The capacitances of the corresponding B&R modules are listed in the certificate.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

At the time a safety function is requested, it is not ensured that the standard outputs are enabled. If an output is disabled at the time of the request, the affected module-internal capacitances remain permanently charged. If the output is enabled by the standard application, an unexpected voltage peak occurs on the output.

In connection with the supply voltage, the total capacitance present in the system results in a charge that must be taken into account during cutoff. In the worst case scenario, it can be assumed that the total capacitance present in the system buffers each output present in the potential group. This behavior is not permitted to lead to a safety-critical state due to actuators in the potential group; alternatives or supplementary corrective measures must be installed.

2.4.10.5.2 Potential group structure

The potential group is only permitted to be made up of modules listed in the following certificate. Modules not listed in this certificate endanger the "absence of feedback" of the external cutoff and therefore the safety function.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

To ensure clarity and that the external cutoff is triggered when a fault occurs, installing multiple power supply sources in a potential group is not permitted.

SELV/PELV power supplies must be used for both the bus supply (X2X) and the I/O power supply; otherwise, safety-related malfunctions can occur due to overvoltages.

For modules with isolated I/O potential for sensors and actuators, the upstream safety relay must shut off the supply for both the sensors and actuators; otherwise, feedback cannot be excluded.

2.4.10.5.3 Circuit examples

Single-channel without feedback

The following example shows a load being cutoff using the emergency stop safety function. Only safe actuators such as motors or input "Enable" of an ACOPOS/ACOPOSmulti drive are permitted to be used as the load in this case.

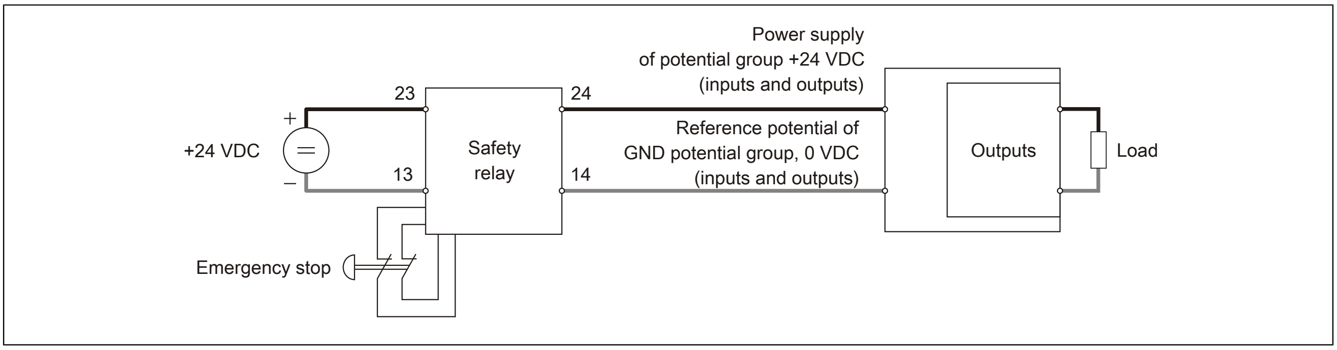


Figure 33: Circuit example: 1-channel without feedback

Provided that the external components being used (emergency stop switch, safety relay, load) satisfy the respective requirements, this example can achieve PL e (performance level as specified in EN ISO 13849-1:2015).

Dual-channel with feedback

The following example shows a load being cutoff using the emergency stop safety function. Feedback allows errors in the actuator to also be detected, and a cutoff is also possible if a fault event occurs due to the full dual-channel design. Whether or not 2 fully isolated potential groups – as shown in the example – are necessary depends on the application and how the safety solution is designed.

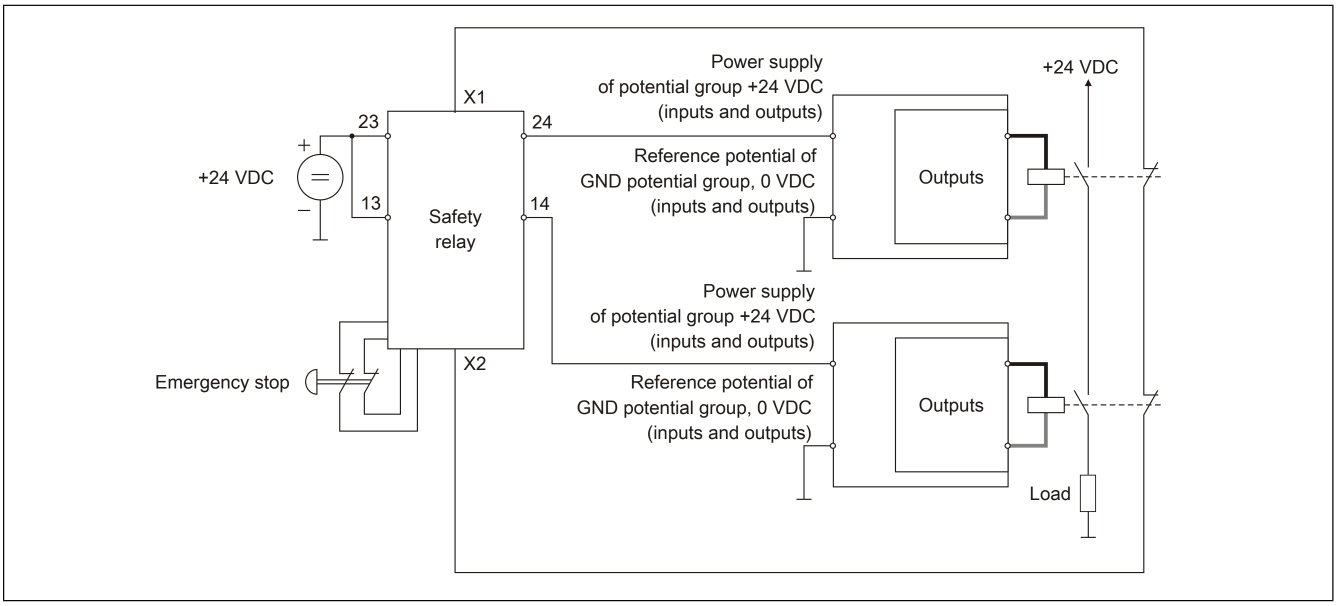


Figure 34: Circuit example: 2-channel with feedback

Provided that the external components being used (emergency stop switch, safety relay, load) satisfy the respective requirements, this example can achieve PL e.

Example with power supply module X20SP1130

The following examples show a load being cut off using safe power supply module X20SP1130 along with safe input module X20SI4100 and the "emergency stop" safety function.

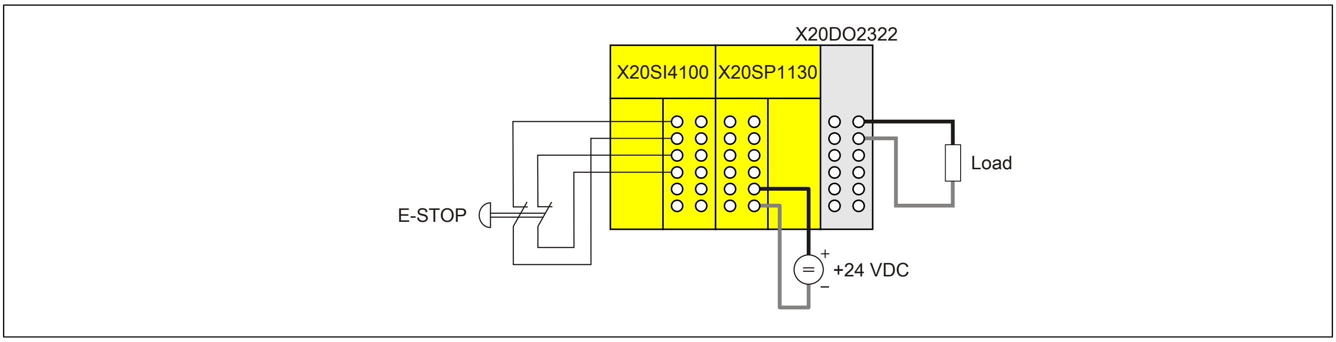


Figure 35: Circuit example with power supply module X20SP1130

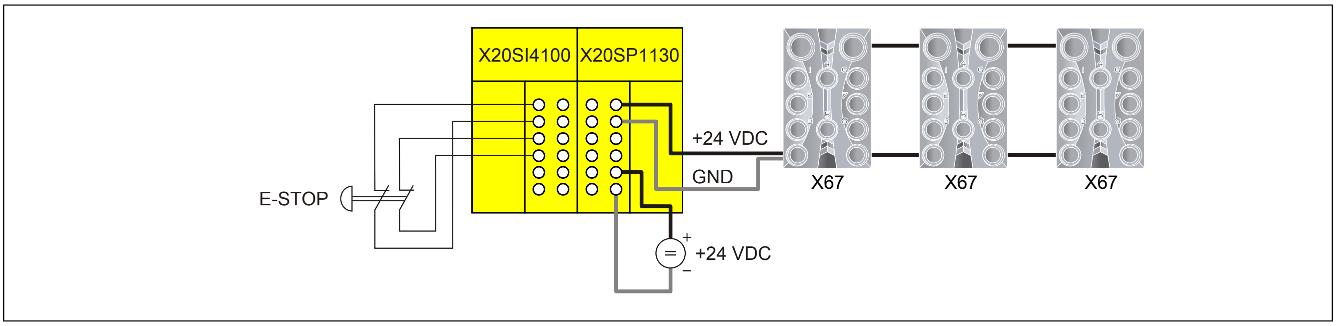


Figure 36: Circuit example with power supply module X20SP1130 and X67

Provided that the external components being used (emergency stop switch, load) satisfy the respective requirements, these examples can achieve PL e.

2.4.10.5.4 Wiring information

The operating principle "Safe cutoff of a potential group" only applies to the B&R modules being used. All other parts of the safety chain such as the application, upstream sensors or downstream actuators are NOT included in this principle.

For this reason, it is important to take the following points into consideration:

- Ensure proper wiring of the safety relay with the I/O supply. A short circuit between the output of the safety relay and an external 24 V voltage source can cause an unintended supply of 24 V to the internal supply voltage of the potential group. As a result, the safety function can no longer be guaranteed, which means that **ALL** of the channels in the potential group can no longer be cut off by the upstream safety relay.
- Make sure that **ALL** of the potential group's input and output channels and the connected sensors and actuators are wired properly. A short circuit between an input or output of the potential group and an external 24 V voltage source can cause the unintended feedback of 24 V to the internal supply voltage of the potential group. As a result, the safety function can no longer be guaranteed, which means that **ALL** of the output channels in the potential group can no longer be cut off by the upstream safety relay.
- In accordance with EN ISO 13849-2:2012, appendix D.2, table D.4, a short circuit between any 2 conductors can be excluded, provided that:
    - They are permanently installed and protected against external damage (e.g. using a cable duct or armored conduit)
    - OR they are in separate plastic-sheathed cables
    - OR they are installed within an electrical enclosure. This requires that the wiring as well as the area for electrical equipment meet the respective requirements [see EN 60204-1]
    - OR they are individually shielded with a ground connection

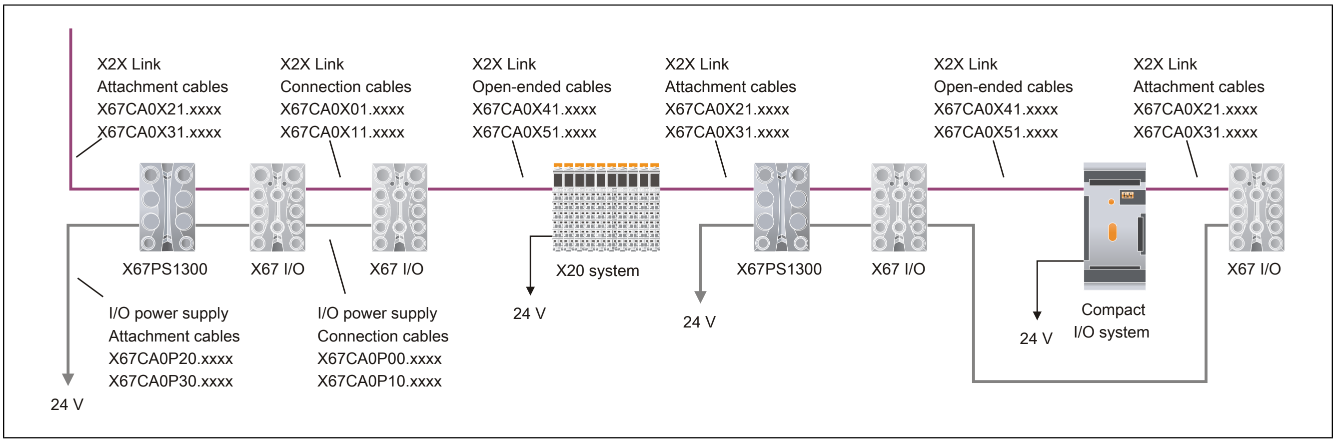
2.4.11 Combining X2X Link systems

The X2X Link provides a complete remote backplane, which is used for communicating between bus modules and over the X2X Link cable. Systems based on X2X Link can be combined with one another as needed.

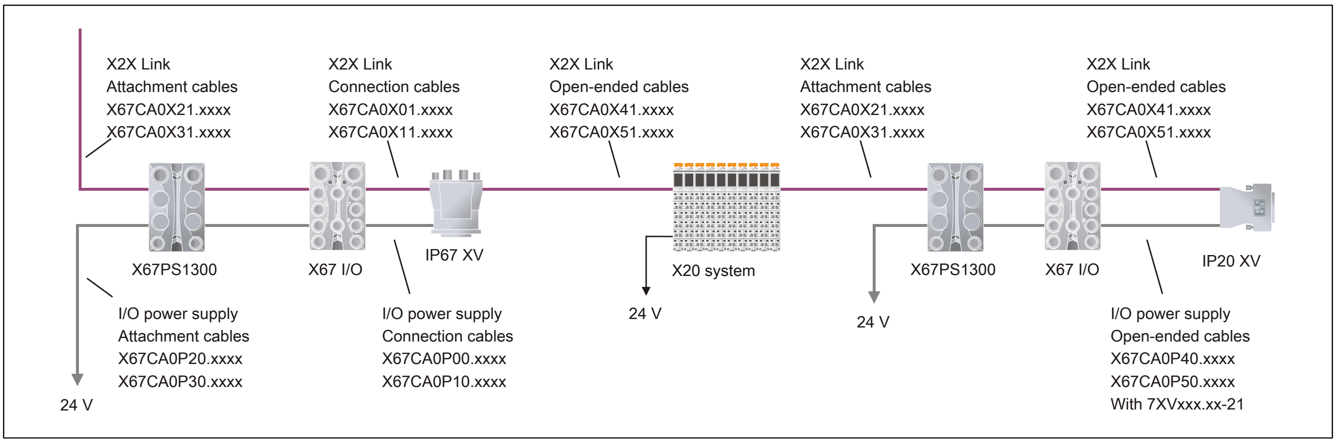
2.4.11.1 Connection overviews

The following connection overviews illustrate combinations of different systems that are based on X2X Link. The model numbers indicate which standard cables available from B&R can be used to connect with one another.

Combining X20, X67 and compact I/O system



Combining X20, X67 and valve terminal connections



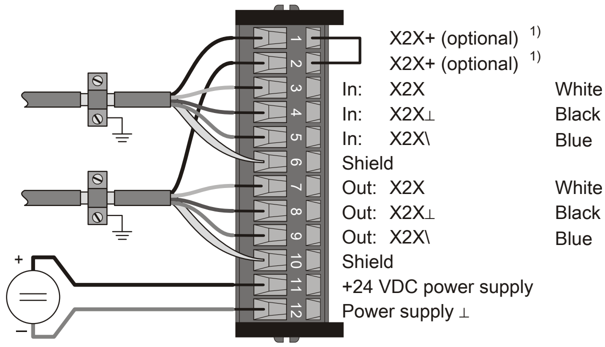
2.4.11.2 Connection examples

2.4.11.2.1 X20 system

Connection examples are listed in the module description:

- X20BR9300 bus receiver: See technical data sheet at www.br-automation.com.
- X20BT9100 bus transmitter: See technical data sheet at www.br-automation.com.

2.4.11.2.2 Compact I/O system



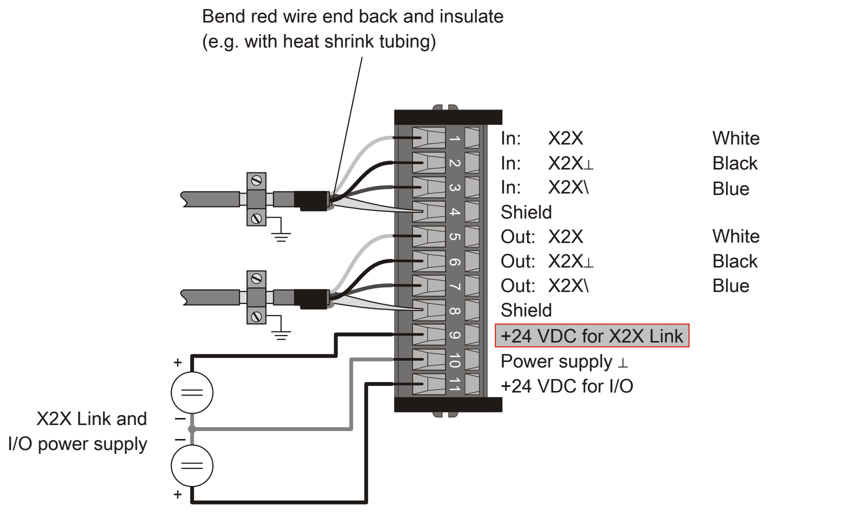
1) Used to forward the X2X Link power supply when using IP67 modules.

Signal	Cable type	Model number
X2X Link In	Open-ended cables[1)]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1)]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.1000

1) Bridge for X2X+ in connection with X67 modules.

2.4.11.2.3 Valve connection

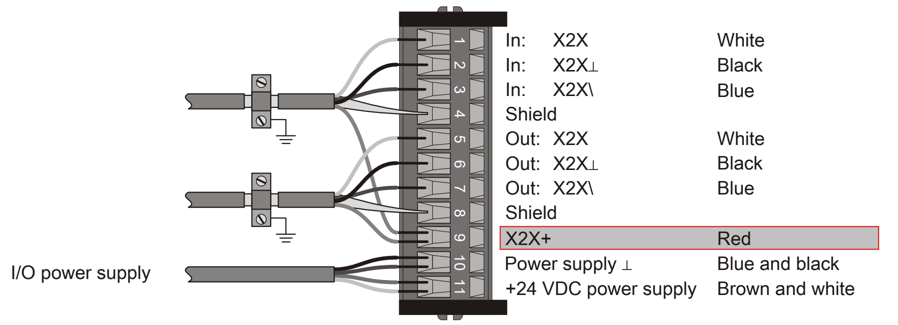
Connection example with 7XVxxx.xx-11/-12



Signal	Cable type	Model number
X2X Link In	Open-ended cables[1]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.xxxx

1) In connection with X67 modules.

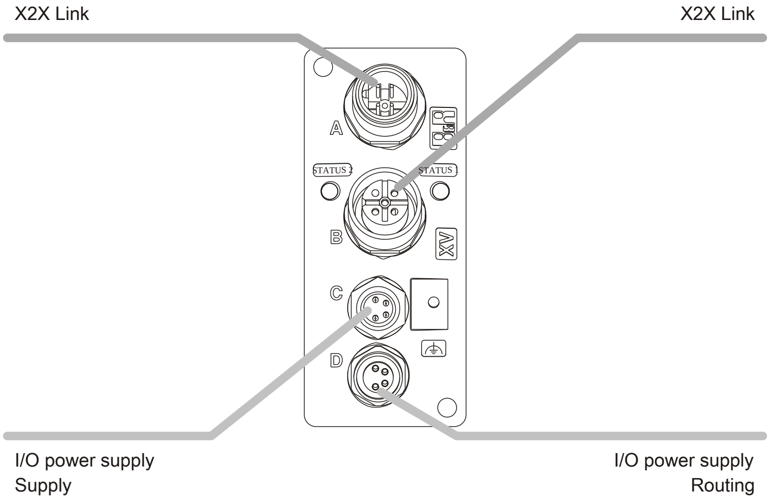
Connection example with 7XVxxx.xx-21



Signal	Cable type	Model number
X2X Link In	Open-ended cables[1]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.xxxx
I/O power supply	Open-ended cables[1]	X67CA0P40.xxxx
		X67CA0P50.xxxx

1) In connection with X67 modules.

Connection example with 7XVxxx.xx-51/-62



Signal	Cable type	Model number
X2X Link	Connection cables[1)]	X67CA0X01.xxxx
		X67CA0X11.xxxx
I/O power supply	Connection cables[1)]	X67CA0P00.xxxx
		X67CA0P10.xxxx

1) In connection with X67 modules.

2.4.12 Calculating the power requirements

The power requirements are calculated based on the power consumption of the individual modules listed in the respective data sheets. Due to the separate power supplies, the power requirements of the **X2X Link power supply** and **I/O power supply** must be calculated independently of one another.

For the power consumption of both areas, see the technical data. This makes it possible to quickly yet clearly establish the power requirements for a certain hardware configuration. The power consumption values of individual modules should then be subtracted from the power provided by the power supply module. The sum is not permitted to be less than zero.

The following entries in the technical data are used for the calculation:

X2X Link power supply

- **General information - Power consumption - Bus**
  This entry refers to the power consumption needed to operate the X2X Link network. The power consumption of the I/O modules and bus base modules are taken into account in the calculation.

I/O power supply

- **General information - Power consumption - Internal I/O**
  This entry refers to the power consumption needed to operate the actual I/O module as well as the inputs and outputs.
- **Sensor power supply - Power consumption**
  This entry contains the power consumption needed by the I/O module to provide power to connected sensors, for example.
- **Actuator power supply - Power consumption**
  This entry contains the power consumption needed by the I/O module to provide power to connected actuators, for example.

The power consumption of I/O modules whose power is supplied externally does not need to be taken into account.

Information:

All modules that require 0.01 W of power on the X2X Link network must be supplied via the internal I/O power supply. If the I/O power supply fails, the module shuts down and communication is lost.
In this case, ModuleOk returns the value "False" and data can no longer be read from the "embedded parameter chip".

Information:

The I/O power actually needed can vary depending on the application. In order to properly calculate the power requirements for the respective application, additional specifications such as simultaneity of the outputs, actual required output current, etc. must be taken into account.

2.4.12.1 Overview of bus and I/O power supplies

The power necessary to operate the X20 system is provided by the power supplied modules, X20 CPUs, bus receivers and bus transmitters.

Module	Power supplied by internal I/O	Power supplied by the bus
X20CP1483, X20CPx58x	+240 W	+7 W
X20CP13xx, X20CP13xx-RT	+240 W	+2 W
X20BR7300	+240 W	+2 W
X20BR9300	+240 W	+7 W
X20PS2100	+240 W	(-0.2 W)[1]
X20PS2110	+144 W	(-0.2 W)[1]
X20PS3300	+240 W	+7 W
X20PS3310	+144 W	+7 W
X20PS9xxx	+240 W	+7 W
X20SP1130	+240 W	(-0.2 W)[1]

1) This module does not provide extra power to the bus (X2X Link) but requires power itself for operation.

Bus transmitters

When calculating the power requirements for bus transmitters, it is important to know whether they are only being used as such or are also being used as an I/O power supply module.

Model number	Power supplied by internal I/O		Bus power
	When operated as a bus transmitter	When operated as a bus transmitter and I/O power supply module	
X20BT9100	(-0.1 W)	+240 W	(-0.5 W)[1]
X20BT9400	(-0.1 W)	+240 W	(-0.5 W)[1]

1) This module does not provide extra power to the bus (X2X Link) but requires power itself for operation.

Information:

If the bus or I/O power consumption of the connected modules exceeds the power actually provided, then additional power supply modules must be added (see "X20 system infrastructure" on page 80).

2.4.12.2 Example: CPU and modules

Calculating the power requirements for the bus and I/O power supply of a module block with X20 CPU.

Power supplied by the CPU

Module	Power supplied by the bus	Power supplied by the I/O power supply
X20CP3585	+7 W	+240 W

Power requirements of the modules

Module	Bus power supply requirements		I/O power supply requirements	
	Bus modules	I/O modules	Internal I/O	Sensor/Actuator power supply
X20CP3585[1]	-	-	-0.60 W	-
X20BM11 + X20DI9371	-0.13 W	-0.18 W	0.00 W	0.00 W
X20BM33 + X20SI9100	-0.13 W	-0.40 W	-1.60 W	0.00 W
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM33 + X20SI4100	-0.13 W	-0.32 W	-1.25 W	0.00 W
X20BM11 + X20DO4322	-0.13 W	-0.16 W	-0.49 W	Max. -12 W (actuator power supply) Max. -48 W (digital outputs)[2]
X20BM11 + X20DI4371	-0.13 W	-0.14 W	-0.59 W	Max. -12 W (sensor power supply)
Subtotal	-0.91 W	-1.22 W	-6.73 W	Max. -72 W
Total power requirements	-2.13 W		-78.73 W	

1) The power requirements of the power supply module (CPU) do not have to be taken into account in this calculation.
2) Power rating at 24 VDC and 2 A.

Calculating the power requirements

	Bus supply	I/O power supply
Power supplied by the CPU	+7 W	+240 W
Total power requirements	-2.13 W	-78.73
Remaining power	+4.87 W	+161.27

The power comparison indicates that the power provided by the CPU power supply module is sufficient. Additional power supply modules are not necessary.

2.4.12.3 Example: Bus controller and modules

Calculating the power requirements for the bus and I/O power supply of a module block with bus controller.

Power supplied by the power supply module

Module	Power supplied by the bus	Power supplied by the I/O power supply
X20PS9400	+7 W	+240 W

Power requirements of the modules

Module	Bus power supply requirements		I/O power supply requirements	
	Bus modules	I/O modules	Internal I/O	Sensor/Actuator power supply
X20PS9400[1)]	-	-	-0.60 W	-
X20BB81 + X20BC8083 + X20HB2880	-0.50 W	-2.00 W -1.17 W	-	0.00 W
X20BM33 + X20SI9100	-0.13 W	-0.40 W	-1.60 W	0.00 W
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM33 + X20SO2530	-0.13 W	-0.26 W	-1.15 W	0.00 W
X20BM33 + X20SO4120	-0.13 W	-0.25 W	-1.30 W	Max. -120 W (digital outputs)[2)]
X20BM33 + X20DO8322	-0.13 W	-0.26 W	-0.80 W	Max. -96 W (digital outputs)[3)]
X20BM11 + X20DI2371	-0.13 W	-0.12 W	-0.29 W	Max. -12 W (sensor power supply)
Subtotal	-1.41 W	-4.48 W	-7.94 W	Max. -228 W
Total power requirements	-5.89 W		-235.94 W	

1) The power requirements of the power supply module do not have to be taken into account in this calculation.
2) Power rating at 24 VDC and 5 A.
3) Power rating at 24 VDC and 4 A.

Calculating the power requirements

	Bus supply	I/O power supply
Power supplied by the power supply module	+7 W	+240 W
Total power requirements	-5.89 W	-235.94
Remaining power	+1.11 W	+4.06

The power comparison indicates that the power provided by the power supply module is sufficient. Additional power supply modules are not necessary.

2.4.12.4 Example: Potential groups

With a large number of I/O modules, the power provided by a power supply module is insufficient to operate all of the modules. In this case, the modules must be divided into potential groups.

Example of module group on bus receiver X20BR9300.

Power supplied by the bus receiver

A power supply module is already integrated in the bus receiver.

Module	Power supplied by the bus	Power supplied by the I/O power supply
X20BR9300	+7 W	+240 W

Power requirements of the entire module group

Module	Bus power supply requirements		I/O power supply requirements	
	Bus modules	I/O modules	Internal I/O	Sensor/Actuator power supply
X20BM01 + X20BR9300[1)]	-0.13 W	-	-0.60 W	-
X20BM11 + X20DI9371	-0.13 W	-0.18 W	0.00 W	0.00 W
X20BM33 + X20SI9100	-0.13 W	-0.40 W	-1.60 W	0.00 W
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM33 + X20SI4100	-0.13 W	-0.32 W	-1.25 W	0.00 W
X20BM11 + X20DO4322	-0.13 W	-0.16 W	-0.49 W	Max. -12 W (actuator power supply) Max. -48 W (digital outputs)[2)]
X20BM11 + X20DI4371	-0.13 W	-0.14 W	-0.59 W	Max. -12 W (sensor power supply)
X20BM33 + X20SI9400	-0.13 W	0.40 W	-1.60 W	0.00 W
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM33 + X20SO2530	-0.13 W	-0.26 W	-1.15 W	0.00 W
X20BM33 + X20SO4120	-0.13 W	-0.25 W	-1.30 W	Max. -120 W (digital outputs)[3)]
X20BM11 + X20DO8322	-0.13 W	-0.26 W	-0.80 W	Max. -96 W (digital outputs)[4)]
X20BM11 + X20DI2371	-0.13 W	-0.12 W	-0.29 W	Max. -12 W (sensor power supply)
Subtotal	-1.95 W	-2.53 W	-14.06 W	Max. -300 W
Total power requirements	-4.48 W		-314.06 W	

1) The power requirements of the bus receiver do not have to be taken into account in this calculation.
2) Power rating at 24 VDC and 2 A.
3) Power rating at 24 VDC and 5 A.
4) Power rating at 24 VDC and 4 A.

Calculating the power requirements

	Bus supply	I/O power supply
Power supplied by the bus receiver	+7 W	+240 W
Total power requirements	-4.48 W	-314.06 W
Remaining power	+2.52 W	-74.06 W

The power comparison indicates that the power provided by the bus receiver is insufficient. As additional power supply module is necessary to ensure the missing 74.06 W for the I/O power supply.

For this reason, the module group is divided into 2 potential groups.

Potential group 1

Power supplied by the bus receiver

A power supply module is already integrated in the bus receiver.

Module	Power supplied by the bus	Power supplied by the I/O power supply
X20BR9300	+7 W	+240 W

Power requirements of the modules

Module	Bus power supply requirements		I/O power supply requirements	
	Bus modules	I/O modules	Internal I/O	Sensor/Actuator power supply
X20BM01 + X20BR9300[1)]	-0.13 W	-	-0.60 W	-
X20BM11 + X20DI9371	-0.13 W	-0.18 W	0.00 W	0.00 W
X20BM33 + X20SI9100	-0.13 W	-0.40 W	-1.60 W	0.00 W
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20DO4322	-0.13 W	-0.16 W	-0.49 W	Max. -12 W (actuator power supply) Max. -48 W (digital outputs)[2)]
X20BM11 + X20DI4371	-0.13 W	-0.14 W	-0.59 W	Max. -12 W (sensor power supply)
X20BM11 + X20AI4622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20AO2622	-0.13 W	-0.01 W	-1.10 W	0.00 W
X20BM11 + X20DO8322	-0.13 W	-0.26 W	-0.80 W	Max. -96 W (digital outputs)[3)]
X20BM11 + X20DI2371	-0.13 W	-0.12 W	-0.29 W	Max. -12 W (sensor power supply)
Subtotal	-1.43 W	-1.30 W	-8.77 W	Max. -180 W
Total power requirements	-2.73 W		-188.77 W	

1) The power requirements of the bus receiver do not have to be taken into account in this calculation.
2) Power rating at 24 VDC and 2 A.
3) Power rating at 24 VDC and 4 A.

Potential group 2

Power supplied by the power supply module

Module	Power supplied by the bus	Power supplied by the I/O power supply
X20SP1130	-	+240 W

Power requirements of the modules

Module	Bus power supply requirements		I/O power supply requirements	
	Bus modules	I/O modules	Internal I/O	Sensor/Actuator power supply
X20BM23 + X20SP1130[1)]	-0.13 W	-0.2 W	-1.50 W	-
X20BM33 + X20SI4100	-0.13 W	-0.32 W	-1.25 W	0.00 W
X20BM33 + X20SI9100	-0.13 W	-0.40 W	-1.60 W	0.00 W
X20BM33 + X20SO2530	-0.13 W	-0.26 W	-1.15 W	0.00 W
X20BM33 + X20SO4120	-0.13 W	-0.25 W	-1.30 W	Max. -120 W (digital outputs)[2)]
Subtotal	-0.65 W	-1.43 W	-6.80 W	Max. -120 W
Total power requirements	-2.08 W		-126.80 W	

1) The power requirements of the power supply module do not have to be taken into account in this calculation.
2) Power rating at 24 VDC and 5 A.

Power balance for the bus power supply

	Bus power supply
Power supplied by the bus receiver	+7 W
Power balance of potential group 1	-2.73 W
Power balance of potential group 2	-2.08 W
Remaining power	+2.19 W

The power comparison indicates that the power provided by the bus receiver is sufficient. Additional power supply modules for the bus power supply are not necessary.

Power balance for the I/O power supply

Potential group 1

	I/O power supply
Power supplied by the bus receiver	+240 W
Total power requirements	-188.77 W
Remaining power	+51.23 W

Potential group 2

	I/O power supply
Power supplied by the power supply module	+240 W
Total power requirements	-126.80 W
Remaining power	+113.20 W

The power comparison indicates that the power provided by the power supply modules is now sufficient for both potential groups.

2.4.13 Power dissipation of power supply modules

Power supply modules are used to supply power to an X20 system. The power supply modules are either a separate module or part of a CPU or bus controller.

The power consumed by the power supply modules is passed on to the X20 system, taking into consideration its own power requirements and the effectiveness of the power supplies. The data sheets for the power supply modules list their own power requirements and power dissipation (as maximum power consumption). With the formulas in the following sections, the exact power consumption can also be calculated. This calculation is explained using an example.

The following image shows where the power supply module uses power for its own requirements. It also shows where the power supply module uses power to supply the system and where power dissipation occurs.

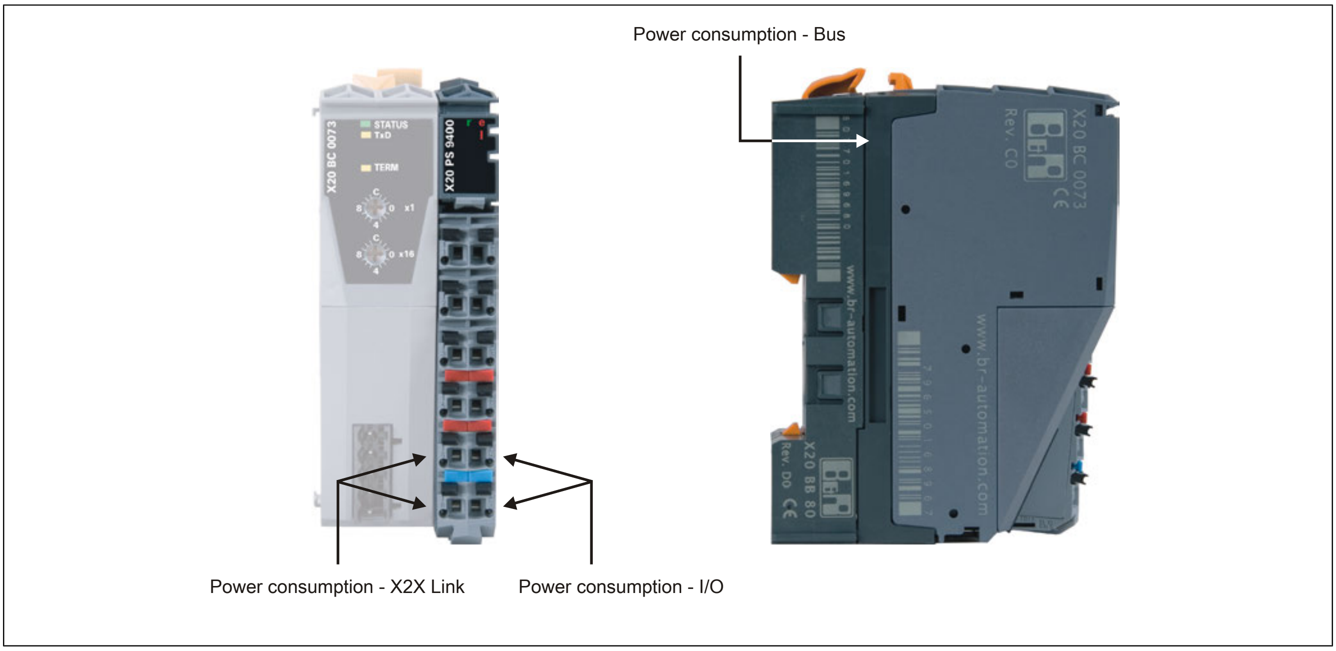


Figure 37: Power supply modules draw power at up to 3 supply points

2.4.13.1 Power consumption of power supply modules

The following table provides an overview of the power consumed by the power supply modules. The exact power requirement can be calculated using the specified formulas.

Module	Bus			Power consumption - I/O-internal [W]
	Power consumption [W]	Power consumption - X2X Link (internal) [W]	Power consumption - Total [W]	
X20PS3300, X20PS9400, X20PS9500, X20PS9600, X20CP1483, X20CP1483-1, X20CP158x, X20CP358x	0.2	$0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$1.42\geq 0.2+0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20PS3310	0.2	$0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$1.42\geq 0.2+0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$0.82 \geq 0.1 + I_{IO}^2 \times 0.02$
X20BR9300	0.4	$0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$1.62\geq 0.4+0.8+\frac{0.06\cdot\sum P_{X2X\,X20}}{n}$	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20PS9402, X20PS9502, X20PS9602	0.2	$0.6+\frac{0.12\cdot\sum P_{X2X\,X20}}{n=1}$	$1.64\geq 0.2+0.6+\frac{0.12\cdot\sum P_{X2X\,X20}}{n=1}$	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20PS2100	0.2	-	0.2	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20PS2110	0.2	-	0.2	$0.82 \geq 0.1 + I_{IO}^2 \times 0.02$
X20BT9100	0.5	-	0.5	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20BT9400	0.5	$0.5+\frac{0.147\cdot\sum P_{X2X\,X67}}{n=1}$	$1.88\geq 0.5+0.5+\frac{0.147\cdot\sum P_{X2X\,X67}}{n=1}$	$0.6 \geq 0.1 + I_{IO}^2 \times 0.005$
X20PS8002	-	$0.5+\frac{0.12\cdot P_{Out}}{n=1}$	$1.34\geq 0.5+\frac{0.12\cdot P_{Out}}{n=1}$	-
X20PD2113	0.12	-	0.12	$1.15 \geq 0.28 + I_{IO}^2 \times 0.02$

$\Sigma P_{X2X\,X20}$ ... Sum of the bus power consumption of all modules in the X20 system (Compact-S CPU, Compact CPU, Fieldbus CPU, BC, BR, I/O, BM, BT)

$\Sigma P_{X2X\,X67}$ ... Sum of the bus power consumption of all I/O modules in the X67 system

P_{Out} ... Sum of the power consumption of all modules (HB) supplied by the power supply module

n ... Number of all power supply modules in the X20 system with X2X Link power supply, including X20BR9300

$I_{I/O}$ ... I/O summation current of all I/O modules supplied by this power supply module (max. 10 A)

X20PS2110 and X20PS3110:
The summation current of these modules is not permitted to exceed 6 A.

X20PD2113:
If the module is used as a power supply module for the I/O power supply, I_{IO} corresponds to the summation current of all I/O modules supplied by the X20PD2113 (max. 10 A).

2.4.13.2 Example

Calculation of the total internal power consumption of bus receiver X20BR9300 using the following hardware configuration:

Module	Bus module - Power	Bus power [W]	I/O-internal power [W]
X20BR9300		0	0
X20DI4371	0.13	0.14	0.59
X20DI2371	0.13	0.12	0.29
X20DO4322	0.13	0.16	0.49
X20DO4322	0.13	0.16	0.49
X20BT9100	0.13	0.50	0.10
Sum	**0.65**	**1.08**	**1.96**

2 power values have to be calculated in order to determine the entire internal power consumption of the bus receiver.

- Internal X2X Link power consumption of the X20BR9300
- Internal I/O power consumption of the X20BR9300

2.4.13.2.1 Calculating the internal X2X Link power consumption of the X20BR9300

Bus power consumption of all modules in the X20 system

In order to calculate the internal X2X Link power consumption of the X20BR9300, the sum of the bus power consumption of all modules in the X20 system is required.

The sum for the example configuration is calculated using the following formula: The bus module of the X20BR9300 does not have to be taken into account in the calculation. The power consumption of the bus module is already included with a factor of 0.8 (see formula below).

A power consumption of 0.13 W for each bus module must be included in the calculation for the 4 I/O modules and the bus transmitter.

$$\sum P_{X2X_{X20}} = P_{X2XBus_{BR9300}} + \sum P_{X2XBus_{IOMod}} + \sum P_{X2XBus_{Busmod}} = 0.4 + 1.08 + 5 \cdot 0.13 = 2.13\,W$$

Internal X2X Link power consumption of the X20BR9300

The internal X2X Link power consumption of the X20BR9300 is calculated using the following formula. Since X2X Link is only supplied by the X20BR9300 bus receiver, the factor is n = 1:

$$P_{X2Xint._{BR9300}} = 0.8 + \frac{0.06 \cdot \sum P_{X2X_{X20}}}{n} = 0.8 + \frac{0.06 \cdot 2.13}{1} = 0.8 + 0.13 = 0.93\,W$$

2.4.13.2.2 Calculating the internal I/O power consumption of the X20BR9300

The I/O summation current of all I/O modules supplied by the X20BR9300 is needed to calculate the internal I/O power consumption. The I/O summation current is composed of 3 parts:

- Internal power consumption of the I/O modules
- Sum of the output currents
- Sum of the actuator currents

Internal power consumption of the I/O modules

The current that results from the internal consumption of the I/O modules is calculated according to the following formula:

$$I_{IO_{int.}} = \frac{P_{IO_{int.}}}{U} = \frac{1.96}{24} = 0.082\,A$$

Sum of output and actuator currents

Two X20DO4322 modules are included in the example configuration. The following images show which outputs are wired and how high the output current and actuator current are per channel.

Connections and currents of the first X20DO4322:

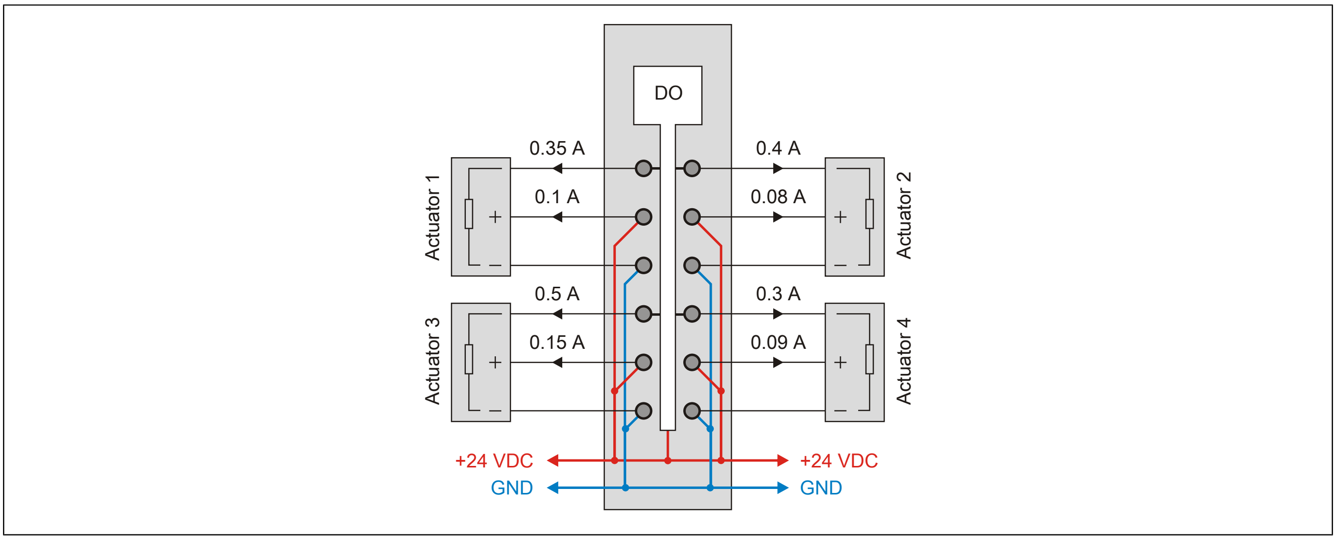


Figure 38: Connections and currents of the first X20DO4322

Connections and currents of the second X20DO4322:

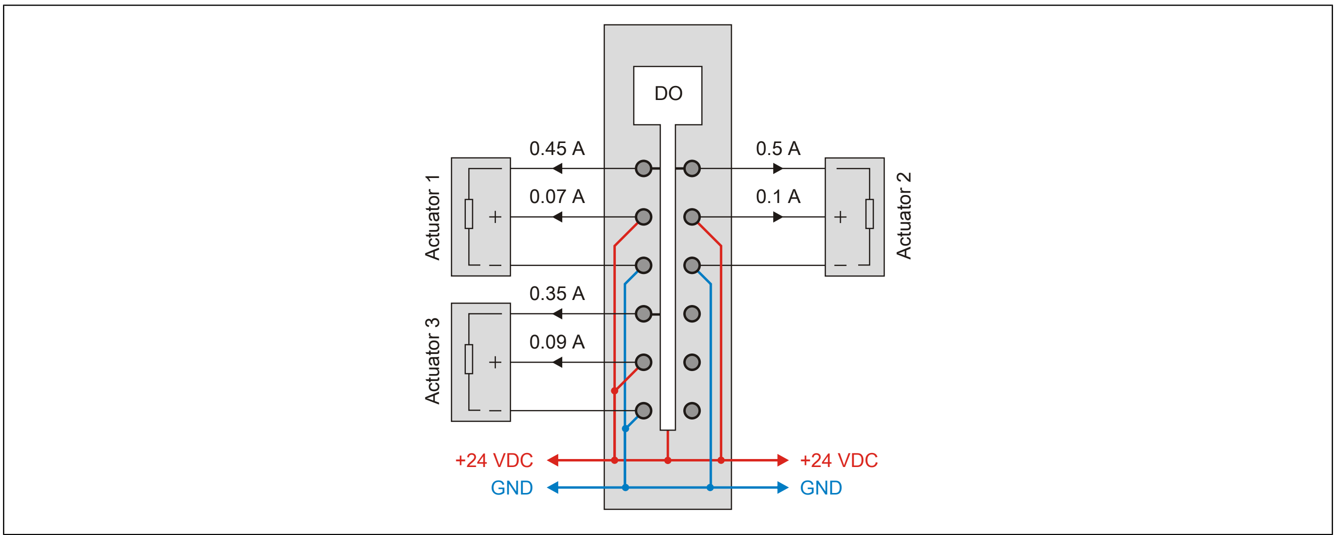


Figure 39: Connections and currents of the second X20DO4322

Calculating the sum of the output currents:

$$I_{DO} = I_{DO_1} + I_{DO_2} = 0.35 + 0.4 + 0.5 + 0.3 + 0.45 + 0.5 + 0.35 = 2.85\,A$$

Calculating the sum of the actuator currents:

$$I_{Actuator} = I_{Actuator_1} + I_{Actuator_2} = 0.1 + 0.08 + 0.15 + 0.09 + 0.07 + 0.1 + 0.09 = 0.68\,A$$

Calculating the I/O summation current

The I/O summation current is calculated from the sum of all 3 partial currents.

$$I_{IO} = I_{IO_{int.}} + I_{DO} + I_{Actuator} = 0.082 + 2.85 + 0.68 = 3.612\,A$$

Calculating the internal I/O power consumption of the X20BR9300

The internal I/O power consumption is calculated using the following formula:

$$P_{IOint._{BR9300}} = 0.1 + I_{IO}^2 \cdot 0.005 = 0.1 + 3.612^2 \cdot 0.005 = 0.17\,W$$

2.4.13.2.3 Total internal power consumption of the X20BR9300

The following 3 power values must be added together to calculate the total internal power consumption of the X20BR9300:

- Power consumption - Bus
- Power consumption - X2X Link (internal)
- Power consumption - I/O-internal

$$P_{BR9300int._{Tot}} = P_{X2XBus_{BR9300}} + P_{X2Xint._{BR9300}} + P_{IOint._{BR9300}} = 0.4 + 0.93 + 0.17 = 1.5\,W$$

2.4.14 Calculation of the additional power dissipation resulting from actuators

Calculation of power dissipation when specifying $R_{DS(on)}$

Explaining output load with an X20DO4332 example

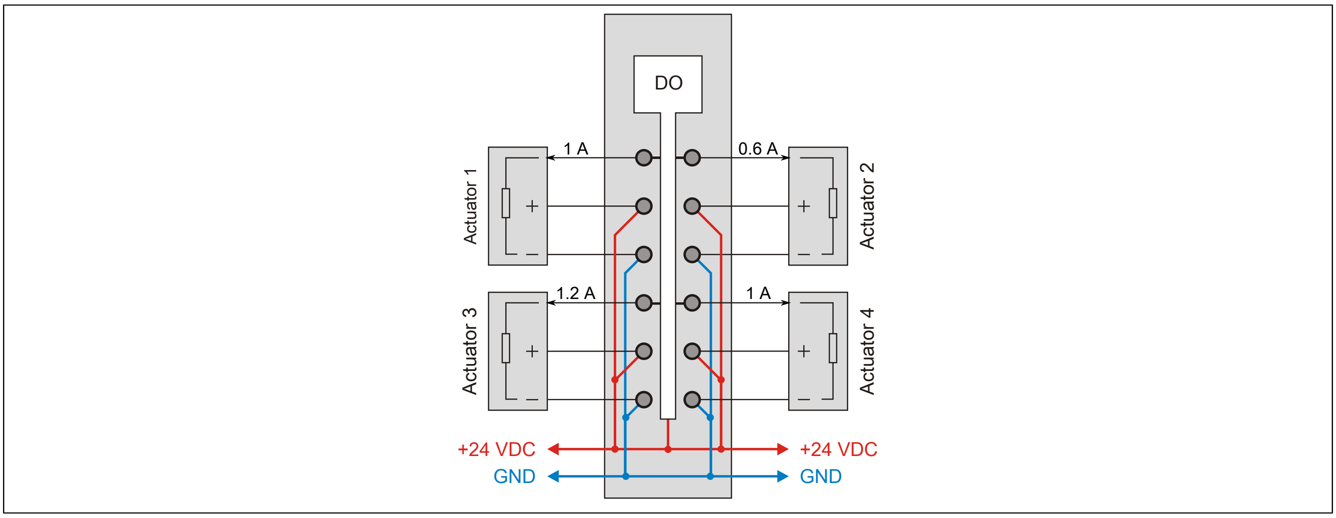


Figure 40: Calculation of power dissipation when specifying $R_{DS(on)}$

Theoretically highest power dissipation resulting from actuators:

Number of outputs * RDS_{on} * nominal output current^2 = power dissipation

4 * 140 mΩ * 2 A^2 = 2.24 W

Power dissipation resulting from actuators in this example:

140 mΩ * (1 A^2 + 0.6 A^2 + 1.2 A^2 + 1 A^2) = 0.532 W

Power dissipation calculation when specifying the residual voltage

Explaining output load with an X20DO4623 example

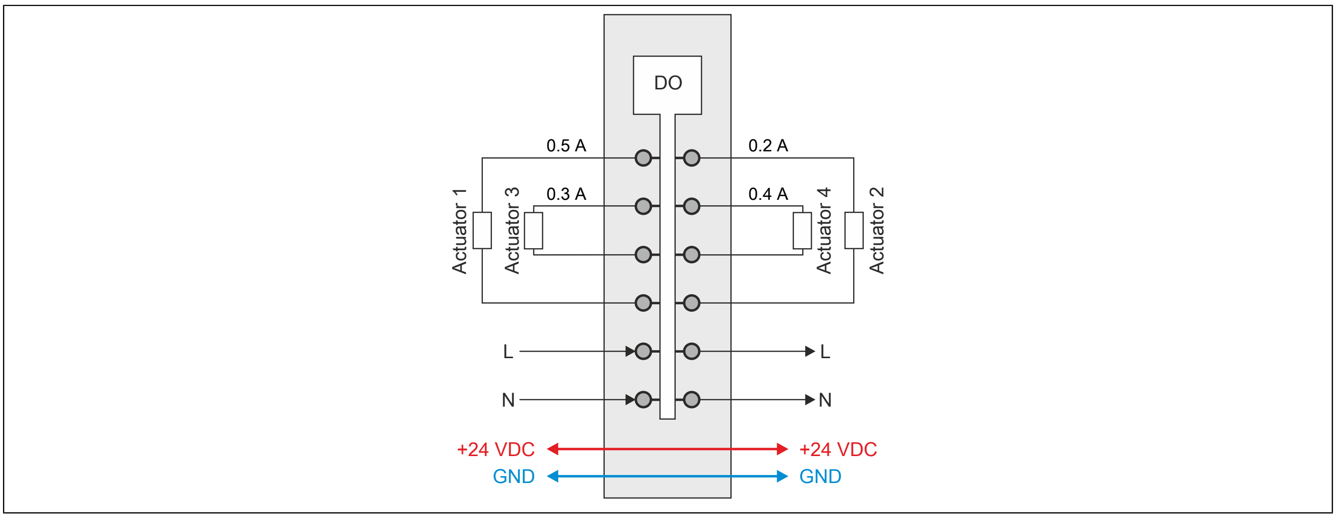


Figure 41: Power dissipation calculation when specifying the residual voltage

Theoretically highest power dissipation resulting from actuators:

Number of outputs * residual voltage * nominal output current = power dissipation

4 * 1.6 V * 0.5 A = 3.2 W

Power dissipation resulting from actuators in this example:

1.6 V * (0.5 A + 0.2 A + 0.3 A + 0.4 A) = 2.24 W

Power dissipation calculation when specifying the contact resistance

Explaining output load with an X20DO4649 example

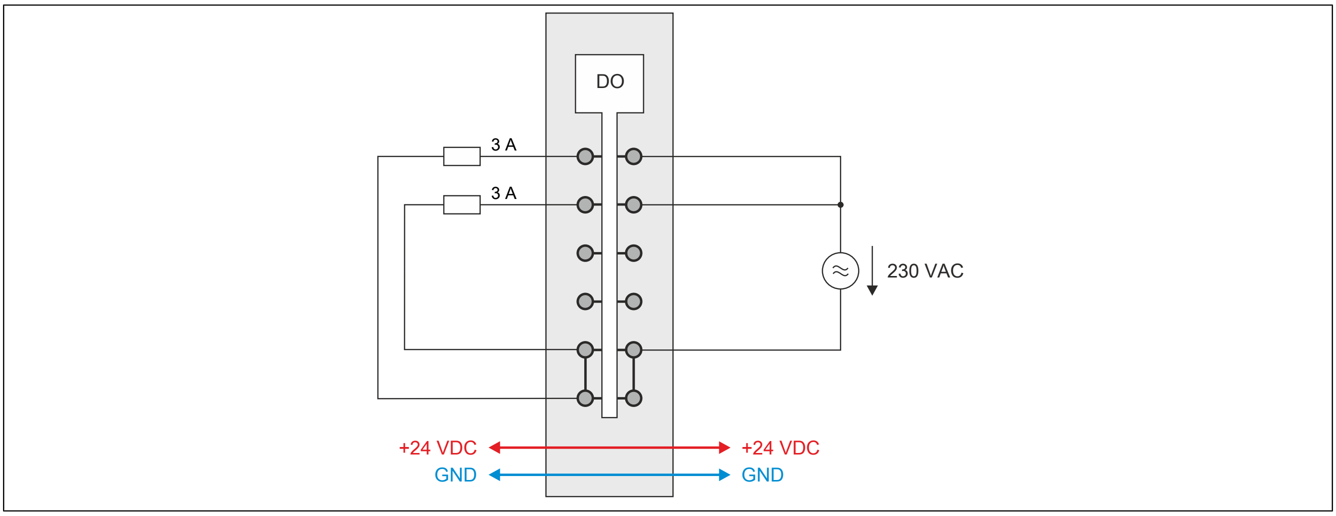


Figure 42: Power dissipation calculation when specifying the contact resistance

Theoretically highest power dissipation resulting from actuators:

Number of outputs * contact resistance * nominal output current² = power dissipation

4 * 15 mΩ * 5 A^2 = 1.5 W

Power dissipation resulting from actuators in this example:

15 mΩ * (3 A^2 + 3 A^2) = 0.27 W

2.4.15 Dimensioning the external 24 VDC power supply

X20 systems are provided with external 24 VDC power supplies. The following examples illustrate how to determine the power to be provided.

2.4.15.1 X20BRx300 bus receivers and X20PS33xx power supply modules

Calculation example with bus receiver X20BR9300

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Example for calculating the power requirements: See "Example: Potential groups" on page 104.	2.73 W	188.77 W
Power consumption of the X20BR9300	For the value, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.62 W	-
Subtotal		4.35 W	188.77 W
Total power requirements		193.12 W	

The external 24 VDC power supply must provide 193.12 W.

2.4.15.2 Power supply modules X20PS9400 and X20PS9402

Calculation example with X20PS9400, X20BC0083 and X20BB80

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Example for calculating the power requirements: See "Example: Bus controller and modules" on page 103. In the example for calculating the power requirements, expandable bus controller X20BC8083 is used with 1-hub expansion module X20HB2880. If X20BC0083 is used, there is no need for a 1-hub expansion module and the following values are included in the calculation: • X20BB80: -0.25 W • X20BC0083: -2 W	4.47 W	235.94 W
Power consumption of the X20PS9400	For the value, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	-
Subtotal		5.89 W	235.94 W
Total power requirements		241.83 W	

The external 24 VDC power supply must provide 241.83 W.

Calculation example with X20PS9400, X20BC8083, X20HB2880 and X20BB81

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Example for calculating the power requirements: See "Example: Bus controller and modules" on page 103.	5.89 W	235.94 W
Power consumption of the X20PS9400	For the value, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	-
Subtotal		7.31 W	235.94 W
Total power requirements		243.25 W	

The external 24 VDC power supply must provide 243.25 W.

2.4.15.3 CPUs X20CP1483 and X20CPx58x

Calculation example with X20CP3585 and 3 interface modules

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Example for calculating the power requirements: See "Example: CPU and modules" on page 102.	2.13 W	78.73 W
Power consumption of X20CP3585 without interface module and USB	For the value, see the technical data in the data sheet: General information - Power consumption without interface module and USB	8.8 W	-
Power consumption for generating the X2X Link power supply	For the value, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	-
2x USB interface	3 W must be taken into account for each USB interface used	6 W	-
Power consumption of the X20IF1091	For the value, see the technical data in the data sheet: General information - Power consumption	0.97 W	-
Power consumption of the X20IF10E1-1		2 W	-
Power consumption of the X20IF10E3-1		2 W	-
Power consumption of X20CP3585 per interface module	The CPU consumes 0.6 W per interface module. The power consumption of 3 interface modules is: 3 x 0.6 W = 1.8 W	1.8 W	-
Subtotal		25.12 W	78.73 W
Total power requirements		103.85 W	

The external 24 VDC power supply must provide 103.85 W.

2.4.15.4 Compact-S CPUs X20CP04xx

2.4.15.4.1 Compact-S CPU without an interface module

The first part of this example shows the calculation of the power requirements for the bus and I/O power supply of a Compact-S CPU without an interface module. Dimensioning the external 24 VDC power supply is explained in the second part.

Power supplied by the power supply module

Module	Power supplied to the bus	Power supplied to the I/O power supply
X20PS9600	+7 W	+240 W

Power requirements of the Compact-S CPU

Module	Bus power supply requirements	I/O power supply requirements
X20PS9600	-	-0.6 W
X20BB52	-0.55 W	-
X20CP0410	-2.2 W	-
2x USB interface	-2 W[1)]	-
Total power requirements	-4.75 W	-0.6 W

1) 2 x 5 V x 0.2 A = 2 W

Power balance for the bus supply

The power requirements of the Compact-S CPU are completely covered by the bus power supply. In a power balance, it must be checked whether the power supply module covers the power requirements of the Compact-S CPU.

	Bus power supply
Power supplied by the power supply module	+7 W
Total power requirements	-4.75 W
Remaining power	+2.25 W

The power comparison indicates that the power provided by the power supply module is sufficient. Additional power supply modules are not necessary.

External 24 VDC power supply

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Examples for calculating the power requirements: See "Calculating the power requirements" on page 101. The following values are assumed for this example: • Bus power supply: 3.67 W • I/O power supply: 192.51 W	3.67 W	192.51 W
Power consumption of the X20PS9600	For the values, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	0.6 W
Power requirements of the Compact-S CPU	The values calculated in section "Power requirements of the Compact-S CPU" on page 116 are used for the bus and I/O power supply.	4.75 W	-
Subtotal		9.84 W	193.11 W
Total power requirements		202.95 W	

The external 24 VDC power supply must provide 202.95 W.

2.4.15.4.2 Compact-S CPU with 1 interface module

The first part of this example shows the calculation of the power requirements for the bus and I/O power supply of a Compact-S CPU with an interface module. Dimensioning the external 24 VDC power supply is explained in the second part.

Power supplied by the power supply module

Module	Power supplied to the bus	Power supplied to the I/O power supply
X20PS9600	+7 W	+240 W

Power requirements of the Compact-S CPU

Module	Bus power supply requirements	I/O power supply requirements
X20PS9600	-	-0.6 W
X20BB62	-0.94 W	-
X20CP0410	-2.2 W	-
2x USB interface	-2 W[1)]	-
X20IF1063-1	-1.8 W	-
Total power requirements	-6.94 W	-0.6 W

1) 2 x 5 V x 0.2 A = 2 W

Power balance for the bus supply

The power requirements of the Compact-S CPU are completely covered by the bus power supply. In a power balance, it must be checked whether the power supply module covers the power requirements of the Compact-S CPU.

	Bus power supply
Power supplied by the power supply module	+7 W
Total power requirements	-6.94 W
Remaining power	+0.06 W

The power comparison indicates that the power provided by the power supply module is sufficient. Additional power supply modules are not necessary.

External 24 VDC power supply

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Examples for calculating the power requirements: See "Calculating the power requirements" on page 101. The following values are assumed for this example: • Bus power supply: 3.67 W • I/O power supply: 192.51 W	3.67 W	192.51 W
Power consumption of the X20PS9600	For the values, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	0.6 W
Power requirements of the Compact-S CPU	The values calculated in section "Power requirements of the Compact-S CPU" on page 117 are used for the bus and I/O power supply.	6.94 W	-
Subtotal		12.03 W	193.11 W
Total power requirements		205.14 W	

The external 24 VDC power supply must provide 205.14 W.

2.4.15.4.3 Compact-S CPU with 2 interface modules

Calculating the power requirements for the bus and I/O power supply of a Compact-S CPU with 2 interface modules is shown in the first part of this example. Dimensioning the external 24 VDC power supply is explained in the second part.

Power supplied by the power supply module

Module	Power supplied to the bus	Power supplied to the I/O power supply
X20PS9600	+7 W	+240 W

Power requirements of the Compact-S CPU

Module	Bus power supply requirements	I/O power supply requirements
X20PS9600	-	-0.6 W
X20BB72	-1.17 W	-
X20CP0410	-2.2 W	-
2x USB interface	-2 W[1)]	-
X20IF1043-1	-1.1 W	-
X20IF1063-1	-1.8 W	-
Total power requirements	-8.27 W	-0.6 W

1) 2 x 5 V x 0.2 A = 2 W

Power balance for the bus supply

The power requirements of the Compact-S CPU are completely covered by the bus power supply. In a power balance, it must be checked whether the power supply module covers the power requirements of the Compact-S CPU.

	Bus power supply
Power supplied by the power supply module	+7 W
Total power requirements	-8.27 W
Remaining power	-1.27 W

The power comparison indicates that the power provided by the power supply module is insufficient. An additional X20PS3300 power supply module is required (see "Hardware configuration" on page 119).

External 24 VDC power supply

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Examples for calculating the power requirements: See "Calculating the power requirements" on page 101. The following values are assumed for this example: • Bus power supply: 3.67 W • I/O power supply: 192.51 W	3.67 W	192.51 W
Power consumption of the X20PS9600	For the values, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	0.6 W
Power requirements of the Compact-S CPU	The values calculated in section "Power requirements of the Compact-S CPU" on page 118 are used for the bus and I/O power supply.	8.27 W	-
Subtotal		13.36 W	193.11 W
Total power requirements		206.47 W	

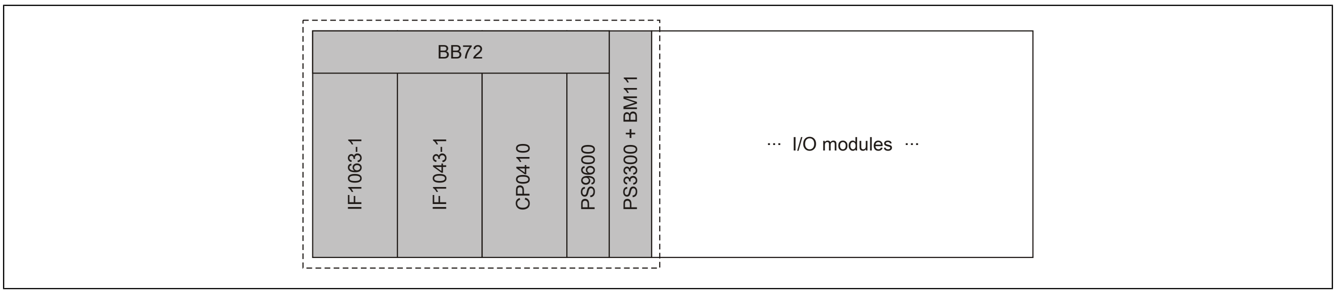
The external 24 VDC power supply must provide 206.47 W.

Hardware configuration

To provide sufficient power for the bus power supply, X20PS3300 power supply modules can be connected in parallel via bus module X20BM11. It is important to note that in order to determine the necessary bus power supply in parallel operation, 75% of the nominal power of the power supply modules must be calculated.

Example for calculating the bus power for 1 X20PS9600 and 1 X20PS3300:

Bus power = 2 x 7 W x 0.75 = 10.5 W



Connection example for power supply modules

X20PS9600

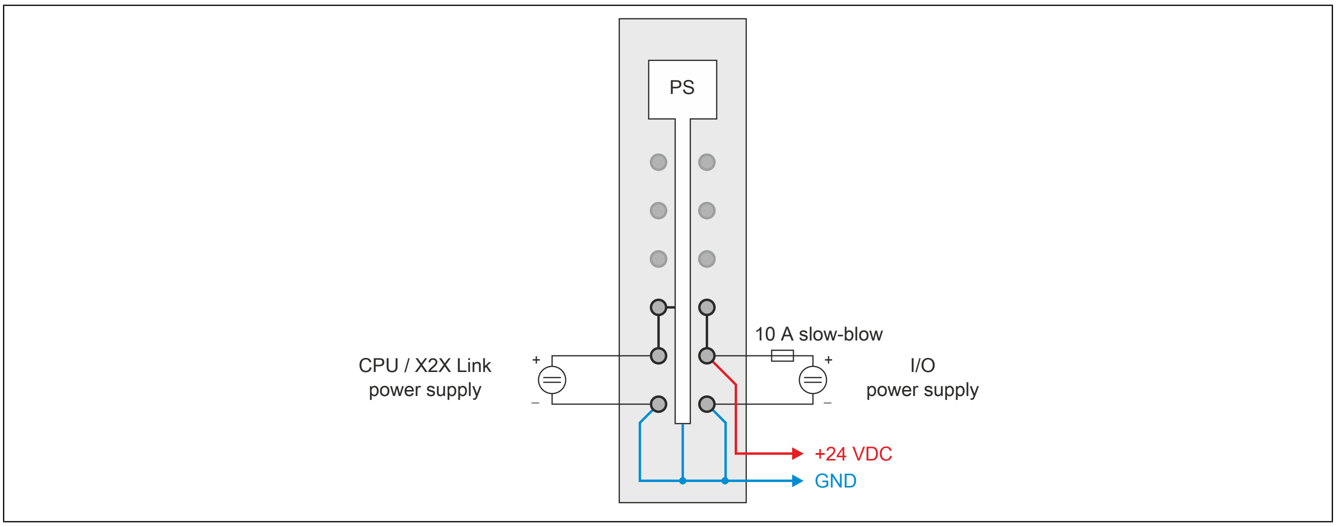


Figure 43: The X20PS9600 is connected as usual.

X20PS3300

Power supply module X20PS3300 is operated with bus module X20BM11. Only the CPU / X2X Link power supply is connected. By using bus module X20BM11, the I/O power supply of power supply module X20PS9600 is connected through to the I/O modules.

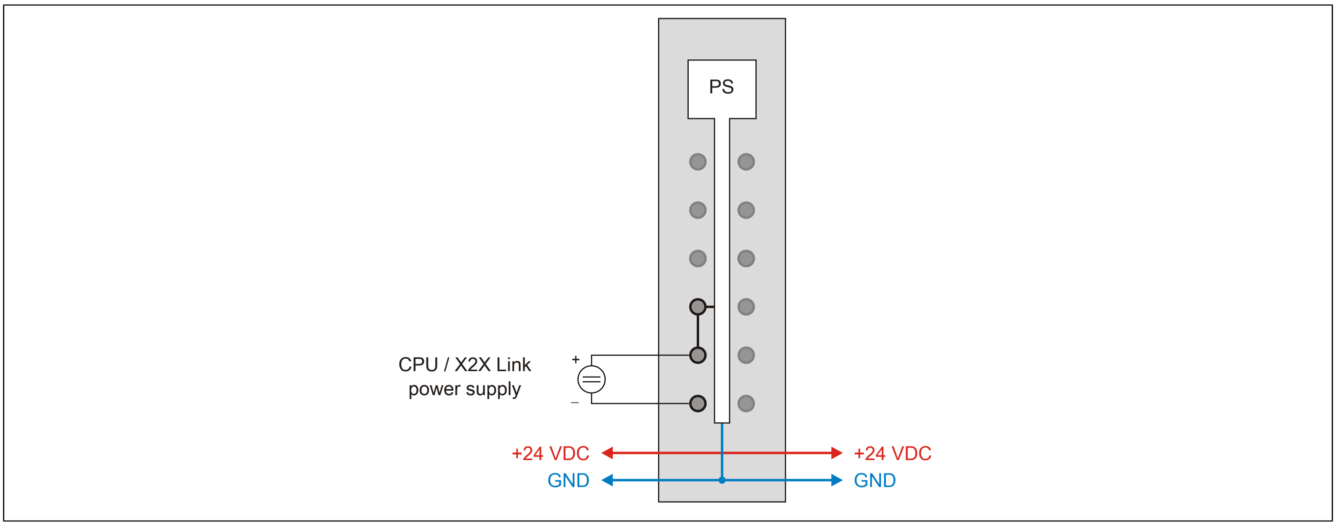


Figure 44: Only the CPU / X2X Link power supply is connected for the X20PS3300.

2.4.15.5 SafeLOGIC controller X20SL81xx

2.4.15.5.1 SafeLOGIC controller X20SL8100

For dimensioning the external 24 VDC power supply, only the power consumption of the SafeLOGIC controller must be taken into account.

Power	Description	Requirements
Power consumption of X20SL8100	For the value, see the technical data in the data sheet: General information - Power consumption	4.3 W

The external 24 VDC power supply must provide 4.3 W.

2.4.15.5.2 SafeLOGIC controller X20SL8110

Calculation example with interface module X20IF10E3-1

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Requirements
Power consumption of X20SL8110	For the value, see the technical data in the data sheet: General information - Power consumption	3.9 W
Power consumption of the X20IF10E3-1	For the value, see the technical data in the data sheet: General information - Power consumption	2 W
Power consumption of the X20SL8110 for the interface module	The CPU consumes 0.6 W for the operation of the interface module.	0.6 W
Total power requirements		6.5 W

The external 24 VDC power supply must provide 6.5 W.

2.4.15.5.3 SafeLOGIC controller X20SL8101

For dimensioning the external 24 VDC power supply, the following power values are included in the calculation:

Power	Description	Bus power supply	I/O power supply
Power consumption of the bus and I/O modules	Example for calculating the power requirements: See "Example: CPU and modules" on page 102	2.13 W	78.73 W
Power consumption of X20SL8101	For the value, see the technical data in the data sheet: General information - Power consumption	5.3 W	-
Power consumption for generating the X2X Link power supply	For the value, see the technical data in the data sheet: General information - Power consumption of the X2X Link power supply	1.42 W	-
Subtotal		8.85 W	78.73 W
Total power requirements		87.58 W	

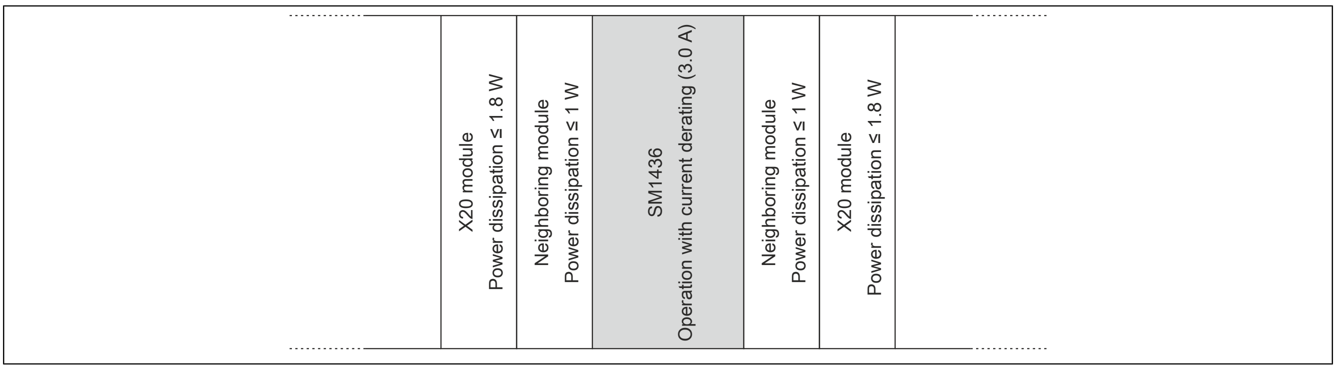
The external 24 VDC power supply must provide 87.58 W.

2.4.16 Power dissipation calculation of I/O modules

Some modules require that neighboring modules are not permitted to exceed a certain power dissipation during operation.

2.4.16.1 Example: Operation of module X20SM1436

The power dissipation of the immediate neighboring modules of the SM module is not permitted to exceed 1 W. A maximum power dissipation of 1.8 W is permitted for the next modules.



2.4.16.2 Calculation of the power dissipation of I/O modules adjacent to the X20SM1436

The power dissipation of I/O modules consists of the following power values:

- Power consumption - Bus module
- Power consumption - Bus
- Power consumption - I/O-internal
- Power consumption - I/O-external
- Additional power dissipation caused by actuators (resistive)

Immediate neighboring modules

The table describes the calculation of the power dissipation of I/O modules that can be operated directly next to the SM module. The power dissipation of these modules is not permitted to exceed 1 W.

Power value	X20AI2622	X20AT2402	X20DI2653	X20DO4322
Power consumption - Bus module [W]	0.13	0.13	0.13	0.13
Power consumption - Bus [W]	0.01	0.01	0.14	0.16
Power consumption - I/O-internal [W]	0.8	0.72	-	0.49
Power consumption - I/O-external [W]	-	-	0.55	-
Additional power dissipation caused by actuators (resistive) [W]	-	-	-	0.21
Power dissipation of the I/O module [W]	0.94	0.86	0.82	0.99

All modules have a power dissipation ≤1 W and can therefore be operated directly next to module X20SM1436.

Modules in the 2nd row

The table describes the calculation of the power dissipation of I/O modules that can be operated in the 2nd row next to the SM module. The power dissipation of these modules is not permitted to exceed 1.8 W.

Power value	X20AI4632	X20AT4222	X20DI8371	X20DO6322
Power consumption - Bus module [W]	0.13	0.13	0.13	0.13
Power consumption - Bus [W]	0.01	0.01	0.18	0.18
Power consumption - I/O-internal [W]	1.5	1.1	-	0.71
Power consumption - I/O-external [W]	-	-	1.2	-
Additional power dissipation caused by actuators (resistive) [W]	-	-	-	0.31
Power dissipation of the I/O module [W]	1.64	1.24	1.51	1.33

All modules have a power dissipation ≤1.8 W and can therefore be operated in the 2nd row next to module X20SM1436.

2.5 Mechanical handling

2.5.1 Solid mechanics

With all the advantages that the three-part modularity of the X20 system offers, one emphasis has always been solid mechanical design.

Its robust design, long guides and strengthened housing guarantee the stability it needs in industrial environments. These features allow the X20 system to be mounted on a top-hat rail with the same ease as a rack system. They also make it just as simple to remove it from the rail.

The following sections describe the mechanical handling of the X20 system step-by-step with the aid of pictures.

2.5.2 Number of connection cycles

The modules of the X20 system are divided into three parts. A module is made up of three basic elements:

- Bus module
- Electronic module
- Terminal block

The number of connection cycles between the respective basic elements is specified at 50.

Basic element	Number of connection cycles
Bus module ↔ Bus module	50
Bus module ↔ Electronic module	
Electronic module ↔ Terminal block	

Table 13: Number of connection cycles between the respective basic elements

2.5.3 Assembling an X20 system

There are several ways to assemble an X20 system. Two methods are described below:

Assembling an X20 system	Description
Variant 1	The X20 system is completely assembled and then installed on the top-hat rail.
Variant 2	The X20 system is installed and assembled directly on the top-hat rail.

Table 14: Two of the several methods for assembling an X20 system

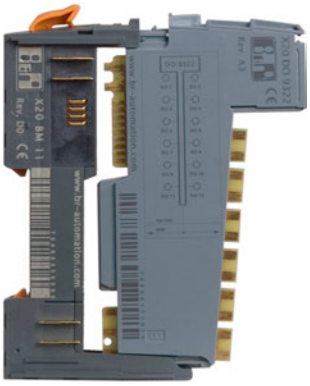
Information:

- **Store X20 modules in the protective packaging until immediately before assembly.**
- **Only touch X20 modules on the housing.**
- **Take the necessary protective measures against electrostatic discharges (see also "Protection against electrostatic discharges" on page 38).**

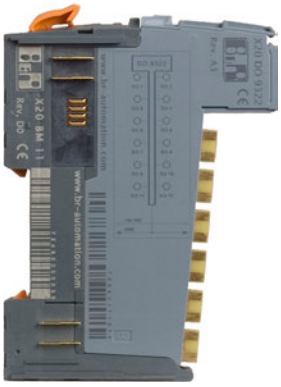
2.5.3.1 Variant 1

The X20 system is completely assembled and then installed on the top-hat rail.

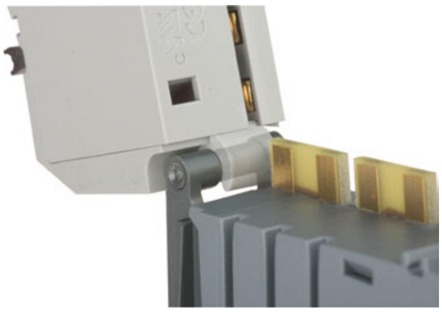
1. Remove X20 modules from protective packaging. Check modules for obvious mechanical damages.
2. Insert electronic module in the guides on the bus module.



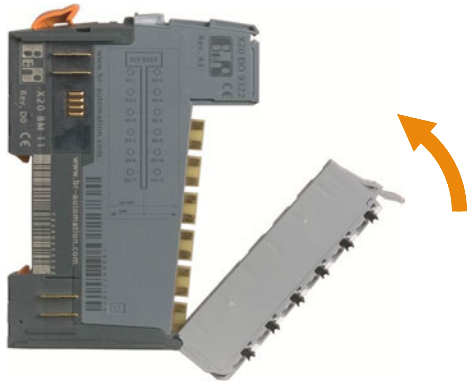
3. Push the electronic module and the bus module flush together.



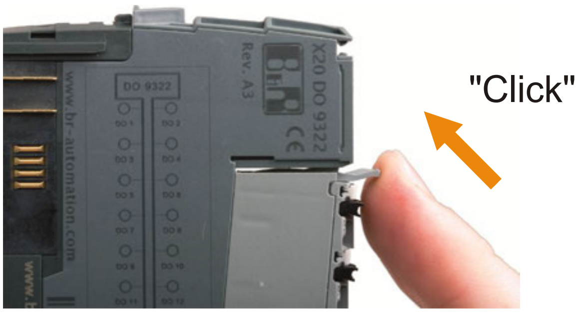
4. Hang the bottom edge of the terminal block in its place on the bus module.



5. Rotate the terminal block up into place.



6. The terminal block latch must close with an audible click. If the latch does not catch, the lever must be pushed up.



7. Individual X20 modules should be assembled from left to right (viewed from front) to form the complete X20 system. To do this, connect the right module from behind to the guides for the left bus module.

8. Slide the right module forward until the two modules fit flush together.
9. Proceed like this until the second to last module.
10. For the last module, only insert the bus module in the guides of the left bus module.
11. Slide the right bus module forward until the two bus modules fit flush together.

12. Insert the right locking plate into the guides on the bus module from the front and push it all the way in.

13. Insert the electronic module into the bus module and push firmly so that the two modules fit flush together.
14. Hang the bottom of the terminal block in its place on the bus module and push it up into place. The terminal block latch must close with an audible click.
15. Lay the left locking plate on the left module and insert it in the guides. Finally, slide the locking plate forward.

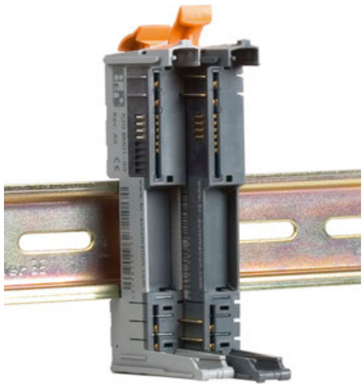
16. The procedure for hanging the X20 system on the top-hat rail is described in section "Installing the X20 system on the top-hat rail" on page 129.

2.5.3.2 Variant 2

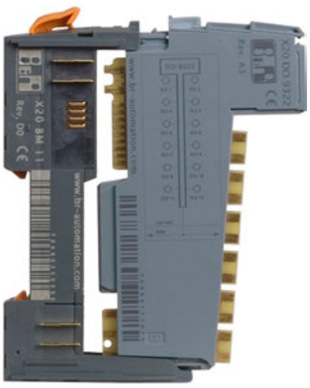
The X20 system is installed and assembled directly on the top-hat rail.

1. Remove X20 modules from protective packaging. Check modules for obvious mechanical damages.
2. Push the locking lever all the way up on all of the bus modules. This opens the locking mechanism for top-hat rail installation.

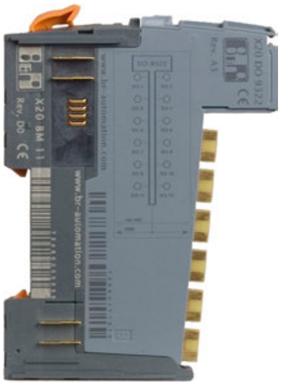
3. Hang the first bus module in the desired position on the top-hat rail and close the locking mechanism by pushing the lever down.
4. Insert the next bus module in the guides of the previously mounted bus module.



5. Slide the bus module in against the top-hat rail and secure it by pushing down the locking lever.
6. Proceed like this with the rest of the bus modules.
7. Insert the corresponding electronic module in the guides on the leftmost bus module.

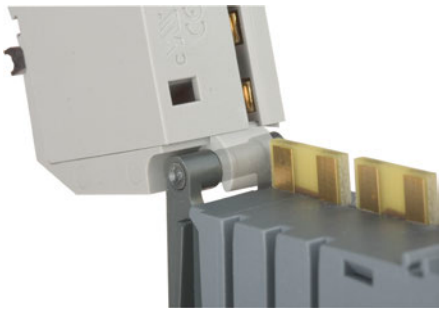


8. Push the electronic module and the bus module flush together.

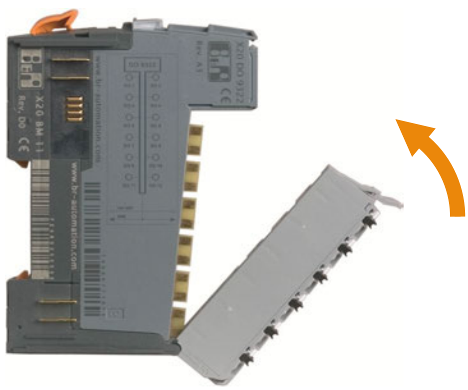


9. Proceed like this until the second to last electronic module.
10. Insert the right locking plate into the guides from the front and push it in all the way.

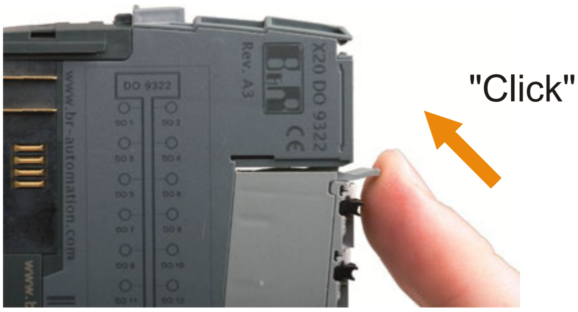
11. Insert the electronic module into the bus module and push firmly so that the two modules fit flush together.
12. Hang the terminal block in its place on the leftmost bus module.



13. Rotate the terminal block up into place.



14. The terminal block latch must close with an audible click. If the latch does not catch, the lever must be pushed up.

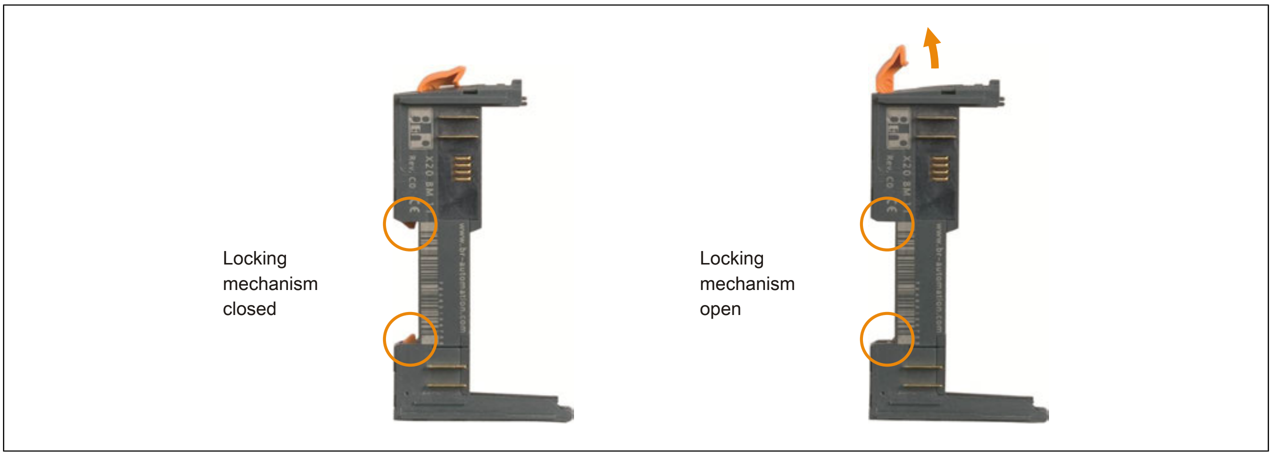


15. Proceed like this with the rest of the terminal blocks.
16. Lay the left locking plate on the left module and insert it in the guides. Finally, slide the locking plate forward.

2.5.4 Installing the X20 system on the top-hat rail

Complete the following steps to install an assembled X20 system on the top-hat rail.

1. Push the locking lever all the way up on all of the bus modules. This opens the locking mechanism for top-hat rail installation.

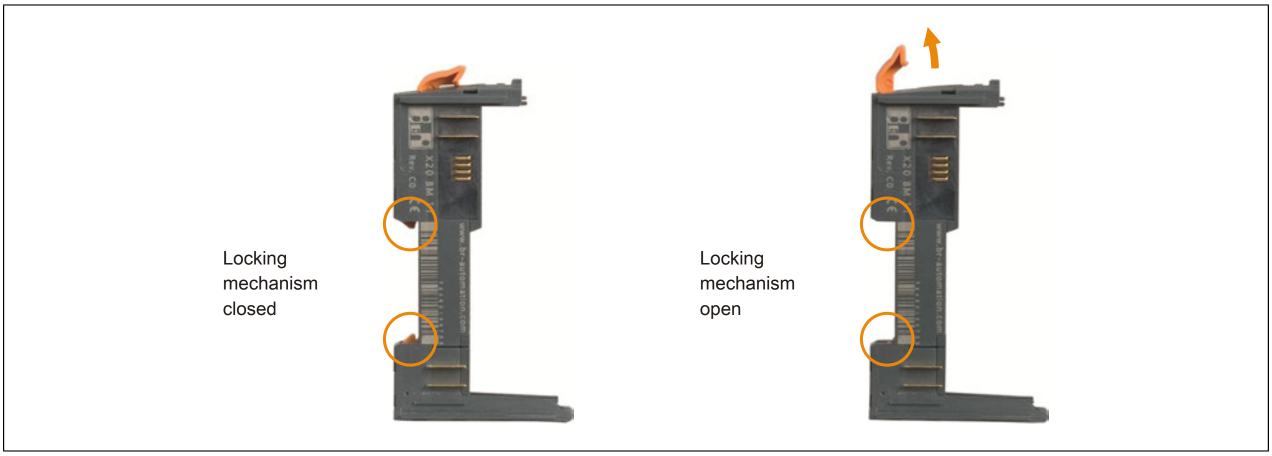


2. Hang the X20 system in the desired position on the top-hat rail and close the locking mechanism by pushing the lever down.

2.5.5 Removing the X20 system from the top-hat rail

2.5.5.1 Remove the entire system from the top-hat rail

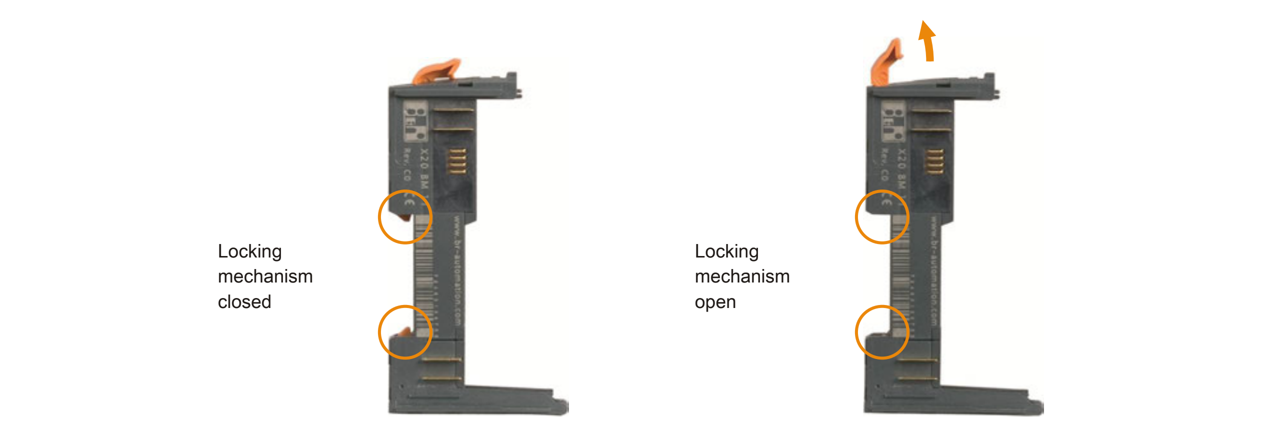
1. Push the locking lever all the way up on all of the bus modules. This opens the locking mechanism for top-hat rail installation.



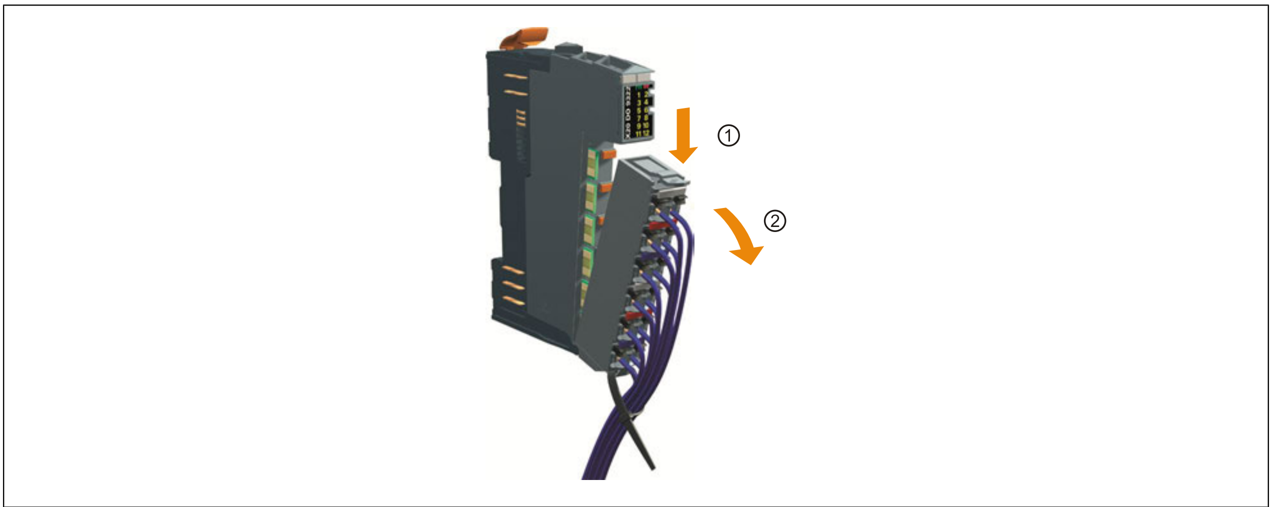
2. Remove the X20 system from the top-hat rail.

2.5.5.2 Removing a block of modules from the top-hat rail

1. Push the locking lever all the way up on all of the modules that you wish to remove from the top-hat rail. This opens the locking mechanism for top-hat rail installation.

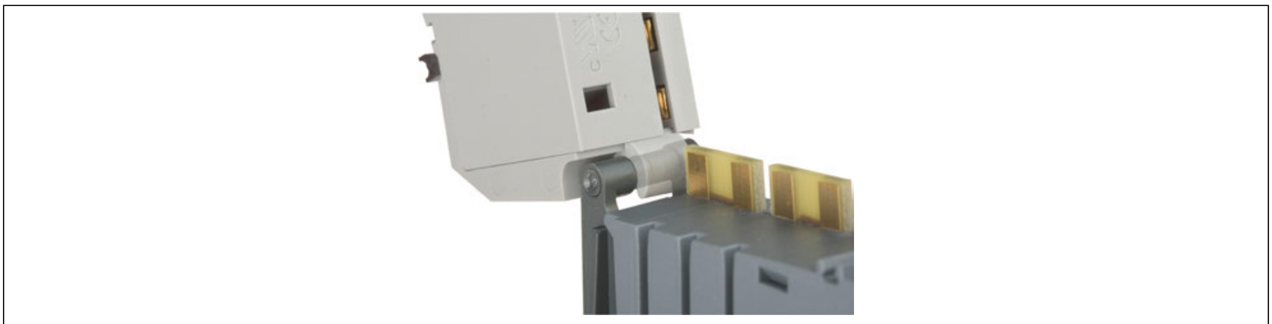


2. The terminal block must be removed from the module to the left of the module block that is to be removed. To do this, push down on the locking lever on the terminal block ① and rotate the terminal block out and down ②.

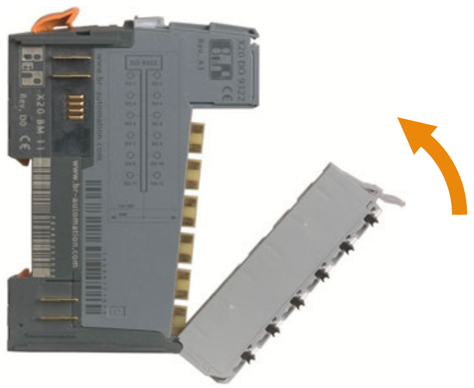


3. Remove the module block from the top-hat rail.

4. Put the removed terminal block back on the module. To do this, hang the bottom in place in the bus module.



5. Rotate the terminal block up into place.

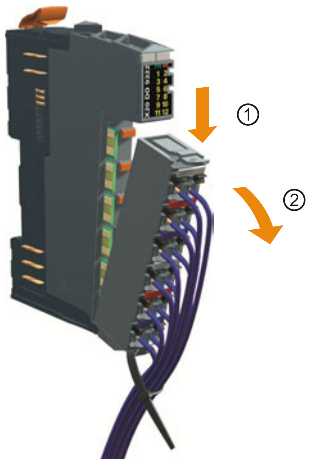


6. The terminal block latch must close with an audible click. If the latch does not catch, the lever must be pushed up.

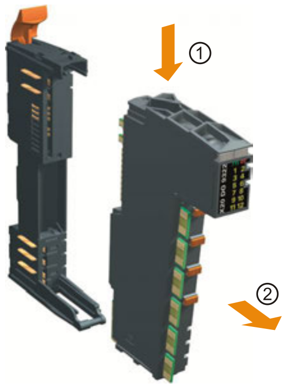
2.5.6 Expanding an X20 system

If you want to expand an existing X20 system to the right, the right locking plate must be removed.

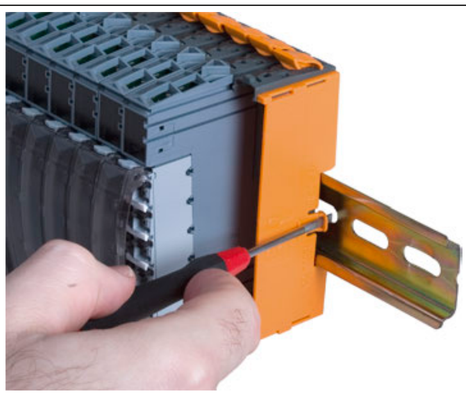
1. Remove the terminal block from the rightmost module. To do this, push down on the locking lever on the terminal block ① and rotate the terminal block out and down ②.



2. Push down on the electronic module's locking lever ① and remove the electronic module ②.



3. Use a screwdriver to lift the locking lever of the right locking plate and pull the locking plate off of the bus module.



4. Additional modules can now also be installed as described in assembly method 2 (see "Variant 2" on page 126).

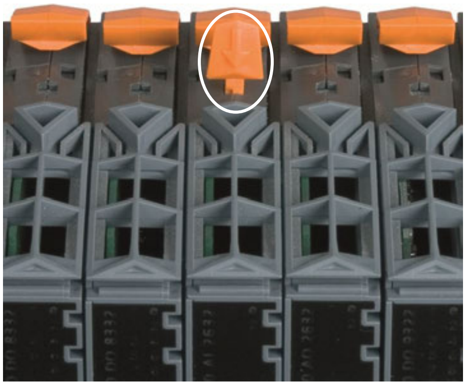
2.5.7 Installing accessories

2.5.7.1 Additional locking mechanisms

Some specific areas require additional locking mechanisms to prevent accidental release of the mechanical components.

2.5.7.1.1 Accessory locking clips

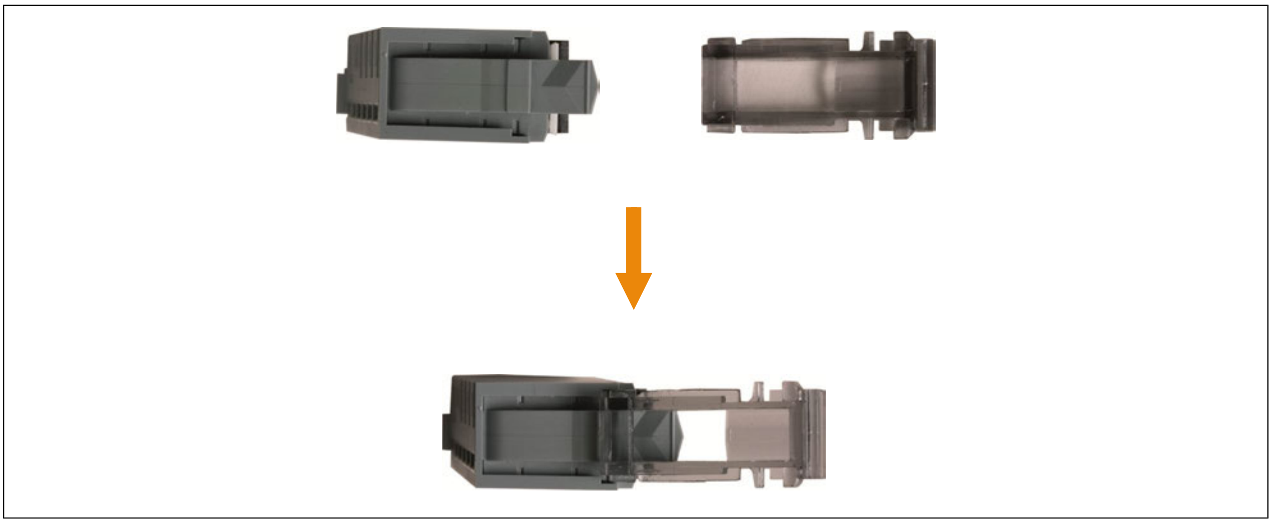
The accessory locking clip attaches the electronic module to the bus module. The locking clip is inserted in the appropriate opening on the module and pushed down.



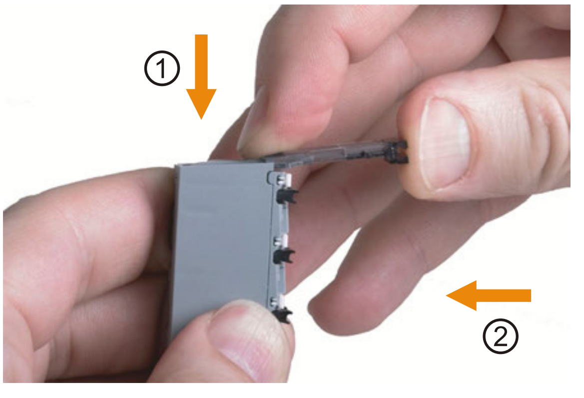
2.5.7.1.2 Terminal locking clip

The terminal locking clip attaches the terminal block securely to the electronic module.

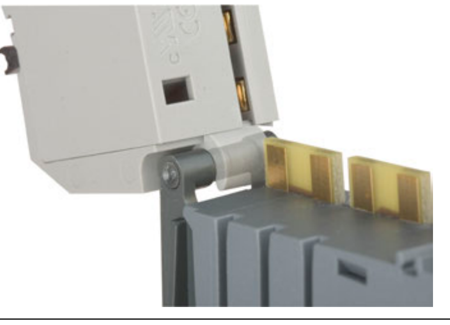
1. Set the terminal locking clip on the terminal block locking lever as shown.



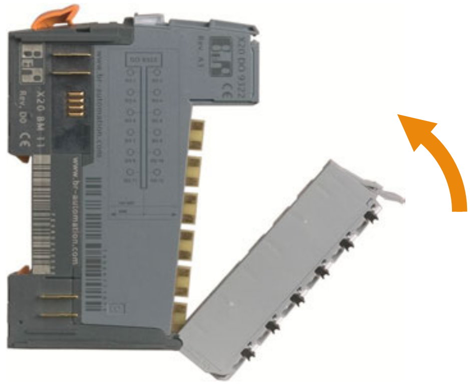
2. Push down and hold the terminal locking clip and the locking lever with your index finger ①. Finally, slide the terminal locking clip forward with your thumb ②.



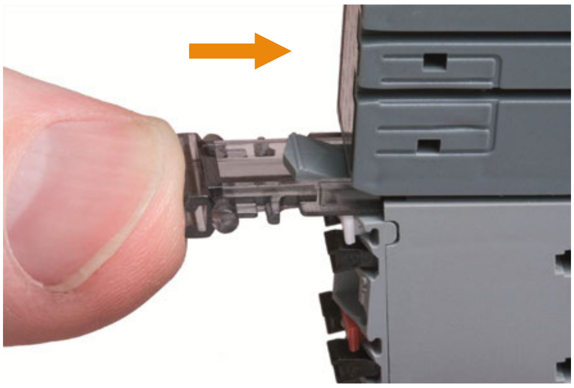
3. Hang the bottom edge of the terminal block in its place on the bus module.



4. Rotate the terminal block up into place.



5. Secure the terminal block in the electronic module by pushing in the terminal locking clip.



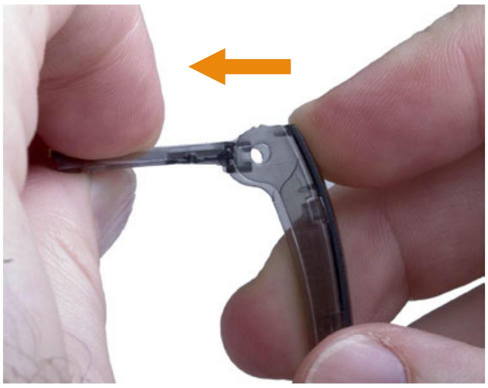
6. Installed terminal locking clip.

7. To remove the terminal block, pull the terminal locking clip out again.

2.5.7.2 Plain text tag for X20 modules

Tags are available for X20 modules into which plain text slide-in labels can be inserted. The tags are attached to the terminal locking clips.

1. Hold the plain text tag at a 90° angle to the terminal locking clip.
2. Push the plain text tag into the terminal locking clip's slot until it clicks into place.

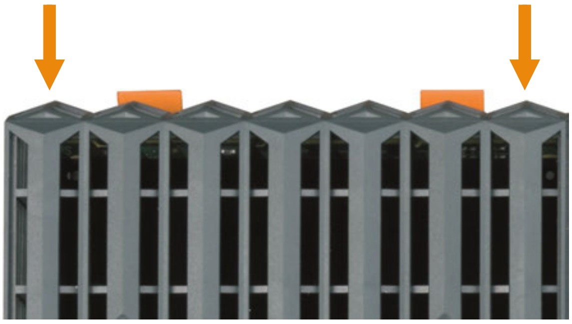


2.5.7.3 Plain text tag for X20 CPUs

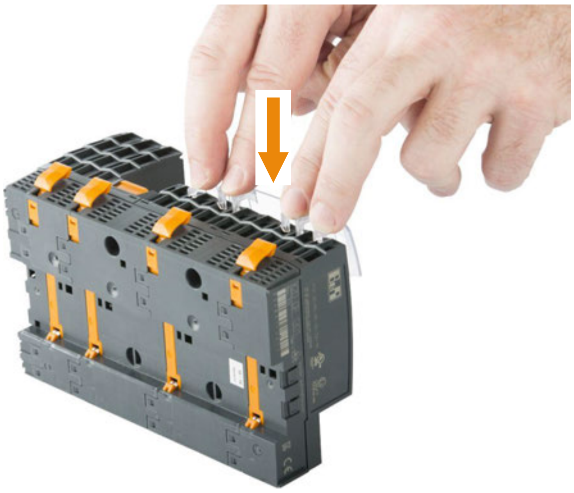
Tags are available for X20 CPUs into which plain text slide-in labels can be inserted. The tags are placed on the housing of the CPU.

Installation

1. Place the plain text tag on the housing so that the outer clips rest on raised edges.

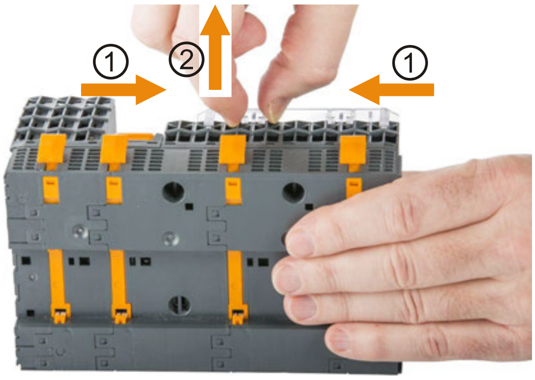


2. Press the plain text tag down until the clips latch into place.



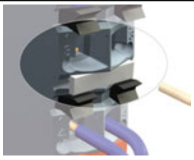
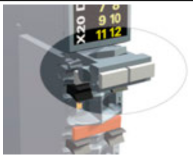
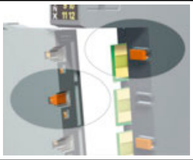
Removal

1. Pinch the clips together ① to release them while lifting the tag off of the housing ②.



2.5.8 Label tags

Label tags can serve the following purposes:

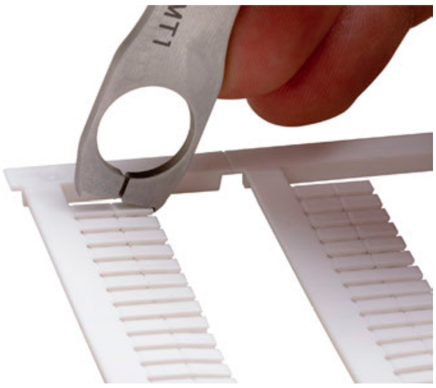
	Labeling the terminal connection		Labeling the module
	Labeling the terminal blocks		Labeling the terminals

The labeling tool is needed to attach the label tags.

2.5.8.1 Labeling the terminal connection

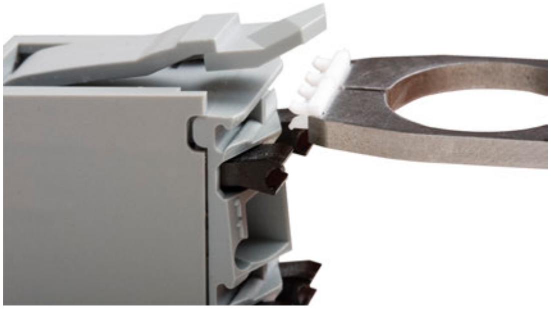
This section explains how to label the terminal connection. The terminal connection, terminal blocks and modules are labeled in a similar manner.

1. Grip the desired label tags with the double-width cutters of the labeling tool.

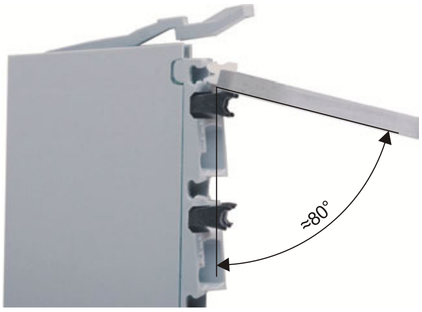


2. Press with the labeling tool to separate the label tags.

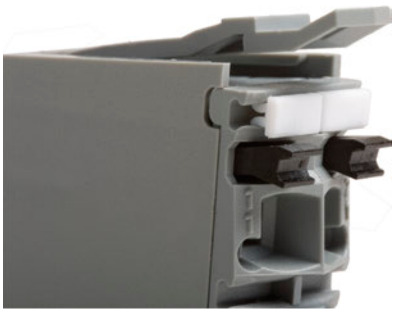
3. Center the label tags over the slot on the terminal block.



4. Hold the labeling tool at approximately an 80° angle to the terminal block.



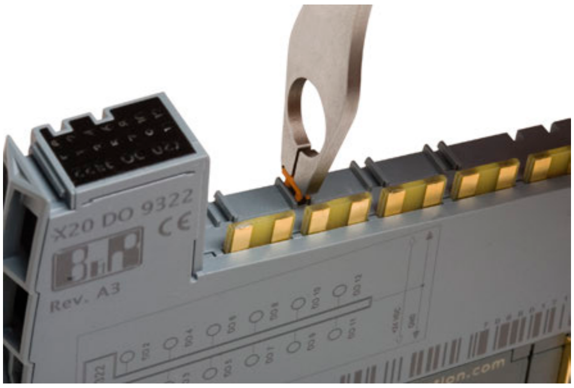
5. Press with the labeling tool to insert the feet of the label tags into the slot.
6. Inserted label tag.



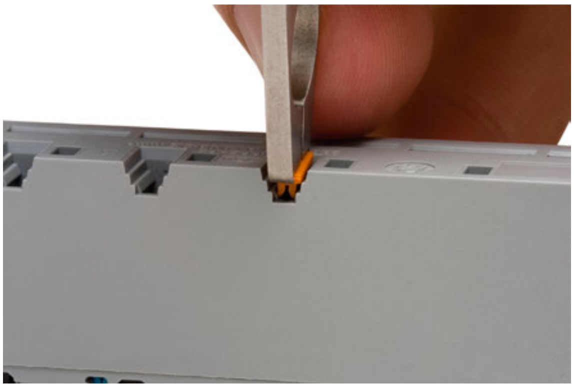
2.5.8.2 Labeling the terminals

To prevent errors, the X20 terminal blocks can be coded. This helps prevent terminal blocks from being inserted in the wrong electronic module.

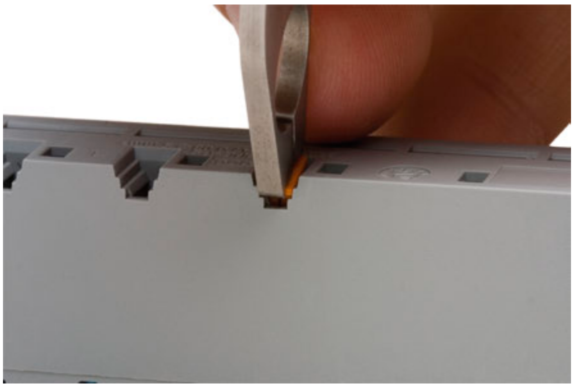
1. Remove a label tag with the single-width cutter of the labeling tool (compare with "Labeling the terminal connection" on page 138).
2. Center the label tag over the slot on the electronic module.



3. Hold the labeling tool at a 90° angle to the electronic module and press to insert the label's feet into the slot.
4. Remove a label tag with the single-width cutter of the labeling tool.
5. Set the label tag in the slot on the back of the terminal block as shown.

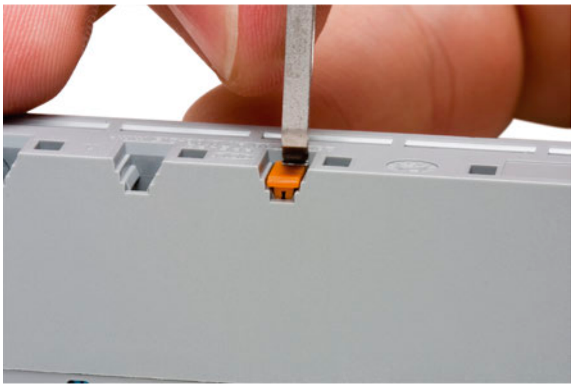


6. Use the labeling tool to push the left feet of the label into the slot.



7. With the labeling tool, press the right feet of the label into the slot.

8. Inserted label for terminal coding.



2.6 Module overviews

2.6.1 Module overview X20 safety: Alphabetical

2.6.2 Module overview X20 safety: Grouped

Analog input modules

Model number	Short description	Page
X20SA4430	X20 safe current input module, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	653

Bus modules

Model number	Short description	Page
X20BM13	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous, single-width	149
X20BM16	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous, single-width	149
X20BM23	X20 power supply bus module, for X20 SafeIO power supply modules, internal I/O power supply interrupted to the left	149
X20BM26	X20 power supply bus module, for X20 SafeIO power supply modules, with node number switch, internal I/O power supply interrupted to the left	149
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	149
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	149

Digital output modules

Model number	Short description	Page
X20SO2110	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 μs	436
X20SO2120	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 μs	436
X20SO2530	X20 safe digital output module, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20SO4110	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 μs	436
X20SO4120	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 μs	436
X20SO6300	X20 safe digital output module, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 μs	405
X20SO6530	X20 safe digital output module, 6 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591

Digital input modules

Model number	Short description	Page
X20SI2100	X20 safe digital input module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC	372
X20SI4100	X20 safe digital input module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20SI8110	X20 safe digital input module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width	372
X20SI9100	X20 safe digital input module, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372

Digital mixed modules

Model number	Short description	Page
X20SC0402	X20 safe digital mixed module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 μs	465
X20SC0806	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 μs	465
X20SC0842	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 μs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 μs	465
X20SC2212	X20 safe digital mixed module, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 μs	510
X20SC2432	X20 safe digital mixed module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555

Power supply modules

Model number	Short description	Page
X20SP1130	X20 power supply module, with integrated safe cutoff function, for internal I/O power supply, 24 VDC, 10 A, 1 safe type B1 digital output, 24 VDC, 10 A, without OSSD, note the list of permitted modules in the potential group	623

Terminal blocks

Model number	Short description	Page
X20TB52	X20 terminal block, 12-pin, safety-keyed	155
X20TB5E	X20 terminal block, 16-pin, safety-keyed, 2x PT1000 integrated for terminal temperature compensation	155
X20TB5F	X20 terminal block, 16-pin, safety-keyed	155
X20TB72	X20 terminal block, 12-pin, safety-keyed, 240 VAC, red	155

Intelligent programmable modules

Model number	Short description	Page
X20SLX210	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC	234
X20SLX402	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20SLX410	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20SLX806	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20SLX811	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width	234
X20SLX842	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs	299
X20SLX910	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234

reACTION modules

Model number	Short description	Page
X20SRT402	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	741
X20SRT806	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	741
X20SRT842	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs	741

Relay modules

Model number	Short description	Page
X20SC2432	X20 safe digital mixed module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555
X20SO2530	X20 safe digital output module, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20SO6530	X20 safe digital output module, 6 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591

Temperature measurement modules

Model number	Short description	Page
X20ST4492	X20 safe temperature input module, 2x 2 safe analog inputs for thermocouples, Type: J, K, N, S, R, C, T, resolution 0.1°C, 1x 2 safe analog inputs for PT100/PT1000 sensors, channel pairs galvanically isolated, integrated compensation of terminal temperature, integrated temperature sensor in terminal block X20TB5E, configurable input filter and switching thresholds	684

Counter and positioning modules

Model number	Short description	Page
X20SD1207	X20 safe digital counter module, 1 safe digital counter channel, 7 kHz, 24 VDC	713

CPUs

Model number	Short description	Page
X20SL8100	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20SL8101	X20 SafeLOGIC with X20 bus controller, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20SL8110	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, 1 slot for X20 interface module, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163

Accessories

Model number	Short description	Page
X20MKXXXX.XXX.XXX	Safety Technology Guarding" defines the range of functions available for applications using X20SL81xx- or X20cSL81xx-series SafeLOGIC controllers. Licenses are stored on a SafeKEY dongle. The functions required for the application must be put together in the X20MK configurator by selecting a SafeKEY with a sufficient amount of memory, a coated/non-coated variant and the necessary technology functions. Each solution is delivered exclusively as a set consisting of the SafeKEY and the activated licenses for the selected technology functions.	787

2.6.3 Module overview X20 coated safety: Alphabetical

Model number	Short description	Page
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	149
X20cSA4430	X20 safe current input module, coated, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	653
X20cSC2212	X20 safe digital mixed module, coated, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 μs	510
X20cSC2432	X20 safe digital mixed module, coated, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555
X20cSD1207	X20 safe digital counter module, coated, 1 safe digital counter channel, 7 kHz, 24 VDC	713
X20cSI4100	X20 safe digital input module, coated, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20cSI9100	X20 safe digital input module, coated, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20cSL8100	X20 SafeLOGIC, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20cSL8101	X20 SafeLOGIC with X20 bus controller, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20cSLX402	X20 safe digital mixed module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 μs	299
X20cSLX410	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20cSLX910	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20cSO2530	X20 safe digital output module, coated, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20cSO4110	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 μs	436
X20cSO4120	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 μs	436
X20cSO6300	X20 safe digital output module, coated, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 μs	405

2.6.4 Module overview X20 coated safety: Grouped

Analog input modules

Model number	Short description	Page
X20cSA4430	X20 safe current input module, coated, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	653

Bus modules

Model number	Short description	Page
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	149

Digital output modules

Model number	Short description	Page
X20cSO2530	X20 safe digital output module, coated, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20cSO4110	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs	436
X20cSO4120	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	436
X20cSO6300	X20 safe digital output module, coated, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	405

Digital input modules

Model number	Short description	Page
X20cSI4100	X20 safe digital input module, coated, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20cSI9100	X20 safe digital input module, coated, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372

Digital mixed modules

Model number	Short description	Page
X20cSC2212	X20 safe digital mixed module, coated, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	510
X20cSC2432	X20 safe digital mixed module, coated, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555

Intelligent programmable modules

Model number	Short description	Page
X20cSLX402	X20 safe digital mixed module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20cSLX410	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20cSLX910	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234

Relay modules

Model number	Short description	Page
X20cSC2432	X20 safe digital mixed module, coated, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555
X20cSO2530	X20 safe digital output module, coated, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591

Counter and positioning modules

Model number	Short description	Page
X20cSD1207	X20 safe digital counter module, coated, 1 safe digital counter channel, 7 kHz, 24 VDC	713

CPUs

Model number	Short description	Page
X20cSL8100	X20 SafeLOGIC, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20cSL8101	X20 SafeLOGIC with X20 bus controller, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163

2.6.5 Bus modules

2.6.5.1 Overview

Model number	Short description	Page
X20BM13	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous, single-width	149
X20BM16	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous, single-width	149
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	149
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	149
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	149
X20BM23	X20 power supply bus module, for X20 SafeIO power supply modules, internal I/O power supply interrupted to the left	149
X20BM26	X20 power supply bus module, for X20 SafeIO power supply modules, with node number switch, internal I/O power supply interrupted to the left	149

2.6.5.2 X20BM13, X20BM16, X20(c)BM33, X20BM36, X20BM23, X20BM26

2.6.5.2.1 General information

Bus modules serve as the basis for all SafeIO modules.
Depending on the bus module type, the internal I/O power supply is connected through or interrupted to the left.

With X20BMx6 bus modules, fixed addresses can be set via node number switches. This type of module at the beginning of an X20 block always generates a unique address. The subsequent modules then automatically increment from this address. This simple feature greatly increases the flexibility of applications.

Another advantage: Addresses can be set independently of specific I/O modules; only the necessary bus modules are required, which is logistically advantageous in terms of the cost and diversity of parts.

	X20BM13	X20BM16	X20BM33	X20BM36	X20BM23	X20BM26
Bus module type	X20 SafeIO modules				X20 SafeIO power supply module	
Formation of potential groups possible	No				Yes	
Internal I/O power supply	Connected through				Interrupted to the left	
Manual node number assignment possible	No	Yes	No	Yes	No	Yes
Single-width	Yes		No			

2.6.5.2.1.1 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.5.2.2 Order data

Model number	Short description
	Bus modules
X20BM13	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous, single-width
X20BM16	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous, single-width
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20BM23	X20 power supply bus module, for X20 SafeIO power supply modules, internal I/O power supply interrupted to the left
X20BM26	X20 power supply bus module, for X20 SafeIO power supply modules, with node number switch, internal I/O power supply interrupted to the left

Table 15: X20BM13, X20BM16, X20BM33, X20cBM33, X20BM36, X20BM23, X20BM26 - Order data

2.6.5.2.3 Technical data

Model number	X20BM13	X20BM16	X20BM33	X20cBM33	X20BM36	X20BM23	X20BM26
Short description							
Bus module	Bus module, for X20 SafeIO modules, internal I/O power supply continuous	Bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	Bus module, for X20 SafeIO modules, internal I/O power supply continuous		Bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	Power supply bus module, for X20 SafeIO power supply modules, internal I/O power supply interrupted to the left	Power supply bus module, for X20 SafeIO power supply modules, with node number switch, internal I/O power supply interrupted to the left
General information							
Power consumption							
Bus	0.13 W						
Internal I/O	-						
Certifications							
CE	Yes						
KC	-		Yes	-		Yes	-
EAC	Yes						
UL	cULus E115267 Industrial control equipment						
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	In preparation	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5				
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	In preparation	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X				
DNV GL	In preparation		Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)		In preparation	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)	In preparation
LR	-		ENV1		-	ENV1	-
I/O power supply							
Nominal voltage	24 VDC						
Permissible contact load	10 A						
Operating conditions							
Mounting orientation							
Horizontal	Yes						
Vertical	Yes						
Installation elevation above sea level	0 to 2000 m, no limitation						
Degree of protection per EN 60529	IP20						
Ambient conditions							
Temperature							
Operation							
Horizontal mounting orientation	-25 to 60°C						
Vertical mounting orientation	-25 to 50°C						
Derating	-						
Storage	-40 to 85°C						
Transport	-40 to 85°C						
Relative humidity							
Operation	5 to 95%, non-condensing			Up to 100%, condensing	5 to 95%, non-condensing		
Storage	5 to 95%, non-condensing						
Transport	5 to 95%, non-condensing						
Mechanical properties							
Spacing	$12.5^{+0.2}$ mm		$25^{+0.2}$ mm				

Table 16: X20BM13, X20BM16, X20BM33, X20cBM33, X20BM36, X20BM23, X20BM26 - Technical data

2.6.5.2.4 Voltage routing

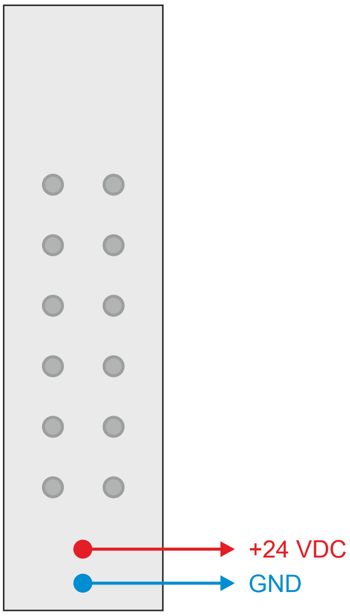


Figure 45: X20BM2x - Voltage routing

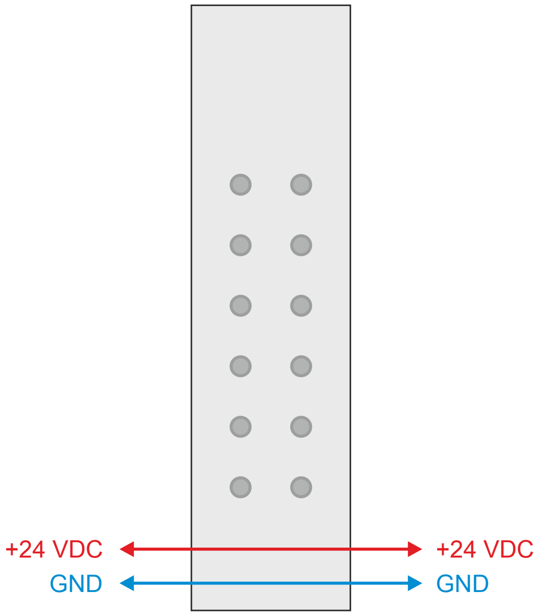


Figure 46: X20BM1x / X20BM3x - Voltage routing

Voltage routing identification

A symbol is printed on the locking lever on bus modules interrupted to the left. This makes it clear from the outside of a fully assembled X20 system that bus modules interrupted to the left are used in this slot.

Figure 47: X20BM2x - Voltage routing identification

2.6.5.2.5 Manual node number assignment in the X20 safe I/O system

With the X20 safety bus modules X20BM16, X20BM26 and X20BM36, permanent addresses can be set using node number switches. One of these modules placed at the beginning of an X20 safety block always creates a unique address. The subsequent module addresses are assigned automatically in ascending order starting with this address. This simple feature greatly increases the flexibility of applications.

Another advantage: Addresses can be set independently of which specific I/O modules are used. All that is required are the respective bus modules. This provides logistical advantages with respect to cost and the variety of parts.

2.6.5.2.5.1 Node number switches

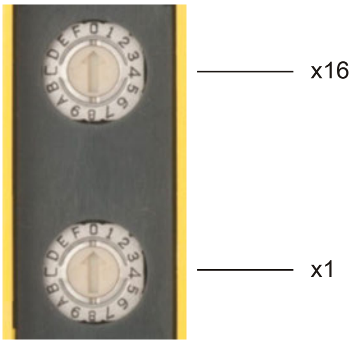


Figure 48: X20BMx6 - Node number switches

The X2X Link address of the module is set using the node number switches (0x01 to 0xFD).
Setting node number 0x00 causes the X2X Link address to be assigned automatically.

Node number switch identification

Symbols are printed on the locking lever of bus modules with node number switches. This provides a way to see from outside that the X20 system mounted in this slot is using node number switches.

Figure 49: X20BMx6 - Node number switch identification

2.6.5.2.6 Version history

Version	Date	Comment
1.141	April 2019	• Chapter 2.6.5.2.3 "Technical data": Updated standards.
1.140	February 2019	• Chapter 2.6.5.2.3 "Technical data": Limited installation elevation to 2000 m. • Updated standards. • Editorial changes.
1.120	January 2018	Added X20BM16 bus module. • Chapter 2.6.5.2.3 "Technical data": Updated standards.
1.101	November 2016	Added bus module X20cBM23. • Chapter 2.6.5.2.3 "Technical data": Updated standards.
1.100	February 2016	Merged coated/uncoated modules. Added bus module X20BM13. • Chapter 2.6.5.2.3 "Technical data": Updated technical data.
1.80	August 2014	• Chapter 2.6.5.2.3 "Technical data": Added I/O power supply. • Chapter 2.6.5.2.4 "Voltage routing": Updated marking for voltage routing. • Chapter 2.6.5.2.5.1 "Node number switches": Updated marking for node number switch.
1.50	March 2012	Bus modules X20BM23, X20BM26 and X20BM36 included
1.00	March 2012	First edition as a product-specific manual

Table 17: Version history

2.6.6 Terminal blocks

2.6.6.1 Overview

Model number	Short description	Page
X20TB52	X20 terminal block, 12-pin, safety-keyed	155
X20TB5E	X20 terminal block, 16-pin, safety-keyed, 2x PT1000 integrated for terminal temperature compensation	155
X20TB5F	X20 terminal block, 16-pin, safety-keyed	155
X20TB72	X20 terminal block, 12-pin, safety-keyed, 240 VAC, red	155

2.6.6.2 X20TB52, X20TB5E, X20TB5F, X20TB72

2.6.6.2.1 General information

The X20 SafeIO modules are wired with the terminal blocks.
Terminal block X20TB52 is available for wiring SafeIO modules with 12 connections.
Terminal blocks X20TB5E and X20TB5F are available for wiring SafeIO modules with 16 connections.

SafeIO modules with 240 VAC are wired to terminal block X20TB72. This is marked by its own color.

Terminal block X20TB5E is equipped with 2 integrated PT1000 sensors. It is therefore ideally suited for internal terminal temperature compensation. The terminal block can be used for all safe thermocouples with 16 connections.

- Tool-free wiring using push-in technology
- Simple wire release with lever or screwdriver
- Labeling option for each terminal connection
- Plain text labeling possible
- Access for standard test probes
- Customized coding possible

2.6.6.2.2 Order data

X20TB52 X20TB5E X20TB5F X20TB72

Model number	Short description
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed
X20TB5E	X20 terminal block, 16-pin, safety-keyed, 2x PT1000 integrated for terminal temperature compensation
X20TB5F	X20 terminal block, 16-pin, safety-keyed
X20TB72	X20 terminal block, 12-pin, safety-keyed, 240 VAC, red

Table 18: X20TB52, X20TB5E, X20TB5F, X20TB72 - Order data

Information:

B&R screwdriver X20AC0SD1 should be used to avoid damaging terminals X20TB5E and X20TB5F.

2.6.6.2.3 Technical data

Model number	X20TB52	X20TB5E	X20TB5F	X20TB72
General information				
Certifications				
CE	Yes			
UL	cULus E115267 Industrial control equipment			
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X			
DNV GL	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)			
LR	ENV1			
Terminal block				
Number of pins	12, safety-keyed	16, safety-keyed		12, safety-keyed
Type of terminal block	Push-in terminal			
Push-in force per contact	Typ. 10 N			
Cable type	Only copper wires (no aluminum wires!)			
Wire stripping length	7 to 9 mm			
Connection cross section				
Solid wires	0.08 to 2.5 mm² / 28 to 14 AWG	0.08 to 1.5 mm² / 28 to 16 AWG		0.08 to 2.5 mm² / 28 to 14 AWG
Fine-stranded wires	0.25 to 2.5 mm² / 24 to 14 AWG	0.25 to 1.5 mm² / 24 to 16 AWG		0.25 to 2.5 mm² / 24 to 14 AWG
With wire end sleeves	0.25 to 1.5 mm² / 24 to 16 AWG	0.25 to 0.75 mm² / 24 to 20 AWG		0.25 to 1.5 mm² / 24 to 16 AWG
With double wire end sleeves	Up to 2x 0.75 mm²	-		Up to 2x 0.75 mm²
Distance between contacts				
Left - Right	4.2 mm			
Above - Below	10.96 mm	8.25 mm		10.96 mm
Terminal temperature compensation	-	2x PT1000 integrated in the terminal	-	
Electrical properties				
Nominal voltage	48 VAC	24 VDC		240 VAC
Max. voltage	48 VAC	50 VDC		300 VAC
Nominal current [1]	10 A / contact	2 A / contact		10 A / contact
Contact resistance	≤5 mΩ			
Ambient conditions [2]				
Temperature				
Operation	Corresponds to the X20 module used			
Relative humidity				
Operation	Corresponds to the X20 module used			

Table 19: X20TB52, X20TB5E, X20TB5F, X20TB72 - Technical data

1) The limit data for each SafeIO module must be taken into consideration.
2) Identical for operation, storage and transport.

Warning!

It is possible to come into contact with parts that carry voltage when the terminal block is disconnected. For this reason, working on a disconnected terminal block is not permitted at voltages starting at 50 V.

Information:

Special care must be exercised during installation when using non-SELV circuits (e.g. 230 V). Local regulations must be observed, particularly with respect to safety precautions.

2.6.6.2.4 Wiring

In order to achieve a secure connection in the terminal blocks, wires must be stripped accordingly.

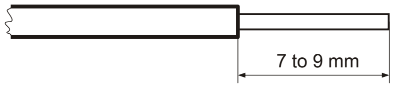


Figure 50: Wire stripping length for a secure connection

Information:

The wire stripping length is not permitted to be more or less than 7 to 9 mm.

2.6.6.2.5 Cable holding force of contacts

To ensure secure contact of a cable with the terminal block, it is not permitted to be subjected to too much tension. If the cable holding force is exceeded, the cable will disconnect from the terminal block and result in a malfunction.

	Fine-stranded wires			Solid wires				With wire end sleeves	
Cable in mm²	0.25	1.5	2.5	0.08	0.25	1.5	2.5	0.25	1.5
Standard specification (min. value in newtons)	12.5	40	50	4	12.5	40	50	12.5	40

Information:

Fine-stranded wires must be twisted in order to maintain the cable holding forces.

Use of wire end sleeves

In order to achieve an optimal cable holding force, the following points must be observed:

- Square crimping with the roughest possible surface should be carried out.
- The end of the wire end sleeve should not be cut in order to avoid a reduction of the cross section.
- No wires should protrude at the end of the sleeve.
- The wire end sleeve must be inserted completely to the end.
- The length of the wire end sleeve corresponds to the wire stripping length.

2.6.6.2.6 Access for test probes

Each contact is equipped with an additional opening for using a test probe.

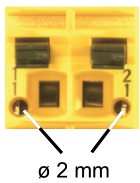


Figure 51: X20TB52 and X20TB72 - Access for test probes

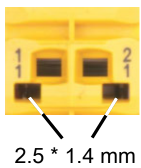


Figure 52: X20TB5E and X20TB5F - Access for test probes

2.6.6.2.7 Version history

Version	Date	Comment
1.141	April 2019	• Editorial changes.
1.140	February 2019	• Chapter 2.6.6.2.2 "Order data": Added information. • Chapter 2.6.6.2.3 "Technical data": Added warning notice. • Added the following chapters: – 2.6.6.2.1 "General information" – 2.6.6.2.4 "Wiring" – 2.6.6.2.5 "Cable holding force of contacts" – 2.6.6.2.6 "Access for test probes" – 2.6.6.2.8 "EC declaration of conformity" • Updated standards. • Editorial changes.
1.100	February 2016	Chapter 2.6.6.2.3 "Technical data": – Added environmental conditions. – Updated technical data.
1.80	August 2014	Updated chapter 2.6.6.2.3 "Technical data".
1.51	January 2013	Chapter 2.6.6.2.3 "Technical data": Added information.
1.50	March 2012	Terminal blocks X20TB72, X20TB5E and X20TB5F included
1.00	February 2012	First edition as a product-specific manual

Table 20: Version history

2.6.7 CPUs

2.6.7.1 Overview

Model number	Short description	Page
X20SL8100	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20cSL8100	X20 SafeLOGIC, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20SL8101	X20 SafeLOGIC with X20 bus controller, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20cSL8101	X20 SafeLOGIC with X20 bus controller, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163
X20SL8110	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, 1 slot for X20 interface module, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!	163

2.6.7.2 X20(c)SL81xx

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 21: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 22: Organization of general notices

2.6.7.2.1 General information

The modules are equipped with SafeLOGIC functionality that allows them to safely execute applications designed in SafeDESIGNER. The modules can be used in safety applications up to PL e or SIL 3.

The SafeLOGIC controller coordinates the safety-related communication of all modules involved in the application. In this context, the SafeLOGIC controller also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided in both variants, however.
In addition, SafeLOGIC-X products have the same I/O properties as the associated SafeIO products.

- openSAFETY manager for up to 10 / 20 / 100 / 280 SafeNODES
- Flexibly programmable using Automation Studio / SafeDESIGNER
- Innovative management of safe machine options (SafeOPTION)
- Parameter and configuration management

2.6.7.2.1.1 Function

SafeLOGIC function

The module is equipped with SafeLOGIC functionality that allows it to safely execute applications designed in SafeDESIGNER. The module can be used in safety-related applications up to PL e or SIL 3.

In addition, the module coordinates the safety-related communication of all modules involved in the application. In this context, the module also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided with both variants, however.

Blackout mode

In blackout mode, module functionality persists even if the network fails. Without this function, the safe state would always be initiated on the affected module if the network fails. In addition, blackout mode can allow partial operation to resume or coordinated shutdown scenarios to be initiated. This mode also makes it possible to boot a module without a network based on a configuration saved on the module beforehand.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.7.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.7.2.2 Order data

X20SL8100 X20SL8101 X20SL8110

Model number	Short description
	CPUs
X20SL8100	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!
X20cSL8100	X20 SafeLOGIC, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!
X20SL8101	X20 SafeLOGIC with X20 bus controller, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!
X20cSL8101	X20 SafeLOGIC with X20 bus controller, coated, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, controlled node, integrated 2-port hub, including power supply module for internal I/O power supply and X2X Link power supply, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!
X20SL8110	X20 SafeLOGIC, safety controller, openSAFETY gateway, removable application memory: SafeKEY, 1 POWERLINK interface, 1 slot for X20 interface module, controlled node, integrated 2-port hub, including power supply module, 1x terminal block X20TB52 and X20 end cover plate X20AC0SR1 (right) included, order SafeKEY and SafeLOGIC range of functions using the X20MK configurator!
	Required accessories
	Accessories
X20MKXXXX.XXX.XXX	"Safety Technology Guarding" defines the range of functions available for applications using X20SL81xx- or X20cSL81xx-series SafeLOGIC controllers. Licenses are stored on a SafeKEY dongle. The functions required for the application must be put together in the X20MK configurator by selecting a SafeKEY with a sufficient amount of memory, a coated/non-coated variant and the necessary technology functions. Each solution is delivered exclusively as a set consisting of the SafeKEY and the activated licenses for the selected technology functions.

Table 23: X20SL8100, X20cSL8100, X20SL8101, X20cSL8101, X20SL8110 - Order data

2.6.7.2.3 Technical data

Model number	X20SL8100	X20cSL8100	X20SL8101	X20cSL8101	X20SL8110
Short description					
Interfaces	POWERLINK				
System module	CPU				
General information					
Cooling	Fanless				
B&R ID code	0xDD61	0xE287	0xE649	0xE926	0xE64A
System requirements					
Automation Studio	4.0.16 or later		4.1.6 or later		V4.2.5 or later
Automation Runtime	V3.08 or later (for AsSafety library F4.06 or later)		F4.09 or later, F4.10 or later, A4.23 or later		B4.25 or later
SafeDESIGNER	3.1.0 or later		4.1.0 or later		V4.2 or later
Safety Release	1.7 or later				1.10 or later
Status indicators	CPU function, POWERLINK, SafeKEY				
Diagnostics					
CPU function	Yes, using status LED				
POWERLINK	Yes, using status LED				
SafeKEY	Yes, using status LED				
Power consumption	4.3 W		5.3 W		3.9 W [1]
Blackout mode					
Scope	-		Network segment		-
Function	-		Programmable		-
Standalone mode	-		Yes		-
Power consumption for X2X Link power supply	-		1.42 W [2]		-
Power consumption					
Internal I/O	-		0.6 W [2]		-
Electrical isolation					
Fieldbus - X2X Link	-		Yes		-
Fieldbus - I/O	-		Yes		-
Certifications					
CE	Yes				
EAC	Yes				
UL	cULus E115267 Industrial control equipment				cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5				-
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X				
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)				In preparation
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013				
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3				
Functional safety	EN 50156-1:2004				
Safety characteristics					
EN ISO 13849-1:2015					
Category	Cat. 4				
PL	PL e				
DC	>94%				
MTTFD	2500 years				
Mission time	Max. 20 years				
IEC 61508:2010, IEC 61511:2004, EN 62061:2013					
SIL CL	SIL 3				
SFF	>90%				
PFH / PFH_d					
Module	$<1*10^{-10}$				
openSAFETY wired	Negligible				
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour				
PFD	$<2*10^{-5}$				
Proof test interval (PT)	20 years				

Table 24: X20SL8100, X20cSL8100, X20SL8101, X20cSL8101, X20SL8110 - Technical data

Model number	X20SL8100	X20cSL8100	X20SL8101	X20cSL8101	X20SL8110
Functionality					
Communication with each other	Yes				
Support for machine options					
BOOL	512				
INT	64				
UINT	64				
DINT	64				
UDINT	64				
SafeMOTION support	Yes, depends on the number of available operating licenses on the SafeKEY				
Timing precision	Time * 0.05 + Cycle time of the safety application				
Shortest task class cycle time	1 ms				
Max. number of openSAFETY nodes	100, depends on the number of available operating licenses on the SafeKEY		280, depends on the number of available operating licenses on the SafeKEY and available resources		
Max. number of POWERLINK controlled nodes	50		100		
Data exchange between CPU and SL					
Max. total data width for each direction	128 bytes				
Max. number of data points for each direction					
BOOL	352 (96 + 256 extended)				
INT	30				
UINT	30				
DINT	15				
UDINT	15				
Data exchange between SL and SL					
Max. total number of data points for each direction 3)	16				
Max. number of data points for each direction					
BOOL	128				
INT	16				
UINT	16				
DINT	16				
UDINT	16				
Limit values for SafeDESIGNER application					
Max. resources available for SafeDESIGNER info window entries 4)					
FB instances	4096				
Marker memory	131,072 bytes				
Stack memory	32,768 bytes				
Memory for safe input data	2048 bytes				
Memory for safe output data	2048 bytes				
Memory for standard input data	1024 bytes				
Memory for standard output data	1024 bytes				
Marker count	8192				
Additional SafeDESIGNER limit values					
Max. number of function block types	512				
Max. number of force variables	64				
Max. number of variable with variable status	1023				
Input SL / BC / X2X Link power supply					
Input voltage	24 VDC -15% / +20%				
Input current	Max. 0.25 A		Max. 0.9 A		Max. 0.25 A
Fuse	-		Integrated, cannot be replaced		-
Reverse polarity protection	Yes				
Output SL / BC / X2X Link power supply					
Nominal output power	-		7 W		-
Parallel connection	-		Yes 5)		-
Redundant operation	-		Yes		-
Overload characteristics	-		Short-circuit proof, temporary overload		-
Input I/O power supply					
Input voltage	-		24 VDC -15% / +20%		-
Fuse	-		Required line fuse: Max. 10 A, slow-blow		-
Reverse polarity protection	-		Yes		-
Output I/O power supply					
Nominal output voltage	-		24 VDC		-
Behavior on short circuit	-		Required line fuse		-
Permissible contact load	-		10 A		-
Interfaces					
Fieldbus	POWERLINK controlled node				
Type	Type 3 6)				
Variant	2x shielded RJ45 port (hub)				

Table 24: X20SL8100, X20cSL8100, X20SL8101, X20cSL8101, X20SL8110 - Technical data

Model number	X20SL8100	X20cSL8100	X20SL8101	X20cSL8101	X20SL8110
Line length	Max. 100 m between 2 nodes (segment length)				
Transfer rate	100 Mbit/s				
Transfer					
Physical layer	100BASE-TX				
Half-duplex	Yes				
Full-duplex	No				
Autonegotiation	Yes				
Auto-MDI / MDIX	Yes				
Min. cycle time 7)					
Fieldbus	200 µs				
X2X Link	-		200 µs		-
Synchronization between bus systems possible	-		Yes		-
Operating conditions					
Mounting orientation					
Horizontal	Yes				
Vertical	Yes				
Installation elevation above sea level	0 to 2000 m, no limitation				
Degree of protection per EN 60529	IP20				
Ambient conditions					
Temperature					
Operation					
Horizontal mounting orientation	0 to 60°C	-40 to 60°C 8)	0 to 60°C	-40 to 60°C 9)	0 to 60°C
Vertical mounting orientation	0 to 45°C	-40 to 45°C 10)	0 to 45°C	-40 to 45°C 11)	0 to 45°C
Derating	-		See section "Derating".		-
Storage	-40 to 85°C				
Transport	-40 to 85°C				
Relative humidity					
Operation	5 to 95%, non-condensing	Up to 100%, condensing	5 to 95%, non-condensing	Up to 100%, condensing	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing				
Transport	5 to 95%, non-condensing				
Mechanical properties					
Note	Order SafeKEY and SafeLOGIC range of functions using the X20MK configurator. X20 end cover plate (right) included in delivery. 12-pin X20 terminal block, safety-keyed, included in delivery. SafeKEY cover included in delivery.				
Dimensions					
Width	$62.5^{+0.2}$ mm				
Height	99 mm				
Depth	75 mm				
Weight	190 g				

Table 24: X20SL8100, X20cSL8100, X20SL8101, X20cSL8101, X20SL8110 - Technical data

1) Power consumption without interface module
2) The specified values are maximum values. For examples of the exact calculation, see section "Mechanical and electrical configuration" of the X20 system user's manual.
3) Keep in mind that 8 BOOL count as 1 data point.
4) For a parameter description, see section "Message window" of the SafeDESIGNER documentation.
5) In parallel operation, it is only permitted to expect 75% of the nominal power. It is important to make sure that all power supplies operated in parallel are switched on and off at the same time.
6) See Automation Help under "Communication / POWERLINK / General information / Hardware - CN" for more information. It is important to note, however, that the SafeLOGIC controller does not support "early writing of output data". The use of "poll-response chaining" is not recommended for controlled nodes in the same POWERLINK line.
7) The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.
8) Up to hardware upgrade <1.10.5.0 and hardware revision <F0: -25 to 60°C
9) Up to hardware upgrade <1.10.5.0 and hardware revision <E0: -25 to 60°C
10) Up to hardware upgrade <1.10.5.0 and hardware revision <F0: -25 to 45°C
11) Up to hardware upgrade <1.10.5.0 and hardware revision <E0: -25 to 45°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

X20SL8101: Derating for SafeLOGIC / Bus controller / X2X Link power supply

The nominal output power for the X2X Link power supply is 7 W.
The nominal output power depends on the operating temperature and the mounting orientation. The resulting nominal output power can be looked up in the following table.

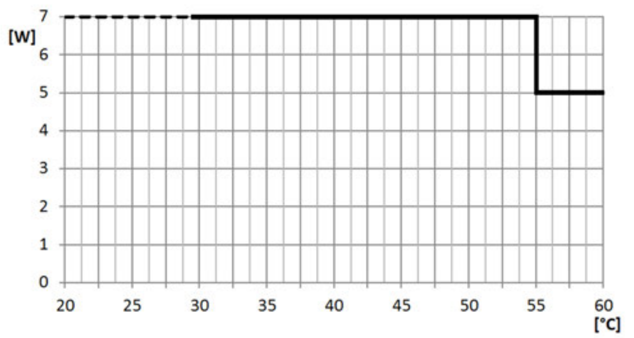
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 45°C, coated: -40 to 45°C)
	No derating

Table 25: Derating for SafeLOGIC / Bus controller / X2X Link power supply

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.7.2.4 Operating and connection elements

LEDs and buttons/switches are provided for operating the SafeLOGIC. These elements can be used to perform the following actions:

- Module exchange, including a test of the complete module configuration (section "Module replacement")
- Firmware replacement (section "Acknowledging a firmware modification")
- SafeKEY replacement, including possible transfer of module configuration from the old SafeKEY (section "Changing the application on the SafeLOGIC controller by replacing the SafeKEY (X20SL8xxx series only)")
- and SafeLOGIC controller replacement (section "Replacing a SafeLOGIC controller")

The AsSafety library (chapter "Operation via the AsSafety library") can also be used to operate the SafeLOGIC controller using the HMI application.

SafeLOGIC has the following operating and connection elements:

X20SL810x

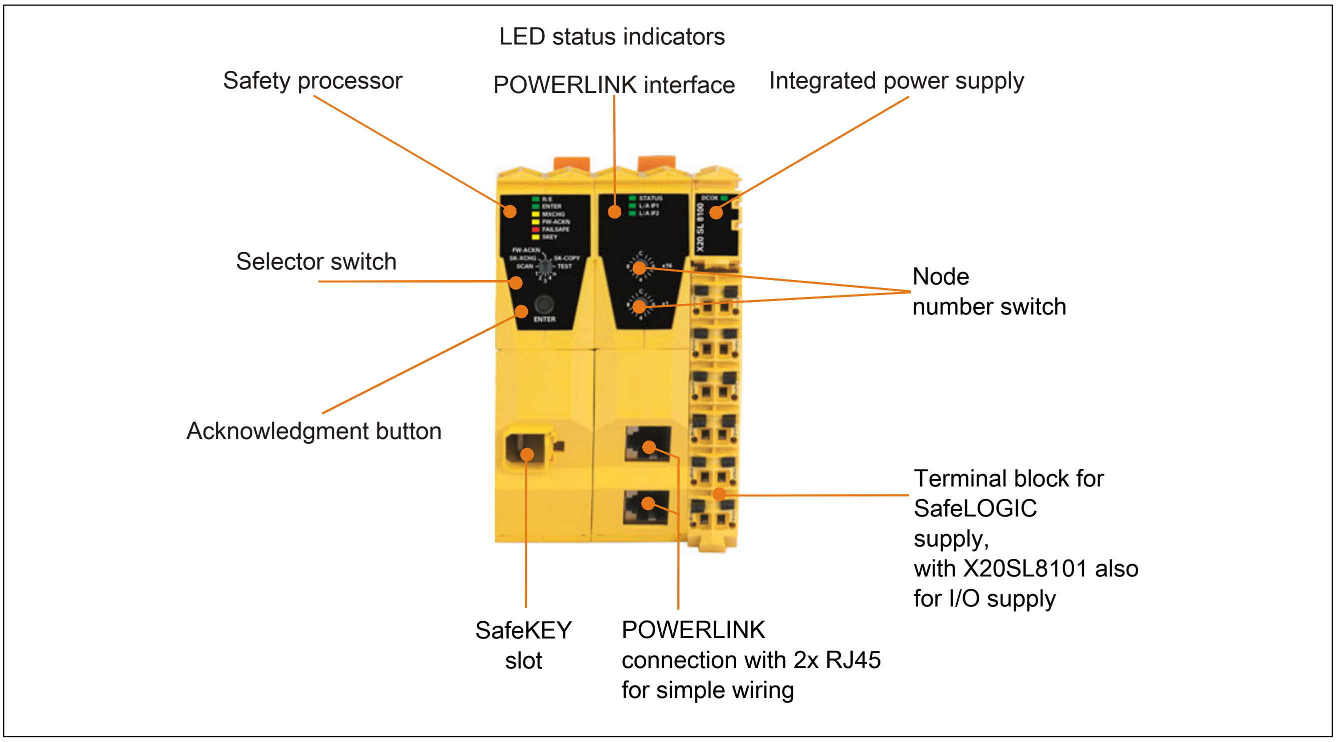


Figure 53: X20SL810x - Operating elements

X20SL8110

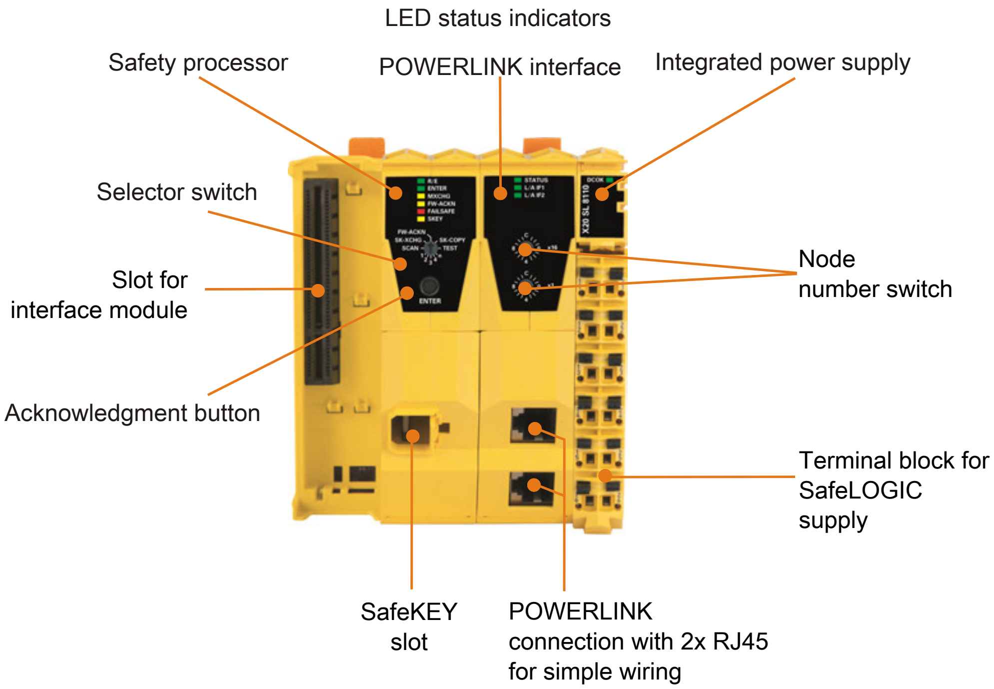


Figure 54: X20SL8110 - Operating elements

Slot for interface modules

The X20SL8110 SafeLOGIC controller is equipped with a slot for interface modules.

Various bus and network systems can easily be integrated into the X20 system by selecting the corresponding interface module.

The following interface modules can be used in the X20SL8110 SafeLOGIC controller:

Module	Description
X20IF10E3-1	X20 interface module for DTM configuration, 1 PROFINET RT device (slave) interface, electrically isolated

2.6.7.2.4.1 Safety processor

LED status indicators of the safety processor

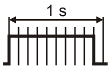
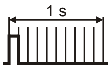
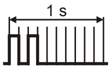
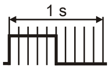
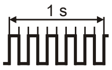
LED	Color	Status	Description
R/E		Off	Boot phase
	Green	On	Application exists and is being executed
		Blinking	Application exists but is not being executed (in the download dialog box for the SafeDESIGNER, "Automatic start" was not selected OR boot phase, i.e. not all necessary safe modules on the network were configured correctly.) In addition, boot states 0x1840 to 0x3440 under index:subindex 0x2410:0x01 must be checked in section "SafeLOGIC - Channel list".
	Orange	On	SafeDESIGNER in "Debug" mode
		Blinks at 0.5 Hz	SafeDESIGNER in "Debug" mode, application in "Stop"
		Blinks at 1 Hz	No application on SafeKEY
ENTER	Green	On	Missing authorization, see "Authorization (X20SL8xxx series only)".
		Blinks 1x for 0.8 s	Confirmation of correct entry
		Blinks (1 Hz) for 5 s	Faulty operation
MXCHG	Orange	Off	Module configuration OK
			Replacement of 1 module detected
			Replacement of 2 modules detected
			Replacement of 3 modules detected
			Replacement of 4 modules detected
			Replacement of more than 4 modules detected
			Missing module detected
FW-ACKN	Orange	Off	Firmware configuration OK
		Blinking	Firmware update carried out
		On	SafeKEY was replaced
ENTER MXCHG FW-ACKN	Green Orange Orange	Running sequence	Performing module scan or boot phase (beginning with Release 1.5 - Note: Check STATUS LED, see section "LED status indicators for the POWERLINK interface"!)
FAILSAFE	Red		The FAILSAFE LED indicates the boot behavior or state "FailSafe" for the entire module after booting.
		Off	Safety firmware OPERATIONAL state
			Boot phase
			Safety firmware PRE_OPERATIONAL state or "SafeOSstate!=RUN"
			Safe communication channel not OK, openSAFETY connection valid problem or "SafeOSstate!=RUN" If the SafeLOGIC controller remains in this state for a longer time, parameter "Default Safe Data Duration" of the "Group: Safety Response Time Defaults" must be checked.
			Boot phase, faulty firmware, setup mode active (hardware upgrade 1.10.2.x and later) For details about setup mode, see section "Setup mode".
			Test/Pilot firmware or safety application created with test/pilot version of SafeDESIGNER

Table 26: Safety processor status indicators

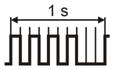
			SafeDESIGNER in "Debug" mode
		On	Safety state active for the entire module (= state "FailSafe")
SKEY	Orange	Off	No access to the SafeKEY
		Blinking	Access to the SafeKEY

Table 26: Safety processor status indicators

Danger!

A constantly lit FAILSAFE LED indicates a possible safety-related system error.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in dangerous situations!

LED test

The functionality of the LEDs can be tested using the following sequence:

- Move the selector switch to TEST.
- Press the ENTER confirmation button.
- All of the safety processor LEDs will turn on (left module of the SafeLOGIC controller) for the exact duration that the confirmation button is pressed.

Selector switch and confirmation button

If configuration confirmations are required for the user, they can be generated by pre-selecting the desired function via the selector switch and then pressing the ENTER confirmation button.

Figure 55: Selector switch and confirmation button

Switch position	Functionality	Description
FW-ACKN	Firmware acknowledgment	Acknowledges a firmware change on one or more modules[1]
Unlabeled position between FW-ACKN and SK-COPY (=0xD)	Setup mode (hardware upgrade 1.10.2.x or later)	Enables/Disables setup mode For details about setup mode, see section "Setup mode".
SK-COPY	SafeKEY copy	Copy of the configuration data from the SafeKEY[2]
TEST	Test	Performs an LED test
Unlabeled position between TEST and n	CLEAR DATA	Deletes the following "user data": • Remanent data • Configuration file from the standard application • Extended machine options • Table objects • Subsequently loadable parameter file - firmware version V322 or later
1,2,3,4,n	Replacing a module	Confirm the replacement of 1, 2, 3, 4 or more than 4 modules
SCAN	Scan	Triggers a module scan
SK-XCHG	SafeKEY exchange	Confirmation of SafeKEY exchange[1]
Unlabeled position between FW-ACKN and SK-XCHG	Format SafeKEY	Formatting SafeKEY (Release 1.4 and later)[2]

Table 27: Confirmation modes

1) Triggers a restart in firmware versions ≤ V322.
2) Triggers an automatic restart.

Confirmation (all functions except for "Format SafeKEY")

The confirmation button must be pressed for 0.5 to 5 s to receive confirmation. After 0.5 s, the LED ENTER (see chapter "LED status indicators of the safety processor") begins to light. After releasing the confirmation button, the ENTER LED remains illuminated for an extra 0.8 s. This sequence indicates a correct entry.

- If the confirmation button is released before 0.5 s, it has no effect.
- If the confirmation button is pressed for longer than 5 s, then the ENTER LED blinks for 5 s to display an error.

Another possible reason for an error is an improper placement of the selector switch. If the user wants to confirm a module replacement for one specific module, for example, then the selector switch must be at position "1" (see section "Replacing an individual module"). In this case, if a placement other than "1" is confirmed with the confirmation button, it is considered an error and the ENTER LED blinks for 5 s.

Confirmation of "Format SafeKEY"

The confirmation button must be pressed for 20 to 30 s to receive a confirmation for "Format SafeKEY". After 20 s, the ENTER LED is illuminated. After releasing the confirmation button, the ENTER LED remains illuminated for an extra 0.8 s. This sequence indicates a correct entry.

- If the confirmation button is released before 20 s, it has no effect.
- If the confirmation button is pressed for longer than 30 s, then the ENTER LED blinks for 5 s to display an error.

All data will be deleted (including password), which is why going online with SafeDESIGNER and assigning a new password is recommended.

2.6.7.2.4.2 Slot for application memory (SafeKEY)

In order to operate the SafeLOGIC controller, application memory (SafeKEY) is required to save the program, the parameters and the system configuration.

The SafeKEY is equipped with a mechanical locking mechanism to make it more difficult to inadvertently remove during operation.

Figure 56: SafeKEY unlocked

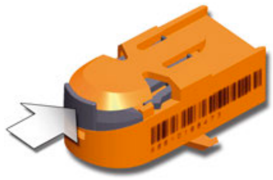


Figure 57: SafeKEY locked

Information:

Removing a SafeKEY during operation causes the SafeLOGIC controller to be restarted and all safety-related actuators to be cut off.

Removing a SafeKEY during operation can destroy the data on the SafeKEY.

Removing a SafeKEY during operation must therefore be avoided at all cost.

The "Backing up the SafeKEY" sequence is not affected by this general rule.

Information:

Note that modules operated on the local X2X bus of the X20SL8101 are only correctly configured if a valid safety project exists on the SafeKEY. Otherwise, "ModuleOk" in Automation Studio remains set to FALSE.

2.6.7.2.4.3 POWERLINK interface

LED status indicators for the POWERLINK interface

Figure	LED	Color	Status	Description
	STATUS[1]	Green/Red		Status/Error LED, The LED states are described in section "LED "STATUS"".
	L/A IFx	Green	On	A link to the peer station has been established.
			Blinking	A link to the peer station has been established. Indicates Ethernet activity is taking place on the bus.

Table 28: POWERLINK interface status indicators

1) The Status/Error LED is a green/red dual LED.

LED "STATUS"

LED "Status/Error" is a green and red dual LED. The color green (status) is superimposed on the color red (error).

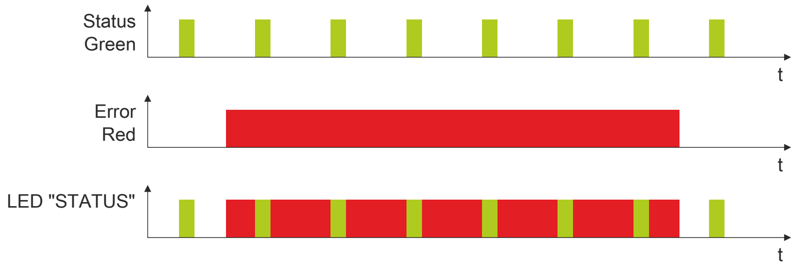
Red - Error	Description
On	The controlled node (CN) is in an error state (failed Ethernet frames, increased number of collisions on the network, etc.). If an error occurs in the following states, then the green LED blinks over the red LED: • PRE_OPERATIONAL_1 • PRE_OPERATIONAL_2 • READY_TO_OPERATE  Note: • Several red blinking signals are displayed immediately after the device is switched on. This is not an error, however. • The LED is lit red for CNs with set physical node number 0 to which no node number has yet been assigned by dynamic node allocation (DNA).

Table 29: Status/Error LED lit red: LED indicating error

Green - Status	Description
Off	No power supply or mode NOT_ACTIVE. The controlled node (CN) is either not supplied with power, or it is in state NOT_ACTIVE. The CN waits in this state for about 5 seconds after a restart. Communication is not possible with the CN. If no POWERLINK communication is detected during these 5 seconds, the CN enters state BASIC_ETHERNET (flickering). If POWERLINK communication is detected before this time expires, however, the CN immediately enters state PRE_OPERATIONAL_1.
Flickering green (approx. 10 Hz)	Mode BASIC_ETHERNET. The CN has not detected any POWERLINK communication. In this state, it is possible to communicate directly with the CN (e.g. with UDP, IP, etc.) If POWERLINK communication is detected while in this state, the CN enters state PRE_OPERATIONAL_1.
Single flash (approx. 1 Hz)	Mode PRE_OPERATIONAL_1. The CN waits until it receives an SoC frame and then switches to state PRE_OPERATIONAL_2.
Double flash (approx. 1 Hz)	Mode PRE_OPERATIONAL_2. The CN is normally configured by the manager in this state. A command then switches the CN to the READY_TO_OPERATE state.
Triple flash (approx. 1 Hz)	Mode READY_TO_OPERATE. The manager switches the CN via command to the OPERATIONAL state.
On	Mode OPERATIONAL. The PDO mapping is active and cyclic data is evaluated.
Blinking (approx. 2.5 Hz)	Mode STOPPED. Output data is not being output, and no input data is being provided. It is only possible to switch to or leave this state after the manager has given the appropriate command.

Table 30: Status/Error LED lit green: LED indicating operating state

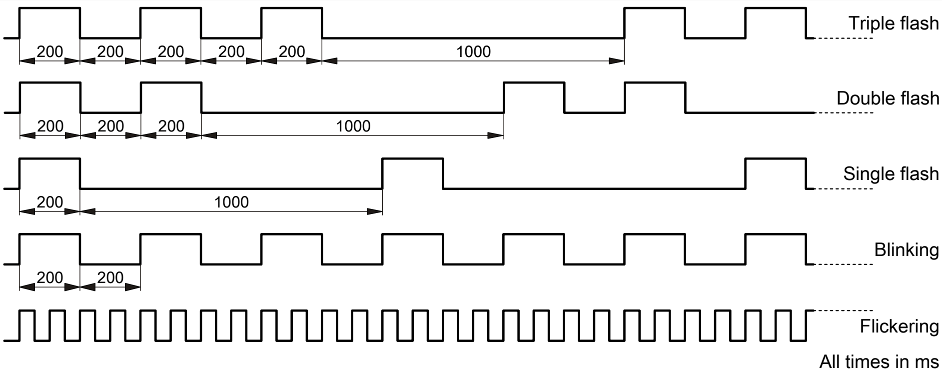


Figure 58: LED status indicators - Blink times

POWERLINK station number

Figure 59: POWERLINK station number switches

The station number of the POWERLINK station is set using the two number switches. Station numbers between 0x01 and 0xEF are permitted.

Switch position	Description
0x00	Reserved, switch position not permitted.
0x01 to 0xEF	Station number of the POWERLINK station, operation as controlled node (CN)
0xF0 to 0xFF	Reserved, switch position not permitted.

Table 31: POWERLINK station number

RJ45 ports

For information about wiring X20 modules with an Ethernet interface, see section "Mechanical and electrical configuration - Cabling guidelines for X20 modules with an Ethernet cable" of the X20 user's manual.

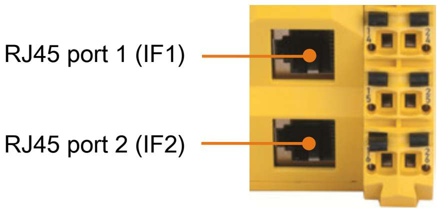


Figure 60: RJ45 ports

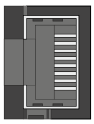
Interface	Pinout		
	Pin	Ethernet	
 1 Shielded RJ45 port	1	RXD	Receive data
	2	RXD\	Receive data\
	3	TXD	Transmit data
	4	Termination	
	5	Termination	
	6	TXD\	Transmit data\
	7	Termination	
	8	Termination	

Table 32: Pinout for RJ45 port

2.6.7.2.4.4 SG support

SG3 / SGC

The SafeLOGIC controller is not currently supported on SG3 and SGC target systems.

SG4

The SafeLOGIC controller comes with preinstalled firmware. In addition, the firmware version appropriate to the Safety Release will also be saved to the standard CPU when the Automation Studio project is downloaded.

If a different firmware version is being used, then the firmware saved on the standard CPU will automatically be loaded to the module.

When changing the safety-related firmware on the SafeLOGIC controller, the measures listed in section "Acknowledging a firmware modification" must be taken.

2.6.7.2.4.5 Integrated power supply

A power supply is integrated in the SafeLOGIC controller.

<u>LED status indicators for the integrated power supply</u>

X20SL81x0

Figure	LED	Color	Status	Description
	DCOK	Green	On	Voltage applied to module
			Off	Voltage not applied to module

Table 33: X20SL81x0 - LED status indicators for the integrated power supply

X20SL8101

Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	RESET mode
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Double flash	LED indicates one of the following states: • The SafeLOGIC controller / bus controller / X2X Link power supply for the power supply is overloaded • I/O power supply too low • Input voltage for the SafeLOGIC controller / bus controller / X2X Link power supply is too low
	e + r	Solid red / Single green flash		Invalid firmware
	l	Red	Off	The SafeLOGIC controller / bus controller / X2X Link power supply is in the valid range.
			On	The SafeLOGIC controller / bus controller / X2X Link power supply for the power supply is overloaded.

Table 34: X20SL8101 - LED status indicators for the integrated power supply

<u>Pinout for the integrated power supply</u>

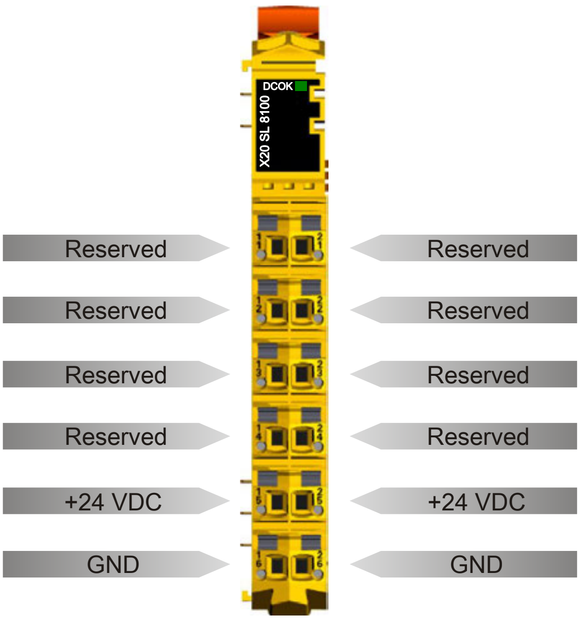


Figure 61: X20SL81x0 - Pinout of the integrated power supply

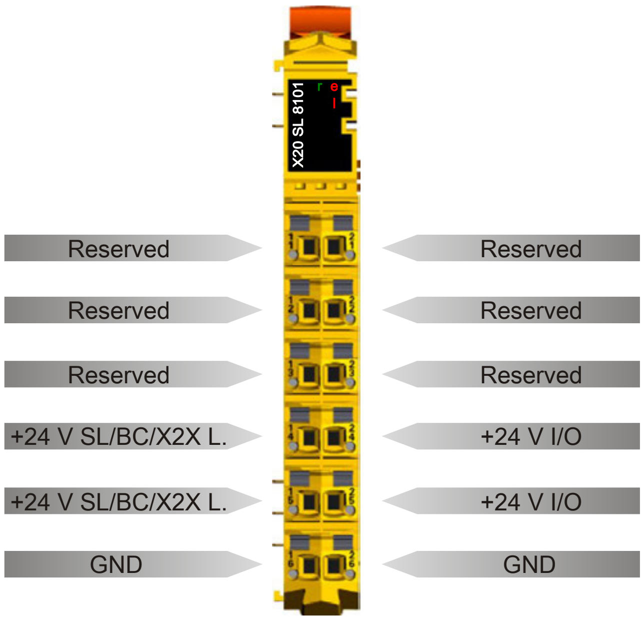


Figure 62: X20SL8101 - Pinout of the integrated power supply

Connection examples

X20SL81x0

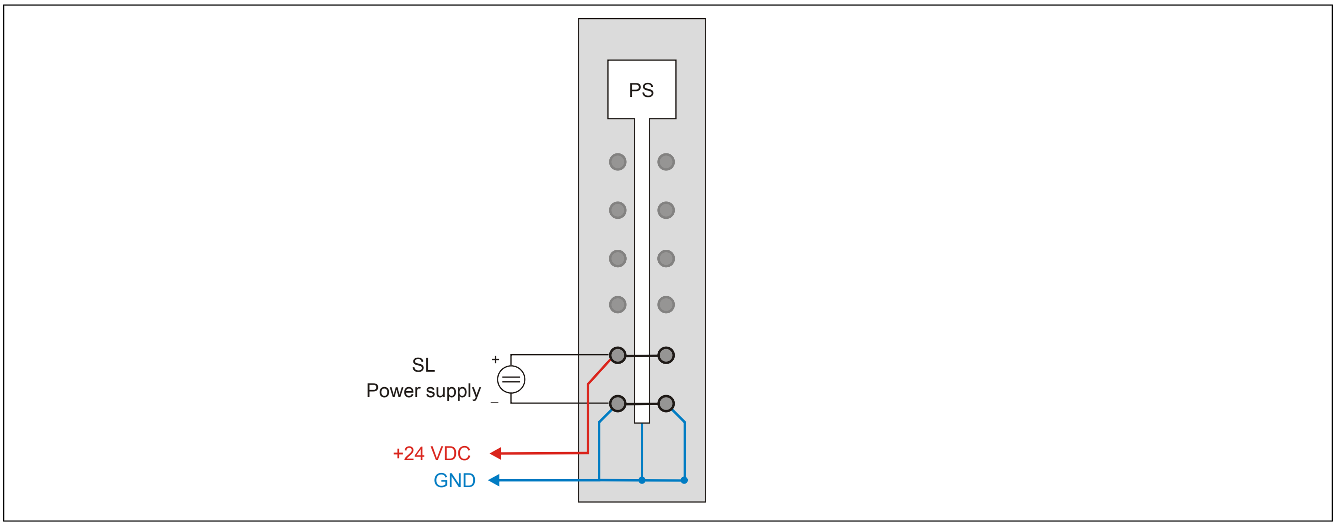


Figure 63: X20SL81x0 - Connection example

X20SL8101 - Connection example with 2 isolated power supplies

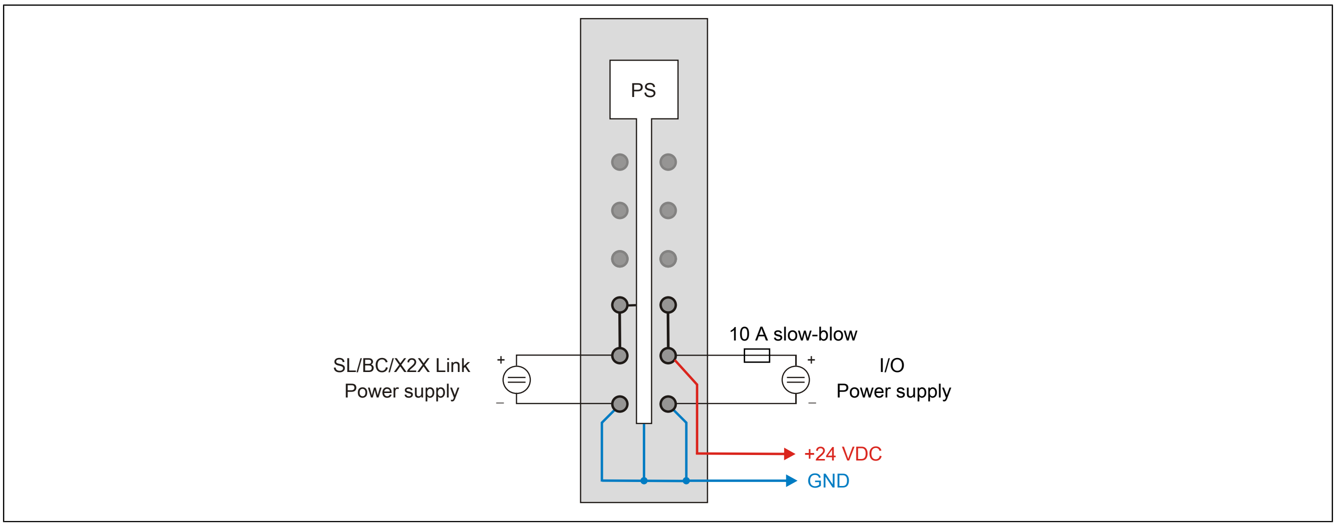


Figure 64: X20SL8101 - Connection example with 2 isolated power supplies

X20SL8101 - With one power supply and jumper

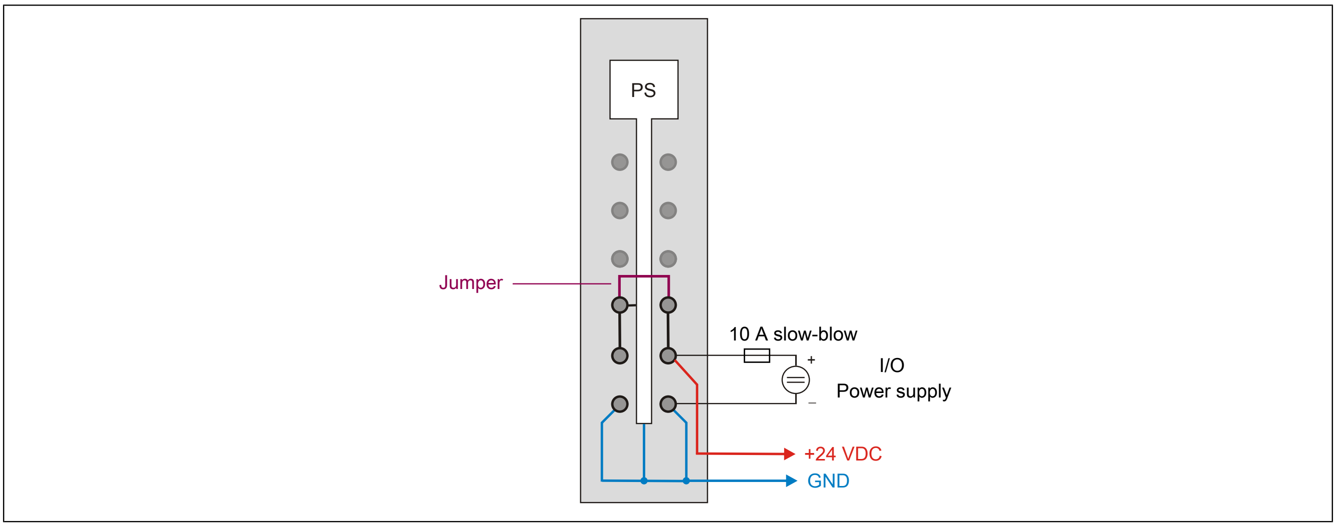


Figure 65: X20SL8101 - Connection example with a supply and jumper

2.6.7.2.5 Register description

2.6.7.2.5.1 Parameters in the I/O configuration

Group: POWERLINK parameters

Parameter	Description	Default value	Unit
Mode	SafeLOGIC can only be operated as a "controlled node" (CN). A "managing node" (MN) is not supported.	Controlled node	-

Table 35: Parameters I/O configuration: POWERLINK parameters

Information:

Additional configuration parameters are available.
For details, see Automation Help under "Communication → POWERLINK → AR configuration → POWERLINK controlled node configuration (SG4)".

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 36: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Missing module triggers service mode</td></tr><tr><td>Off</td><td>Missing module is ignored</td></tr></table></td></tr>
<tr><td rowspan="2">Interface Slot Enable
(only X20SL8110, hardware upgrade 1.10.1.3 or later)</td><td>This parameter enables data transfer to the interface card.</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Data transfer to the interface card is enabled.</td></tr><tr><td>Off</td><td>Data transfer to the interface card is disabled.</td></tr></table></td></tr>
<tr><td>Node used as IP gateway</td><td>This parameter is reserved for future functional expansions.</td><td>240</td><td>-</td></tr>
<tr><td rowspan="2">Standalone mode
(only X20SL8101, hardware upgrade 1.10.2.x or later and Automation Runtime A4.32 or later)</td><td>This parameter enables standalone mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode) and allows the SafeLOGIC controller to be started up without an active master.</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Standalone mode is enabled.</td></tr><tr><td>Off</td><td>Standalone mode is disabled.</td></tr></table></td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the unique SafeLOGIC address.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 1</td><td>1</td><td>-</td></tr>
<tr><td>SafeDESIGNER project</td><td>Name of the safety project</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeDESIGNER version</td><td>SafeDESIGNER version of the safety project for this SafeLOGIC controller.</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td rowspan="2">Authorization</td><td>For information about activating the "Authorization" function, see "Authorization (X20SL8xxx series only)".</td><td>Disabled</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Enabled</td><td>The "Authorization" function is enabled; the standard CPU can block acknowledgment actions from the SafeLOGIC controller.</td></tr><tr><td>Disabled</td><td>The "Authorization" function is disabled; the standard CPU has no effect on acknowledgment functions.</td></tr></table></td></tr>
</table>

Table 37: Parameters I/O configuration: General

Group: SafeDESIGNER to SafeLOGIC communication

Starting with SafeLOGIC V1.4.0.0 and Automation Runtime V3.04:

When SPROXY is enabled, the SafeLOGIC controller can be accessed via a TCP/IP port on the standard CPU.

This uses the SafeDESIGNER setting "SL communication via the CPU" (SafeDESIGNER V2.80 or higher).

Parameter	Description	Default value	Unit
Activate SPROXY	Enables the SafeDESIGNER online connection	On	-
Server communication port	TCP/IP port number used to access the SafeLOGIC controller • Recommended values: 50,000 to 50,100 **Note:** If multiple SafeLOGIC controllers are being used in the project, then a different port number must be configured for each one!	50000	-

Table 38: I/O configuration parameters: SafeDESIGNER to SafeLOGIC communication

Group: CPU to SafeLOGIC communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the CPU to the SafeLOGIC controller • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96.	8	-
Number of extended BOOL channels	Number of BOOL channels from the CPU to the SafeLOGIC controller • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256.	0	-
Number of INT channels	Number of INT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 30.	0	-
Number of UINT channels	Number of UINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 30.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 15.	0	-
Number of UDINT channels	Number of UDINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 15.	0	-

Table 39: Parameters I/O configuration: CPU to SafeLOGIC communication

Group: SafeLOGIC to CPU communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the SafeLOGIC controller to the CPU • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96.	8	-
Number of extended BOOL channels	Number of BOOL channels from the SafeLOGIC controller to the CPU • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256.	0	-
Number of INT channels	Number of INT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 30.	0	-
Number of UINT channels	Number of UINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 30.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 15.	0	-
Number of UDINT channels	Number of UDINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 15.	0	-

Table 40: Parameters I/O configuration: SafeLOGIC to CPU communication

Group: SafeLOGIC to SafeLOGIC communication

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Use as source SafeLOGIC</td><td>This parameter configures this SafeLOGIC controller as a data source for another SafeLOGIC controller.</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>This SafeLOGIC controller is available as a data source for another SafeLOGIC controller.</td></tr><tr><td>Off</td><td>This SafeLOGIC controller is not available as a data source for other SafeLOGIC controllers.</td></tr></table></td></tr>
<tr><td>Extended source SafeLOGIC communication (Safety Release 1.4 and Automation Runtime V3.08 required)</td><td>Enables the option of configuring the number of data points for "SafeLOGIC to SafeLOGIC communication" (for connections where this SafeLOGIC controller serves as a data source for another SafeLOGIC controller).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="4">Group: Connected SafeLOGIC modules
(Safety Release 1.4 and later)</td></tr>
<tr><td>Group: Connection xx</td><td colspan="3">Configuration of the maximum SafeLOGIC controllers to which this SafeLOGIC controller will establish a connection.</td></tr>
<tr><td>SafeLOGIC ID of connection xx</td><td>SafeLOGIC ID to which the connection should be established</td><td>0</td><td>-</td></tr>
<tr><td>Group: Output channels
(Safety Release 1.4 and Automation Runtime V3.08 required)</td><td colspan="3"></td></tr>
<tr><td>Number of BOOL channels</td><td rowspan="5">Number of channels with the respective data type</td><td>8</td><td>-</td></tr>
<tr><td>Number of INT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of UINT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of DINT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of UDINT channels</td><td>0</td><td>-</td></tr>
<tr><td>Group: Input channels
(Safety Release 1.4 and Automation Runtime V3.08 required)</td><td colspan="3"></td></tr>
<tr><td>Number of BOOL channels</td><td rowspan="5">Number of channels with the respective data type</td><td>8</td><td>-</td></tr>
<tr><td>Number of INT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of UINT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of DINT channels</td><td>0</td><td>-</td></tr>
<tr><td>Number of UDINT channels</td><td>0</td><td>-</td></tr>
</table>

Table 41: Parameters I/O configuration: SafeLOGIC to SafeLOGIC communication

Group: Power Supply Parameter (X20SL8101 only)

Parameter	Description	Default value	Unit
Module status information	This parameter enables/disables additional status information in the I/O mapping.	On	-
Current/voltage information	This parameter enables/disables additional current and voltage information in the I/O mapping.	Off	-

Table 42: I/O configuration parameters: Power Supply Parameter

2.6.7.2.5.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td>Cycle_Time_us</td><td>This parameter determines the cycle time of the SafeLOGIC controller.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)
The set value is internally rounded up to the next whole number multiple of the POWERLINK cycle time.</td><td>2000</td><td>µs</td></tr>
<tr><td>Cycle_Time_max_us
(Release 1.5 and later)</td><td>Parameter for checking whether a maximum time between 2 SafeLOGIC cycles is exceeded.
• Permissible values: 800 to 21,000 µs (corresponds to 0.8 to 21 ms)
IMPORTANT:
This value should not be the same as the actual cycle time. Network jitter must also be taken into account.
The actual cycle time is affected by parameter "Cycle_Time_us".</td><td>21000</td><td>µs</td></tr>
<tr><td rowspan="2">SSDO_Creation</td><td>This parameter defines the number of asynchronous processing steps per SafeLOGIC cycle.
This parameter can be used to optimize the startup behavior of the system.</td><td>Time dependent</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Time dependent</td><td>Depends on the SafeLOGIC cycle time
• Cycle times ≤3 ms = 1 per 5 cycles
• Cycle times >3 ms = 1 per cycle</td></tr>
<tr><td>1 per 5 cycles</td><td>One asynchronous processing step is distributed over 5 SafeLOGIC cycles
• Can lead to long startup times
• Minimum possible communication overhead in each cycle</td></tr>
<tr><td>1 per cycle</td><td>One asynchronous processing step per SafeLOGIC cycle
• Average startup times
• Average communication overhead in each cycle</td></tr>
<tr><td>5 per cycle</td><td>5 asynchronous processing steps per SafeLOGIC cycle
• Minimum startup times
• Maximum possible communication overhead in each cycle</td></tr>
</table></td></tr>
<tr><td>Node_Guarding_Timeout_s</td><td>Timeout for changing the safety modules to the PRE_OPERATIONAL state after the SafeLOGIC controller drops out or if there is a communication problem between the safety module and the SafeLOGIC controller.
This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module.
• Permissible values: 30 to 3000 s
Notes
• The shorter the time, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time".</td><td>60</td><td>s</td></tr>
<tr><td>Number_of_scans</td><td>This parameter defines the number of module search scans completed during startup.
This parameter is used to optimize the startup behavior of the system, especially if optional modules are configured but not available.
• Permissible values: 1 to 10</td><td>5</td><td>-</td></tr>
<tr><td rowspan="2">ExternalMachineOptions
(Release 1.4 and later)</td><td>Enables external machine options</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>External machine options are enabled.</td></tr>
<tr><td>No</td><td>External machine options are disabled.</td></tr>
</table></td></tr>
<tr><td rowspan="2">ExternalStartupFlags
(Release 1.4 and later)</td><td>Enables external startup flags</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>External startup flags are enabled.</td></tr>
<tr><td>No</td><td>External startup flags are disabled.</td></tr>
</table></td></tr>
<tr><td rowspan="2">KeepRemanent</td><td>Automatically resets the remanent data (see Automation Help for SafeDESIGNER function block "SF_RemanentData_SAFEDINT" or "SF_RemanentData_SAFEDWORD")</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Remanent data not automatically reset</td></tr>
<tr><td>No</td><td>Remanent data is automatically reset if a modified SafeDESIGNER project (modified CRC and/or timestamp) is loaded to the SafeLOGIC controller.</td></tr>
</table></td></tr>
</table>

Table 43: SafeDESIGNER parameters: Basic

Information:

Parameter "Cycle_Time_us" must be greater than the processing time for the safety application. The processing time can be determined in the online dialog window using function "Info". If parameter "Cycle_Time_us" is less than or too close to the necessary processing time, a cycle time violation can occur.

Additional information can also be found in section "SafeLOGIC "Info" dialog box in SafeDESIGNER".

Danger!

If parameter "ExternalMachineOptions" or "ExternalStartupFlags" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Danger!

If parameter "KeepRemanent" is set to "Yes-ATTENTION", it is important when saving data after a project download to note that the data still has the same meaning in the application program.

Group: Safety_Response_Time_Defaults

The parameters for the safety response time are generally set in the same way for all stations involved in the application. This is why these parameters are configured for the SafeLOGIC controller in the "Safety_Response_Time_Defaults" group in SafeDESIGNER.

If "Manual_Configuration = No" is set for the individual modules, then these default values are used.

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Default_Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks.</td></tr></table></td></tr>
<tr><td>Default_Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Default_Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Default_Max_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 30,000 µs (corresponds to 0 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Default_Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Default_Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Default_Min_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 30,000 µs (corresponds to 0 to 30 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Default_Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 500,000 µs (corresponds to 3 to 500 ms)</td><td>50000</td><td>µs</td></tr>
<tr><td>Default_Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 44: SafeDESIGNER parameters: Safety_Response_Time_Defaults

Group: Commissioning

Parameters "SafeMachineOption00" to "SafeMachineOption31" make it possible to activate or deactivate dedicated machine options during commissioning.

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">SafeMachineOptionXX</td><td>With this parameter, individual machine options can be enabled or disabled during commissioning.</td><td>OFF</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>ON</td><td>Enables machine option XX.
The "SafeMachineOptionXX" channel is constantly set to SAFETRUE.</td></tr><tr><td>OFF</td><td>Disables machine option XX.
The "SafeMachineOptionXX" channel is constantly set to SAFEFALSE.</td></tr></table></td></tr>
</table>

Table 45: SafeDESIGNER parameters: Commissioning

2.6.7.2.5.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic release</td><td>-</td></tr>
<tr><td>SSDO Creation</td><td>This parameter defines the number of asynchronous processing steps per SafeLOGIC cycle.
It can be used to optimize the boot behavior of the system.
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Time dependent</td><td>Depends on the SafeLOGIC cycle time
• Cycle times ≤3 ms = 1 per 5 cycles
• Cycle times >3 ms = 1 per cycle</td></tr>
<tr><td>1 per 5 cycles</td><td>One asynchronous processing step is distributed over 5 SafeLOGIC cycles
• Can lead to long boot times
• Minimum possible communication overhead in each cycle</td></tr>
<tr><td>1 per cycle</td><td>One asynchronous processing step per SafeLOGIC cycle
• Average boot times
• Average communication overhead in each cycle</td></tr>
<tr><td>5 per cycle</td><td>5 asynchronous processing steps per SafeLOGIC cycle
• Minimum boot times
• Maximum possible communication overhead in each cycle</td></tr>
</table></td><td>Time dependent</td><td>-</td></tr>
<tr><td>Node Guarding Timeout</td><td>Timeout for changing the safety modules to the PRE_OPERATIONAL state after the SafeLOGIC controller drops out or if there is a communication problem between the safety module and the SafeLOGIC controller.
This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module.
• Permissible values: 30 to 300 s
Notes
• The shorter the time, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>60</td><td>s</td></tr>
<tr><td>Number of scans</td><td>This parameter defines the number of module search scans completed while booting.
This parameter is used to optimize the startup behavior of the system, especially if optional modules are configured but not available.
• Permissible values: 1 to 10</td><td>5.
Hardware upgrade 1.10.1.0 or later: 3</td><td>-</td></tr>
<tr><td>Activate Setup Mode on empty SafeKEY
(hardware upgrade 1.10.2.x or later)</td><td>This parameter enables setup mode after downloading a project to a blank SafeKEY.
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Setup mode is enabled.</td></tr>
<tr><td>No</td><td>Setup mode is disabled.</td></tr>
</table></td><td>No</td><td>-</td></tr>
<tr><td>Auto acknowledge firmware mismatch
(hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a firmware exchange (acknowledgment request "Firmware Acknowledge").
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of firmware exchange is enabled.</td></tr>
<tr><td>No</td><td>Automatic acknowledgment of firmware exchange is not enabled.</td></tr>
</table></td><td>No</td><td>-</td></tr>
<tr><td>Auto acknowledge SafeKEY exchange
(hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a SafeKEY exchange (acknowledgment request "SafeKEY Exchange").
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of SafeKEY exchange is enabled.</td></tr>
<tr><td>No</td><td>Automatic acknowledgment of SafeKEY exchange is not enabled.</td></tr>
</table></td><td>No</td><td>-</td></tr>
<tr><td>Process Data Transmission Rate
(hardware upgrade 1.10.5.x or later)</td><td>This parameter defines the base transfer rate for process data.
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>High</td><td>Normal transfer rate.</td></tr>
<tr><td>Low</td><td>Reduced transfer rate to support networks with low transfer rates (data transmission time >1 s).</td></tr>
</table></td><td>High</td><td>-</td></tr>
</table>

Table 46: SafeDESIGNER parameters: Basic

Information:

Startup time is also affected by the asynchronous bandwidth on the POWERLINK network. For optimization options, see Automation Help under Communication → POWERLINK → General information → Multiple asynchronous send.

Information:

The information in section "Setup mode" on page 229 must be observed when using parameter "Activate Setup Mode on empty SafeKEY". The information in section "Automatic acknowledgment" on page 202 must be observed when using parameters "Auto acknowledge firmware mismatch" and "Auto acknowledge SafeKEY exchange".

Group: Safety Response Time Defaults

The parameters for the safety response time are generally set in the same way for all stations involved in the application. This is why these parameters are configured for the SafeLOGIC controller in group "Safety Response Time Defaults" in SafeDESIGNER.

If "Manual Configuration = No" is set for the individual modules, then these default values are used.

Parameter	Description	Default value	Unit
Default Safe Data Duration	This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller SafeIO module. For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added. • Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)	20000	µs
Default Additional Tolerated Packet Loss	This parameter specifies the number of additionally tolerated lost packets during data transfer. • Permissible values: 0 to 10	0	Packets
Default Packets per Node Guarding	This parameter specifies the maximum number of packets used for node guarding. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.	5	Packets

Table 47: SafeDESIGNER parameters: Safety Response Time Defaults

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">External Machine Options</td><td>Enables external machine options</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Enables external machine options</td></tr><tr><td>No</td><td>Disables external machine options</td></tr></table></td></tr>
<tr><td rowspan="2">External Startup Flags</td><td>Enables external startup flags</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Enables external startup flags</td></tr><tr><td>No</td><td>Disables external startup flags</td></tr></table></td></tr>
<tr><td rowspan="2">Keep Remanent</td><td>Automatically resets the remanent data (see Automation Help for SafeDESIGNER function block "SF_RemanentData_SAFEDINT" or "SF_RemanentData_SAFEDWORD")</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Remanent data not automatically reset</td></tr><tr><td>No</td><td>Remanent data is automatically reset if a modified SafeDESIGNER project (modified CRC and/or timestamp) is loaded to the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Cycle Time</td><td>This parameter determines the cycle time of the safety application.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)
The configured value is internally rounded up to the next whole number multiple of the POWERLINK cycle time.</td><td>2000</td><td>µs</td></tr>
<tr><td>Cycle Time max
(up to hardware upgrade 1.10.1.0)</td><td>Parameter for checking whether a maximum time between 2 SafeLOGIC cycles is exceeded.
• Permissible values: 800 to 21,000 µs (corresponds to 0.8 to 21 ms)
Important:
This value should not be the same as the actual cycle time. Network jitter must also be taken into account.
The actual cycle time is affected by the "Cycle Time" parameter.</td><td>21000</td><td>µs</td></tr>
</table>

Table 48: SafeDESIGNER parameters: Basic

Information:

The parameter "Cycle Time" must be greater than the processing time for the safety application. The processing time can be determined in the online dialog window using function "Info". If the parameter "Cycle Time" is less than or too close to the necessary processing time, a cycle time violation can occur.

Additional information can also be found in section "SafeLOGIC "Info" dialog box in SafeDESIGNER".

Danger!

If parameter "External Machine Options" or "External Startup Flags" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Danger!

If parameter "Keep Remanent" is set to Yes-ATTENTION, it is important when saving data after a project download to note that the data still has the same meaning in the application program.

Group: Commissioning

Parameters "SafeMachineOption00" to "SafeMachineOption31" make it possible to activate or deactivate dedicated machine options during commissioning.

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">SafeMachineOptionXX</td><td>With this parameter, individual machine options can be enabled or disabled during commissioning.</td><td>OFF</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>ON</td><td>Enables machine option XX.
The "SafeMachineOptionXX" channel is constantly set to SAFETRUE.</td></tr>
<tr><td>OFF</td><td>Disables machine option XX.
The "SafeMachineOptionXX" channel is constantly set to SAFEFALSE.</td></tr>
</table></td></tr>
</table>

Table 49: SafeDESIGNER parameters: Commissioning

2.6.7.2.5.4 SafeLOGIC - Channel list

Channel name	Access via Automation Studio	Access via SafeDESIGNER	Data type	Description
ModuleOk	Read	-	BOOL	Indicates if the module is OK
SerialNumber	Read	-	UDINT	Module serial number
ModuleID	Read	-	UDINT	Module ID
HardwareVariant	Read	-	UDINT	Hardware variant
FirmwareVersion	Read	-	UDINT	Firmware version of the module
SafeFirmwareVersion	Read	-	UINT	Hardware upgrade 1.10.1.4 or later: Channel for reading the version of the safe firmware
UDID_low	Read	-	UDINT	UDID, lower 4 bytes
UDID_high	Read	-	UINT	UDID, upper 2 bytes
BOOL1xx	Write	Read	BOOL	CPU to SafeLOGIC communication channel
BOOLext1xxx	Write	Read	BOOL	CPU to SafeLOGIC communication channel
INT1xx	Write	Read	INT	CPU to SafeLOGIC communication channel
UINT1xx	Write	Read	UINT	CPU to SafeLOGIC communication channel
DINT1xx	Write	Read	DINT	CPU to SafeLOGIC communication channel
UDINT1xx	Write	Read	UDINT	CPU to SafeLOGIC communication channel
BOOL0xx	Read	Write	BOOL	SafeLOGIC to CPU communication channel
BOOLext0xxx	Read	Write	BOOL	SafeLOGIC to CPU communication channel
INT0xx	Read	Write	INT	SafeLOGIC to CPU communication channel
UINT0xx	Read	Write	UINT	SafeLOGIC to CPU communication channel
DINT0xx	Read	Write	DINT	SafeLOGIC to CPU communication channel
UDINT0xx	Read	Write	UDINT	SafeLOGIC to CPU communication channel
SafeBOOLx	-	Write	SAFEBOOL	SafeLOGIC to SafeLOGIC communication channel
SafeMachineOptionxx	-	Read	SAFEBOOL	Internal channel for machine options
ExternalMachineOptionsBITxxx	-	Read	SAFEBOOL	Internal channels for external machine options
ExternalMachineOptionsINTxx	-	Read	SAFEINT	Internal channels for external machine options
ExternalMachineOptionsUINTxx	-	Read	SAFEWORD	Internal channels for external machine options
ExternalMachineOptionsDINTxx	-	Read	SAFEDINT	Internal channels for external machine options
ExternalMachineOptionsUDINTxx	-	Read	SAFEDWORD	Internal channels for external machine options

Table 50: SafeLOGIC - Channel list

Information:

Channels for SafeLOGIC to SafeLOGIC communication: See Display in SafeDESIGNER

Information:

Additional diagnostic data points are available on the X20SL8101 and the X20SL8110.

For details, see Communication → POWERLINK → Diagnostics → Diagnostic data points → Bus controller in Automation Help.

In addition, the following data can be read via POWERLINK registers:

Index:Subindex	Object name	Data type	Access	Values	Description
0x2000:0x04	SafetyFWversion1	UDINT	Read	-	Higher-order 2 bytes: Hardware variant of the module Lower-order 2 bytes: Firmware version - Safety processor 1
0x2000:0x05	SafetyFWversion2	UDINT	Read	-	Higher-order 2 bytes: Hardware variant of the module Lower-order 2 bytes: Firmware version - Safety processor 2
0x2000:0x08	Project_CRC	UDINT	Read	-	CRC of the SafeDESIGNER project
0x2000:0x09	Project_Time	DATE_AND_TIME	Read	-	Timestamp of the SafeDESIGNER project
0x2000:0x0C	Project_Name	STRING (without zero termination)	Read	-	Project name of the SafeDESIGNER project
0x2000:0x0D	Project_Author	STRING (without zero termination)	Read	-	Name of the author of the SafeDESIGNER project
0x2000:0x0E	SafeOS_RUN_STATE	BOOL	Read	0	SafeOS is not in RUN (identical to SafeOSstate!=0x66)
				1	SafeOS is in RUN (identical to SafeOSstate==0x66)
0x2000:0x0F	BOOT_STATE	UDINT	Read	General firmware startup status. Using the updated "Bootstate" object (0x2410:0x01) is recommended.	
				0x00	Startup not yet begun
				0x01	Initialization started
				0x10	Cyclic hardware tests running
				0x11	openSAFETY stack running
				0x12	SafeOS running
0x2000:0x10	openSAFETYstate	UDINT	Read	0	PREOPERATIONAL state (all cyclic safe data zeroed out)
				1	OPERATIONAL state
0x2000:0x11	SafeOsState	UDINT	Read	Status of the safety application, corresponds to the R/E LED on the SafeLOGIC controller. For details, see "SafeLOGIC "Info" dialog box in SafeDESIGNER".	
				0x00	Invalid (e.g. SafeKEY blank) or startup still active (BOOT_STATE!=0x12)
				0x0F	ON (startup / internal initialization) or error (check logbook)
				0x33	Loading (startup / internal initialization)
				0x55	Stop [Safe]
				0x66	Run [Safe]
				0x99	Halt [Debug]
				0xAA	Stop [Debug]
				0xCC	Run [Debug]
				0xF0	No execution
0x2000:0x12	Temperature	INT	Read	-	Measured temperature in 0.1°C

The following objects are available in hardware upgrade 1.10.4.0 and later:

Index:Subindex	Data type	Access	Values	Description
0x2410:0x01	UDINT	Read	Boot state. Startup state of the SafeLOGIC controller. Notes: • Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally. • The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.	
			0x0003	Startup communication processor OK, no communication to the safety processors
			0x0008	SafeKEY check (valid SafeKEY not connected)
			0x0010	FAILSAFE. At least one of the safety processors is in the safe state.
			0x0020	Internal communication to safety processors started
			0x0024	Firmware update of safety processors
			0x0030	Startup of safety processors
			0x0040	Firmware of safety processors started
			0x0440	Firmware of safety processors running
			0x0840	Loads the SafeDESIGNER application or valid SafeDESIGNER application not found
			0x1840	Waiting for acknowledgments (e.g. module replacement)
			0x2040 ... 0x2A40	SCAN: The safety modules being used are being looked for in the network and configured. Multiple scan cycles are carried out based on SafeDESIGNER parameter "Number of Scans" until all modules are found: 0x2040: First cycle 0x2140: Second cycle 0x2240: Third cycle ...
			0x3040	Missing modules. Startup cannot be resumed since modules are missing that are configured with "Optional = No".
			0x3440	Configuration of existing safety modules completed. Stabilizing cyclic openSAFETY data exchange. **Note:** If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".
			0x4040	RUN. Final state, startup completed.
0x2410:0x02	UDINT	Read	-	SCAN progress (how many modules have already been processed in the current scan)

Index:Subindex	Data type	Access	Values	Description
0x2410:0x03	UDINT	Read	-	Supply voltage (in mV)
0x2410:0x04	UDINT	Read	-	CRC of firmware header on safety processor 1
0x2410:0x05	UDINT	Read	-	CRC of firmware header on safety processor 2
0x2410:0x06	UDINT	Read	-	Maximum cycle time (time from cycle start to cycle end)
0x2410:0x07	UDINT	Read	-	Cycle start interval (time from one cycle start to next cycle start)
0x2410:0x08	UDINT	Read	-	SafeLOGIC status word
0x2410:0x09	UDINT	Read	-	Number of missing modules
0x2410:0x0A	UDINT	Read	-	Number of UDID mismatches
0x2410:0x0B	UDINT	Read	-	Number of firmware mismatches
0x2410:0x0C	UDINT	Read	-	Number of configured modules
0x2410:0x0D	UDINT	Read	-	Flag for missing subsequently loadable files: Bit 0: Machine options missing in AUTOCNF.BIN Bit 1: Startup flags missing in AUTOCNF.BIN Bit 2: EMODATA1.BIN missing Bit 3: TABDATA1.BIN
0x2410:0x0E	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_LENGTH
0x2410:0x0F	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_TOO_LONG
0x2410:0x10	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_FRM_ID
0x2410:0x11	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_SADR_INV
0x2410:0x12	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_SDN_INV
0x2410:0x13	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_TADR_INV
0x2410:0x14	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_CRC1
0x2410:0x15	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_CRC2
0x2410:0x16	UDINT	Read	-	openSAFETY common event counter SERR_k_SFS_DATA
0x2410:0x17	UDINT	Read	-	openSAFETY common event counter SERR_k_CYC_REJECT
0x2410:0x18	UDINT	Read	-	openSAFETY common event counter SERR_k_CYC_ERROR
0x2410:0x19	UDINT	Read	-	openSAFETY common event counter SERR_k_ACYC_REJECT
0x2410:0x1A	UDINT	Read	-	openSAFETY common event counter SERR_k_ACYC_RETRY
0x2410:0x1B to 0x2410:0x1F	UDINT	Read	-	Reserved for future openSAFETY common event counters
0x2410:0x20	UDINT	Read	-	Number of SCFM errors
0x2410:0x21	UDINT	Read	-	Number of SCM errors
0x2410:0x22	UDINT	Read	-	Number of SDN errors
0x2410:0x23	UDINT	Read	-	Number of SFS errors
0x2410:0x24	UDINT	Read	-	Number of SHNF errors
0x2410:0x25	UDINT	Read	-	Number of SNMTM errors
0x2410:0x26	UDINT	Read	-	Number of SNMTS errors
0x2410:0x27	UDINT	Read	-	Number of SOD errors
0x2410:0x28	UDINT	Read	-	Number of SPDO errors
0x2410:0x29	UDINT	Read	-	Number of SSC errors
0x2410:0x2A	UDINT	Read	-	Number of SSDOC errors
0x2410:0x2B	UDINT	Read	-	Number of SSDOS errors
0x2410:0x2C to 0x2410:0xFE	UDINT	Read	-	Reserved for future expansions
0x2424:0x01	UDINT	Read	-	AutoCnf.bin - Timestamp
0x2424:0x02	UDINT	Read	-	AutoCnf.bin - Number of CRCs
0x2424:0x03	UDINT	Read	-	AutoCnf.bin - Size of file in bytes
0x2424:0x04 to 0x2424:0x0A	UDINT	Read	-	AutoCnf.bin - Reserved for future expansions
0x2424:0x0B to 0x2424:0xn	UDINT	Read	-	AutoCnf.bin - CRC 1 to N
0x2424:0xn+1 to 0x2424:0xFE	UDINT	Read	-	AutoCnf.bin - Reserved for future expansions
0x2425:0x01	UDINT	Read	-	EmoData1.bin - Timestamp
0x2425:0x02	UDINT	Read	-	EmoData1.bin - Number of CRCs
0x2425:0x03	UDINT	Read	-	EmoData1.bin - Size of file in bytes
0x2425:0x04 to 0x2425:0x0A	UDINT	Read	-	EmoData1.bin - Reserved for future expansions
0x2425:0x0B to 0x2425:0xn	UDINT	Read	-	EmoData1.bin - CRC 1 to N
0x2425:0xn+1 to 0x2425:0xFE	UDINT	Read	-	EmoData1.bin - Reserved for future expansions
0x2426:0x01	UDINT	Read	-	TabData1.bin - Timestamp
0x2426:0x02	UDINT	Read	-	TabData1.bin - Number of CRCs
0x2426:0x03	UDINT	Read	-	TabData1.bin - Size of file in bytes
0x2426:0x04 to 0x2426:0x0A	UDINT	Read	-	TabData1.bin - Reserved for future expansions
0x2426:0x0B to 0x2426:0xn	UDINT	Read	-	TabData1.bin - CRC 1 to N
0x2426:0xn+1 to 0x2426:0xFE	UDINT	Read	-	TabData1.bin - Reserved for future expansions
0x2427:0x01	UDINT	Read	-	ParData1.bin - Timestamp
0x2427:0x02	UDINT	Read	-	ParData1.bin - Number of CRCs

Index:Subindex	Data type	Access	Values	Description
0x2427:0x03	UDINT	Read	-	ParData1.bin - Size of file in bytes
0x2427:0x04 to 0x2427:0x0A	UDINT	Read	-	ParData1.bin - Reserved for future expansions
0x2427:0x0B to 0x2427:0xn	UDINT	Read	-	ParData1.bin - CRC 1 to N
0x2427:0xn+1 to 0x2427:0xFE	UDINT	Read	-	ParData1.bin - Reserved for future expansions

The following information about each openSAFETY node can be retrieved in object range 0x2416 to 0x2423 (data type: UDINT, Access: Read):

Parameter ID	Value
0	SafeModule ID
1	Status word Bit 0: Missing module Bit 1: Firmware mismatch on module Bit 2: UDID mismatch on module Bit 3: Reserved Bit 4: Reserved Bit 5: "Connection valid" bit of module Bit 6 to 31: Reserved
2	Connection valid statistics (number of negative edges of the connection valid bit)
3	Propagation delay statistics (average value of the data transmission time). Unit: 100 µs

The following formulas must be used to calculate the index/subindex.

$$Index = \frac{Module\ \ number}{23} + 0x2416$$

$$Subindex = Parameter\ \ ID + \{[(Module\ \ number - 1)\ \%\ 23] \times 11\}\ \%\ 254 + 1$$

Module number: Sequential number of the desired module
Parameter ID: See previous table

2.6.7.2.5.5 Power supply module (X20SL8101 only) - Channel list

A power supply module is already integrated on station 1 on the X2X Link.

Channel name	Access via Automation Studio	Access via SafeDESIGNER	Data type	Description
ModuleOk	Read	-	BOOL	Indicates if the module is OK
ModuleID	Read	-	UINT	Module code
HardwareVariant	Read	-	UINT	Hardware variant
FirmwareVersion	Read	-	UINT	Firmware version of the module
StatusInput01	Read	-	BOOL	Warning if overcurrent (>2.3 A) or undervoltage (<4.7 V)
StatusInput02	Read	-	BOOL	I/O power supply below the warning level of 20.4 V
SupplyCurrent	Read	-	USINT	Bus supply current with a resolution of 0.1 A
SupplyVoltage	Read	-	USINT	Bus supply voltage with a resolution of 0.1 V

Table 51: Power supply module channel list

2.6.7.2.5.6 SafeLOGIC "Info" dialog box in SafeDESIGNER

Dialog box "SafePLC info" appears if the "Info" button in dialog box "SafePLC" (control dialog box) or in dialog box "Debug" is pressed.

The dialog box shows information about the current project in the safe programming system, the project stored/running on the safety controller, the current status of the safety controller, debugging information, etc.

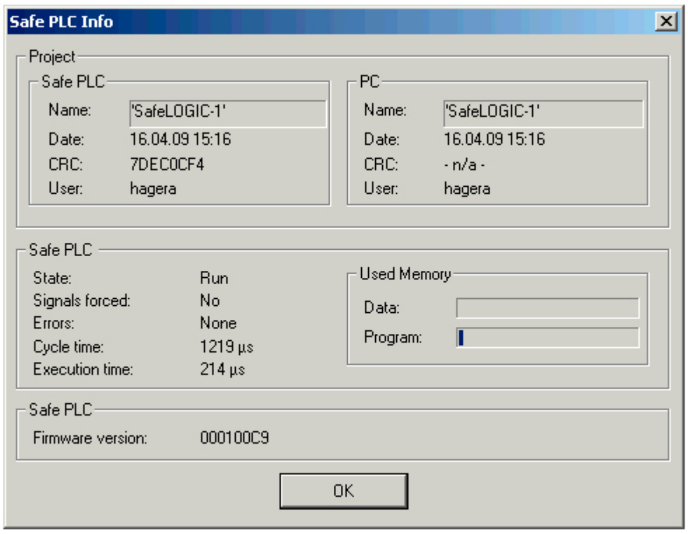


Figure 66: SafeLOGIC "Info" dialog box

Project	**Project-defining data**	
Safe PLC	Project data saved on the SafeKEY being used for the SafeLOGIC controller	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC
	User	User who made the last change
PC	SafeDESIGNER project data on the PC	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC, "- n/a -" if the project is not yet compiled
	User	User who made the last change
Safe PLC	**Status and information about the SafeLOGIC controller**	
State	Indicates the operating states of the safety controller.	
Signals forced	No	No variables are forced.
	Yes	Variables are forced.
Errors	Information regarding error messages present in the SafeDESIGNER message window	
Cycle time	Cycle time that is actually required, maximum value since the last power up This value is only relevant if "Safe PLC state = Run".	
Execution time	Actual application execution time This value corresponds to the "Safe PLC Cycle time" minus system and communication overhead.	
Used memory	Bar that shows the system resources being used	
	Data	Data memory for the safety application
	Program	Application memory for the safety application
Firmware version	Firmware version	

2.6.7.2.6 Maintenance scenarios

The operating elements on the SafeLOGIC controller (X20SL8xxx series) or the operating elements of the "Remote Control" in SafeDESIGNER (X20SL8xxx series and X20SLXxxx series) are available to handle the following maintenance scenarios.

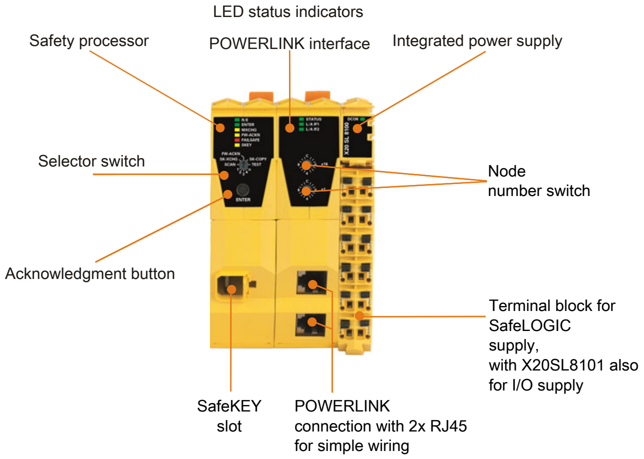


Figure 67: X20SL810x - Operating elements

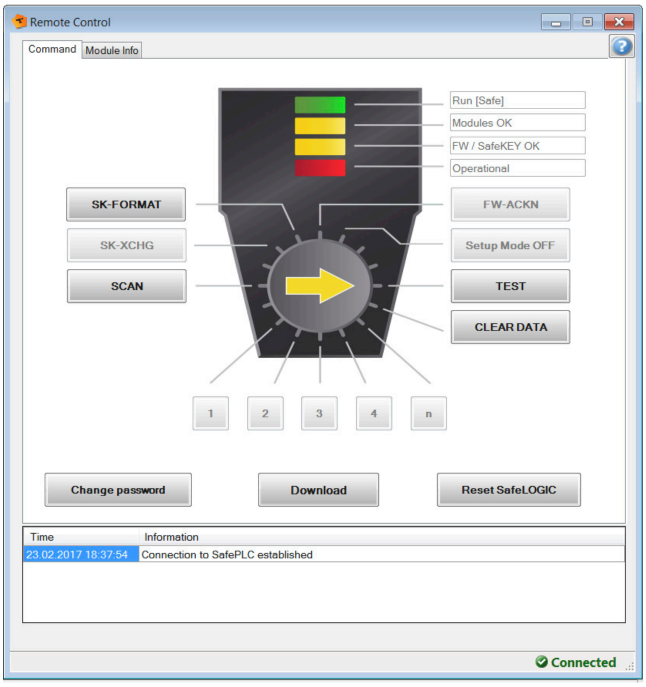


Figure 68: SafeDESIGNER - "Remote control" operating elements

For a detailed description of operating elements, see section Operating and connection elements of the technical data sheet for X20SL8xxx-series devices.

For a detailed description of operating elements, see SafeDESIGNER section Operating elements of the Remote Control in Automation Help.

2.6.7.2.6.1 Module replacement

The SafeLOGIC controller recognizes on its own when safe modules have been replaced. Following a module replacement, the complete system (SafeLOGIC, SafeLOGIC-X system components, openSAFETY) automatically ensures that the module operates again using the correct parameters and that incompatible modules are rejected. Nevertheless, the following errors are still possible after a module replacement:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

Terminals swapped between several modules

To determine whether terminals have been swapped between several modules, the user must test the safety function by performing a wiring test.

Danger!

The user must ensure that the wiring test can detect when terminals have been swapped.

Be sure to validate the entire safety function!

Wiring errors

A wiring error can occur if the wiring between the sensor or actuator and the X20 terminal is disconnected. To detect this sort of error in the wiring, the user must test the safety function by performing a wiring test.

Danger!

The user must make sure that the wiring test can detect wiring errors.

Be sure to validate the entire safety function!

SafeIO modules swapped with each other

Errors in the standard application can cause SafeIO modules to become swapped, which appears identical to a module replacement to the SafeLOGIC controller. To detect this error, the user must confirm the number of replaced modules. This links the number of modules replaced by the user and the replacements recognized by the system so that any additional replacements can be detected.

The user is informed of the number of detected module replacements via the MXCHG status. In the process, the module identifiers (UDIDs) on the SafeKEY or in the safety section of the CompactFlash card are compared to the UDIDs of the modules in the network.

If there are 1, 2, 3 or 4 different UDIDs, the user is provided information about the exact number of differences. The user must then check whether the number of replaced modules recognized by the SafeLOGIC controller corresponds to the actual number of replaced modules. If the values are the same, the user must confirm the number and perform a wiring test. This wiring test can be limited specifically to the modules that have been replaced.

If there are more than 4 different UDIDs, a standard message is provided indicating that there are differences on more than 4 modules. In this case, the user must perform a comprehensive wiring test for all modules.

If the number of modules indicated and the actual number of replaced modules do not match, the user must confirm the number of replacements determined by the SafeLOGIC controller and perform a comprehensive wire test for all modules.

Danger!

Be sure to validate the entire safety function!

Replacing an individual module

If only one module was replaced (MXCHG status indicates 1 module was replaced) and the wiring was not changed, the user can skip the wiring test because in this case the following errors can be ruled out:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

Danger!

The wiring test can only be excluded if no additional changes are made when replacing an individual module (e.g. unplugging terminals, removing the wiring, etc.).

Confirming a module replacement

To confirm the number of the replaced modules, the correct number of modules must be selected:

- 1 - One module replaced
- 2 - Two modules replaced
- 3 - Three modules replaced
- 4 - Four modules replaced
- n - Five or more modules replaced

The replacement can be confirmed and the accompanying wiring test can be limited to the replaced modules when up to four modules are replaced. When more than four modules are replaced, a comprehensive wiring test must be performed for all modules.

Following confirmation of the module replacement, the SafeLOGIC controller immediately commences a module scan.

Danger!

The user must ensure that the wiring test can detect a wiring error or when terminals have been swapped.

Be sure to validate the entire safety function!

2.6.7.2.6.2 Other errors in module configuration

The aforementioned differences are limited exclusively to module replacements. An error – "Missing module" status – is reported if a device is missing (except when the device is defined as optional), has an incorrect hardware code or other problems are present on the module (e.g. incorrect parameters that may not be changed by the SafeLOGIC controller). This status is only indicated if a module or firmware replacement is not being indicated. This status cannot be acknowledged.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

2.6.7.2.6.3 Acknowledging a firmware modification

A change to the firmware is indicated by the FW-ACKN status and must be confirmed using the FW-ACKN action. A firmware modification must always be concluded with full functional testing.

> ## Danger!
>
> **Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions and trained in the procedure of exchanging firmware.**
>
> **Be sure to validate the entire safety function!**

> ## Danger!
>
> **Only use firmware versions listed in the FS certificates for B&R safety technology. These FS certificates are available for download from the B&R website at http://www.br-automation.com.**

2.6.7.2.6.4 Triggering a module scan

A module scan determines if all configured modules are present in the application and if they correspond to the project configuration. The module scan runs automatically but at large time intervals. To minimize the time it takes for the SafeLOGIC controller to recognize a newly replaced module, this function can also be triggered manually by the user. The result of the scan is described in the following sections:

- "Module replacement"
- "Other errors in module configuration"
- "Acknowledging a firmware modification"

The process itself is started using the SCAN function and indicated using the "Scanning" status. The results are reported after the "Scanning" status is completed (e.g. three modules replaced).

2.6.7.2.6.5 SafeKEY or safety section of the CompactFlash card

The following data is stored on the SafeKEY (X20SL8xxx series) or in the safety section of the CompactFlash card (X20SLXxxx series):

- SafeDESIGNER application (application and all SafeDESIGNER parameters for the modules)
- Configuration (unique module code (UDID), firmware versions of modules)
- Subsequently loadable data elements (machine options, tables, etc.)

Size of the SafeDESIGNER application on the SafeKEY

The size of the current application on the SafeKEY is calculated by SafeDESIGNER during compilation and displayed in the message window (e.g. "The safety application uses 0.688 MB (11 sectors) memory.").

Notes:

- The output only takes the size of the SafeDESIGNER application into account. Space on the data storage device used by firmware or subsequently loadable data (tables, machine options, etc.) is not taken into account.
- If the online project comparison is not needed (see Automation Help → SafeDESIGNER), the download size of the application can be reduced by disabling the following communication setting: Online → Communication settings → Download project source to SL.

Removing a SafeKEY (X20SL8xxx series only)

Removing a SafeKEY always results in a change to BOOT mode, and the safety application is completely shut down.

Information:

Removing a SafeKEY during operation causes the SafeLOGIC controller to be restarted and all safety-related actuators to be cut off.

Removing a SafeKEY during operation can destroy the data on the SafeKEY.

Removing a SafeKEY during operation must therefore be avoided at all cost.

The "Backing up the SafeKEY" sequence is not affected by this general rule.

Acknowledging a SafeKEY replacement

Replacing a SafeKEY or replacing a CompactFlash card with a CompactFlash that has a modified safety section is indicated by the "FW-ACKN" status and must be acknowledged with the SK-XCHG function. Complete functional testing is then required.

Information:

A SafeKEY replacement can only be acknowledged if a valid SafeDESIGNER project has already been transferred to the SafeKEY or CompactFlash card.

Danger!

Replacing a SafeKEY or CompactFlash card will enable the safety application stored on the SafeKEY or CompactFlash card. Always check the project CRC and date that the safety application project was saved on the SafeKEY or CompactFlash card.

Danger!

Be sure to validate the entire safety function!

Changing the application on the SafeLOGIC controller by replacing the SafeKEY (X20SL8xxx series only)

All relevant configuration data and all application data and parameters are stored on the SafeKEY. In order to transfer the previous configuration data to a new SafeKEY when changing the application, the following sequence must be carried out.

- Set the selector switch to the SK-COPY position.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- The SafeKEY configuration data is saved on the SafeLOGIC controller. The SKEY LED blinks with each access.
- The FW-ACKN LED will flash after the copying procedure. This SafeKEY can now be replaced by the SafeKEY with the new application. 30 seconds are provided to do this. The FW-ACKN LED blink frequency increases after 20 seconds to signal the end of the replacement phase.
- The acknowledgment button must be pressed again after the new SafeKEY has been inserted. The selector switch remains on the setting SK-COPY.
- The internal, temporarily saved configuration data is saved on the new SafeKEY. A reset is then triggered automatically, and the data from the new SafeKEY is applied.
- Following the reset, the SafeKEY replacement must be acknowledged. To do this, move the selector switch to the setting SK-XCHG.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- Perform complete functional testing.

Information:

If the new SafeKEY is not acknowledged after 30 seconds, the function will end, i.e. if the function is triggered inadvertently, the copy function ends automatically after 30 seconds. If a SafeKEY is not inserted after 30 seconds, the SafeLOGIC controller switches to BOOT mode.

Danger!

This procedure enables the safety application stored on the new SafeKEY. Always check the project CRC and date that the safety application project was saved on the SafeKEY.

Danger!

Be sure to validate the entire safety function!

Information:

This sequence can also be used to create a SafeKEY backup using a second SafeKEY with an identical safety application. After executing the sequence, two identical SafeKEYs are available (backup copy).

Information:

Only data relevant to the machine is copied, not all of the safety application data.

2.6.7.2.6.6 Replacing a SafeLOGIC controller

Replacing a SafeLOGIC controller involves the same procedures as a normal module replacement. When replacing a SafeLOGIC controller, the SafeKEY from the SafeLOGIC controller being replaced must be kept in order to avoid activating an old safety-related application.

Danger!

Be sure to validate the entire safety function!

2.6.7.2.6.7 Authorization (X20SL8xxx series only)

The following functions can be blocked by the standard CPU:

- Confirming a module replacement
- Acknowledging a firmware modification
- Acknowledging a SafeKEY replacement
- Backing up the SafeKEY
- Replacing a SafeLOGIC controller

This allows actions to be executed in accordance with an application-specific user concept. This option is not possible from a safety perspective, however, since these functions are executed on the standard CPU.

The following table lists the associated objects in Index "0x2402" that can be accessed using the POWERLINK library.

Index:Subindex	Object description	Data type	Access	Value	Description
0x2402:0x00	NumberOfEntries	USINT	R	0x22	Number of entries in this index
0x2402:0x01	EnableAuthorization	UDINT	RW	"AENA", 0x41454E41	Enables authorization
				"ADIS", 0x41444953	Disables authorization
0x2402:0x04	EnableModuleExchange	UDINT	RW	"UDID", 0x55444944	Provides authorization to acknowledge a module replacement
				All other values	Does not provide authorization to acknowledge a module replacement
0x2402:0x05	EnableFWMismatch	UDINT	RW	"FWAC", 0x46574143	Provides authorization to acknowledge a firmware replacement
				All other values	Does not provide authorization to acknowledge a firmware replacement
0x2402:0x06	EnableSKeyExchange	UDINT	RW	"SKEY", 0x534B4559	Provides authorization to acknowledge a SafeKEY replacement
				All other values	Does not provide authorization to acknowledge a SafeKEY replacement

User requests made to the SafeLOGIC controller that are not authorized by the CPU are indicated by a steadily lit ENTER LED.

2.6.7.2.7 Software functions

2.6.7.2.7.1 Operation via the AsSafety library

Information about using library "AsSafety" is available under Programming -> Libraries -> Safety -> AsSafety in Automation Help.

2.6.7.2.7.2 Automatic acknowledgment

As specified in previous chapters, automatic acknowledgment is usually not permitted. Provided that the user implements appropriate quality assurance measures and/or constraints, it is nevertheless possible to deviate from this to permit the following automatic acknowledgment.

> **Danger!**
>
> **The automatic acknowledgment of SafeLOGIC controller acknowledgment requests under improper circumstances is not permitted and can lead to dangerous states.**
>
> **It is the sole responsibility of the user to assess the requirements of the safety application in order to determine whether additional measures are necessary.**

"SafeKEY exchange" acknowledgment request

The SafeDESIGNER application and machine option are saved in the safety section of the CompactFlash card (X20SLXxxx series) or on the SafeKEY (X20SL8xxx series). Replacing the CompactFlash card or SafeKEY may result in the unintended exchange of this data. The "SafeKEY exchange" acknowledgment request is meant to prevent this unintentional exchange of data.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards or SafeKEYs:

- The SafeDESIGNER application must be completely validated on a reference machine.
- The machine options file must be completely validated on a reference machine.
- Sufficient measures must be implemented to prevent the SafeDESIGNER application or machine options file from being mixed up across different machine types.
- No test versions of the SafeDESIGNER application or machine options file are permitted.

Under the conditions specified, an automated update of the SafeDESIGNER application or machine options file is permitted to be implemented on the SafeLOGIC/SafeLOGIC-X controller.

"Firmware acknowledge" acknowledgment request

B&R Automation Runtime sees to it independently that the firmware versions stored on the CompactFlash card are transferred to the automation components in the network. This mechanism may cause other firmware versions to be enabled in the system than those that were active when the SafeDESIGNER application was validated. A change to the firmware of the safety modules always requires revalidation of the SafeDESIGNER application. The "Firmware acknowledge" acknowledgment request is meant to prevent an unintentional exchange of firmware versions.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards:

- The firmware files installed on the safety modules must be completely validated together with the SafeDESIGNER application on a reference machine.

"UDID mismatch" acknowledgment request

The "UDID mismatch" request occurs in the following situations:

- When modules are exchanged by the user (e.g. during a service call). In this case, it is possible for the connection lines to be mixed up.
- When errors occur in the standard application that lead to a mix-up of modules.

To rule out these mix-ups, a wiring test must be performed after a "UDID mismatch" request is acknowledged.

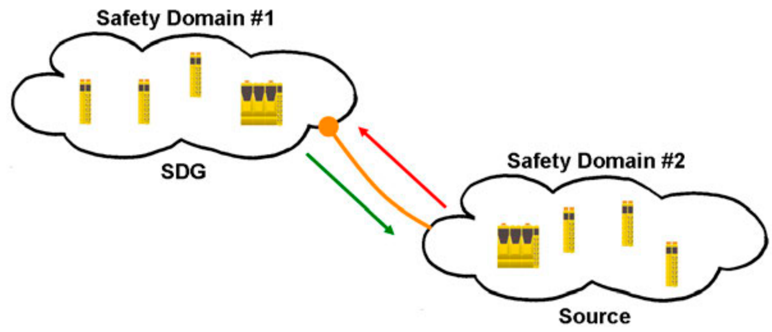
The "UDID mismatch" acknowledgment request is meant to prevent the unintentional mix-up of signals caused by exchanging a module or errors in the standard application.

- Service personnel are to be informed that the mandatory wiring test when exchanging modules must be performed independently of the automatic acknowledgment of the "UDID mismatch" request.
- It is not permitted to use more than 1 module per module type in the Automation Studio application or SafeDESIGNER application.

If the last requirement cannot be met, a "UDID mismatch" acknowledgment request is not permitted to be acknowledged automatically since it would not cover the possible mix-up of signals caused by errors in the standard application.

2.6.7.2.7.3 SafeLOGIC to SafeLOGIC communication

The safety system makes it possible to exchange safety-related information between two safety controllers (SafeLOGIC). SafeLOGIC to SafeLOGIC communication can be used to implement functions such as a global E-stop across a machine network or if a dependency exists between the safety applications on two or more machines. This makes it possible to establish a central collection point for safety information that will be responsible for distributing current values to all relevant locations.



Information:

The safety domain number is taken from the SafeLOGIC ID. In order to use SafeLOGIC to SafeLOGIC communication, the SafeLOGIC IDs must be unique. This uniqueness should be taken into consideration from the very beginning.

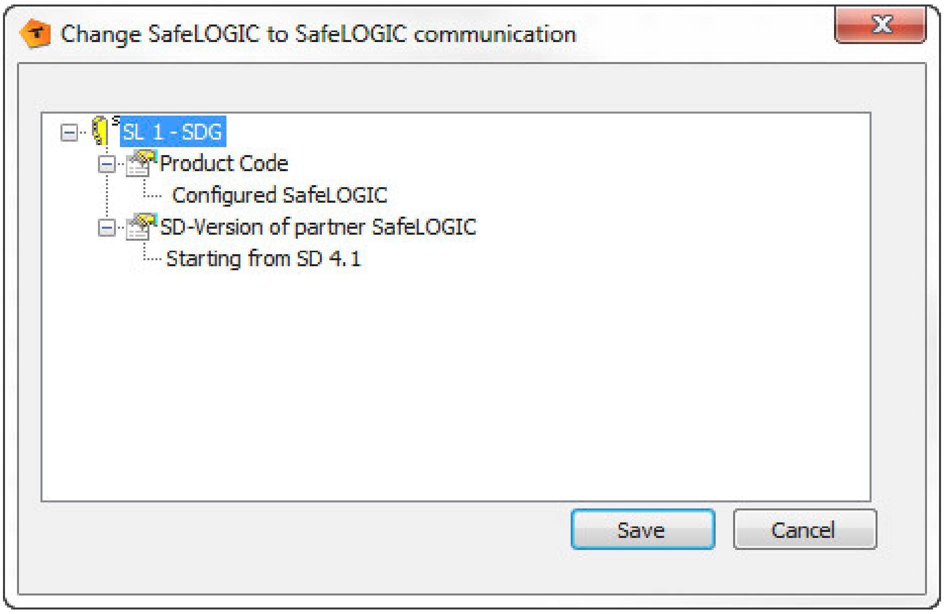
To help with this, a SafeLOGIC controller provides a Safety Domain Gateway (SDG) that can be used to connect additional SafeLOGIC controllers (source controllers). This gateway functionality ensures communication between several safety domains. The connection between source SafeLOGIC controllers and SDG SafeLOGIC controller is indicated in the source SafeLOGIC controller's project as an additional safety module that provides additional communication channels. An SDG SL controller itself can also be used as a source controller and connected to another SDG SL controller. This can be done to achieve cascading communication relationships.

A source SL controller can also be connected several times to the same SDG SL controller, just as it is possible for the source SL controller to communicate with several SDG SL controllers. This results in several ways for SafeLOGIC to SafeLOGIC communication to take place.

System requirements

The following points must be taken into account for safe data exchange between at least 2 SafeLOGIC controllers:

- SafeDESIGNER <4.1: The same SafeDESIGNER versions must be used.
- SafeDESIGNER 4.1 to 4.2.1: The SafeDESIGNER versions must be within this version range.
- SafeDESIGNER 4.2.2 and later: SafeDESIGNER 3.0 or later is permitted to be used.
  The corresponding parameters in the following dialog box must be configured in order to establish a connection to the remote station.



  - Configured SafeLOGIC: Remote station with which communication takes place (e.g. X20SL8100)
  - SD-Version of partner SafeLOGIC: Version with which the application on the remote station was created

Possibilities

The system supports various communication options. The corresponding communication type is defined via parameters in Automation Studio (see "Group: SafeLOGIC to SafeLOGIC communication").

Fixed communication

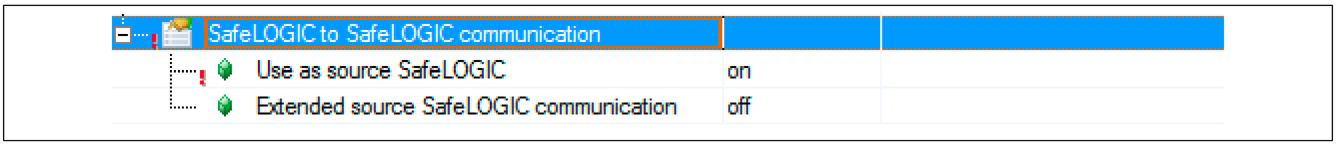
- 8 BOOL channels (1 byte) per communication direction
- One source SL controller can only communicate with one SDG SL controller
- No "any to any" constellation
- Cannot be used with SafeLOGIC-X

Extended communication (Release 1.4 or later and Automation Studio 3.0.90 or later)

- Freely configurable communication channels
- Limited to 16 channels (where 8 BOOLs count as 1 channel; other data types are calculated 1:1).
- One source SL controller can communicate with several SDG SL controllers
- "Any to any" constellation possible

Configuration in Automation Studio

To use SafeLOGIC to SafeLOGIC communication, a SafeLOGIC controller first needs to be configured as a source SL controller. This is done in the I/O configuration.

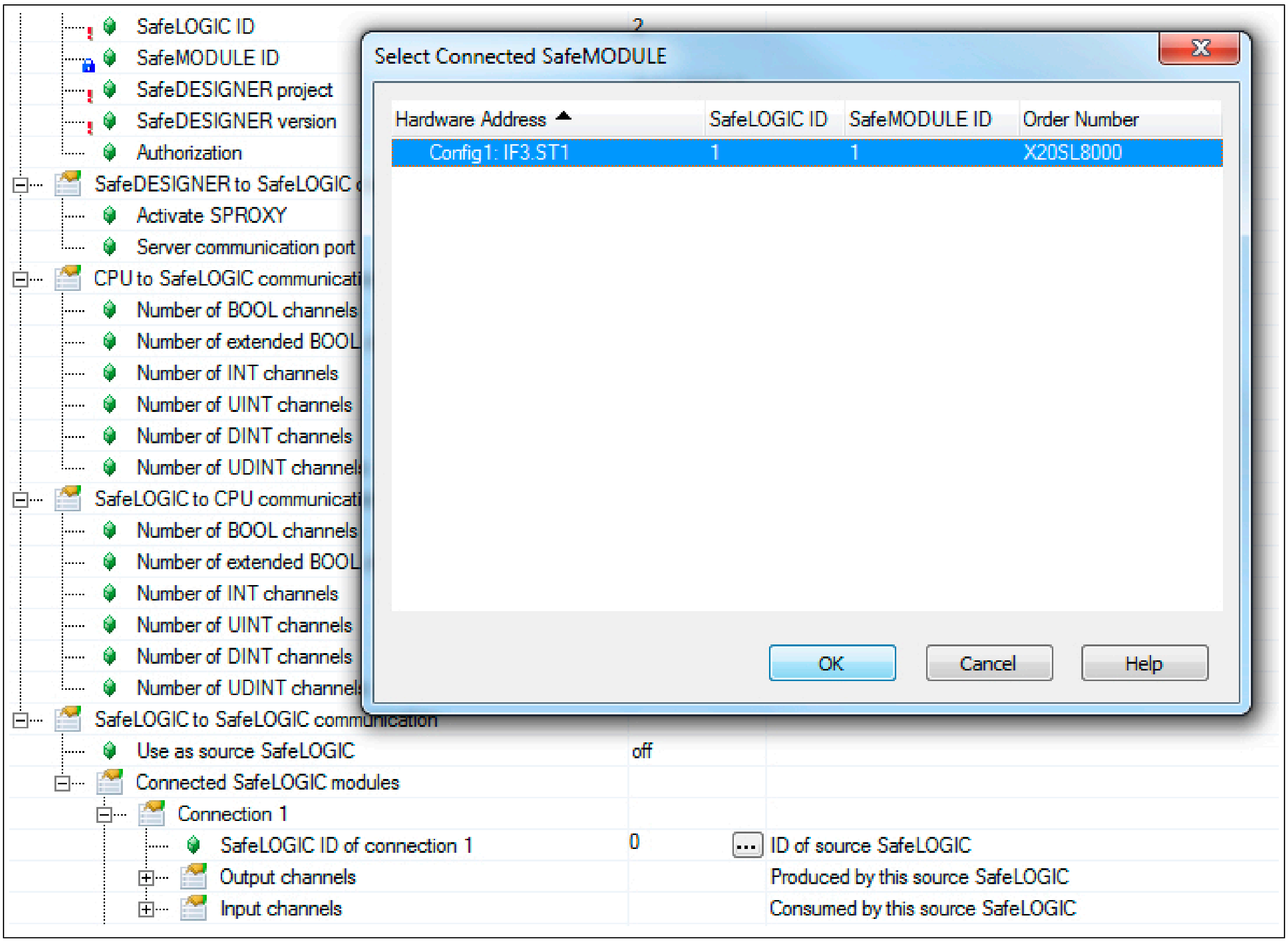


After the "Use as source SafeLOGIC" parameter has been enabled, it is possible to define the type of SafeLOGIC to SafeLOGIC communication as fixed or extended. If the "Extended source SafeLOGIC communication" parameter is not enabled, then fixed communication is used.

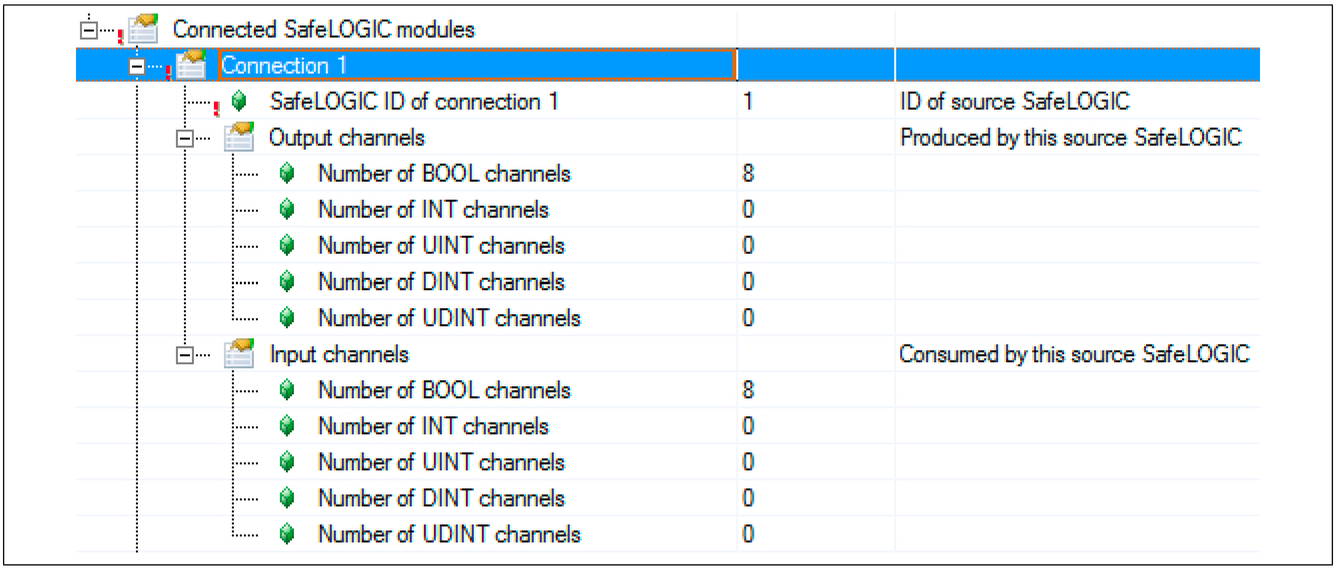
Information:

Changing the type of communication (fixed or extended) at a later time may result in channel overlaps in SafeDESIGNER; the communication channels must therefore be reconnected.

The source SL controller is then connected to the SDG SL controller in the next step. This is done using the connection points in Automation Studio under the I/O configuration of a SafeLOGIC controller (X20SL80x1 and X20SL81xx). Each SafeLOGIC ID (safety domain) is specified from the connection sections using the wizard in Automation Studio.



The necessary communication channels must be defined under each connection. With fixed communication, they are limited to 8 BOOL channels in each direction.



If SafeLOGIC to SafeLOGIC communication should be established between existing or separate Automation Studio projects, several things must be taken into consideration:

- SafeLOGIC IDs must be unique.
- A dummy configuration that includes all safety components must be created on the peer station.
- The dummy configuration must match the real configuration - the SafeMODULE IDs are important here.
- If the projects have multiple iCNs (intelligent controlled nodes), all iCNs must always be taken into account in the iCN project.

Display in SafeDESIGNER

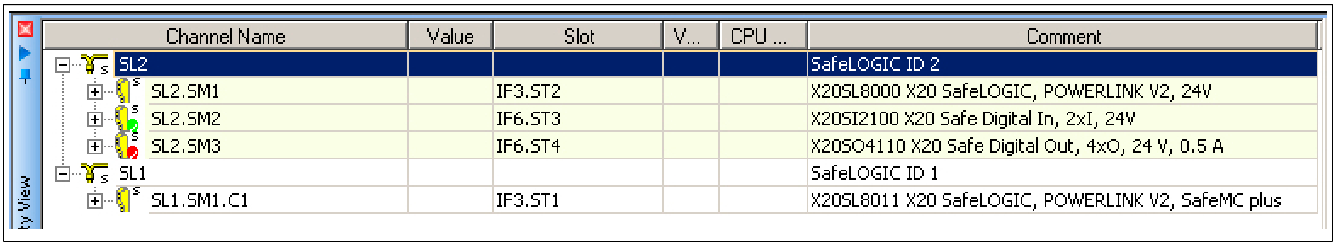
The communication channels are also shown in the SafeDESIGNER project for the respective SafeLOGIC controller (source or SDG).

Danger!

All of the communication channels being used in the project must be mapped in both SafeDESIGNER projects using the same variable names. Channels and variable names are used to calculate a checksum that is then checked at runtime. If the checksum does not match, then the system issues a corresponding logger message in the Safety Logger and communication does not take place.

SafeDESIGNER project – Source SL controller

In the source SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.

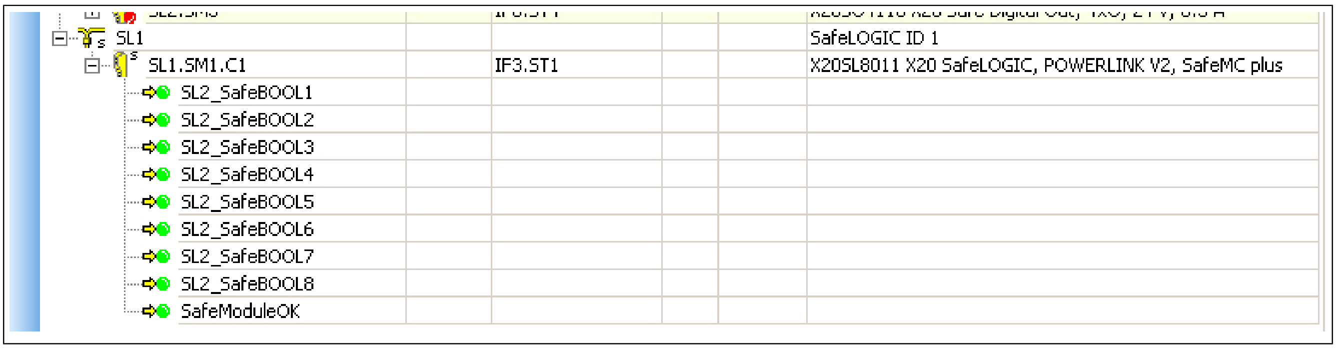


Channel Name	Value	Slot	V...	CPU ...	Comment
SL2					SafeLOGIC ID 2
SL2.SM1		IF3.ST2			X20SL8000 X20 SafeLOGIC, POWERLINK V2, 24V
SL2.SM2		IF6.ST3			X20SI2100 X20 Safe Digital In, 2xI, 24V
SL2.SM3		IF6.ST4			X20SO4110 X20 Safe Digital Out, 4xO, 24 V, 0.5 A
SL1					SafeLOGIC ID 1
SL1.SM1.C1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus

If this module is selected, it is possible to configure its safety-related parameters (see section "Parameters for connection - Release 1.10 and later").

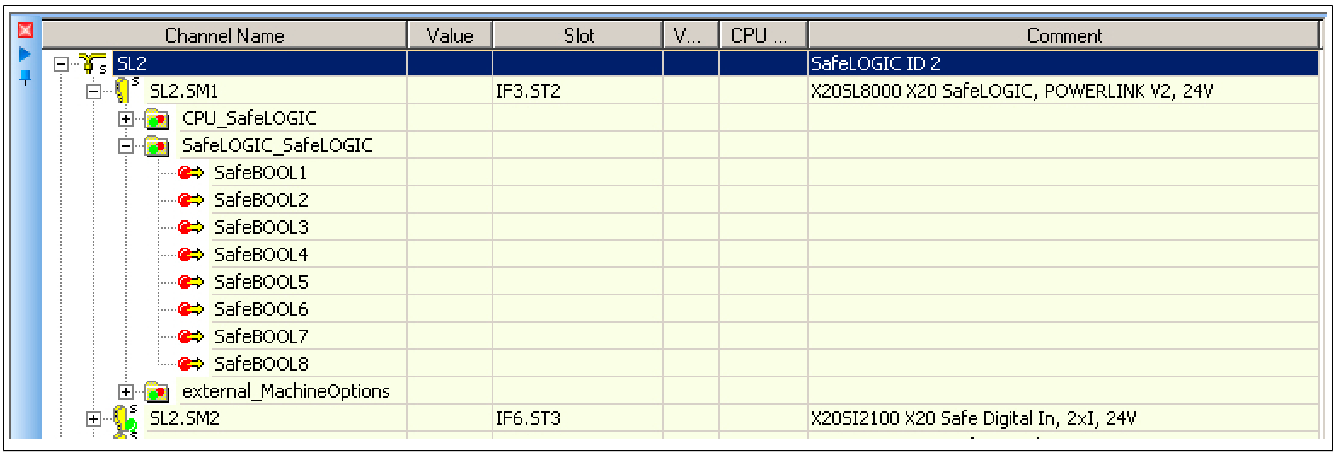
Fixed communication

The input channels sent from the SDG SL controller to the source SL controller and bit information about the status of the connection are listed under the module.



SL1					SafeLOGIC ID 1
SL1.SM1.C1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
SL2_SafeBOOL1					
SL2_SafeBOOL2					
SL2_SafeBOOL3					
SL2_SafeBOOL4					
SL2_SafeBOOL5					
SL2_SafeBOOL6					
SL2_SafeBOOL7					
SL2_SafeBOOL8					
SafeModuleOK					

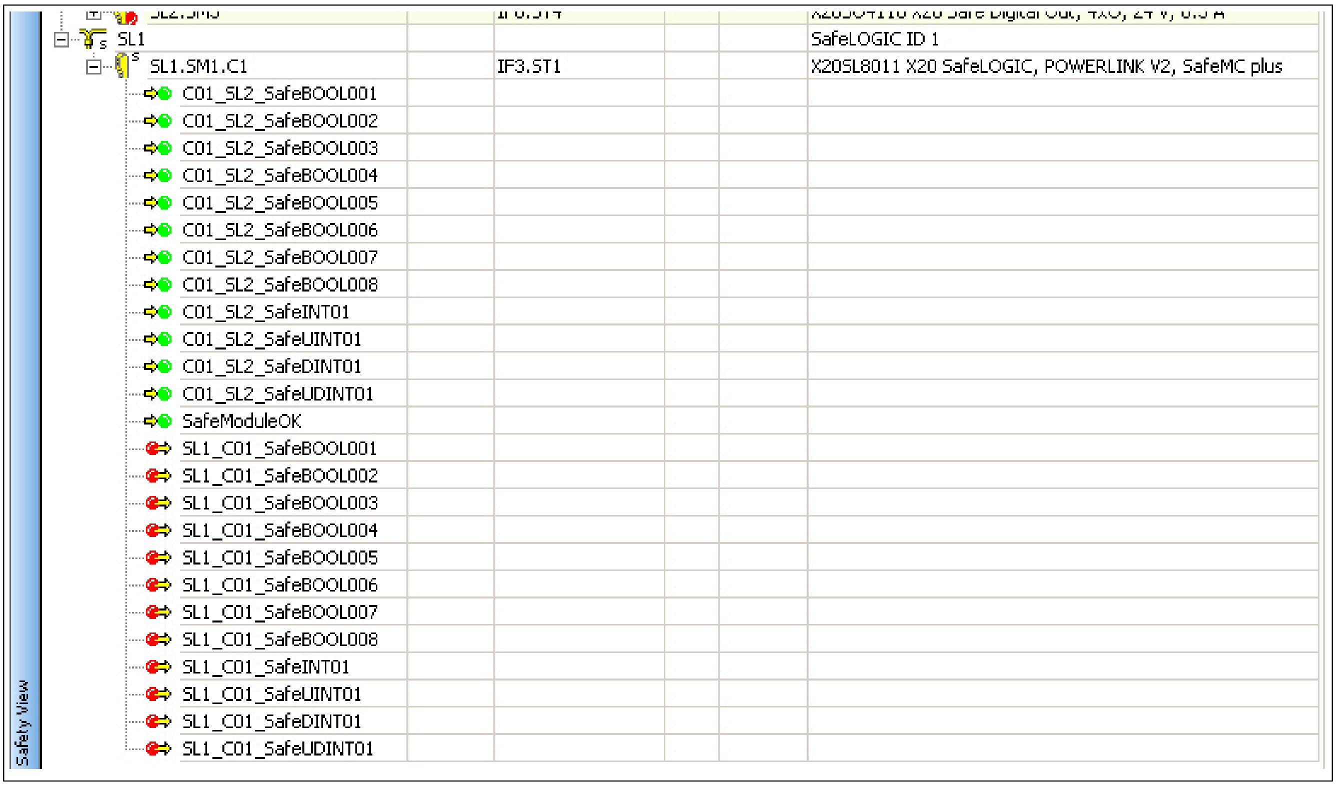
The output channels sent from the source SL controller to the SDG SL controller are listed under the actual SL controller in the project in section "SafeLOGIC_SafeLOGIC".



Channel Name	Value	Slot	V...	CPU ...	Comment
SL2					SafeLOGIC ID 2
SL2.SM1		IF3.ST2			X20SL8000 X20 SafeLOGIC, POWERLINK V2, 24V
CPU_SafeLOGIC					
SafeLOGIC_SafeLOGIC					
SafeBOOL1					
SafeBOOL2					
SafeBOOL3					
SafeBOOL4					
SafeBOOL5					
SafeBOOL6					
SafeBOOL7					
SafeBOOL8					
external_MachineOptions					
SL2.SM2		IF6.ST3			X20SI2100 X20 Safe Digital In, 2xI, 24V

Extended communication

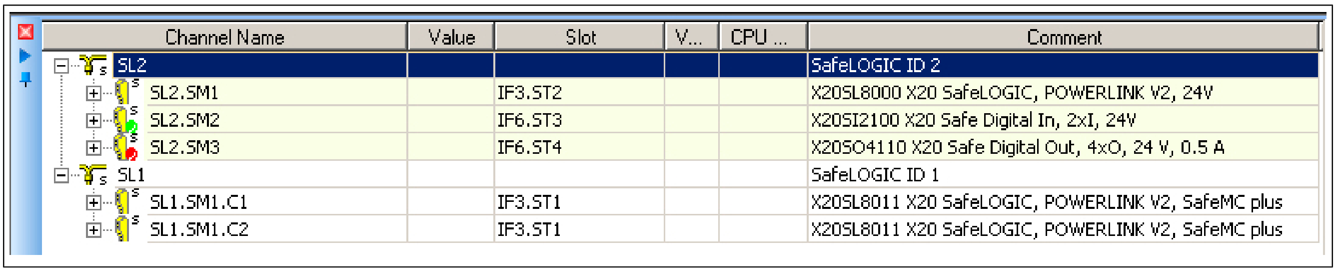
The input channels, output channels and bit information regarding the status of the connection are listed under the module.



SL2.SM3		IF6.ST4			X20SO4110 X20 Safe Digital Out, 4xO, 24 V, 0.5 A
SL1					SafeLOGIC ID 1
SL1.SM1.C1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
C01_SL2_SafeBOOL001					
C01_SL2_SafeBOOL002					
C01_SL2_SafeBOOL003					
C01_SL2_SafeBOOL004					
C01_SL2_SafeBOOL005					
C01_SL2_SafeBOOL006					
C01_SL2_SafeBOOL007					
C01_SL2_SafeBOOL008					
C01_SL2_SafeINT01					
C01_SL2_SafeUINT01					
C01_SL2_SafeDINT01					
C01_SL2_SafeUDINT01					
SafeModuleOK					
SL1_C01_SafeBOOL001					
SL1_C01_SafeBOOL002					
SL1_C01_SafeBOOL003					
SL1_C01_SafeBOOL004					
SL1_C01_SafeBOOL005					
SL1_C01_SafeBOOL006					
SL1_C01_SafeBOOL007					
SL1_C01_SafeBOOL008					
SL1_C01_SafeINT01					
SL1_C01_SafeUINT01					
SL1_C01_SafeDINT01					
SL1_C01_SafeUDINT01					

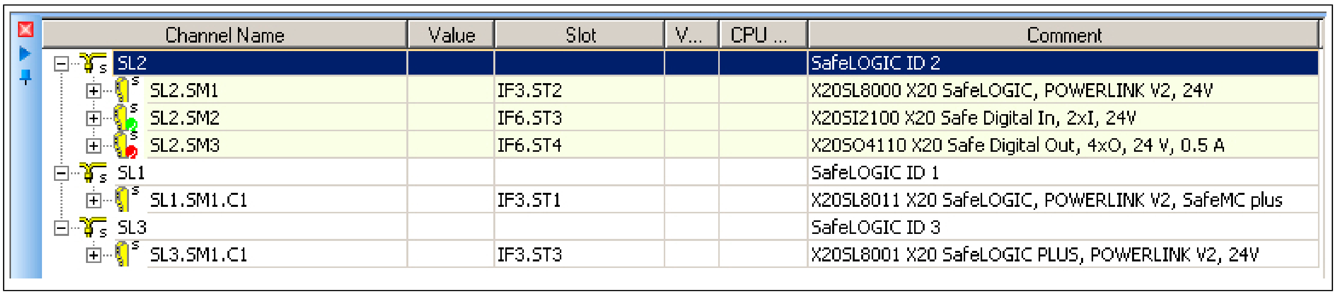
Additional connection

If the source SL controller should be connected once again to the same SDG SL controller, an additional module underneath the same node is available with the necessary parameters and communication channels.



Channel Name	Value	Slot	V...	CPU ...	Comment
SL2					SafeLOGIC ID 2
SL2.SM1		IF3.ST2			X20SL8000 X20 SafeLOGIC, POWERLINK V2, 24V
SL2.SM2		IF6.ST3			X20SI2100 X20 Safe Digital In, 2xI, 24V
SL2.SM3		IF6.ST4			X20SO4110 X20 Safe Digital Out, 4xO, 24 V, 0.5 A
SL1					SafeLOGIC ID 1
SL1.SM1.C1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
SL1.SM1.C2		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus

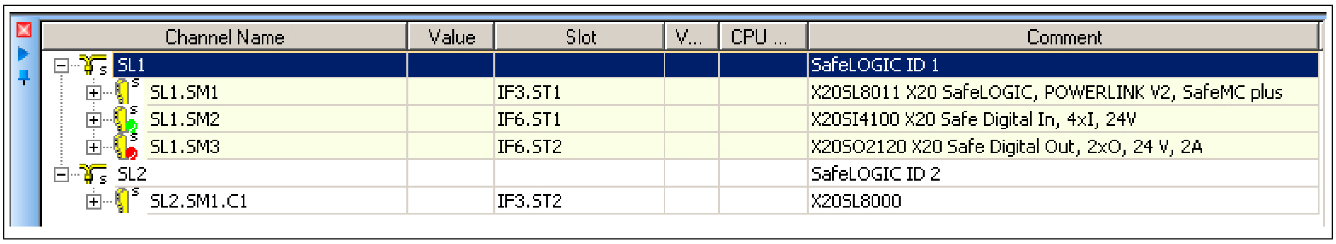
If the source SL controller should be connected to another SDG SL controller, an additional node for the safety domain as well as a module with the necessary parameters and communication channels is available.



Channel Name	Value	Slot	V...	CPU ...	Comment
SL2					SafeLOGIC ID 2
SL2.SM1		IF3.ST2			X20SL8000 X20 SafeLOGIC, POWERLINK V2, 24V
SL2.SM2		IF6.ST3			X20SI2100 X20 Safe Digital In, 2xI, 24V
SL2.SM3		IF6.ST4			X20SO4110 X20 Safe Digital Out, 4xO, 24 V, 0.5 A
SL1					SafeLOGIC ID 1
SL1.SM1.C1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
SL3					SafeLOGIC ID 3
SL3.SM1.C1		IF3.ST3			X20SL8001 X20 SafeLOGIC PLUS, POWERLINK V2, 24V

SafeDESIGNER project – SDG SL controller

In the SDG SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.



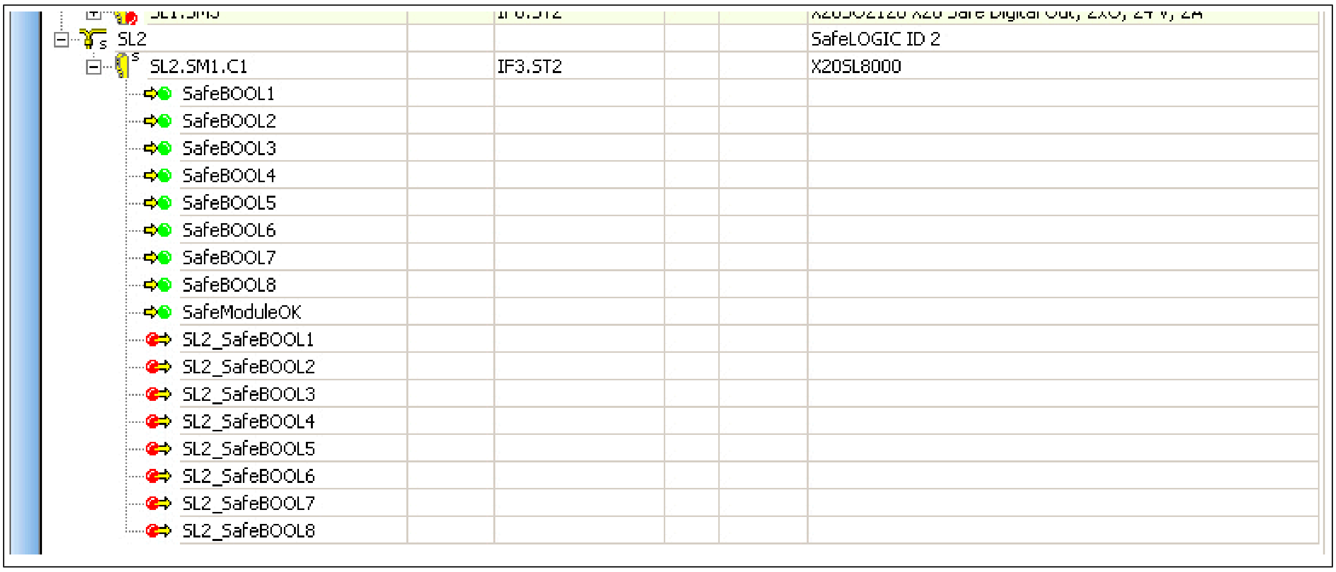
Channel Name	Value	Slot	V...	CPU ...	Comment
SL1					SafeLOGIC ID 1
SL1.SM1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
SL1.SM2		IF6.ST1			X20SI4100 X20 Safe Digital In, 4xI, 24V
SL1.SM3		IF6.ST2			X20SO2120 X20 Safe Digital Out, 2xO, 24 V, 2A
SL2					SafeLOGIC ID 2
SL2.SM1.C1		IF3.ST2			X20SL8000

Information:

No connection parameters are available in the SDG SL controller's project. They must be configured in the source SL controller's project.

Fixed communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.



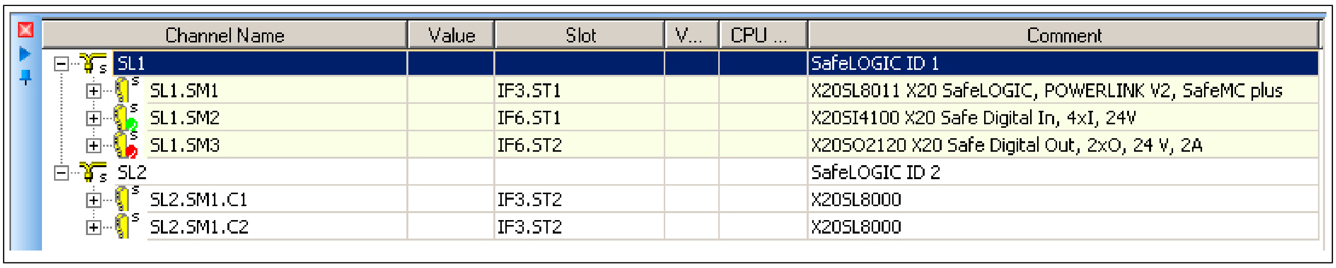
SL2					SafeLOGIC ID 2
SL2.SM1.C1		IF3.ST2			X20SL8000
SafeBOOL1					
SafeBOOL2					
SafeBOOL3					
SafeBOOL4					
SafeBOOL5					
SafeBOOL6					
SafeBOOL7					
SafeBOOL8					
SafeModuleOK					
SL2_SafeBOOL1					
SL2_SafeBOOL2					
SL2_SafeBOOL3					
SL2_SafeBOOL4					
SL2_SafeBOOL5					
SL2_SafeBOOL6					
SL2_SafeBOOL7					
SL2_SafeBOOL8					

Extended communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

Additional connection

If the source SL controller should be connected once again to the SDG SL controller, an additional module underneath the same node is available with the necessary communication channels.



Parameters for connection - up to Release 1.9

Safety Release 1.4 or higher:

Cycle time parameters are also available for communication in order to define the "Worst_Case_Response_Time_us". As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Parameter	Value
Basic	
Min_required_FW_Rev	Basic Release
Optional	No
External_UDID	No
Safety_Response_Time	
Synchronous_Network_Only	Yes
Max_SDG_Powerlink_CycleTime_us	5000
Max_Powerlink_CycleTime_us	5000
Max_CPU_CrossLinkTask_CycleTime_us	5000
Min_SDG_Powerlink_CycleTime_us	200
Min_Powerlink_CycleTime_us	200
Min_CPU_CrossLinkTask_CycleTime_us	0
Worst_Case_Response_Time_us	100000
Max_SDG_Cycle_Time_us	5000
Min_SDG_Cycle_Time_us	1600
Slow_Connection	No

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 52: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks.</td></tr></table></td></tr>
<tr><td>Max_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the maximum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
• Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the minimum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
• Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 12,500,000 µs (corresponds to 3 ms to 12.5 s)
Note:
Keep parameter "Slow_Connection" in mind when entering large values here!</td><td>100000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
<tr><td>Max_SDG_Cycle_Time_us</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Cycle_Time_us</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
<tr><td rowspan="2">Slow_Connection</td><td>This parameter specifies whether this connection is a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr><tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr></table></td></tr>
</table>

Table 53: SafeDESIGNER parameters: Safety_Response_Time

Information:

Parameter "CPU_CrossLinkTask_CycleTime_us" is needed if the source SL and SDG SL controllers are in different networks or located on different controllers. If this is not the case, the minimum and maximum value must be set to "0".
For this parameter, the entire connection distance between the controllers must be taken into account – including copy times between the interfaces involved.

Information:

Parameter "Slow_Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow_Connection = Yes").

Parameters for connection - Release 1.10 and later

Cycle time parameters are also available for communication in order to define the maximum data transmission time. As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Materialnummer: **X20SL8100**
Description: **X20 SafeLOGIC, POWERLINK V2, 24V, univ.**
SafeMODULE ID: **3**
Import file: -

Parameter	Value	Unit
Basic		
Min required FW Rev	Basic Release	
Optional	No	
External UDID	No	
Safety Response Time		
Synchronous Network Only	Yes	
Safe Data Duration	20000	us
Additional Tolerated Packed Loss	0	packets
Slow Connection	No	
Node Guarding Lifetime	5	iterations
Max SDG Cycle Time	5000	us
Min SDG Cycle Time	1600	us

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 54: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td rowspan="2">Slow Connection</td><td>This parameter specifies whether this connection is classified as a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr>
<tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr>
</table></td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
<tr><td>Max SDG Cycletime</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min SDG Cycletime</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
</table>

Table 55: SafeDESIGNER parameters: Safety Response Time

Information:

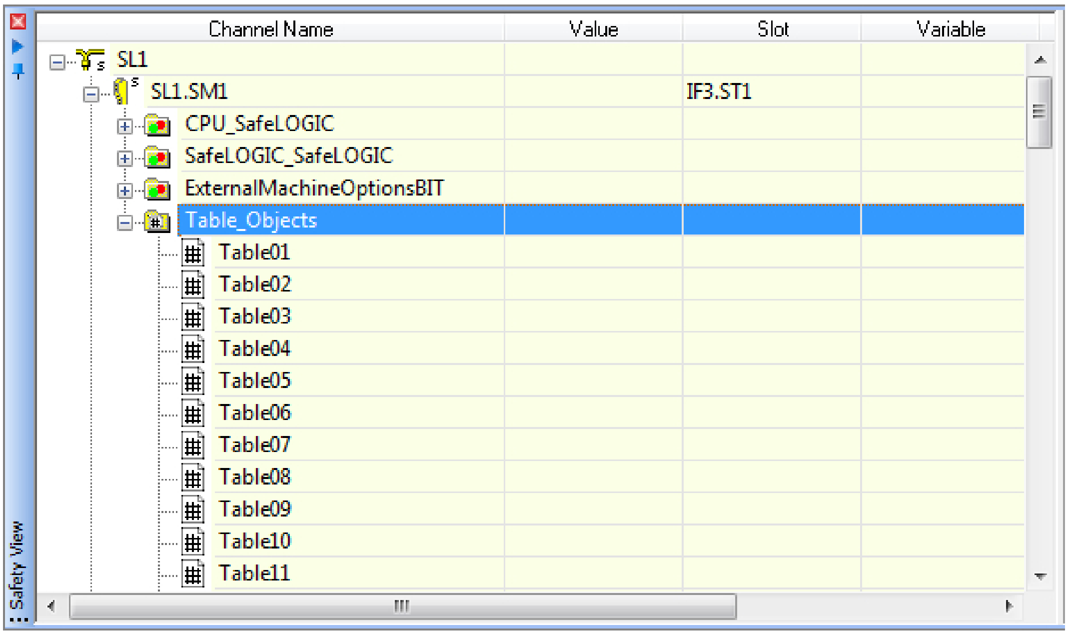
Parameter "Slow Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow Connection = Yes").

2.6.7.2.7.4 Table objects

A table object is a CSV file with a certain structure and certain data. Up to 99 table objects are available in SafeDESIGNER under the SafeLOGIC controller. Each object represents the connection to a CSV file with the corresponding data. In addition, SafeDESIGNER contains library "Table_SF" for evaluating the various table objects. The function blocks of this library must be linked to a table object.

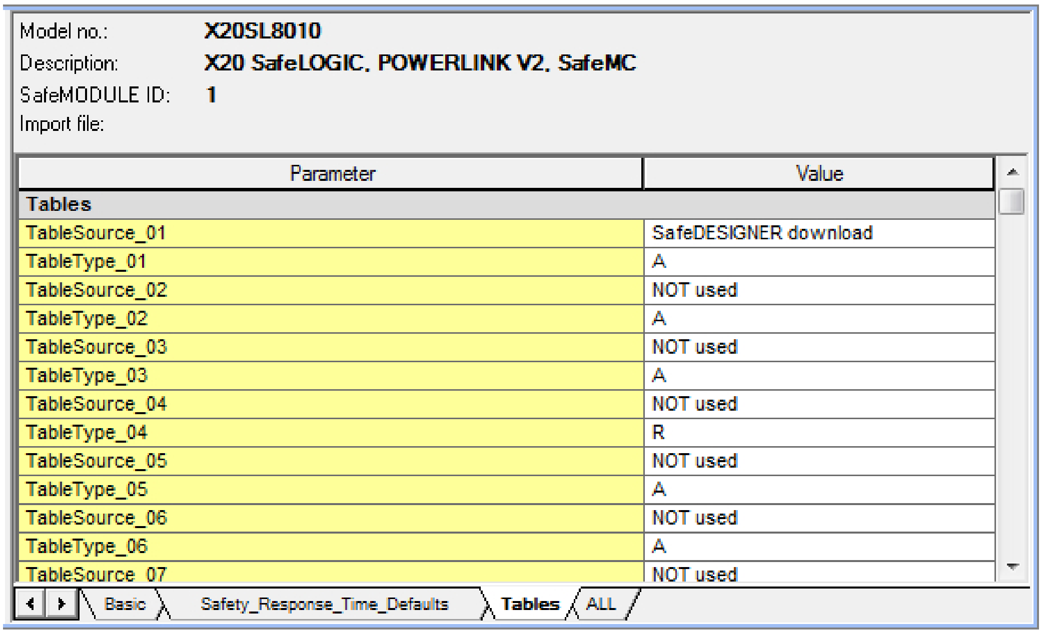
Information:

The validation and lock functions implemented in SafeDESIGNER, together with the validation of the table data by the user, allow the use of commercial off-the-shelf (COTS) editors for table data.



The necessary settings for these table objects can be controlled using SafeLOGIC controller parameters. There is a tab called "Tables" available for this. The following settings can then be made for each table object:

- TableSource → Where the table data is coming from
  - NOT used → Table object not used
  - SafeDESIGNER download → Data transferred with the application
  - Remote download → Data not transferred with the application. It must be transferred subsequently using the AsSafety library.
- TableType → The type of table
  - A - Q
  - R - Z → Table types for SafeROBOTIC



Information:

For details about the structure of the table objects or data, see the help documentation of the function block to be used.

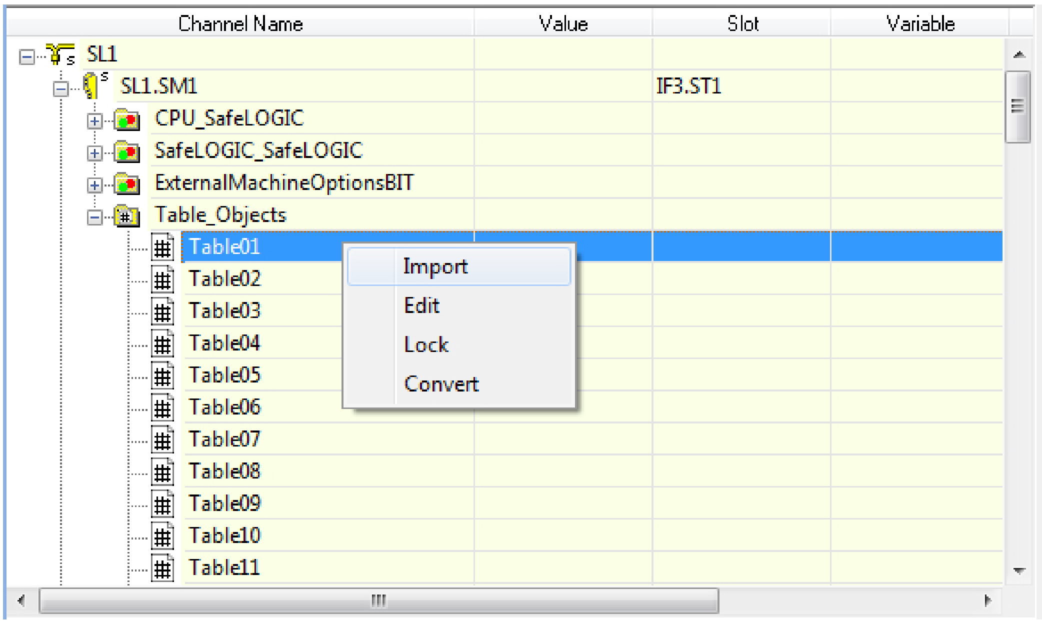
Procedure

To start, each table object must be assigned the proper type and source.

Information:

If a table object is being used in the application but parameter "TableSource" is set to "NOT used", then an error message will be generated when the project is compiled.

Several different actions can be carried out from the table object's shortcut menu (right-click on the table object).



Import

This menu item can be used to import an existing CSV file with corresponding data suitable to the selected table type.

Information:

If a file that does not match the table type is imported, then an error message will be generated when the project is compiled.

Edit

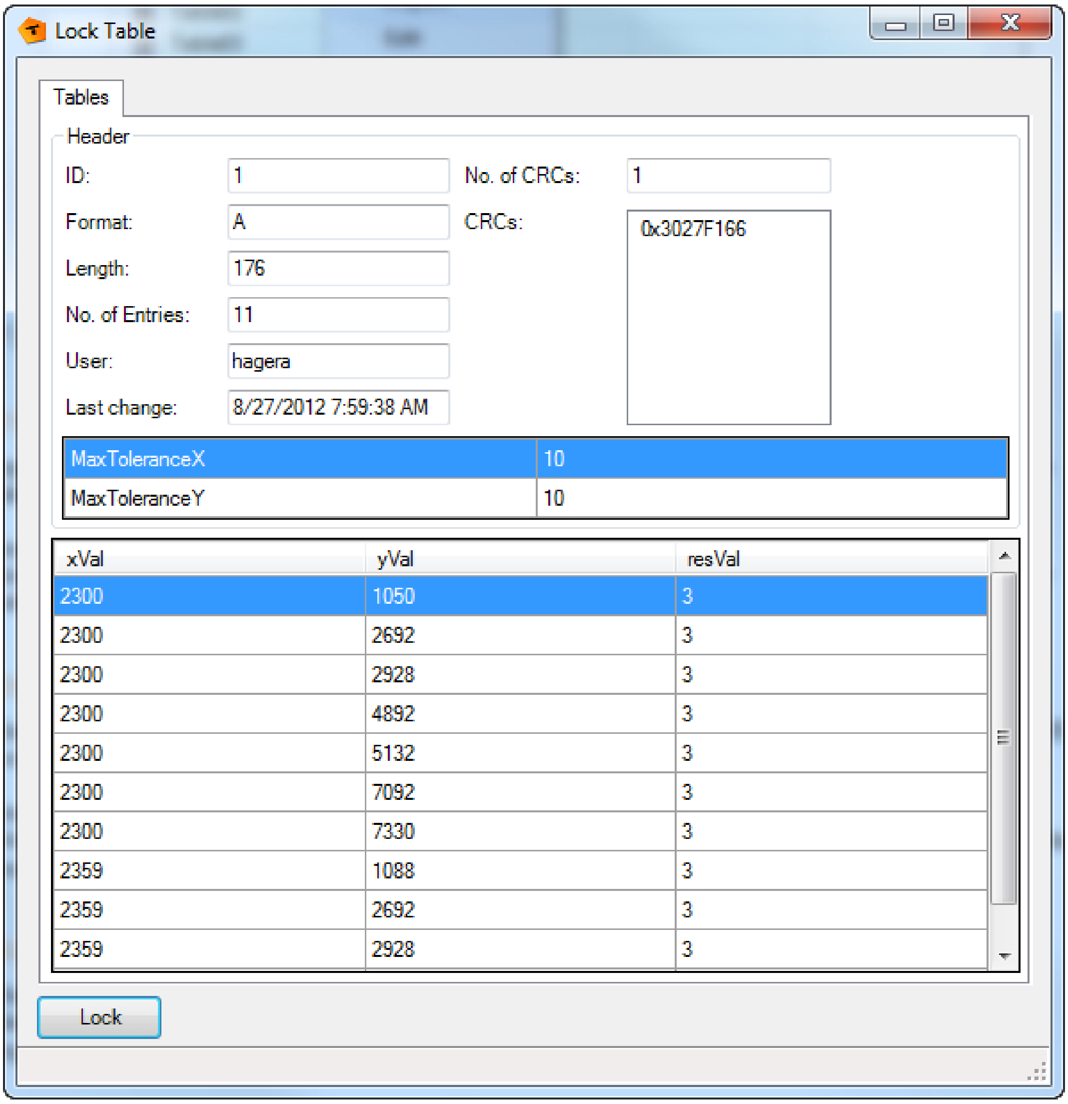
This menu item allows the file to be edited using the default program for CSV files (e.g. MS Excel).

Information:

If a file is being edited, it is required that it is locked again ("Lock" action). Otherwise, the file's CRC will be invalid.

Lock

This menu item locks the file and calculates a CRC for its current content. The data is also displayed once more in a new window according to the specified table type.

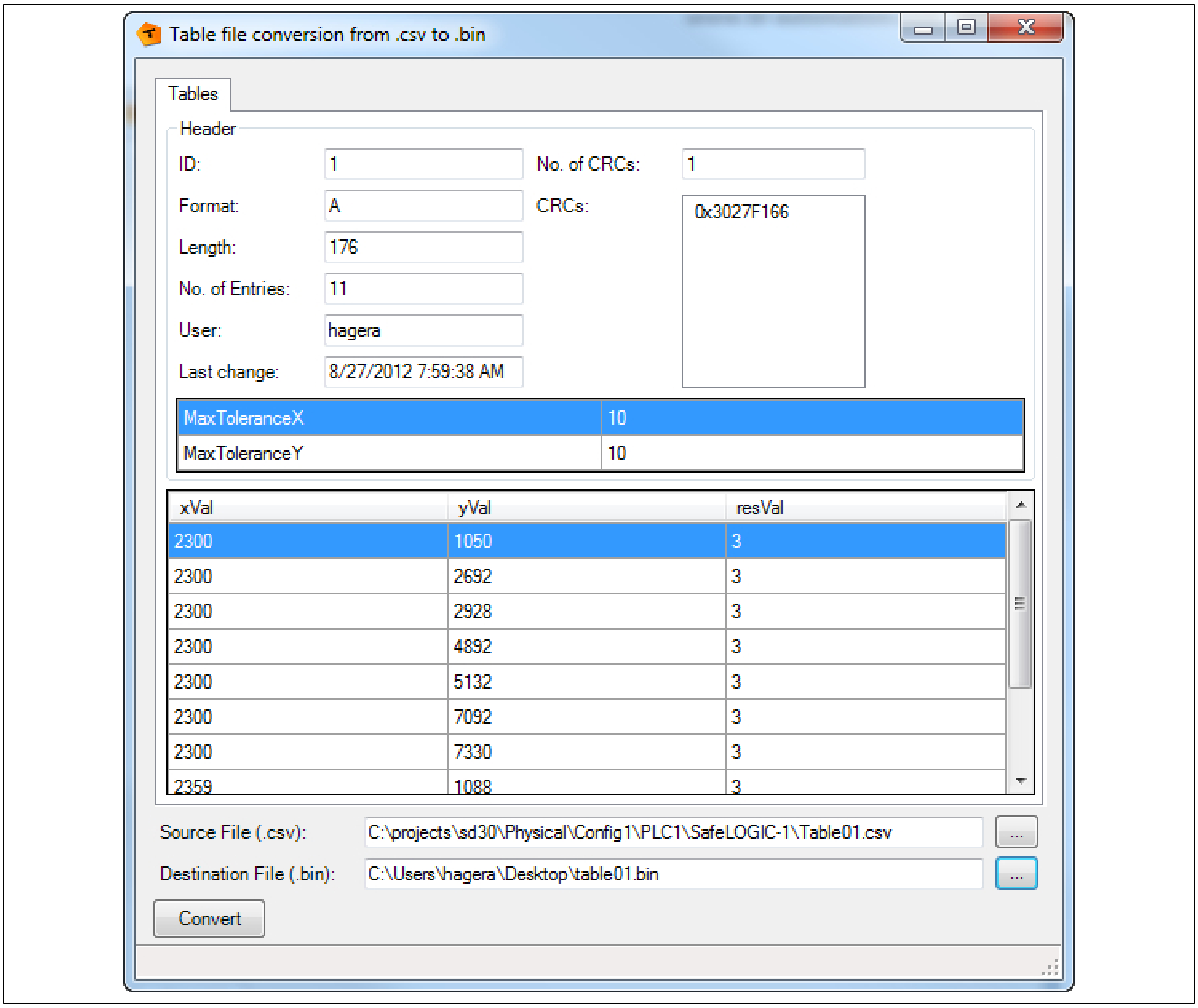


Information:

Error messages will also be displayed in this window if there are any problems with the file (e.g. invalid format, cannot open file, etc.).

Convert

This menu item can be used to convert the file to binary format for the SafeLOGIC controller. The path where the binary file is to be saved must be specified.



Information:

This binary file can then be used for downloading via the standard CPU.

Usage in the application

To use table objects, an associated function block must first be used in the application (see library "Table_SF").

Input "S_TableID" must be linked to a table object. This is done in the safety view by selecting the table object and dragging it into the application. It is also possible to provide a meaningful name for the connection.

Information:

An error message will be output during compilation if there are any problems or errors.

2.6.7.2.7.5 Blackout mode

Blackout mode allows users to continue execution of the application in lower-level subsystems if components of the B&R system fail. In this way, the B&R system – independently of redundancy technology – makes it possible to respond to system-critical situations based on the specific application.

The use of blackout-capable modules is recommended for the following requirements:

- Exit routines on system failure, e.g. to enable the opening of a press if the system fails.
- Stopping or controlled setting of an output on system failure, e.g. to automatically close inflow valves.
- Deceleration sequences on system failure, e.g. to reduce motor speeds before transmitting a stop command.

If blackout-capable modules are configured accordingly, blackout mode will be carried out if the network connection to the higher-level controller or CPU is interrupted.

As soon as the network disturbance has been corrected, blackout mode is stopped by the modules and bumpless synchronization with the network takes place.

Requirements for operation

The following requirements must be met in order to use blackout mode:

- The module being used must support blackout mode.
- Parameter "Blackout mode" must be enabled in Automation Studio.

Areas of use

Through the use of blackout-capable modules, a part of the control system can also remain functional if a disturbance in the network or X2X Link connection between the modules occurs.

Loss of POWERLINK connection

Initial situation

Several stations in an application are connected to the CPU via network cables. A fault occurs that interrupts data transfer between the CPU and stations.

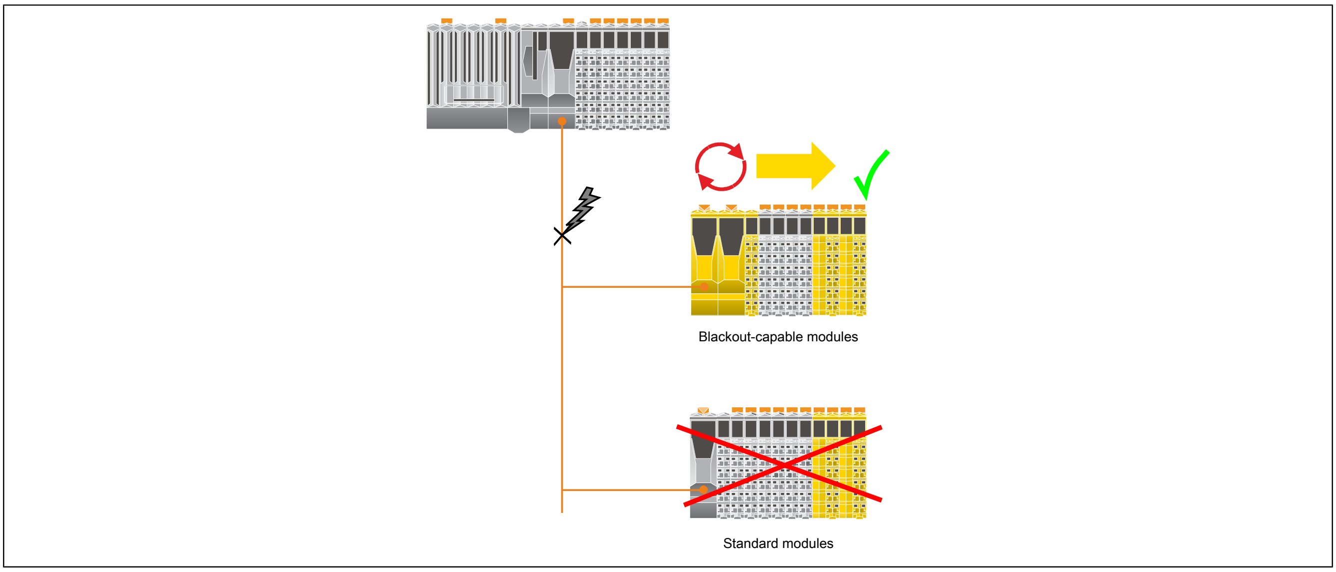
Effect

Non-blackout modules are reset and operated according to their default characteristics.

Blackout-capable modules show the following behavior:

- The programmed function continues to be executed.
- Subordinate networks continue to work.
- Data from the CPU is initialized with "0".
- After the disturbance has been corrected, the module bumplessly returns to the higher-level network.

> **Warning!**
>
> **Blackout mode causes data from the CPU to be initialized with "0". If blackout mode is used in combination with "output inversion", this can lead to the unwanted setting of outputs.**



Loss of X2X Link connection

Initial situation

Modules in an application are connected to the network via X2X Link cables. A defect in the X2X Link cable causes the data transfer between the CPU and modules to be interrupted.

Effect

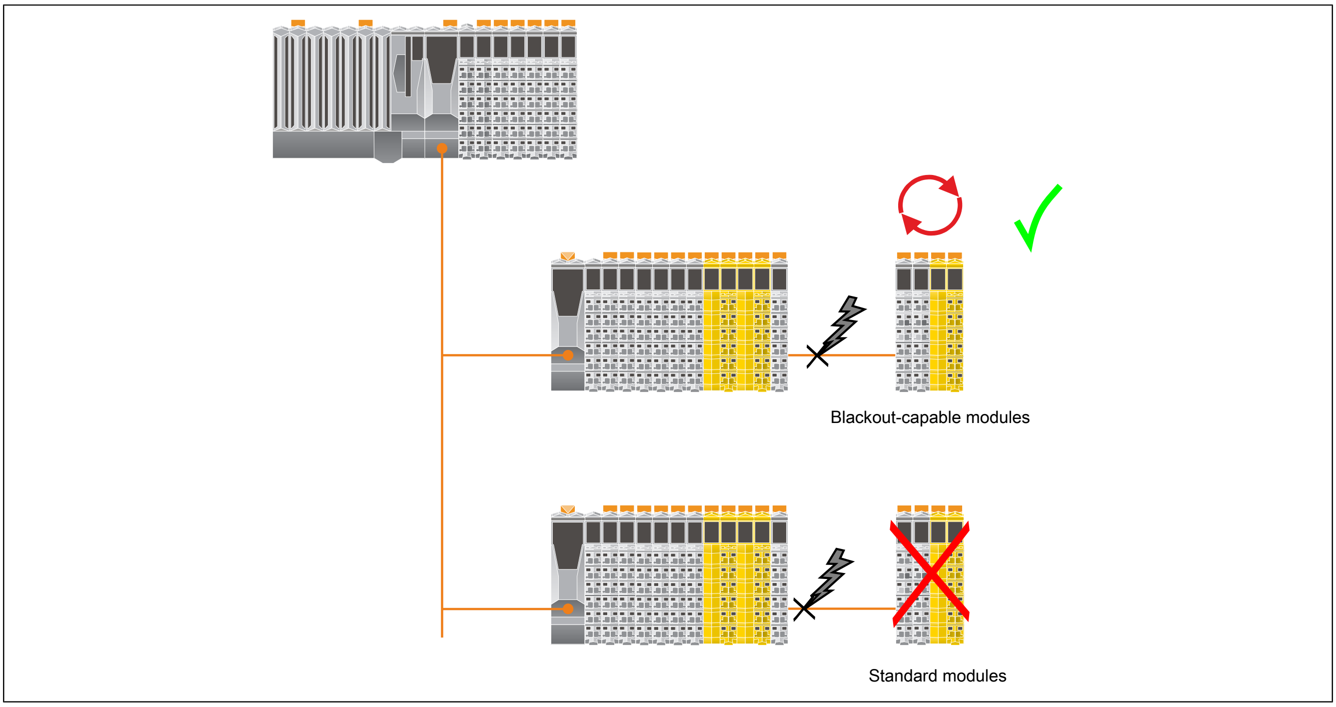
Non-blackout modules are reset and operated according to their default characteristics.

Blackout-capable modules show the following behavior:

- The programmed function continues to be executed.
- Subordinate networks continue to work.
- Data from the CPU is initialized with "0".
- After the disturbance has been corrected, the module bumplessly returns to the higher-level network.

Warning!

Blackout mode causes data from the CPU to be initialized with "0". If blackout mode is used in combination with "output inversion", this can lead to the unwanted setting of outputs.



Programming blackout mode

Blackout mode cannot be detected by the blackout-capable modules themselves. If it is necessary to program specific blackout behavior in an application, an indirect method must therefore be chosen.

One possibility is to implement a counter in the blackout-capable module's higher-level CPU and query it cyclically. Blackout mode would make itself noticeable in this case by a counter value that no longer changes or a counter value of zero.

Blackout-capable modules can be divided into 2 categories:

- **Programmable modules**
  The blackout function is programmed using existing function blocks. In other words, the existing technologies for application programming or reACTION Technology are used.
  The blackout function is executed largely independently of other system components.
- **Standard function modules**
  These modules are not programmable and maintain their default behavior in blackout mode.

Standalone function

The standalone function is an extension of blackout mode. After switching on the power supply, blackout mode is enabled immediately regardless of whether a network connection exists. This means that after switching on the power supply, the module begins executing the most recently saved configuration or application without waiting for activity or synchronization with a higher-level CPU or SafeLOGIC controller.

As soon as the network is active, bumpless synchronization between the module and existing network takes place.

Warning!

Standalone modules act identically to blackout mode on system startup and until the network connection is established. Their use therefore requires extreme caution!

Requirements for operation

The following requirements must be met in order to use the standalone function:

- The module being used must support the standalone function.
- Parameter "Standalone mode" must be enabled in Automation Studio.
- For the standalone function on the bus controller (e.g. X20SL8101), blackout mode is enabled for at least 1 module on the local X2X Link network.
- The module must have been operated with a CPU at least once in order to have a valid configuration.

Information:

The use of the standalone function in connection with DNA is not permitted. Static addresses must be used.

Warning!

The following aspects need to be taken into account in particular:

- **The module must be clearly (and permanently) identified to highlight its distinctive behavior from the standard.**
- **Service technicians must be well-versed with the special characteristics of these modules.**
- **Before connecting the terminal block to a module with an enabled standalone function, at least one of the following conditions must be met:**
    - **It must be ensured that the module is really meant to be operated with the standalone function and the configuration on the module has been checked for correctness.**
    - **The flashing sequence of the module indicates the "normal, network-connected operational state" of the module.**

Area of application

Initial situation

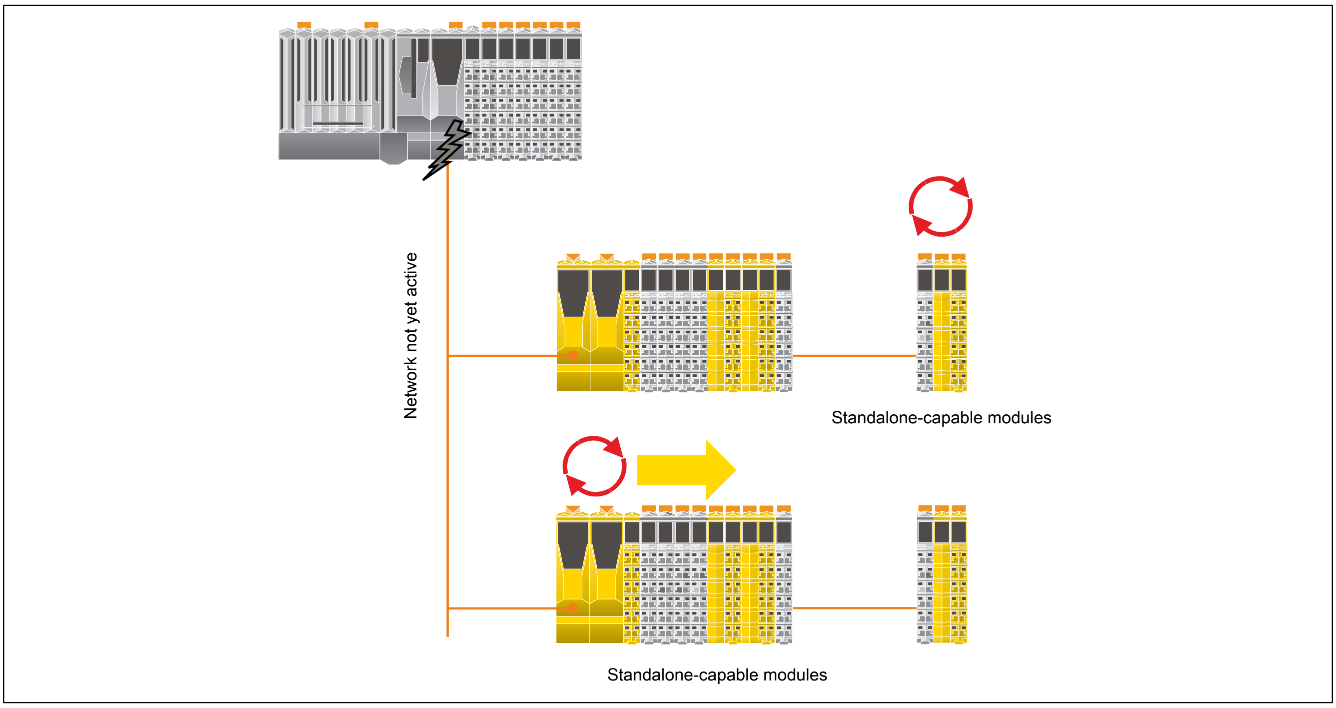
Several stations in an application are connected to the CPU via network cables. After the entire system has been switched off and on, a fault results in the network connection not being established.

Effect

Non-standalone modules are put into the active state only after the application starts up.

Standalone-capable modules show the following behavior:

- The boot procedure is started without waiting on a higher-level network.
- The module behaves identically to blackout mode.
- As soon as the network becomes active, it is bumplessly added to the higher-level network.



2.6.7.2.7.6 Setup mode

Setup mode supports the user during commissioning.

Setup mode is supported in hardware upgrade 1.10.2.x and later.
Automation Runtime B4.26 or higher is required to use setup mode.

Active setup mode is indicated by both the FAILSAFE LED (X20SL81xx series) or SE LED (X20SLXxxx series) as well as an entry in the logbook.

When setup mode is active, acknowledgment requests "SafeKEY exchange", "Firmware acknowledge" and "UDID mismatch" are no longer necessary.

Setup mode can be enabled and disabled using the operating elements of the "Remote Control" in SafeDESIGNER (X20SL81xx and X20SLXxxx series) or using the selector switch and acknowledgment button (X20SL81xx series).

Danger!

Setup mode is only permitted to be enabled during the commissioning of the machine/system.
Setup mode must be disabled during operation.

Danger!

After setup mode is ended, functional testing including a wiring test must be carried out.
If a SafeKEY or SafeLOGIC controller is replaced while setup mode is active, then setup mode will be disabled.
Functional testing must also be carried out in this case.

Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions.

Be sure to validate the entire safety function!

2.6.8 Intelligent programmable modules

2.6.8.1 Overview

Model number	Short description	Page
X20SLX210	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC	234
X20SLX402	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20cSLX402	X20 safe digital mixed module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20SLX410	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20cSLX410	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20SLX806	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	299
X20SLX811	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width	234
X20SLX842	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs	299
X20SLX910	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234
X20cSLX910	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	234

2.6.8.2 X20(c)SLXx1x

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 56: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 57: Organization of general notices

2.6.8.2.1 General information

The modules are equipped with SafeLOGIC functionality that allows them to safely execute applications designed in SafeDESIGNER. The modules can be used in safety applications up to PL e or SIL 3.

The SafeLOGIC controller coordinates the safety-related communication of all modules involved in the application. In this context, the SafeLOGIC controller also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided in both variants, however.
In addition, SafeLOGIC-X products have the same I/O properties as the associated SafeIO products.

- openSAFETY manager for up to 10 / 20 / 100 / 280 SafeNODES
- Flexibly programmable using Automation Studio / SafeDESIGNER
- Innovative management of safe machine options (SafeOPTION)
- Parameter and configuration management

2.6.8.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

SafeLOGIC function

The module is equipped with SafeLOGIC functionality that allows it to safely execute applications designed in SafeDESIGNER. The module can be used in safety-related applications up to PL e or SIL 3.

In addition, the module coordinates the safety-related communication of all modules involved in the application. In this context, the module also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided with both variants, however.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

2.6.8.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.8.2.2 Overview

Module	X20SLX210	X20SLX410	X20SLX811	X20SLX910
Safe digital inputs				
Number of inputs	2	4	8	20
Nominal voltage	24 VDC			
Input filter				
Hardware	≤150 µs			
Software	Default 0 ms, configurable between 0 and 500 ms			
Input circuit	Sink			
Pulse outputs				
Design	Push-Pull			
Switching voltage	I/O power supply minus residual voltage			

Table 58: Digital input modules

2.6.8.2.3 Order data

X20SLX210 / X20SLX410 X20SLX811 X20SLX910

Model number	Short description
	Intelligent programmable modules
X20SLX210	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC
X20SLX410	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20cSLX410	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20SLX811	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width
X20SLX910	X20 safe digital input module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20cSLX910	X20 safe digital input module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
	Required accessories
	Bus modules
X20BM13	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous, single-width
X20BM16	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous, single-width
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 59: X20SLX210, X20SLX410, X20cSLX410, X20SLX811, X20SLX910, X20cSLX910 - Order data

2.6.8.2.4 Technical data

Model number	**X20SLX210**	**X20SLX410**	**X20cSLX410**	**X20SLX811**	**X20SLX910**	**X20cSLX910**
Short description						
I/O module	2 safe digital inputs, 2 pulse outputs, 24 VDC, SafeLOGIC-X technology	4 safe digital inputs, 4 pulse outputs, 24 VDC, SafeLOGIC-X technology		8 safe digital inputs, 4 pulse outputs, 24 VDC, SafeLOGIC-X technology	20 safe digital inputs, 4 pulse outputs, 24 VDC, SafeLOGIC-X technology	
General information						
B&R ID code	0xC5B0	0xC5B2	0xE288	0xE757	0xC5B1	0xE4D1
System requirements						
Automation Studio	4.0.16 or later			4.0 or later	4.0.16 or later	
Automation Runtime	From Safety Release 1.7 to 1.9: F4.06 or later Safety Release 1.10 or later: B4.25 or later			B4.25 or later	From Safety Release 1.7 to 1.9: F4.06 or later Safety Release 1.10 or later: B4.25 or later	
SafeDESIGNER	3.1.0 or later			4.2.0 or later	3.1.0 or later	
Safety Release	1.7 or later			1.10 or later	1.7 or later	
Status indicators	I/O function per channel, operating state, module status					
Diagnostics						
Module run/error	Yes, using status LED and software					
Inputs	Yes, using status LED and software					
Max. I/O cycle time	1600 µs			1 ms	1600 µs	
Power consumption						
Bus	0.25 W	0.32 W		0.4 W		
Internal I/O	1 W	1.25 W		2.5 W	1.6 W	
Electrical isolation						
Channel - Bus	Yes					
Channel - Channel	No					
Certifications						
CE	Yes					
KC	Yes		-		Yes	-
EAC	Yes					
UL	cULus E115267 Industrial control equipment			cULus E115267 Industrial control equipment	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5			-	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X					
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)			In preparation	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013					
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3					
Functional safety	EN 50156-1:2004					
Safety characteristics						
EN ISO 13849-1:2015						
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1)]					
PL	PL e					
DC	>94%					
MTTFD	2500 years					
Mission time	Max. 20 years					
IEC 61508:2010, IEC 61511:2004, EN 62061:2013						
SIL CL	SIL 3					
SFF	>90%					
PFH / PFH_d						
Module	$<1*10^{-10}$					
openSAFETY wired	Negligible					
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour					
PFD	$<2*10^{-5}$					
Proof test interval (PT)	20 years					

Table 60: X20SLX210, X20SLX410, X20cSLX410, X20SLX811, X20SLX910, X20cSLX910 - Technical data

Model number	X20SLX210	X20SLX410	X20cSLX410	X20SLX811	X20SLX910	X20cSLX910
Functionality						
Communication with each other	Communication only possible with SafeLOGIC controller X20(c)SL81xx Max. 1 active SafeLOGIC-X controller per standard X20(c)CPxxxx CPU [2]					
Support for machine options						
BOOL	64					
INT	-					
UINT	-					
DINT	-					
UDINT	-					
SafeMOTION support	Yes					
Max. number of SafeMOTION axes	4, depends on the data width of the modules used					
Timing precision	Time * 0.05 + Cycle time of the safety application					
Shortest task class cycle time	2 ms					
Max. number of openSAFETY nodes	10, depends on the data width of the modules used					
Data exchange between CPU and SL						
Max. total data width for each direction	8 bytes					
Max. number of data points for each direction						
BOOL	64					
INT	4					
UINT	4					
DINT	2					
UDINT	2					
Data exchange between SL and SL						
Max. total number of data points for each direction [3]	2					
Max. number of data points for each direction						
BOOL	16					
INT	2					
UINT	2					
DINT	2					
UDINT	2					
Limit values for SafeDESIGNER application						
Max. resources available for SafeDESIGNER info window entries [4]						
FB instances	V1.7.xx: 128, V1.8.xx and later: 256					
Marker memory	5120 bytes (0x1400)					
Stack memory	2048 bytes					
Memory for safe input data	128 bytes, 68 bytes of which are usable for modules					
Memory for safe output data	64 bytes					
Memory for standard input data	64 bytes					
Memory for standard output data	64 bytes					
Marker count	V1.7.xx: 128, V1.8.xx and later: 256					
Additional SafeDESIGNER limit values						
Max. number of function block types	64					
Max. number of force variables	8					
Max. number of variable with variable status	V1.7.xx: 64, V1.8.xx and later: 128					
I/O power supply						
Nominal voltage	24 VDC					
Voltage range	24 VDC -15% / +20%					
Integrated protection	Reverse polarity protection					
Safe digital inputs						
Nominal voltage	24 VDC					
Input characteristics per EN 61131-2	Type 1					
Input filter						
Hardware	≤150 µs					
Software	Configurable between 0 and 500 ms					
Input circuit	Sink					
Input voltage	24 VDC -15% / +20%					
Input current at 24 VDC	Max. 3.28 mA					
Input resistance	Min. 7.33 kΩ					
Error detection time	200 ms			100 ms	200 ms	
Isolation voltage between channel and bus	500 V_{eff}					
Switching threshold						
Low	<5 VDC					
High	>15 VDC					
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line					

Table 60: X20SLX210, X20SLX410, X20cSLX410, X20SLX811, X20SLX910, X20cSLX910 - Technical data

Model number	X20SLX210	X20SLX410	X20cSLX410	X20SLX811	X20SLX910	X20cSLX910
Pulse outputs						
Variant	Push-Pull					
Nominal output current	50 mA					
Output protection	Shutdown of individual channels in the event of overload or short circuit [5]					
Peak short-circuit current	25 A for 15 µs			0.5 A for 120 µs	25 A for 15 µs	
Short-circuit current	100 mA_{eff}			15 mA_{eff}	100 mA_{eff}	
Leakage current when switched off	0.1 mA					
Residual voltage	2 VDC			≤4 VDC	2 VDC	
Switching voltage	I/O power supply minus residual voltage					
Total nominal current	100 mA	200 mA				
Operating conditions						
Mounting orientation						
Horizontal	Yes					
Vertical	Yes					
Installation elevation above sea level	0 to 2000 m, no limitation					
Degree of protection per EN 60529	IP20					
Ambient conditions						
Temperature						
Operation						
Horizontal mounting orientation	0 to 60°C		-40 to 60°C [6]	0 to 60°C		-40 to 60°C [6]
Vertical mounting orientation	0 to 50°C		-40 to 50°C [7]	0 to 50°C		-40 to 50°C [7]
Derating	See section "Derating".					
Storage	-40 to 85°C					
Transport	-40 to 85°C					
Relative humidity						
Operation	5 to 95%, non-condensing		Up to 100%, condensing	5 to 95%, non-condensing		Up to 100%, condensing
Storage	5 to 95%, non-condensing					
Transport	5 to 95%, non-condensing					
Mechanical properties						
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.			Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module (single-width) separately.	Order 2x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm			$12.5^{+0.2}$ mm	$25^{+0.2}$ mm	

Table 60: X20SLX210, X20SLX410, X20cSLX410, X20SLX811, X20SLX910, X20cSLX910 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) If there are multiple SafeLOGIC-X controllers in the Automation Studio hardware tree, all but 1 must be disabled.
3) Keep in mind that 8 BOOL count as 1 data point.
4) For a parameter description, see section "Message window" of the SafeDESIGNER documentation.
5) The protective function is provided for max. 30 minutes for a continuous short circuit.
6) Up to hardware upgrade <1.10.5.0: -25 to 60°C
7) Up to hardware upgrade <1.10.5.0: -25 to 50°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SLX210	X20SLX410	X20SLX811	X20SLX910
Derating bonus				
At 24 VDC	+2.5°C		+0°C	+5°C
Dummy module on the left	+0°C			
Dummy module on the right	+2.5°C		+0°C	+2.5°C
Dummy module on the left and right	+5°C		+0°C	+5°C
With double PFH / PFH_d	+0°C			

Table 61: Derating bonus

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

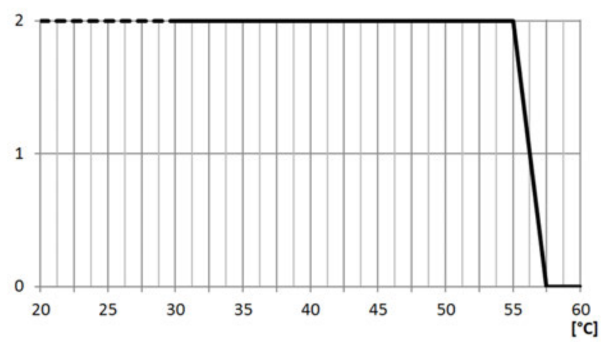
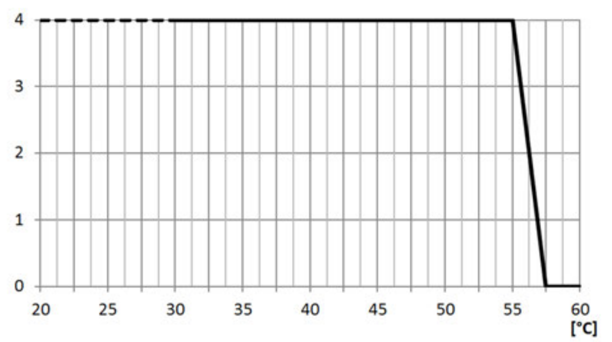
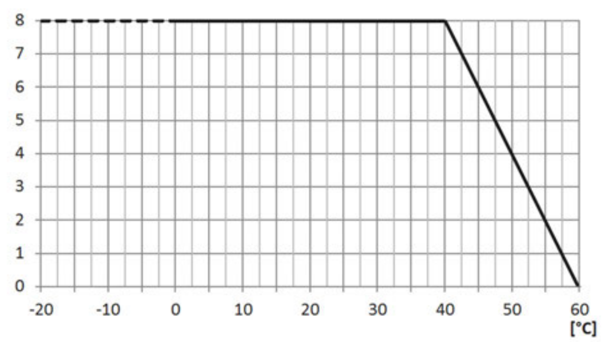
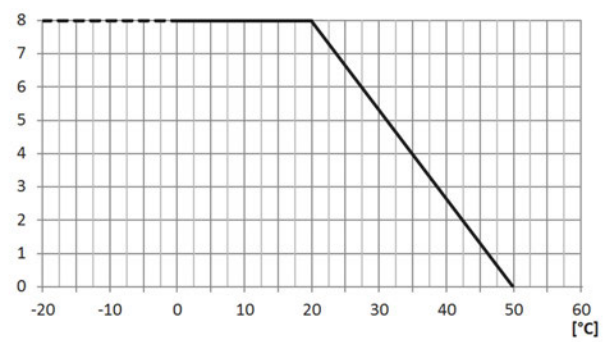
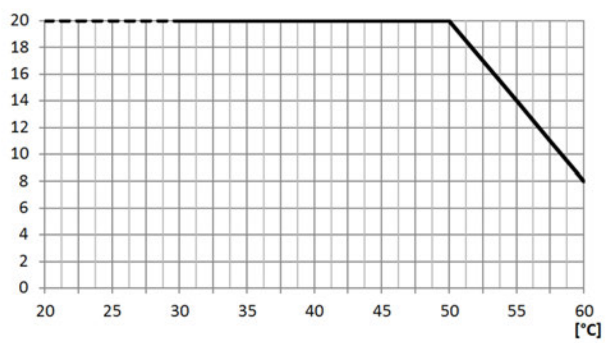
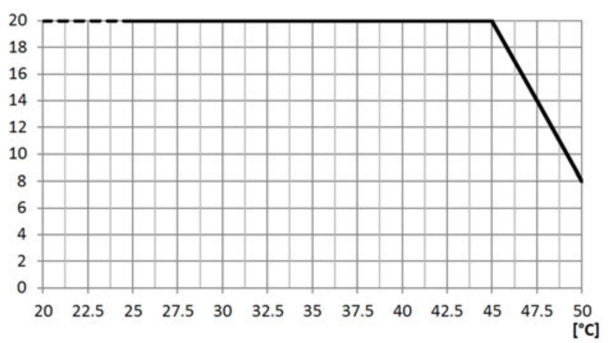
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
X20SLX210	
	No derating
X20SLX410	
	No derating
X20SLX811	
	
X20SLX910	
	

Table 62: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.8.2.5 LED status indicators

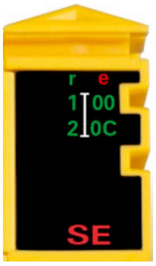
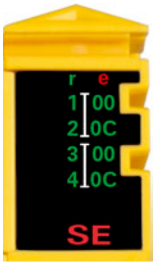
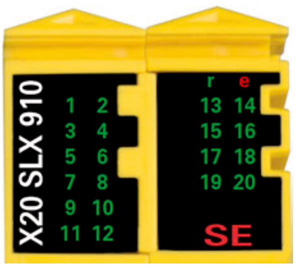
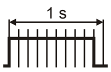
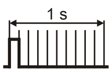
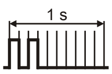
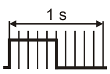
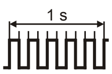
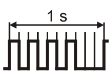
Figure	LED	Color	Status	Description
 X20SLX210  X20SLX410  X20SLX811  X20SLX910	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 20	Input state of the corresponding digital input The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an input channel
			Blinking (only for X20SLX910 and X20SLX811)	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels or booting not yet completed
		Green	On	Input set
	OO	These LEDs may not be available depending on the module type. Errors in dual-channel evaluation are indicated by channel LEDs 1 to 20 in this case.		Open - Open: Dual-channel evaluation on channels 1 and 2 using the "Equivalent" function block
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels or startup not yet completed
		Green	On	Evaluation channel set
	OC	These LEDs may not be available depending on the module type. Errors in dual-channel evaluation are indicated by channel LEDs 1 to 20 in this case.		Open - Closed: Dual-channel evaluation on channels 1 and 2 using the "Antivalent" function block
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels or booting not yet completed
		Green	On	Evaluation channel set
	SE	Red	Off	RUN mode or I/O component not supplied with voltage, safety firmware in OPERATIONAL state
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state or "SafeOSstate!=RUN"
				Safe communication channel not OK, openSAFETY connection valid problem or "SafeOSstate!=RUN"
				Boot phase, faulty firmware, setup mode active (hardware upgrade 1.10.2.x and later) For details about setup mode, see section "Setup mode" on page 294.
				Test/Pilot firmware or safety application created with test/pilot version of SafeDESIGNER
				SafeDESIGNER in "Debug" mode
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 63: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.8.2.6 Pinouts

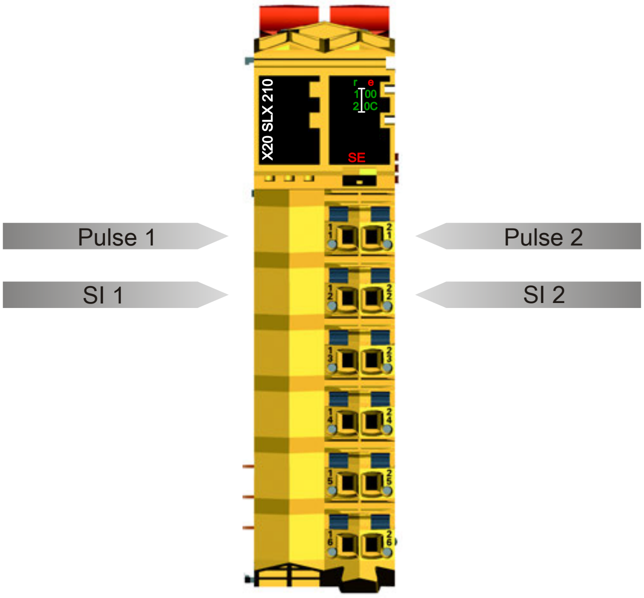


Figure 69: X20SLX210 - Pinout

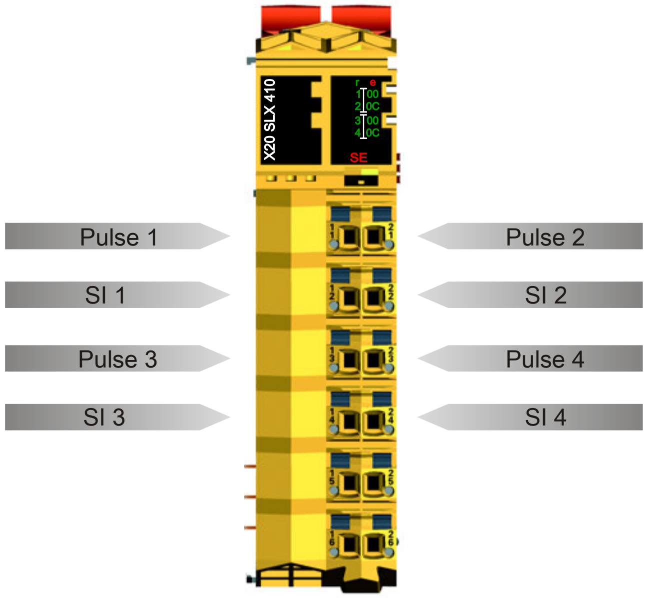


Figure 70: X20SLX410 - Pinout

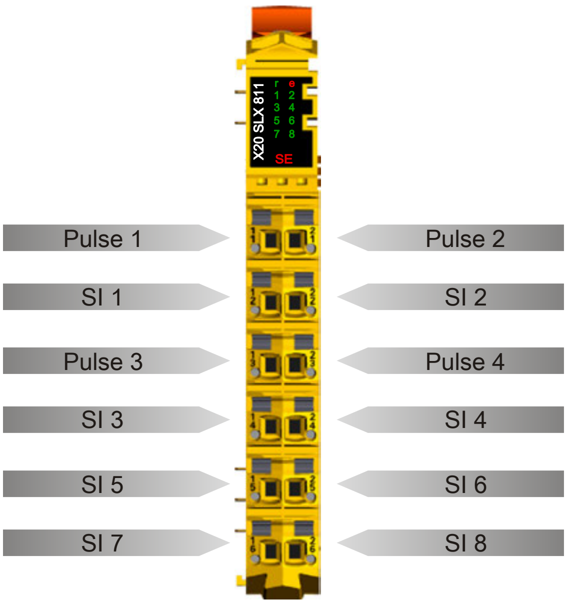


Figure 71: X20SLX811 - Pinout

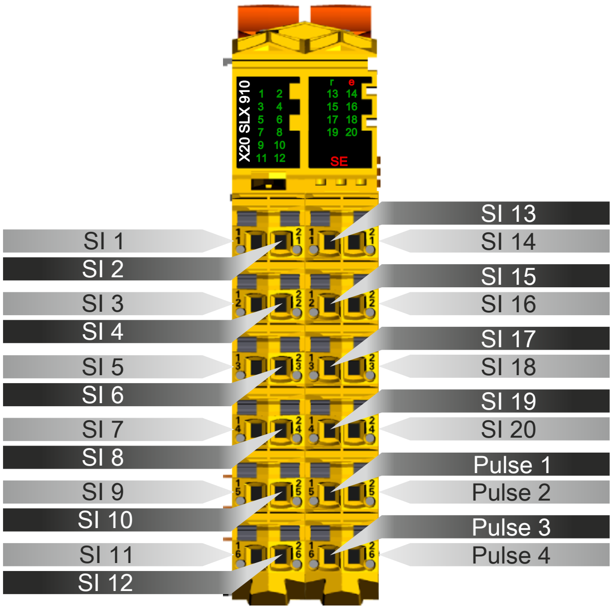


Figure 72: X20SLX910 - Pinout

2.6.8.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.8.2.7.1 Connecting single-channel sensors with contacts

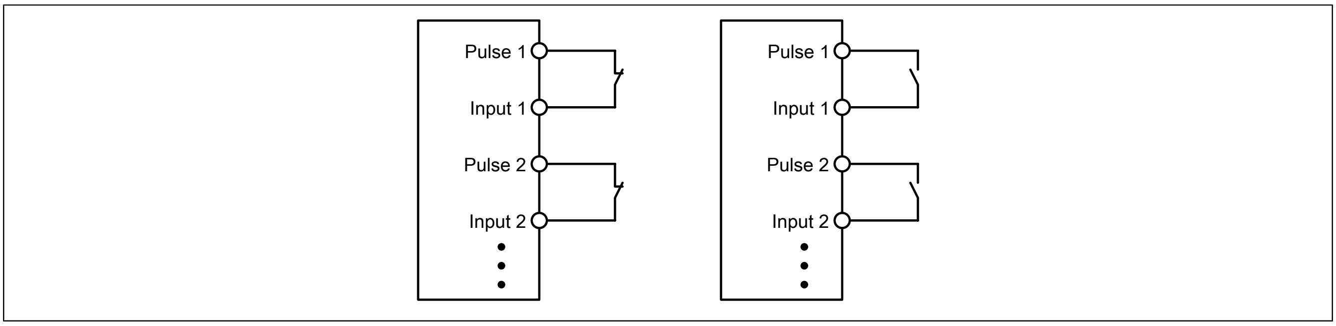


Figure 73: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.8.2.7.2 Connecting two-channel sensors with contacts

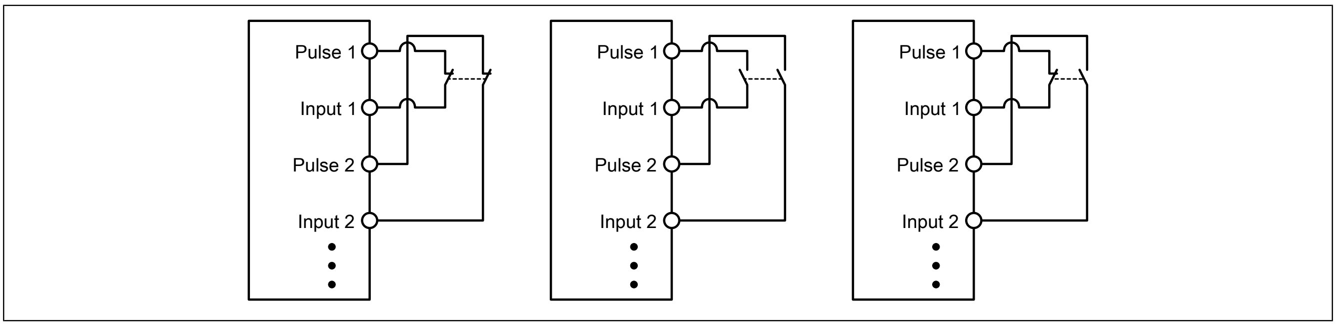


Figure 74: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.8.2.7.3 Connecting multi-channel sensors with contacts

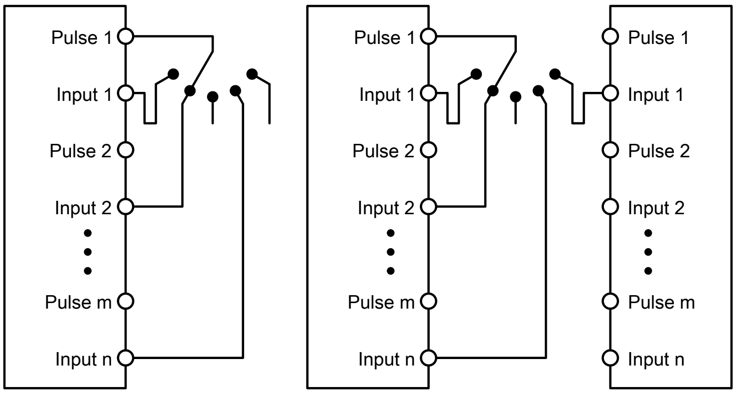


Figure 75: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.8.2.7.4 Connecting electronic sensors

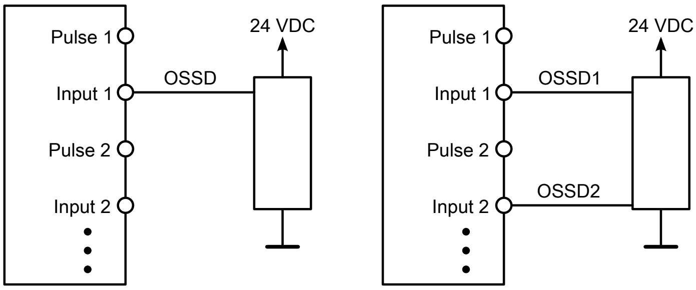


Figure 76: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.8.2.7.5 Using the same pulse signals

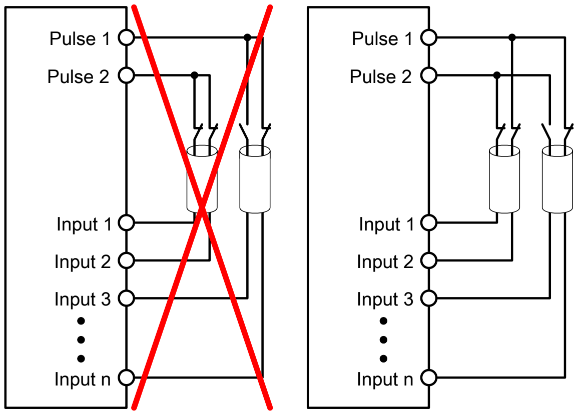


Figure 77: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.8.2.8 Error detection

2.6.8.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.8.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

> **Danger!**
>
> **Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.**

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 64: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 65: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 66: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

2.6.8.2.9 Input circuit diagram

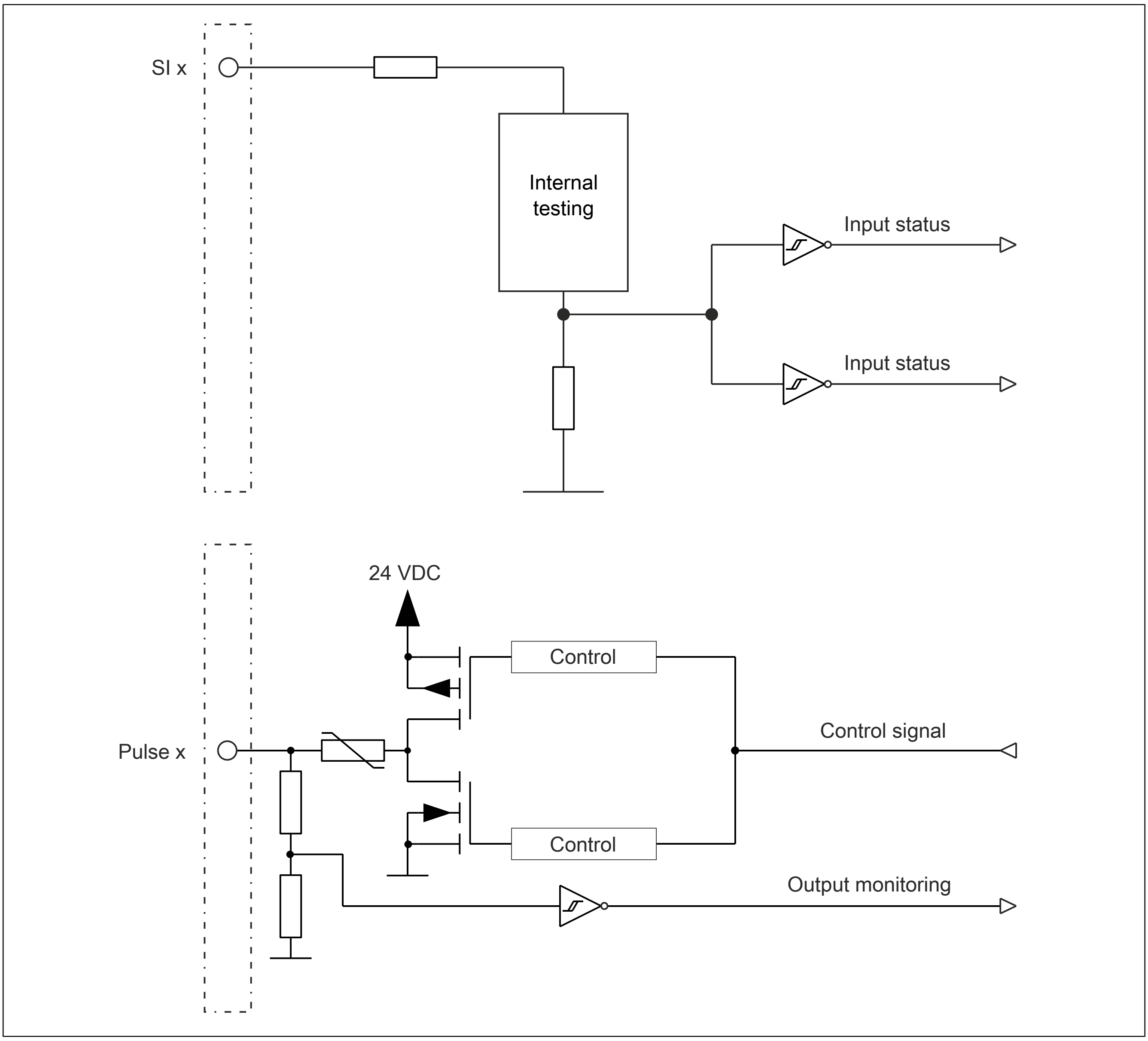


Figure 78: Input circuit diagram

2.6.8.2.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.8.2.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time			
X20SLX210	X20SLX410	X20SLX811	X20SLX910
800 µs	800 µs	500 µs	800 µs

Maximum I/O update time			
X20SLX210	X20SLX410	X20SLX811	X20SLX910
3350 µs + Filter time (see chapter "Filter")	3350 µs + Filter time (see chapter "Filter")	1150 µs + Filter time (see chapter "Filter")	3350 µs + Filter time (see chapter "Filter")

2.6.8.2.12 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

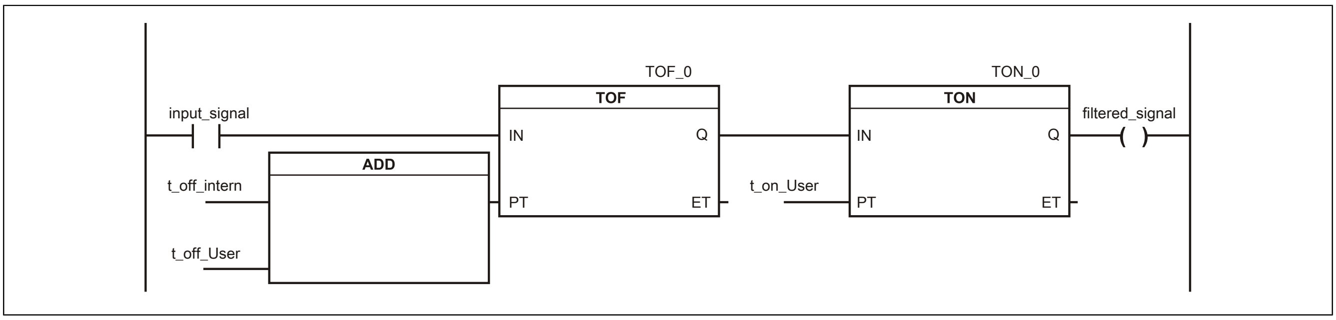


Figure 79: SI input filter - Diagram 1

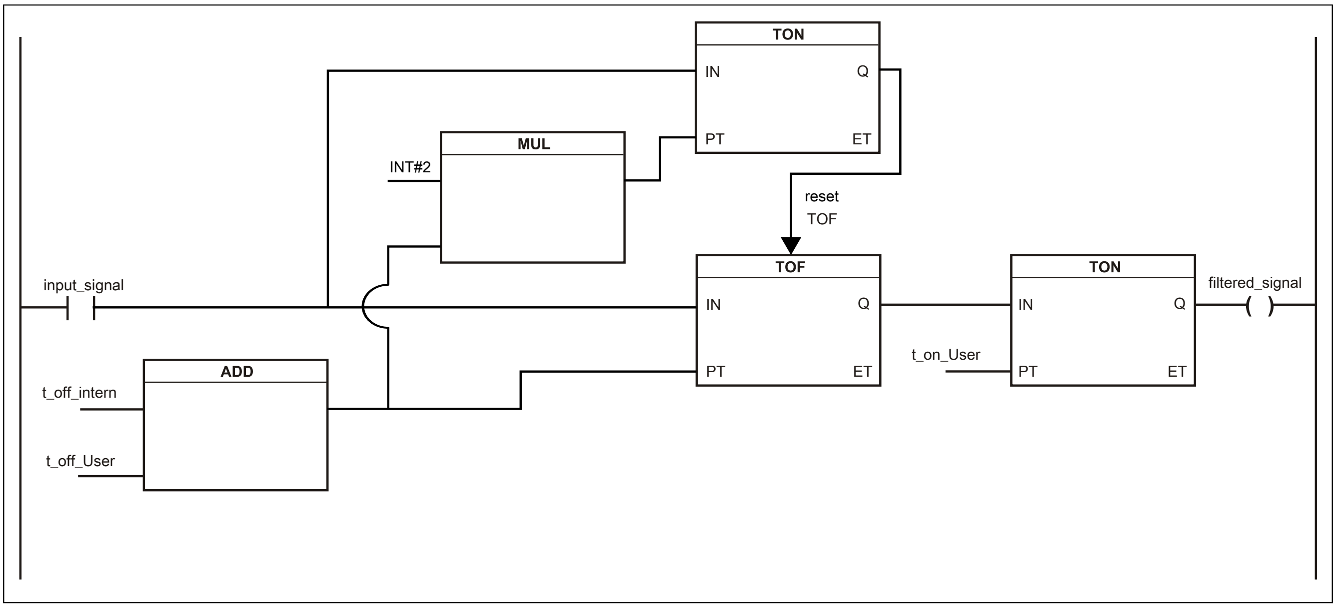


Figure 80: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.8.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.8.2.14 Register description

2.6.8.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 67: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Input status information</td><td>This parameter enables/disables channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.
This parameter may not be available depending on the module type.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 1</td><td>1</td><td>-</td></tr>
<tr><td>SafeDESIGNER project</td><td>Name of the safety project</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeDESIGNER version</td><td>SafeDESIGNER version of the safety project</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 68: I/O configuration parameters: General

Group: SafeDESIGNER to SafeLOGIC communication

Starting with SafeLOGIC V1.4.0.0 and Automation Runtime V3.04:

When SPROXY is enabled, the SafeLOGIC controller can be accessed via a TCP/IP port on the standard CPU.

This uses the SafeDESIGNER setting "SL communication via the CPU" (SafeDESIGNER V2.80 or higher).

Parameter	Description	Default value	Unit
Activate SPROXY	Enables the SafeDESIGNER online connection	On	-
Server communication port	TCP/IP port number used to access the SafeLOGIC controller • Recommended values: 50,000 to 50,100 **Note:** If multiple SafeLOGIC controllers are being used in the project, then a different port number must be configured for each one!	50000	-

Table 69: I/O configuration parameters: SafeDESIGNER to SafeLOGIC communication

Group: CPU to SafeLOGIC communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the CPU to the SafeLOGIC controller • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64.	8	-
Number of INT channels	Number of INT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 4.	0	-
Number of UINT channels	Number of UINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 4.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 2.	0	-
Number of UDINT channels	Number of UDINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 2.	0	-

Table 70: I/O configuration parameters: CPU to SafeLOGIC communication

Group: SafeLOGIC to CPU communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the SafeLOGIC controller to the CPU • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64.	8	-
Number of INT channels	Number of INT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 4.	0	-
Number of UINT channels	Number of UINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 4.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 2.	0	-
Number of UDINT channels	Number of UDINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 2.	0	-

Table 71: I/O configuration parameters: SafeLOGIC to CPU communication

Group: SafeLOGIC to SafeLOGIC communication

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Use as source SafeLOGIC</td><td>This parameter configures this SafeLOGIC controller as a data source for another SafeLOGIC controller.</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>On</td><td>This SafeLOGIC controller is available as a data source for another SafeLOGIC controller.</td></tr>
<tr><td>Off</td><td>This SafeLOGIC controller is not available as a data source for other SafeLOGIC controllers.</td></tr>
</table></td></tr>
<tr><td>Extended source SafeLOGIC communication (Safety Release 1.4 and Automation Runtime V3.08 required)</td><td>This parameter enables the option of configuring the number of data points for "SafeLOGIC to SafeLOGIC communication" for connections where this SafeLOGIC controller serves as a data source for another SafeLOGIC controller.</td><td>Off</td><td>-</td></tr>
</table>

Table 72: I/O configuration parameters: SafeLOGIC to SafeLOGIC communication

2.6.8.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td>Cycle_Time_typical_us</td><td>This parameter specifies the cycle time of the SafeDESIGNER application. If set correctly, this can improve the stability of the cyclic data connection. The correct value is application-dependent and can be read in the "Cycle time" field of the SafeLOGIC controller's Info dialog box in the RUN [Safe] state of the SafeLOGIC-X controller.
• Permissible values: 2000 to 20,000 µs (corresponds to 2 to 20 ms)</td><td>20000</td><td>µs</td></tr>
<tr><td>Cycle_Time_max_us
(Release 1.5 and later)</td><td>Parameter for checking whether a maximum time between 2 SafeLOGIC cycles is exceeded.
• Permissible values: 2100 to 41,000 µs (corresponds to 2.1 to 41 ms)
IMPORTANT:
This value should not be the same as the actual cycle time; jitter must also be taken into account.
The actual cycle time is influenced by the SafeDESIGNER application and the "SLXioCycle" data point.
The actual cycle time can be seen in the SafeLOGIC "Info" dialog box.</td><td>40000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Timeout_s</td><td>Timeout for changing the safety modules to the PREOPERATIONAL state after the SafeLOGIC controller drops out or if there is a communication problem between the safety module and the SafeLOGIC controller.
This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module.
• Permissible values: 30 to 3000 s
Notes
• The shorter the time, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>60</td><td>s</td></tr>
<tr><td>Number_of_scans</td><td>This parameter defines the number of module search scans completed during startup.
This parameter is used to optimize the startup behavior of the system, especially if optional modules are configured.
• Permissible values: 1 to 10</td><td>5</td><td>-</td></tr>
<tr><td rowspan="2">ExternalMachineOptions
(Release 1.4 and later)</td><td>Enables external machine options</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>External machine options are enabled.</td></tr><tr><td>No</td><td>External machine options are disabled.</td></tr></table></td></tr>
<tr><td rowspan="2">ExternalStartupFlags
(Release 1.4 and later)</td><td>Enables external startup flags</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>External startup flags are enabled.</td></tr><tr><td>No</td><td>External startup flags are disabled.</td></tr></table></td></tr>
</table>

Table 73: SafeDESIGNER parameters: Basic

Danger!

If parameter "ExternalMachineOptions" or "ExternalStartupFlags" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Group: Safety_Response_Time_Defaults

The parameters for the safety response time are generally set in the same way for all stations involved in the application. This is why these parameters are configured for the SafeLOGIC controller in group "Safety_Response_Time_Defaults" in SafeDESIGNER.

If "Manual_Configuration = No" is set for the modules in the network, then these default values are used.

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Default_Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Default_Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>12000</td><td>µs</td></tr>
<tr><td>Default_Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Default_Max_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 30,000 µs (corresponds to 0 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Default_Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Default_Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Default_Min_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 30,000 µs (corresponds to 0 to 30 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Default_Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 10,000 to 5,000,000 µs (corresponds to 10 ms to 5 s)</td><td>150000</td><td>µs</td></tr>
<tr><td>Default_Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 74: SafeDESIGNER parameters: Safety_Response_Time_Defaults

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Pulse_Source
(Release 1.4 and later)</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>See table.</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="4">Possible "Pulse_Source" for X20SLX210 and X20SLX410</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td></tr>
</table>
All available pulse outputs on the X20SLX811 and X20SLX910 can be specified as pulse sources.
The default values can be determined using the following tables:
<table>
<tr><th>Channel</th><th>Default "Pulse_Source" for X20SLX811</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table>
<table>
<tr><th>Channel</th><th>Default "Pulse_Source" for X20SLX910</th></tr>
<tr><td>1, 3, 5, 7, 9, 11</td><td>Channel 1</td></tr>
<tr><td>2, 4, 6, 8, 10, 12</td><td>Channel 2</td></tr>
<tr><td>13, 15, 17, 19</td><td>Channel 3</td></tr>
<tr><td>14, 16, 18, 20</td><td>Channel 4</td></tr>
</table>
Note:
If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".
</td></tr>
<tr><td rowspan="2">Pulse_Mode</td><td>This parameter can be used to specify the "Pulse_Mode" for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.
Release 1.4 and later:
The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured to "Pulse_Mode = External" (X20SLX210 and X20SLX410 only).</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter_On_us</td><td>Switch-on filter for the channel used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)
(up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>TwoChannelProcessingMode
(only for X20SLX811 and X20SLX910)</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 75: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

2.6.8.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic release</td><td>-</td></tr>
<tr><td>Node Guarding Timeout</td><td>Timeout for changing the safety modules to the PRE_OPERATIONAL state after the SafeLOGIC controller drops out or if there is a communication problem between the safety module and the SafeLOGIC controller.
This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module.
• Permissible values: 30 to 300 s
Notes
• The shorter the time, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>60</td><td>s</td></tr>
<tr><td rowspan="2">External Startup Flags</td><td>Enables external startup flags</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Enables external startup flags</td></tr><tr><td>No</td><td>Disables external startup flags</td></tr></table></td></tr>
<tr><td>Number of scans</td><td>This parameter defines the number of module search scans completed while booting.
This parameter is used to optimize the startup behavior of the system, especially if optional modules are configured but not available.
• Permissible values: 1 to 10</td><td>5.
Hardware upgrade 1.10.2.0 or later: 3</td><td>-</td></tr>
<tr><td rowspan="2">Activate Setup Mode on empty SafeKEY (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables setup mode after downloading a project to a blank SafeKEY / blank section of the CompactFlash card.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Setup mode is enabled.</td></tr><tr><td>No</td><td>Setup mode is disabled.</td></tr></table></td></tr>
<tr><td rowspan="2">Auto acknowledge firmware mismatch (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a firmware exchange (acknowledgment request "Firmware Acknowledge").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of firmware exchange is enabled.</td></tr><tr><td>No</td><td>Automatic acknowledgment of firmware exchange is not enabled.</td></tr></table></td></tr>
<tr><td rowspan="2">Auto acknowledge SafeKEY exchange (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a SafeKEY exchange (acknowledgment request "SafeKEY Exchange").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of SafeKEY exchange is enabled.</td></tr><tr><td>No</td><td>Automatic acknowledgment of SafeKEY exchange is not enabled.</td></tr></table></td></tr>
</table>

Table 76: SafeDESIGNER parameters: Basic

Danger!

If parameter "External Startup Flags" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Information:

Startup time is also affected by the asynchronous bandwidth on the POWERLINK network. For optimization options, see Automation Help under Communication → POWERLINK → General information → Multiple asynchronous send.

Information:

The information in section "Setup mode" on page 294 must be observed when using parameter "Activate Setup Mode on empty SafeKEY". The information in section "Automatic acknowledgment" on page 277 must be observed when using parameters "Auto acknowledge firmware mismatch" and "Auto acknowledge SafeKEY exchange".

Group: Safety Response Time Defaults

The parameters for the safety response time are generally set in the same way for all stations involved in the application. This is why these parameters are configured for the SafeLOGIC controller in group "Safety Response Time Defaults" in SafeDESIGNER.

If "Manual Configuration = No" is set for the individual modules, then these default values are used.

Parameter	Description	Default value	Unit
Default Safe Data Duration	This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller and SafeIO module. For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added. • Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)	150000	µs
Default Additional Tolerated Packet Loss	This parameter specifies the number of additional tolerated lost packets during data transfer. • Permissible values: 0 to 10	0	Packets
Default Packets per Node Guarding	This parameter specifies the maximum number of packets used for node guarding. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.	5	Packets

Table 77: SafeDESIGNER parameters: Safety Response Time Defaults

Group: Module Configuration

Parameter	Description	Default value	Unit
External Machine Options	Enables external machine options **Parameter value** / **Description** Yes-ATTENTION: Enables external machine options No: Disables external machine options	No	-
Cycle Time max	Parameter for checking whether a maximum time between 2 SafeLOGIC cycles is exceeded. • Permissible values: 2100 to 41,000 μs (corresponds to 2.1 to 41 ms) **Important:** This value should not be the same as the actual cycle time; jitter must also be taken into account. The actual cycle time is influenced by the SafeDESIGNER application and the "SLXioCycle" data point. The actual cycle time of the safety application can be seen in the SafeLOGIC "Info" dialog box.	40000	μs

Table 78: SafeDESIGNER parameters: Module Configuration

Danger!

If parameter "External Machine Options" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>See table</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="4">Possible "Pulse Source" for X20SLX210 and X20SLX410</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td></tr>
</table>
All available pulse outputs on the X20SLX811 and X20SLX910 can be specified as pulse sources.
The default values can be determined using the following tables:
<table>
<tr><th>Channel</th><th>Default "Pulse Source" for X20SLX811</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table>
<table>
<tr><th>Channel</th><th>Default "Pulse Source" for X20SLX910</th></tr>
<tr><td>1, 3, 5, 7, 9, 11</td><td>Channel 1</td></tr>
<tr><td>2, 4, 6, 8, 10, 12</td><td>Channel 2</td></tr>
<tr><td>13, 15, 17, 19</td><td>Channel 3</td></tr>
<tr><td>14, 16, 18, 20</td><td>Channel 4</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the "Pulse Mode" for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured to "Pulse Mode = External" (X20SLX210 and X20SLX410 only).</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential "low phases" of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Two-Channel Processing Mode
(only for X20SLX811 and X20SLX910)</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 79: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

2.6.8.2.14.4 Channel list

Channel name	SLX210 SLX410	SLX811	SLX910	Access via Automation Studio	Access via Safe-DESIGNER	Data type	Description
ModuleOk	●	●	●	Read	-	BOOL	Indicates if the module is OK
SerialNumber	●	●	●	Read	-	UDINT	Module serial number
ModuleID	●	●	●	Read	-	UINT	Module ID
HardwareVariant	●	●	●	Read	-	UINT	Hardware variant
FirmwareVersion	●	●	●	Read	-	UINT	Firmware version of the module
SLXioCycle	●	●	●	Read	-	UDINT	Exchanging cyclic data between the SafeLOGIC-X controller and CPU (time in µs). This value is influenced by: • Quantity and data width of SafeNODEs • Cycle times set in Automation Studio (POWERLINK, X2X, Crosslink task) • Automation Studio configuration (see items above) Up to Safety Release 1.9: The value must be <12 ms since the openSAFETY data connection cannot be configured for larger values. Safety Release 1.10 and later: The value must be <30 ms; otherwise, the max. SafeLOGIC-X cycle time (parameter "Cycle Time max") is exceeded. In addition, values <15 ms are recommended since large values slow down the SafeDESIGNER online connection.
UDID_low	●	●	●	(Read) [1)]	-	UDINT	UDID, lower 4 bytes
UDID_high	●	●	●	(Read) [1)]	-	UINT	UDID, upper 2 bytes
SafetyFWversion1	●	●	●	(Read) [1)]	-	UINT	Firmware version - Safety processor 1
SafetyFWversion2	●	●	●	(Read) [1)]	-	UINT	Firmware version - Safety processor 2
SafetyFWversionSCM	●	●	●	(Read) [1)]	-	UINT	Firmware version - SCMar
SafetyFWcrc1 (hardware upgrade 1.10.5.0 or later)	●	●	●	(Read) [1)]	-	UINT	CRC of firmware header on safety processor 1
SafetyFWcrc2 (hardware upgrade 1.10.5.0 or later)	●	●	●	(Read) [1)]	-	UINT	CRC of firmware header on safety processor 2
ApplSDcrc	●	●	●	(Read) [1)]	-	UDINT	CRC of the SafeDESIGNER application on the module
ApplSDtime	●	●	●	(Read) [1)]	-	UDINT	Timestamp of the SafeDESIGNER application on the module in UNIX format
ApplMOptCRC	●	●	●	(Read) [1)]	-	UDINT	CRC of the external machine options on the module
ApplMOptTime	●	●	●	(Read) [1)]	-	UDINT	Timestamp of the external machine options on the module in UNIX format

Table 80: Channel list

<table>
<tr><th>Channel name</th><th>SLX210
SLX410</th><th>SLX811</th><th>SLX910</th><th>Access via Automation Studio</th><th>Access via Safe-DESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.5.0 or later)</td><td>●</td><td>●</td><td>●</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors or download of the SafeDESIGNER application to the safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
</table>

Table 80: Channel list

<table>
<tr><th>Channel name</th><th>SLX210
SLX410</th><th>SLX811</th><th>SLX910</th><th>Access via Automation Studio</th><th>Access via Safe-DESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>SLXbootState</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>Startup state of the SafeLOGIC-X system
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>0</td><td>Invalid - Firmware not yet running</td></tr>
<tr><td>1</td><td>Start - Waiting for synchronization of internal cyclic systems</td></tr>
<tr><td>4</td><td>Up to Safety Release 1.9:
Start OK - Waiting for SOD data from SCMar
Safety Release 1.10 and later:
Start OK - Application data valid</td></tr>
<tr><td>5 to 7[2]</td><td>Download of SOD data from SCMar active</td></tr>
<tr><td>8[2]</td><td>Waiting for download of SD application from SCMar</td></tr>
<tr><td>9 to 11[2]</td><td>Download of SD application from SCMar active</td></tr>
<tr><td>16[2]</td><td>Waiting for external machine option data 1 from SCMar</td></tr>
<tr><td>17 to 19[2]</td><td>Download of external machine option data 1 from SCMar active</td></tr>
<tr><td>20[2]</td><td>Waiting for external machine option data 2 from SCMar</td></tr>
<tr><td>21 to 23[2]</td><td>Download of external machine option data 2 from SCMar active</td></tr>
<tr><td>25</td><td>Safety Release 1.10 and later:
Safety PREOPERATIONAL state or "SafeOSstate!=RUN"</td></tr>
<tr><td>30[2]</td><td>Downloads OK - All data received from SCMar</td></tr>
<tr><td>32[2]</td><td>Write data received from the SCMar to flash memory</td></tr>
<tr><td>34</td><td>Waiting on X2X parameters from Automation Runtime</td></tr>
<tr><td>40[2]</td><td>Start initialization of the SD application</td></tr>
<tr><td>50[3]</td><td>Ready for RUN - Waiting on "SafeModuleOK" for the modules</td></tr>
<tr><td>52[3]</td><td>Waiting period for stable valid "SafeModuleOK" active</td></tr>
<tr><td>54[3]</td><td>Startup complete - SafeRUN</td></tr>
</table>
[2] Up to Safety Release 1.9
[3] Possible to establish connection to the SafeLOGIC-X controller via the SafePLC window in SafeDESIGNER (see dialog box "SafePLC" (control dialog box) in Automation Help).</td></tr>
<tr><td>SafeOsState</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>Status of the safety application. For details, see "SafeLOGIC "Info" dialog box in SafeDESIGNER".
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>0x00</td><td>Invalid (e.g. SafeKEY blank) or startup still active (BOOT_STATE!=0x12)</td></tr>
<tr><td>0x0F</td><td>ON (startup / internal initialization) or error (check logbook)</td></tr>
<tr><td>0x33</td><td>Loading (startup / internal initialization)</td></tr>
<tr><td>0x55</td><td>Stop [Safe]</td></tr>
<tr><td>0x66</td><td>Run [Safe]</td></tr>
<tr><td>0x99</td><td>Halt [Debug]</td></tr>
<tr><td>0xAA</td><td>Stop [Debug]</td></tr>
<tr><td>0xCC</td><td>Run [Debug]</td></tr>
<tr><td>0xF0</td><td>No execution</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxy_state</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>-</td><td>●</td><td>-</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>-</td><td>-</td><td>●</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
</table>

Table 80: Channel list

<table>
<tr><th>Channel name</th><th>SLX210
SLX410</th><th>SLX811</th><th>SLX910</th><th>Access via Automation Studio</th><th>Access via Safe-DESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>InputErrorStates</td><td>●</td><td>-</td><td>-</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="3">Type of error</th></tr>
<tr><th>Inputs</th><th colspan="2">Pulse outputs</th></tr>
<tr><th>Input stuck at high</th><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 4 to 7 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>InputErrorStates</td><td>-</td><td>●</td><td>●</td><td>(Read) 1)</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 19 = Channel 1 to 20</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>-</td><td>●</td><td>●</td><td>(Read) 1)</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeDigitalInputxx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeEquivalentInputxxyy</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentInputxxyy</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>SafeTwoChannelInputxxyy</td><td>-</td><td>●</td><td>●</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>SafeChannelOKxx</td><td>●</td><td>-</td><td>●</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeInputOKxx</td><td>-</td><td>●</td><td>-</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeEquivalentOKxxyy</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentOKxxyy</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>SafeTwoChannelOkxxyy</td><td>-</td><td>●</td><td>●</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>BOOL1xx</td><td>●</td><td>●</td><td>●</td><td>Write</td><td>Read</td><td>BOOL</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>INT1xx</td><td>●</td><td>●</td><td>●</td><td>Write</td><td>Read</td><td>INT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>UINT1xx</td><td>●</td><td>●</td><td>●</td><td>Write</td><td>Read</td><td>UINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>DINT1xx</td><td>●</td><td>●</td><td>●</td><td>Write</td><td>Read</td><td>DINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>UDINT1xx</td><td>●</td><td>●</td><td>●</td><td>Write</td><td>Read</td><td>UDINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>BOOL0xx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Write</td><td>BOOL</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>INT0xx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Write</td><td>INT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>UINT0xx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Write</td><td>UINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>DINT0xx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Write</td><td>DINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>UDINT0xx</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>Write</td><td>UDINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>SafeBOOLx</td><td>●</td><td>●</td><td>●</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>SafeLOGIC to SafeLOGIC communication channel</td></tr>
<tr><td>SafeMachineOptionxx</td><td>●</td><td>●</td><td>●</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Internal channel for machine options</td></tr>
</table>

Table 80: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Information:

Channels for SafeLOGIC to SafeLOGIC communication: See section "Display in SafeDESIGNER"

PLCopen state diagrams

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

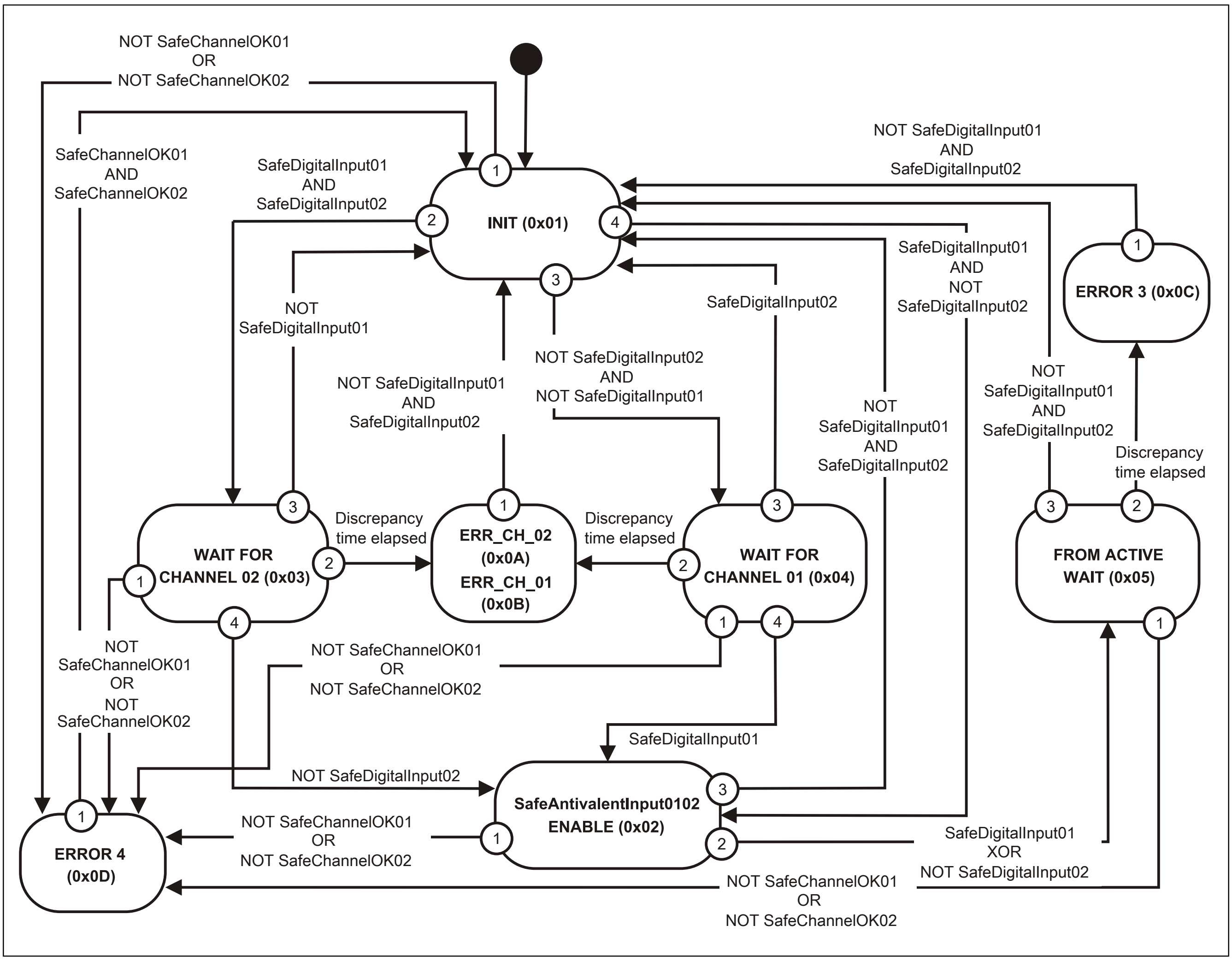


Figure 81: "Antivalent" function block - State diagram

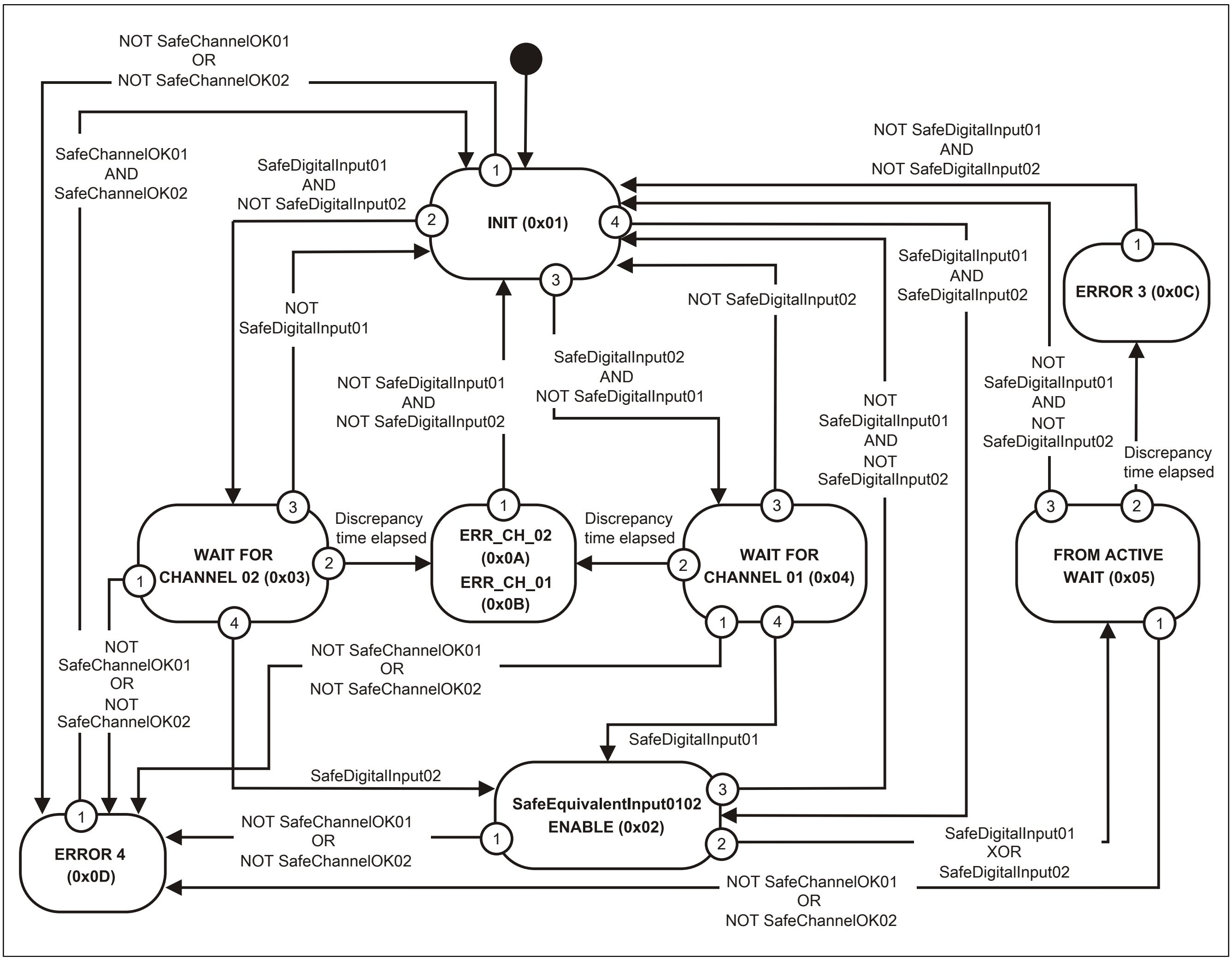


Figure 82: "Equivalent" function block - State diagram

2.6.8.2.14.5 SafeLOGIC "Info" dialog box in SafeDESIGNER

Dialog box "SafePLC info" appears if the "Info" button in dialog box "SafePLC" (control dialog box) or in dialog box "Debug" is pressed.

The dialog box shows information about the current project in the safe programming system, the project stored/ running on the safety controller, the current status of the safety controller, debugging information, etc.

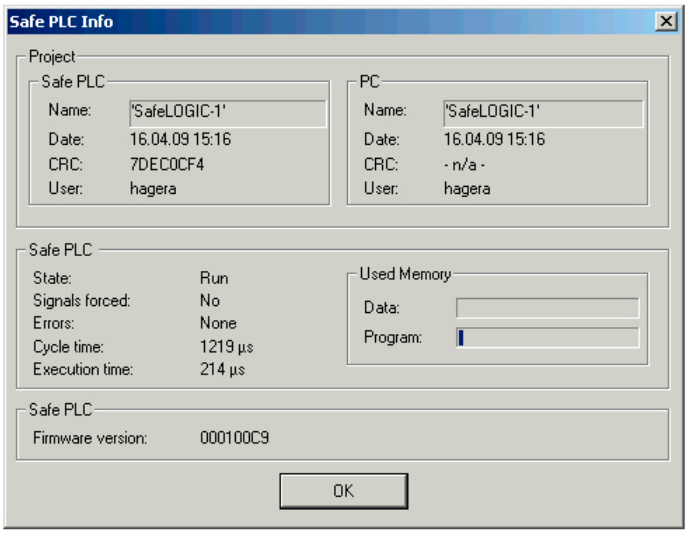


Figure 83: SafeLOGIC "Info" dialog box

Project	**Project-defining data**	
Safe PLC	Project data saved on the SafeKEY being used for the SafeLOGIC controller	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC
	User	User who made the last change
PC	SafeDESIGNER project data on the PC	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC, "- n/a -" if the project is not yet compiled
	User	User who made the last change
Safe PLC	**Status and information about the SafeLOGIC controller**	
State	Indicates the operating states of the safety controller.	
Signals forced	No	No variables are forced.
	Yes	Variables are forced.
Errors	Information regarding error messages present in the SafeDESIGNER message window	
Cycle time	Cycle time that is actually required, maximum value since the last power up This value is only relevant if "Safe PLC state = Run".	
Execution time	Actual application execution time This value corresponds to the "Safe PLC Cycle time" minus system and communication overhead.	
Used memory	Bar that shows the system resources being used	
	Data	Data memory for the safety application
	Program	Application memory for the safety application
Firmware version	Firmware version	

2.6.8.2.15 Maintenance scenarios

The operating elements on the SafeLOGIC controller (X20SL8xxx series) or the operating elements of the "Remote Control" in SafeDESIGNER (X20SL8xxx series and X20SLXxxx series) are available to handle the following maintenance scenarios.

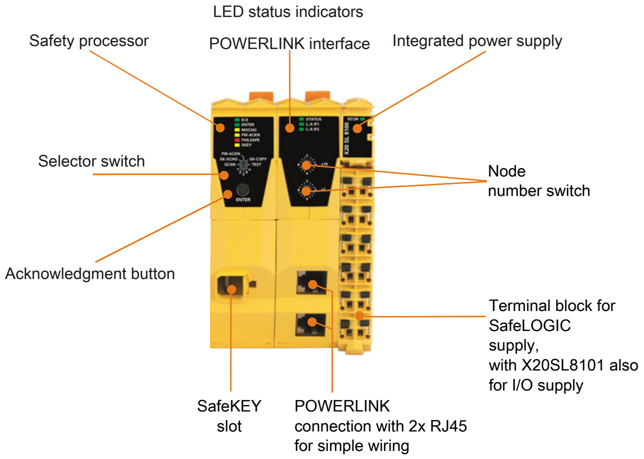


Figure 84: X20SL810x - Operating elements

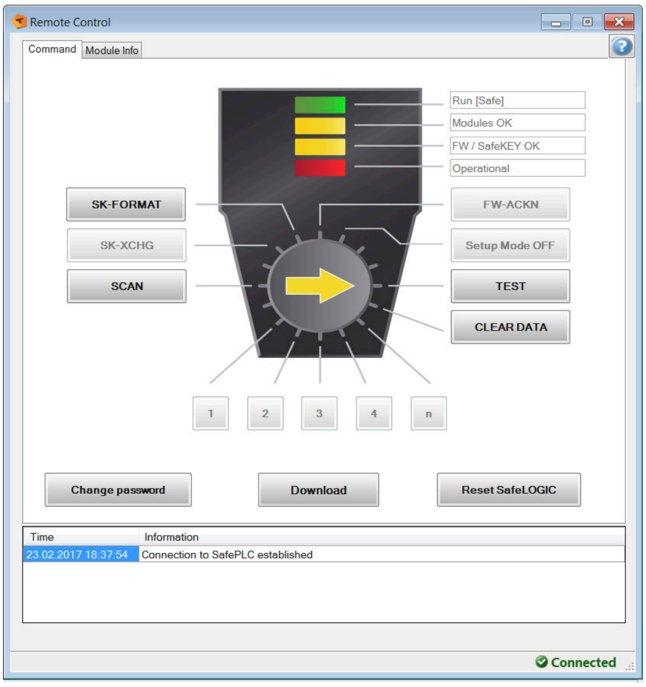


Figure 85: SafeDESIGNER - "Remote control" operating elements

For a detailed description of operating elements, see section Operating and connection elements of the technical data sheet for X20SL8xxx-series devices.

For a detailed description of operating elements, see SafeDESIGNER section Operating elements of the Remote Control in Automation Help.

2.6.8.2.15.1 Module replacement

The SafeLOGIC controller recognizes on its own when safe modules have been replaced. Following a module replacement, the complete system (SafeLOGIC, SafeLOGIC-X system components, openSAFETY) automatically ensures that the module operates again using the correct parameters and that incompatible modules are rejected. Nevertheless, the following errors are still possible after a module replacement:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

<u>Terminals swapped between several modules</u>

To determine whether terminals have been swapped between several modules, the user must test the safety function by performing a wiring test.

Danger!

The user must ensure that the wiring test can detect when terminals have been swapped.

Be sure to validate the entire safety function!

Wiring errors

A wiring error can occur if the wiring between the sensor or actuator and the X20 terminal is disconnected. To detect this sort of error in the wiring, the user must test the safety function by performing a wiring test.

Danger!

The user must make sure that the wiring test can detect wiring errors.

Be sure to validate the entire safety function!

SafeIO modules swapped with each other

Errors in the standard application can cause SafeIO modules to become swapped, which appears identical to a module replacement to the SafeLOGIC controller. To detect this error, the user must confirm the number of replaced modules. This links the number of modules replaced by the user and the replacements recognized by the system so that any additional replacements can be detected.

The user is informed of the number of detected module replacements via the MXCHG status. In the process, the module identifiers (UDIDs) on the SafeKEY or in the safety section of the CompactFlash card are compared to the UDIDs of the modules in the network.

If there are 1, 2, 3 or 4 different UDIDs, the user is provided information about the exact number of differences. The user must then check whether the number of replaced modules recognized by the SafeLOGIC controller corresponds to the actual number of replaced modules. If the values are the same, the user must confirm the number and perform a wiring test. This wiring test can be limited specifically to the modules that have been replaced.

If there are more than 4 different UDIDs, a standard message is provided indicating that there are differences on more than 4 modules. In this case, the user must perform a comprehensive wiring test for all modules.

If the number of modules indicated and the actual number of replaced modules do not match, the user must confirm the number of replacements determined by the SafeLOGIC controller and perform a comprehensive wire test for all modules.

Danger!

Be sure to validate the entire safety function!

Replacing an individual module

If only one module was replaced (MXCHG status indicates 1 module was replaced) and the wiring was not changed, the user can skip the wiring test because in this case the following errors can be ruled out:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

Danger!

The wiring test can only be excluded if no additional changes are made when replacing an individual module (e.g. unplugging terminals, removing the wiring, etc.).

Confirming a module replacement

To confirm the number of the replaced modules, the correct number of modules must be selected:

- 1 - One module replaced
- 2 - Two modules replaced
- 3 - Three modules replaced
- 4 - Four modules replaced
- n - Five or more modules replaced

The replacement can be confirmed and the accompanying wiring test can be limited to the replaced modules when up to four modules are replaced. When more than four modules are replaced, a comprehensive wiring test must be performed for all modules.

Following confirmation of the module replacement, the SafeLOGIC controller immediately commences a module scan.

Danger!

The user must ensure that the wiring test can detect a wiring error or when terminals have been swapped.

Be sure to validate the entire safety function!

2.6.8.2.15.2 Other errors in module configuration

The aforementioned differences are limited exclusively to module replacements. An error – "Missing module" status – is reported if a device is missing (except when the device is defined as optional), has an incorrect hardware code or other problems are present on the module (e.g. incorrect parameters that may not be changed by the SafeLOGIC controller). This status is only indicated if a module or firmware replacement is not being indicated. This status cannot be acknowledged.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.8.2.15.3 Acknowledging a firmware modification

A change to the firmware is indicated by the FW-ACKN status and must be confirmed using the FW-ACKN action. A firmware modification must always be concluded with full functional testing.

Danger!

Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions and trained in the procedure of exchanging firmware.

Be sure to validate the entire safety function!

Danger!

Only use firmware versions listed in the FS certificates for B&R safety technology. These FS certificates are available for download from the B&R website at http://www.br-automation.com.

2.6.8.2.15.4 Triggering a module scan

A module scan determines if all configured modules are present in the application and if they correspond to the project configuration. The module scan runs automatically but at large time intervals. To minimize the time it takes for the SafeLOGIC controller to recognize a newly replaced module, this function can also be triggered manually by the user. The result of the scan is described in the following sections:

- "Module replacement"
- "Other errors in module configuration"
- "Acknowledging a firmware modification"

The process itself is started using the SCAN function and indicated using the "Scanning" status. The results are reported after the "Scanning" status is completed (e.g. three modules replaced).

2.6.8.2.15.5 SafeKEY or safety section of the CompactFlash card

The following data is stored on the SafeKEY (X20SL8xxx series) or in the safety section of the CompactFlash card (X20SLXxxx series):

- SafeDESIGNER application (application and all SafeDESIGNER parameters for the modules)
- Configuration (unique module code (UDID), firmware versions of modules)
- Subsequently loadable data elements (machine options, tables, etc.)

Size of the SafeDESIGNER application on the SafeKEY

The size of the current application on the SafeKEY is calculated by SafeDESIGNER during compilation and displayed in the message window (e.g. "The safety application uses 0.688 MB (11 sectors) memory.").

Notes:

- The output only takes the size of the SafeDESIGNER application into account. Space on the data storage device used by firmware or subsequently loadable data (tables, machine options, etc.) is not taken into account.
- If the online project comparison is not needed (see Automation Help → SafeDESIGNER), the download size of the application can be reduced by disabling the following communication setting: Online → Communication settings → Download project source to SL.

Removing a SafeKEY (X20SL8xxx series only)

Removing a SafeKEY always results in a change to BOOT mode, and the safety application is completely shut down.

Information:

Removing a SafeKEY during operation causes the SafeLOGIC controller to be restarted and all safety-related actuators to be cut off.

Removing a SafeKEY during operation can destroy the data on the SafeKEY.

Removing a SafeKEY during operation must therefore be avoided at all cost.

The "Backing up the SafeKEY" sequence is not affected by this general rule.

Acknowledging a SafeKEY replacement

Replacing a SafeKEY or replacing a CompactFlash card with a CompactFlash that has a modified safety section is indicated by the "FW-ACKN" status and must be acknowledged with the SK-XCHG function. Complete functional testing is then required.

Information:

A SafeKEY replacement can only be acknowledged if a valid SafeDESIGNER project has already been transferred to the SafeKEY or CompactFlash card.

Danger!

Replacing a SafeKEY or CompactFlash card will enable the safety application stored on the SafeKEY or CompactFlash card. Always check the project CRC and date that the safety application project was saved on the SafeKEY or CompactFlash card.

Danger!

Be sure to validate the entire safety function!

Changing the application on the SafeLOGIC controller by replacing the SafeKEY (X20SL8xxx series only)

All relevant configuration data and all application data and parameters are stored on the SafeKEY. In order to transfer the previous configuration data to a new SafeKEY when changing the application, the following sequence must be carried out.

- Set the selector switch to the SK-COPY position.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- The SafeKEY configuration data is saved on the SafeLOGIC controller. The SKEY LED blinks with each access.
- The FW-ACKN LED will flash after the copying procedure. This SafeKEY can now be replaced by the SafeKEY with the new application. 30 seconds are provided to do this. The FW-ACKN LED blink frequency increases after 20 seconds to signal the end of the replacement phase.
- The acknowledgment button must be pressed again after the new SafeKEY has been inserted. The selector switch remains on the setting SK-COPY.
- The internal, temporarily saved configuration data is saved on the new SafeKEY. A reset is then triggered automatically, and the data from the new SafeKEY is applied.
- Following the reset, the SafeKEY replacement must be acknowledged. To do this, move the selector switch to the setting SK-XCHG.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- Perform complete functional testing.

Information:

If the new SafeKEY is not acknowledged after 30 seconds, the function will end, i.e. if the function is triggered inadvertently, the copy function ends automatically after 30 seconds. If a SafeKEY is not inserted after 30 seconds, the SafeLOGIC controller switches to BOOT mode.

Danger!

This procedure enables the safety application stored on the new SafeKEY. Always check the project CRC and date that the safety application project was saved on the SafeKEY.

Danger!

Be sure to validate the entire safety function!

Information:

This sequence can also be used to create a SafeKEY backup using a second SafeKEY with an identical safety application. After executing the sequence, two identical SafeKEYs are available (backup copy).

Information:

Only data relevant to the machine is copied, not all of the safety application data.

2.6.8.2.15.6 Replacing a SafeLOGIC controller

Replacing a SafeLOGIC controller involves the same procedures as a normal module replacement. When replacing a SafeLOGIC controller, the SafeKEY from the SafeLOGIC controller being replaced must be kept in order to avoid activating an old safety-related application.

Danger!

Be sure to validate the entire safety function!

2.6.8.2.15.7 Authorization (X20SL8xxx series only)

The following functions can be blocked by the standard CPU:

- Confirming a module replacement
- Acknowledging a firmware modification
- Acknowledging a SafeKEY replacement
- Backing up the SafeKEY
- Replacing a SafeLOGIC controller

This allows actions to be executed in accordance with an application-specific user concept. This option is not possible from a safety perspective, however, since these functions are executed on the standard CPU.

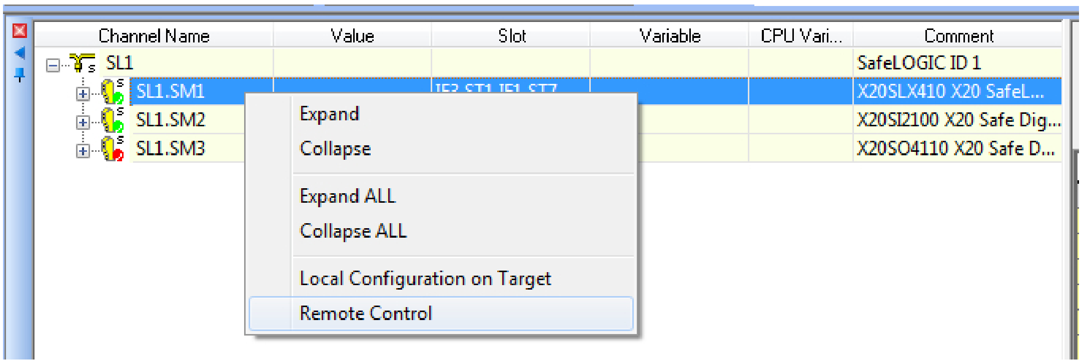
The following table lists the associated objects in Index "0x2402" that can be accessed using the POWERLINK library.

Index:Subindex	Object description	Data type	Access	Value	Description
0x2402:0x00	NumberOfEntries	USINT	R	0x22	Number of entries in this index
0x2402:0x01	EnableAuthorization	UDINT	RW	"AENA", 0x41454E41	Enables authorization
				"ADIS", 0x41444953	Disables authorization
0x2402:0x04	EnableModuleExchange	UDINT	RW	"UDID", 0x55444944	Provides authorization to acknowledge a module replacement
				All other values	Does not provide authorization to acknowledge a module replacement
0x2402:0x05	EnableFWMismatch	UDINT	RW	"FWAC", 0x46574143	Provides authorization to acknowledge a firmware replacement
				All other values	Does not provide authorization to acknowledge a firmware replacement
0x2402:0x06	EnableSKeyExchange	UDINT	RW	"SKEY", 0x534B4559	Provides authorization to acknowledge a SafeKEY replacement
				All other values	Does not provide authorization to acknowledge a SafeKEY replacement

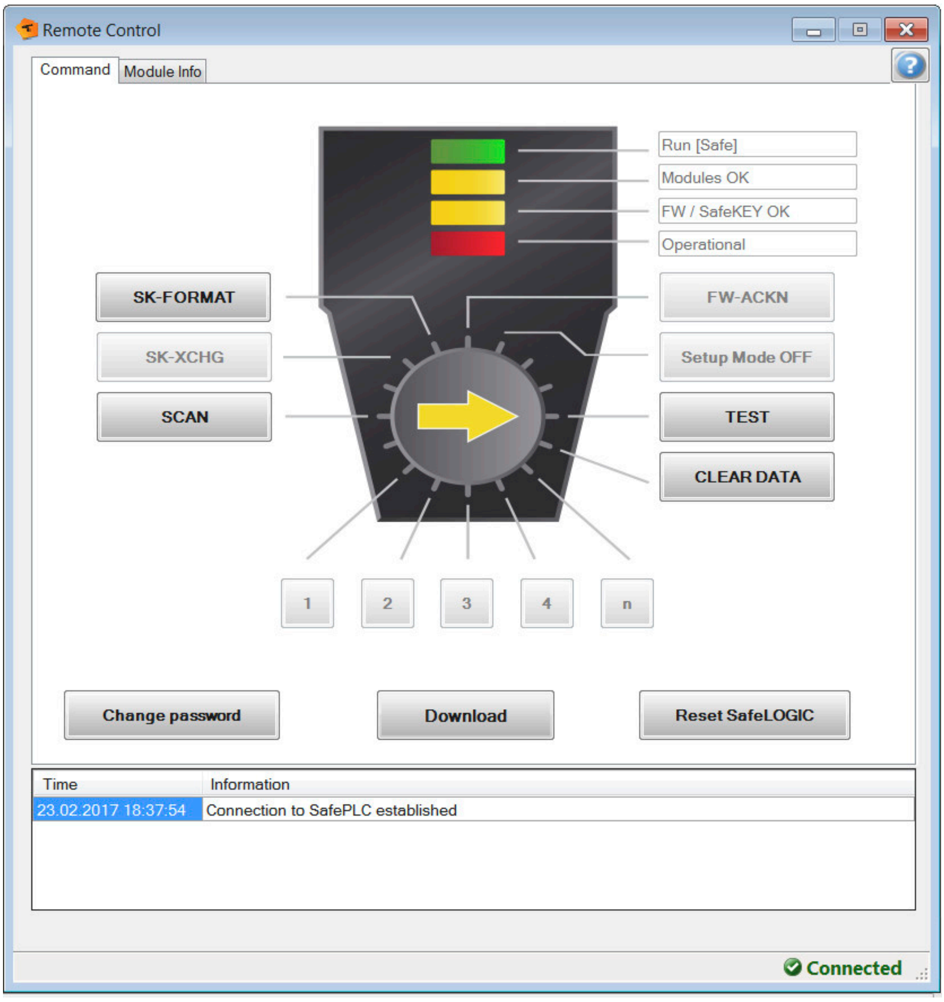
User requests made to the SafeLOGIC controller that are not authorized by the CPU are indicated by a steadily lit ENTER LED.

2.6.8.2.16 Quick start

With the X20SLXxxx series, it is not possible to perform a direct download via the SafePLC window in SafeDESIGNER. The application must be downloaded using the remote interface instead. The remote interface can be accessed from the Safety View.



The password must be entered in order to gain access (or a new password defined at the beginning). Startup must be carried out via the remote interface, as is the case with the X20SL8xxx series via its operating elements.



In addition, the AsSafety library can also be used for startup (see section "Operation via the AsSafety library").

Information:

The possibilities listed above are also available for the X20SL8xxx series starting with Safety Release 1.7.

2.6.8.2.16.1 Download mechanism

Downloading takes place in 2 steps – first to the CompactFlash card and then to the SafeLOGIC-X controller. "Download completed" indicates that the data has been applied during a download to the CompactFlash card.

> ## Information:
> **The "Download completed" window in SafeDESIGNER is displayed already after downloading to the CompactFlash card. The download to the SafeLOGIC-X controller takes place afterward; it is completed by restarting the SafeLOGIC-X controller.**

2.6.8.2.16.2 Visualization

In order to carry out maintenance scenarios, an HMI application must be created using library "AsSafety".

> ## Information:
> **For details, see Solutions -> Technology Solutions in Automation Help.**

2.6.8.2.16.3 Possible data loss

Data for the SafeLOGIC-X controller is stored on the CompactFlash card.

> ## Information:
> **Note that this data can be lost when reformatting the CompactFlash card, for example.**

2.6.8.2.16.4 Necessary resources

Automation Runtime resources are necessary for the safety system.

> ## Information:
> **When converting from a SafeLOGIC controller to a SafeLOGIC-X controller, note that more Automation Runtime resources are needed for the SafeLOGIC-X controller.**

2.6.8.2.17 Software functions

2.6.8.2.17.1 Operation via the AsSafety library

Information about using library "AsSafety" is available under Programming -> Libraries -> Safety -> AsSafety in Automation Help.

2.6.8.2.17.2 Automatic acknowledgment

As specified in previous chapters, automatic acknowledgment is usually not permitted. Provided that the user implements appropriate quality assurance measures and/or constraints, it is nevertheless possible to deviate from this to permit the following automatic acknowledgment.

> **Danger!**
>
> **The automatic acknowledgment of SafeLOGIC controller acknowledgment requests under improper circumstances is not permitted and can lead to dangerous states.**
>
> **It is the sole responsibility of the user to assess the requirements of the safety application in order to determine whether additional measures are necessary.**

"SafeKEY exchange" acknowledgment request

The SafeDESIGNER application and machine option are saved in the safety section of the CompactFlash card (X20SLXxxx series) or on the SafeKEY (X20SL8xxx series). Replacing the CompactFlash card or SafeKEY may result in the unintended exchange of this data. The "SafeKEY exchange" acknowledgment request is meant to prevent this unintentional exchange of data.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards or SafeKEYs:

- The SafeDESIGNER application must be completely validated on a reference machine.
- The machine options file must be completely validated on a reference machine.
- Sufficient measures must be implemented to prevent the SafeDESIGNER application or machine options file from being mixed up across different machine types.
- No test versions of the SafeDESIGNER application or machine options file are permitted.

Under the conditions specified, an automated update of the SafeDESIGNER application or machine options file is permitted to be implemented on the SafeLOGIC/SafeLOGIC-X controller.

"Firmware acknowledge" acknowledgment request

B&R Automation Runtime sees to it independently that the firmware versions stored on the CompactFlash card are transferred to the automation components in the network. This mechanism may cause other firmware versions to be enabled in the system than those that were active when the SafeDESIGNER application was validated. A change to the firmware of the safety modules always requires revalidation of the SafeDESIGNER application. The "Firmware acknowledge" acknowledgment request is meant to prevent an unintentional exchange of firmware versions.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards:

- The firmware files installed on the safety modules must be completely validated together with the SafeDESIGNER application on a reference machine.

"UDID mismatch" acknowledgment request

The "UDID mismatch" request occurs in the following situations:

- When modules are exchanged by the user (e.g. during a service call). In this case, it is possible for the connection lines to be mixed up.
- When errors occur in the standard application that lead to a mix-up of modules.

To rule out these mix-ups, a wiring test must be performed after a "UDID mismatch" request is acknowledged.

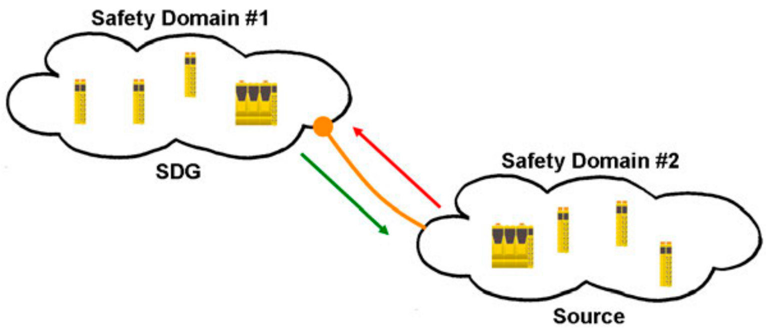
The "UDID mismatch" acknowledgment request is meant to prevent the unintentional mix-up of signals caused by exchanging a module or errors in the standard application.

- Service personnel are to be informed that the mandatory wiring test when exchanging modules must be performed independently of the automatic acknowledgment of the "UDID mismatch" request.
- It is not permitted to use more than 1 module per module type in the Automation Studio application or SafeDESIGNER application.

If the last requirement cannot be met, a "UDID mismatch" acknowledgment request is not permitted to be acknowledged automatically since it would not cover the possible mix-up of signals caused by errors in the standard application.

2.6.8.2.17.3 SafeLOGIC to SafeLOGIC communication

The safety system makes it possible to exchange safety-related information between two safety controllers (SafeLOGIC). SafeLOGIC to SafeLOGIC communication can be used to implement functions such as a global E-stop across a machine network or if a dependency exists between the safety applications on two or more machines. This makes it possible to establish a central collection point for safety information that will be responsible for distributing current values to all relevant locations.



Information:

The safety domain number is taken from the SafeLOGIC ID. In order to use SafeLOGIC to SafeLOGIC communication, the SafeLOGIC IDs must be unique. This uniqueness should be taken into consideration from the very beginning.

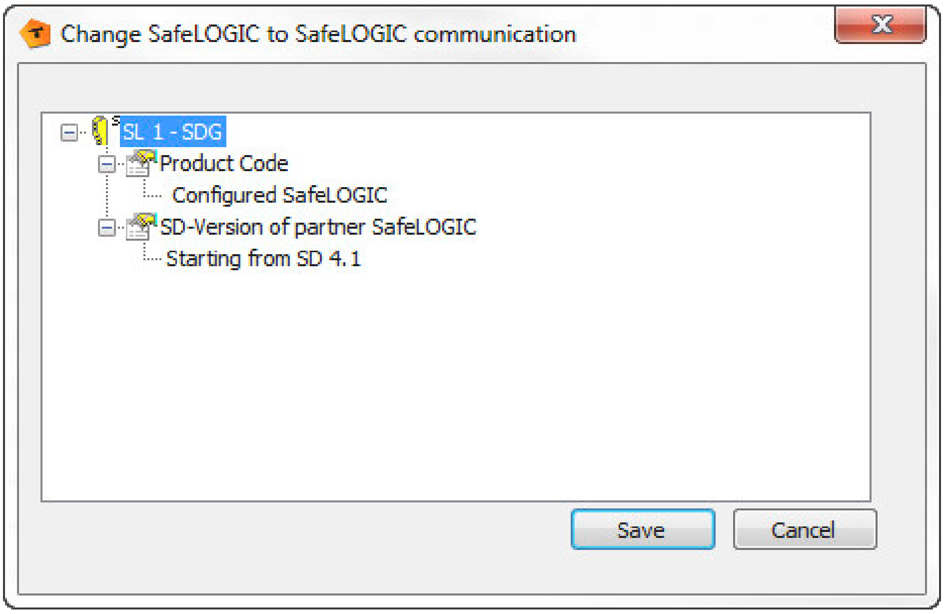
To help with this, a SafeLOGIC controller provides a Safety Domain Gateway (SDG) that can be used to connect additional SafeLOGIC controllers (source controllers). This gateway functionality ensures communication between several safety domains. The connection between source SafeLOGIC controllers and SDG SafeLOGIC controller is indicated in the source SafeLOGIC controller's project as an additional safety module that provides additional communication channels. An SDG SL controller itself can also be used as a source controller and connected to another SDG SL controller. This can be done to achieve cascading communication relationships.

A source SL controller can also be connected several times to the same SDG SL controller, just as it is possible for the source SL controller to communicate with several SDG SL controllers. This results in several ways for SafeLOGIC to SafeLOGIC communication to take place.

System requirements

The following points must be taken into account for safe data exchange between at least 2 SafeLOGIC controllers:

- SafeDESIGNER <4.1: The same SafeDESIGNER versions must be used.
- SafeDESIGNER 4.1 to 4.2.1: The SafeDESIGNER versions must be within this version range.
- SafeDESIGNER 4.2.2 and later: SafeDESIGNER 3.0 or later is permitted to be used.
  The corresponding parameters in the following dialog box must be configured in order to establish a connection to the remote station.



  - Configured SafeLOGIC: Remote station with which communication takes place (e.g. X20SL8100)
  - SD-Version of partner SafeLOGIC: Version with which the application on the remote station was created

Possibilities

The system supports various communication options. The corresponding communication type is defined via parameters in Automation Studio (see "Group: SafeLOGIC to SafeLOGIC communication").

Fixed communication

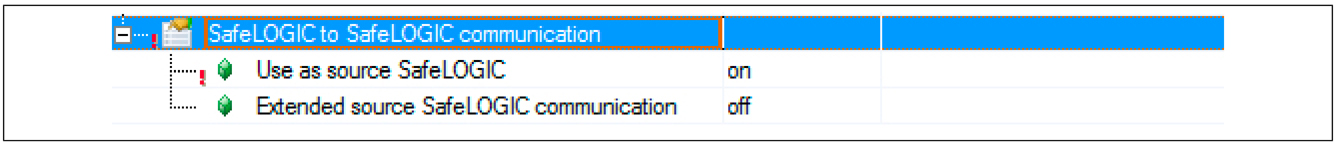
- 8 BOOL channels (1 byte) per communication direction
- One source SL controller can only communicate with one SDG SL controller
- No "any to any" constellation
- Cannot be used with SafeLOGIC-X

Extended communication (Release 1.4 or later and Automation Studio 3.0.90 or later)

- Freely configurable communication channels
- Limited to 16 channels (where 8 BOOLs count as 1 channel; other data types are calculated 1:1).
- One source SL controller can communicate with several SDG SL controllers
- "Any to any" constellation possible

Configuration in Automation Studio

To use SafeLOGIC to SafeLOGIC communication, a SafeLOGIC controller first needs to be configured as a source SL controller. This is done in the I/O configuration.

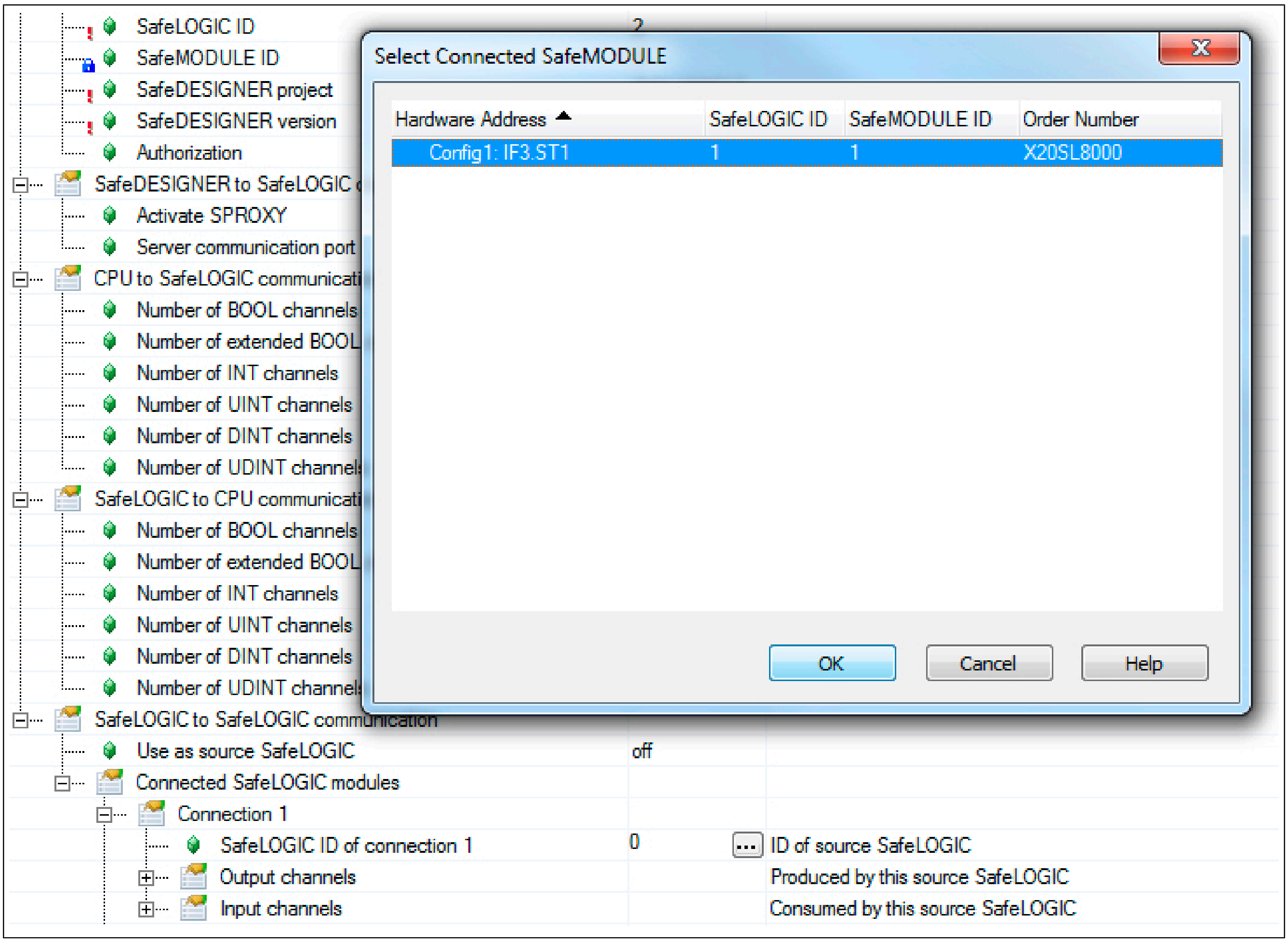


After the "Use as source SafeLOGIC" parameter has been enabled, it is possible to define the type of SafeLOGIC to SafeLOGIC communication as fixed or extended. If the "Extended source SafeLOGIC communication" parameter is not enabled, then fixed communication is used.

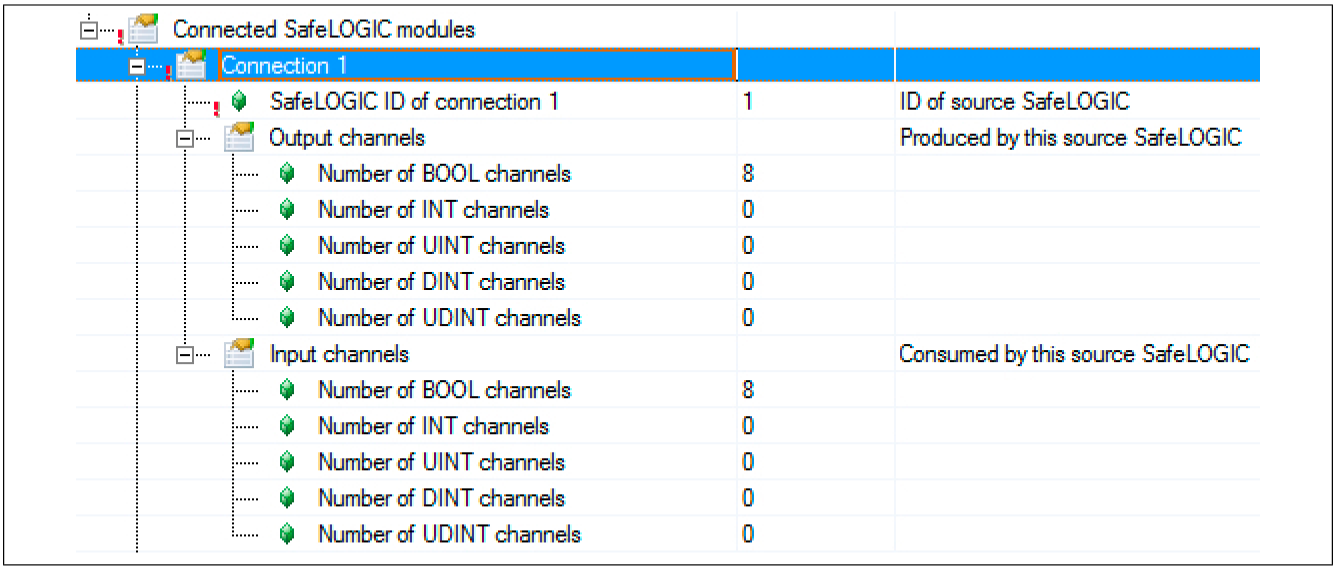
Information:

Changing the type of communication (fixed or extended) at a later time may result in channel overlaps in SafeDESIGNER; the communication channels must therefore be reconnected.

The source SL controller is then connected to the SDG SL controller in the next step. This is done using the connection points in Automation Studio under the I/O configuration of a SafeLOGIC controller (X20SL80x1 and X20SL81xx). Each SafeLOGIC ID (safety domain) is specified from the connection sections using the wizard in Automation Studio.



The necessary communication channels must be defined under each connection. With fixed communication, they are limited to 8 BOOL channels in each direction.



If SafeLOGIC to SafeLOGIC communication should be established between existing or separate Automation Studio projects, several things must be taken into consideration:

- SafeLOGIC IDs must be unique.
- A dummy configuration that includes all safety components must be created on the peer station.
- The dummy configuration must match the real configuration - the SafeMODULE IDs are important here.
- If the projects have multiple iCNs (intelligent controlled nodes), all iCNs must always be taken into account in the iCN project.

Display in SafeDESIGNER

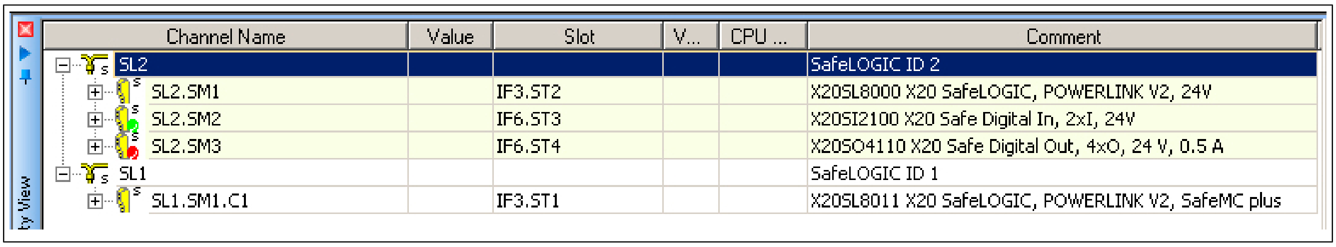
The communication channels are also shown in the SafeDESIGNER project for the respective SafeLOGIC controller (source or SDG).

Danger!

All of the communication channels being used in the project must be mapped in both SafeDESIGNER projects using the same variable names. Channels and variable names are used to calculate a checksum that is then checked at runtime. If the checksum does not match, then the system issues a corresponding logger message in the Safety Logger and communication does not take place.

SafeDESIGNER project – Source SL controller

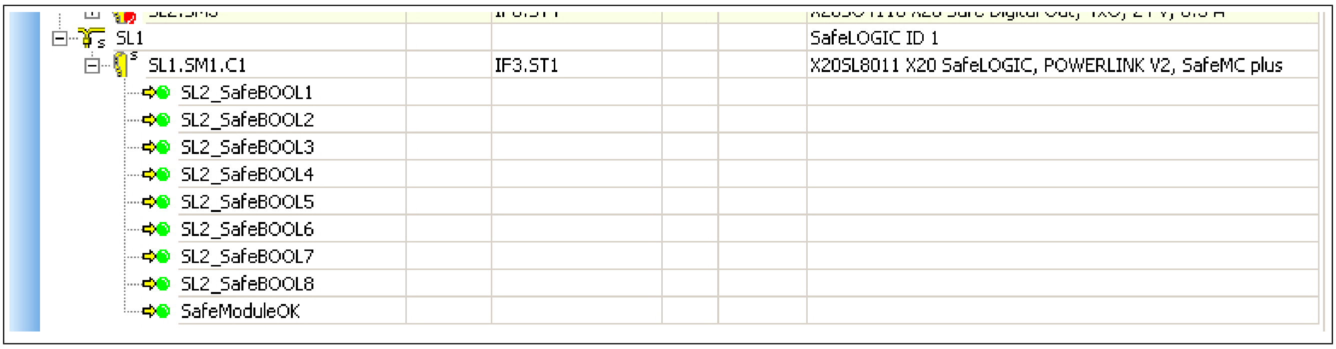
In the source SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.



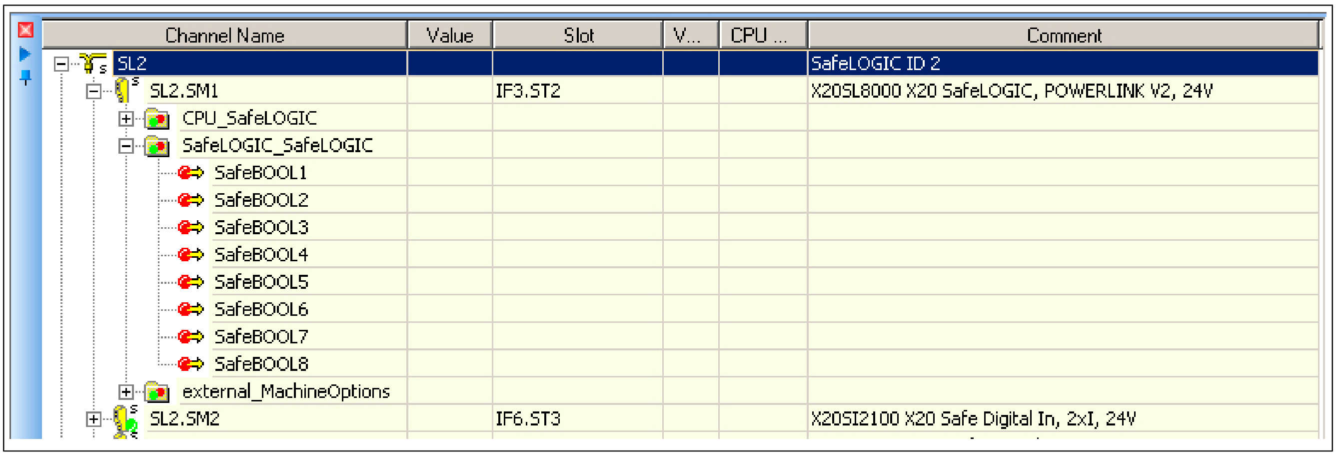
If this module is selected, it is possible to configure its safety-related parameters (see section "Parameters for connection - Release 1.10 and later").

Fixed communication

The input channels sent from the SDG SL controller to the source SL controller and bit information about the status of the connection are listed under the module.

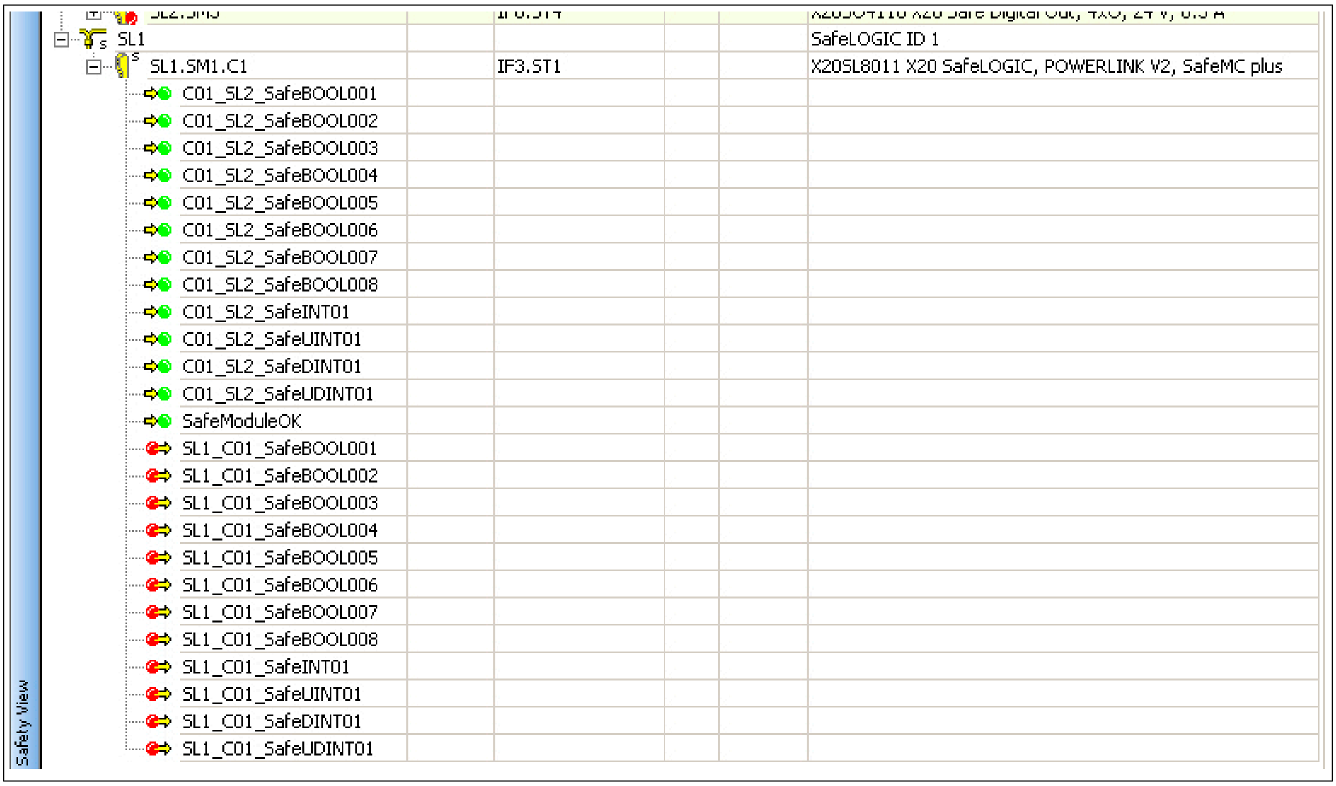


The output channels sent from the source SL controller to the SDG SL controller are listed under the actual SL controller in the project in section "SafeLOGIC_SafeLOGIC".



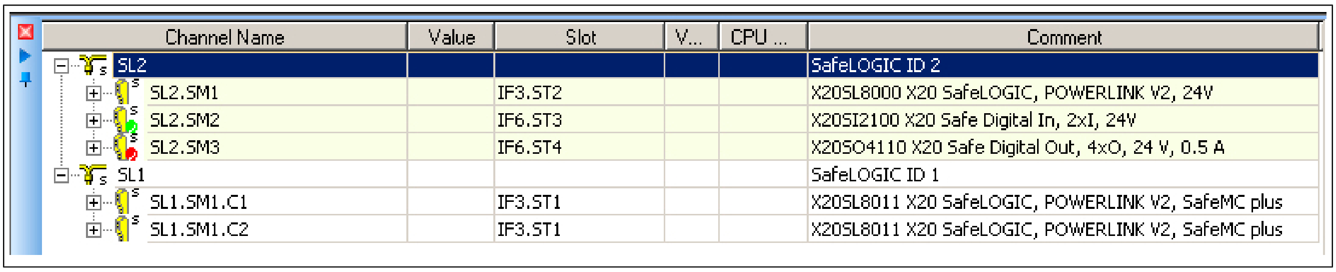
Extended communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

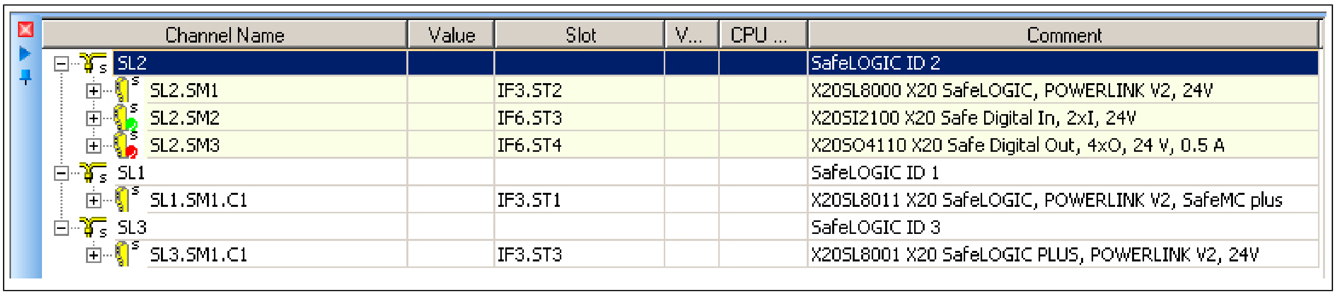


Additional connection

If the source SL controller should be connected once again to the same SDG SL controller, an additional module underneath the same node is available with the necessary parameters and communication channels.

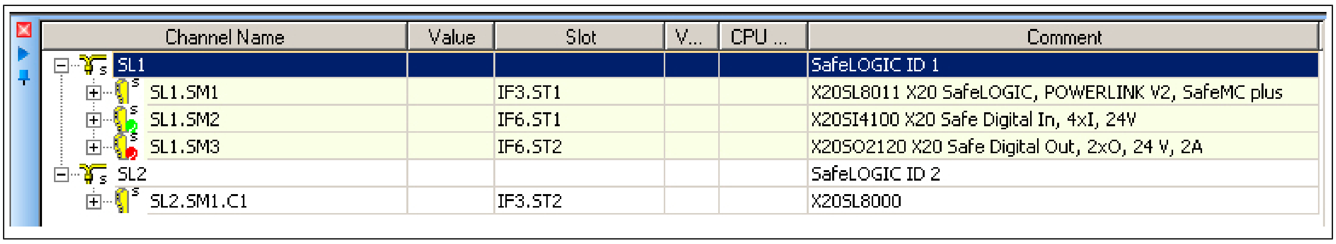


If the source SL controller should be connected to another SDG SL controller, an additional node for the safety domain as well as a module with the necessary parameters and communication channels is available.



SafeDESIGNER project – SDG SL controller

In the SDG SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.

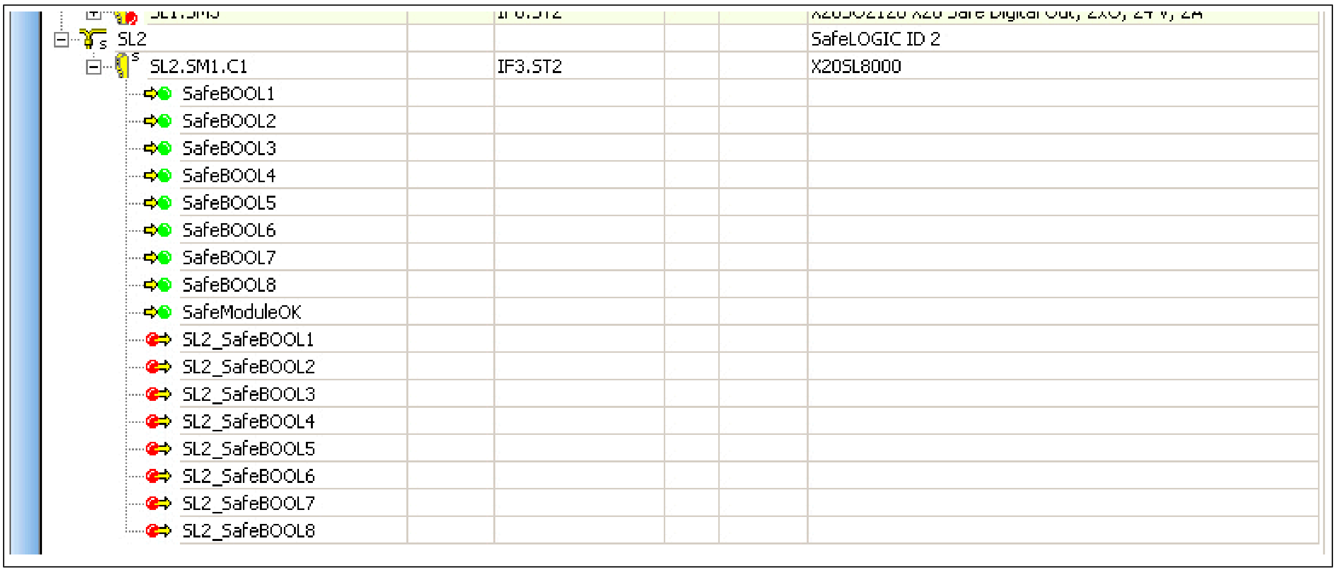


Information:

No connection parameters are available in the SDG SL controller's project. They must be configured in the source SL controller's project.

Fixed communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

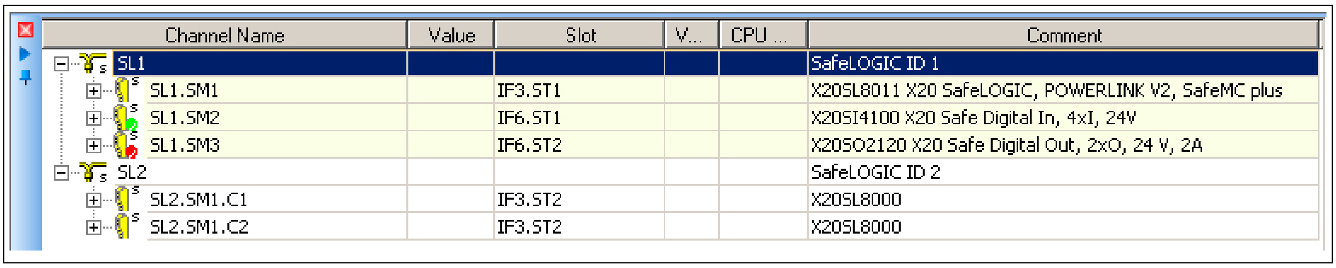


Extended communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

Additional connection

If the source SL controller should be connected once again to the SDG SL controller, an additional module underneath the same node is available with the necessary communication channels.



Parameters for connection - up to Release 1.9

Safety Release 1.4 or higher:

Cycle time parameters are also available for communication in order to define the "Worst_Case_Response_Time_us". As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Parameter	Value
Basic	
Min_required_FW_Rev	Basic Release
Optional	No
External_UDID	No
Safety_Response_Time	
Synchronous_Network_Only	Yes
Max_SDG_Powerlink_CycleTime_us	5000
Max_Powerlink_CycleTime_us	5000
Max_CPU_CrossLinkTask_CycleTime_us	5000
Min_SDG_Powerlink_CycleTime_us	200
Min_Powerlink_CycleTime_us	200
Min_CPU_CrossLinkTask_CycleTime_us	0
Worst_Case_Response_Time_us	100000
Max_SDG_Cycle_Time_us	5000
Min_SDG_Cycle_Time_us	1600
Slow_Connection	No

Group: Basic

Parameter	Description	Default value	Unit
Min_required_FW_Rev	This parameter is reserved for future functional expansions.	Basic Release	-
Optional	This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.	No	-

Parameter value	Description
No	This module is mandatory for the application. The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No". After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.
Yes	The module is not required for the application. The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller. After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.
Startup	This module is optional. The system determines how the module will proceed during startup. If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set. If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.
Not_Present (Release 1.9 and later)	The module is not required for the application. The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present. Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior. After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.

Parameter	Description	Default value	Unit
External_UDID	This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.	No	-

Parameter value	Description
Yes-ATTENTION	The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.
No	The UDID is specified by a teach-in procedure during startup.

Table 81: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr>
<tr><td>No</td><td>No requirement for synchronization of the networks.</td></tr>
</table></td></tr>
<tr><td>Max_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the maximum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
• Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the minimum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
• Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 12,500,000 µs (corresponds to 3 ms to 12.5 s)
Note:
Keep parameter "Slow_Connection" in mind when entering large values here!</td><td>100000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
<tr><td>Max_SDG_Cycle_Time_us</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Cycle_Time_us</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
<tr><td rowspan="2">Slow_Connection</td><td>This parameter specifies whether this connection is a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr>
<tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr>
</table></td></tr>
</table>

Table 82: SafeDESIGNER parameters: Safety_Response_Time

Information:

Parameter "CPU_CrossLinkTask_CycleTime_us" is needed if the source SL and SDG SL controllers are in different networks or located on different controllers. If this is not the case, the minimum and maximum value must be set to "0".
For this parameter, the entire connection distance between the controllers must be taken into account – including copy times between the interfaces involved.

Information:

Parameter "Slow_Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow_Connection = Yes").

Parameters for connection - Release 1.10 and later

Cycle time parameters are also available for communication in order to define the maximum data transmission time. As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Materialnummer: **X20SL8100**
Description: **X20 SafeLOGIC, POWERLINK V2, 24V, univ.**
SafeMODULE ID: **3**
Import file: -

Parameter	Value	Unit
Basic		
Min required FW Rev	Basic Release	
Optional	No	
External UDID	No	
Safety Response Time		
Synchronous Network Only	Yes	
Safe Data Duration	20000	us
Additional Tolerated Packed Loss	0	packets
Slow Connection	No	
Node Guarding Lifetime	5	iterations
Max SDG Cycle Time	5000	us
Min SDG Cycle Time	1600	us

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 83: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td rowspan="2">Slow Connection</td><td>This parameter specifies whether this connection is classified as a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr>
<tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr>
</table></td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
<tr><td>Max SDG Cycletime</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min SDG Cycletime</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
</table>

Table 84: SafeDESIGNER parameters: Safety Response Time

Information:

Parameter "Slow Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow Connection = Yes").

2.6.8.2.17.4 Setup mode

Setup mode supports the user during commissioning.

Setup mode is supported in hardware upgrade 1.10.2.x and later.
Automation Runtime B4.26 or higher is required to use setup mode.

Active setup mode is indicated by both the FAILSAFE LED (X20SL81xx series) or SE LED (X20SLXxxx series) as well as an entry in the logbook.

When setup mode is active, acknowledgment requests "SafeKEY exchange", "Firmware acknowledge" and "UDID mismatch" are no longer necessary.

Setup mode can be enabled and disabled using the operating elements of the "Remote Control" in SafeDESIGNER (X20SL81xx and X20SLXxxx series) or using the selector switch and acknowledgment button (X20SL81xx series).

Danger!

Setup mode is only permitted to be enabled during the commissioning of the machine/system.
Setup mode must be disabled during operation.

Danger!

After setup mode is ended, functional testing including a wiring test must be carried out.
If a SafeKEY or SafeLOGIC controller is replaced while setup mode is active, then setup mode will be disabled.
Functional testing must also be carried out in this case.

Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions.

Be sure to validate the entire safety function!

2.6.8.3 X20(c)SLXxxx

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 85: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 86: Organization of general notices

2.6.8.3.1 General information

The modules are equipped with SafeLOGIC functionality that allows them to safely execute applications designed in SafeDESIGNER. The modules can be used in safety applications up to PL e or SIL 3.

The SafeLOGIC controller coordinates the safety-related communication of all modules involved in the application. In this context, the SafeLOGIC controller also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided in both variants, however.
In addition, SafeLOGIC-X products have the same I/O properties as the associated SafeIO products.

- openSAFETY manager for up to 10 / 20 / 100 / 280 SafeNODES
- Flexibly programmable using Automation Studio / SafeDESIGNER
- Innovative management of safe machine options (SafeOPTION)
- Parameter and configuration management

2.6.8.3.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

SafeLOGIC function

The module is equipped with SafeLOGIC functionality that allows it to safely execute applications designed in SafeDESIGNER. The module can be used in safety-related applications up to PL e or SIL 3.

In addition, the module coordinates the safety-related communication of all modules involved in the application. In this context, the module also monitors the configuration of these modules and autonomously carries out parameter downloads to the modules if necessary. This guarantees a consistent and correct module configuration in the network from a safety point of view in all scenarios involving module replacement and service. For SafeLOGIC products, these services are executed by the SafeLOGIC controller. For SafeLOGIC-X products, these services are executed on the standard CPU in interaction with Automation Runtime. The safety-related characteristics up to PL e or SIL 3 for applications are provided with both variants, however.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.8.3.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.8.3.2 Overview

Module	X20SLX402	X20SLX806	X20SLX842
Safe digital inputs			
Number of inputs	4	8	8
Nominal voltage	24 VDC		
Input filter Hardware Software	≤150 µs Configurable between 0 and 500 ms		
Input circuit	Sink		
Pulse outputs			
Design	Push-Pull		
Switching voltage	I/O power supply minus residual voltage		
Safe digital HS-LS outputs			
Number of outputs	-		4
Nominal voltage	-		24 VDC
Nominal output current	-		3 A
Total nominal current	-		10 A [1)]
Output protection	-		Thermal short circuit shut-down, integrated protection for switching inductive loads
Safe digital HS-HS outputs			
Number of outputs	2	6	2
Nominal voltage	24 VDC		
Nominal output current	0.2 A		50 mA
Total nominal current	0.4 A	1.2 A	100 mA
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads		

Table 87: Digital mixed modules

1) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.

2.6.8.3.3 Order data

X20SLX402 X20SLX806 X20SLX842

Model number	Short description
	Intelligent programmable modules
X20SLX402	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20cSLX402	X20 safe digital mixed module, coated, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SLX806	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SLX842	X20 safe digital mixed module, safety controller, openSAFETY, 11 openSAFETY nodes, 4 SafeMOTION axes, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs
	Required accessories
	Bus modules
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 88: X20SLX402, X20cSLX402, X20SLX806, X20SLX842 - Order data

2.6.8.3.4 Technical data

Model number	X20SLX402	X20cSLX402	X20SLX806	X20SLX842
Short description				
I/O module	4 safe digital inputs, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs, SafeLOGIC-X technology		8 safe digital inputs, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs, SafeLOGIC-X technology	8 safe digital inputs, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs, SafeLOGIC-X technology
General information				
B&R ID code	0xE7EA	0xF210	0xE758	0xE7EB
System requirements				
Automation Studio	4.2 or later			
Automation Runtime	B4.25 or later			
SafeDESIGNER	4.2.1 or later			
Safety Release	1.10 or later			
Status indicators	I/O function per channel, operating state, module status			
Diagnostics				
Module run/error	Yes, using status LED and software			
Outputs	Yes, using status LED and software			
Inputs	Yes, using status LED and software			
Blackout mode				
Scope	Module			
Function	Programmable			
Standalone mode	Yes			
Max. I/O cycle time	1 ms			
Power consumption				
Bus	0.4 W			
Internal I/O	2.5 W			
Electrical isolation				
Channel - Bus	Yes			
Channel - Channel	No			
Certifications				
CE	Yes			
EAC	Yes			
UL	cULus E115267 Industrial control equipment			
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	In preparation	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	In preparation			
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013			
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3			
Functional safety	EN 50156-1:2004	EN 50156-1 in preparation	EN 50156-1:2004	
Safety characteristics				
EN ISO 13849-1:2015				
MTTFD	2500 years			
Mission time	Max. 20 years			
IEC 61508:2010, IEC 61511:2004, EN 62061:2013				
PFH / PFH_d				
Module	$<1*10^{-10}$			
openSAFETY wired	Negligible			
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour			
PFD	$<2*10^{-5}$			
Proof test interval (PT)	20 years			

Table 89: X20SLX402, X20cSLX402, X20SLX806, X20SLX842 - Technical data

Model number	X20SLX402	X20cSLX402	X20SLX806	X20SLX842
Safe digital inputs				
EN ISO 13849-1:2015				
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1]			
PL	PL e			
DC	>94%			
IEC 61508:2010, IEC 61511:2004, EN 62061:2013				
SIL CL	SIL 3			
SFF	>90%			
Safe digital outputs				
EN ISO 13849-1:2015				
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" [1]			
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" [1]			
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" [1]			
IEC 61508:2010, IEC 61511:2004, EN 62061:2013				
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" [1]			
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" [1]			
Functionality				
Communication with each other	Communication only possible with SafeLOGIC controller X20(c)SL81xx Max. 1 active SafeLOGIC-X controller per standard X20(c)CPxxxx CPU [2]			
Support for machine options				
BOOL	64			
INT	-			
UINT	-			
DINT	-			
UDINT	-			
SafeMOTION support	Yes			
Max. number of SafeMOTION axes	4, depends on the data width of the modules used			
Timing precision	Time * 0.05 + Cycle time of the safety application			
Max. number of openSAFETY nodes	10, depends on the data width of the modules used			
Data exchange between CPU and SL				
Max. total data width for each direction	8 bytes			
Max. number of data points for each direction				
BOOL	64			
INT	4			
UINT	4			
DINT	2			
UDINT	2			
Data exchange between SL and SL				
Max. total number of data points for each direction [3]	2			
Max. number of data points for each direction				
BOOL	16			
INT	2			
UINT	2			
DINT	2			
UDINT	2			
Limit values for SafeDESIGNER application				
Max. resources available for SafeDESIGNER info window entries [4]				
FB instances	256			
Marker memory	5120 bytes (0x1400)			
Stack memory	2048 bytes			
Memory for safe input data	128 bytes, 68 bytes of which are usable for modules			
Memory for safe output data	64 bytes			
Memory for standard input data	64 bytes			
Memory for standard output data	64 bytes			
Marker count	256			

Table 89: X20SLX402, X20cSLX402, X20SLX806, X20SLX842 - Technical data

Model number	X20SLX402	X20cSLX402	X20SLX806	X20SLX842
Additional SafeDESIGNER limit values				
Max. number of function block types	64	64	64	64
Max. number of force variables	8	8	8	8
Max. number of variable with variable status	128	128	128	128
I/O power supply				
Nominal voltage	24 VDC	24 VDC	24 VDC	24 VDC
Voltage range	24 VDC -15% / +20%	24 VDC -15% / +20%	24 VDC -15% / +20%	24 VDC -15% / +20%
Integrated protection	Reverse polarity protection	Reverse polarity protection	Reverse polarity protection	Reverse polarity protection
Safe digital inputs				
Nominal voltage	24 VDC	24 VDC	24 VDC	24 VDC
Input characteristics per EN 61131-2	Type 1	Type 1	Type 1	Type 1
Input filter				
Hardware	≤150 µs	≤150 µs	≤150 µs	≤150 µs
Software	Configurable between 0 and 500 ms	Configurable between 0 and 500 ms	Configurable between 0 and 500 ms	Configurable between 0 and 500 ms
Input circuit	Sink	Sink	Sink	Sink
Input voltage	24 VDC -15% / +20%	24 VDC -15% / +20%	24 VDC -15% / +20%	24 VDC -15% / +20%
Input current at 24 VDC	Max. 3.28 mA	Max. 3.28 mA	Max. 3.28 mA	Max. 3.28 mA
Input resistance	Min. 7.33 kΩ	Min. 7.33 kΩ	Min. 7.33 kΩ	Min. 7.33 kΩ
Error detection time	100 ms	100 ms	100 ms	100 ms
Isolation voltage between channel and bus	500 V_{eff}	500 V_{eff}	500 V_{eff}	500 V_{eff}
Switching threshold				
Low	<5 VDC	<5 VDC	<5 VDC	<5 VDC
High	>15 VDC	>15 VDC	>15 VDC	>15 VDC
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line	Max. 60 m with unshielded line Max. 400 m with shielded line	Max. 60 m with unshielded line Max. 400 m with shielded line	Max. 60 m with unshielded line Max. 400 m with shielded line
Safe digital HS-LS outputs				
Variant	-	-	-	FET, 1x positive switching, 1x negative switching, type A, output level readable
Nominal voltage	-	-	-	24 VDC
Nominal output current	-	-	-	3 A
Total nominal current	-	-	-	10 A [5)]
Output protection	-	-	-	Thermal short-circuit shutdown, integrated protection for switching inductive loads [6)]
Braking voltage when switching off inductive loads	-	-	-	Max. 90 VDC [7)]
Error detection	-	-	-	1 s
Isolation voltage between channel and bus	-	-	-	500 V_{eff}
Peak short-circuit current	-	-	-	Max. 100 A
Leakage current when switched off	-	-	-	<1 mA
Residual voltage	-	-	-	≤1 VDC at nominal current
Switching voltage	-	-	-	I/O power supply minus residual voltage
Max. switching frequency	-	-	-	1000 Hz
Test pulse length	-	-	-	Max. 500 µs
Max. capacitive load	-	-	-	100 nF
Safe digital HS-HS outputs				
Variant	FET, 2x positive switching, type B2, output level readable	FET, 2x positive switching, type B2, output level readable	FET, 2x positive switching, type B2, output level readable	FET, 2x positive switching, type B2, output level readable
Nominal voltage	24 VDC	24 VDC	24 VDC	24 VDC
Nominal output current	0.2 A	0.2 A	0.2 A	50 mA
Total nominal current	0.4 A	0.4 A	1.2 A	100 mA
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads [6)]	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads [6)]	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads [6)]	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads [6)]
Braking voltage when switching off inductive loads	Max. 45 VDC	Max. 45 VDC	Max. 45 VDC	Max. 45 VDC
Error detection time	1 s	1 s	1 s	1 s
Isolation voltage between channel and bus	500 V_{eff}	500 V_{eff}	500 V_{eff}	500 V_{eff}
Peak short-circuit current	Max. 10 A	Max. 10 A	Max. 10 A	500 mA
Leakage current when switched off	<100 µA	<100 µA	<100 µA	<1 mA
Residual voltage	≤1.2 VDC at nominal current	≤1.2 VDC at nominal current	≤1.2 VDC at nominal current	≤3 VDC at nominal current
Switching voltage	I/O power supply minus residual voltage	I/O power supply minus residual voltage	I/O power supply minus residual voltage	I/O power supply minus residual voltage
Max. switching frequency	100 Hz	100 Hz	100 Hz	100 Hz
Test pulse length	Max. 10 µs	Max. 10 µs	Max. 10 µs	Max. 500 µs
Max. capacitive load	100 nF	100 nF	100 nF	100 nF
Current on loss of ground				
I_{OUT}	<100 µA	<100 µA	<100 µA	<100 µA
I_{GND}	<200 mA	<200 mA	<200 mA	<50 mA [8)]
Pulse outputs				
Variant	Push-Pull	Push-Pull	Push-Pull	Push-Pull

Table 89: X20SLX402, X20cSLX402, X20SLX806, X20SLX842 - Technical data

Model number	X20SLX402	X20cSLX402	X20SLX806	X20SLX842
Nominal output current	50 mA			
Output protection	Shutdown of individual channels in the event of overload or short circuit [6]			
Peak short-circuit current	0.5 A for 120 µs			
Short-circuit current	15 mA_{eff}			
Leakage current when switched off	0.1 mA			
Residual voltage	≤4 VDC			
Switching voltage	I/O power supply minus residual voltage			
Total nominal current	200 mA			
Operating conditions				
Mounting orientation				
Horizontal	Yes			
Vertical	Yes			
Installation elevation above sea level	0 to 2000 m, no limitation			
Degree of protection per EN 60529	IP20			
Ambient conditions				
Temperature				
Operation				
Horizontal mounting orientation	0 to 60°C	-40 to 60°C	0 to 60°C	
Vertical mounting orientation	0 to 50°C	-40 to 50°C	0 to 50°C	
Derating	See section "Derating".			
Storage	-40 to 85°C			
Transport	-40 to 85°C			
Relative humidity				
Operation	5 to 95%, non-condensing	Up to 100%, condensing	5 to 95%, non-condensing	
Storage	5 to 95%, non-condensing			
Transport	5 to 95%, non-condensing			
Mechanical properties				
Note	Order 2x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.			
Spacing	$25^{+0.2}$ mm			

Table 89: X20SLX402, X20cSLX402, X20SLX806, X20SLX842 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) If there are multiple SafeLOGIC-X controllers in the Automation Studio hardware tree, all but 1 must be disabled.
3) Keep in mind that 8 BOOL count as 1 data point.
4) For a parameter description, see section "Message window" of the SafeDESIGNER documentation.
5) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.
6) The protective function is provided for max. 30 minutes for a continuous short circuit.
7) Due to the internal protective circuit, this braking voltage only takes effect starting at a load of typ. 250 mA.
8) The value for this module is limited to 50 mA by the nominal output current of the HS-HS outputs.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SLX402	X20SLX806	X20SLX842
Derating bonus			
At 24 VDC	+0°C		
Dummy module on the left	+0°C		
Dummy module on the right	+0°C		
Dummy module on the left and right	+0°C		
With double PFH / PFH_d	+0°C		

Table 90: Derating bonus

Inputs

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

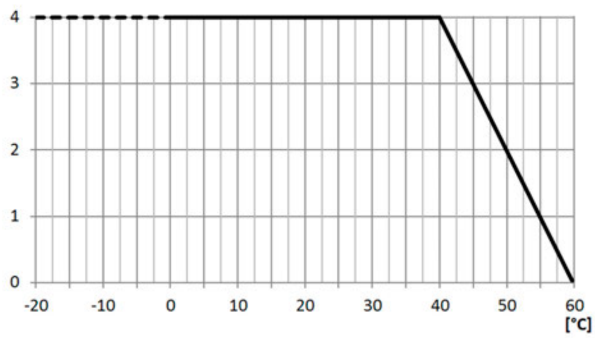
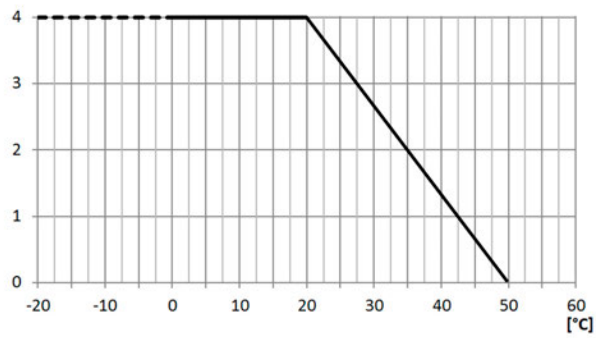
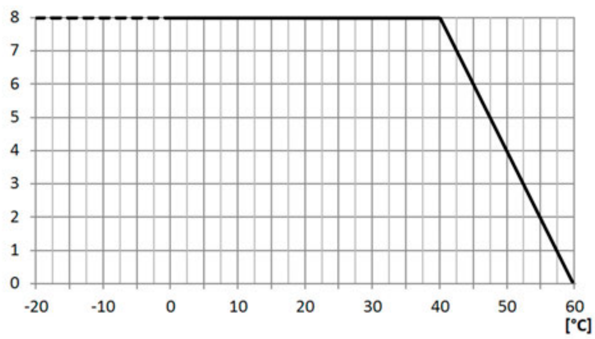
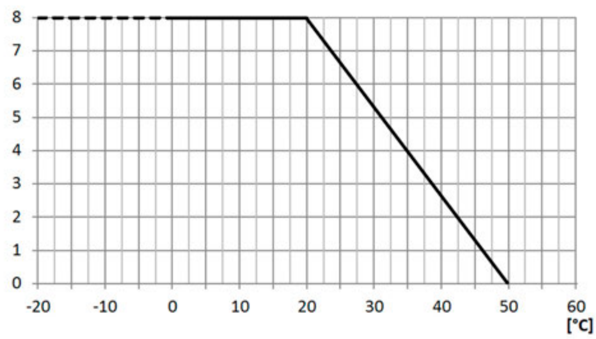
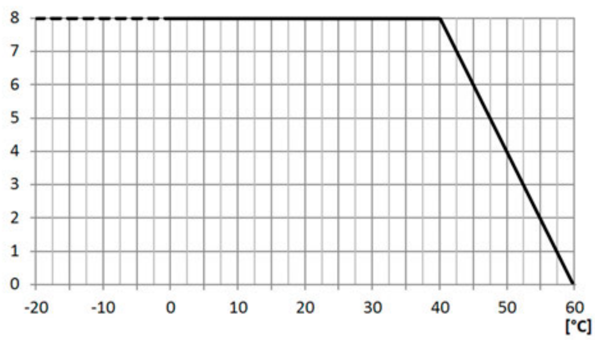
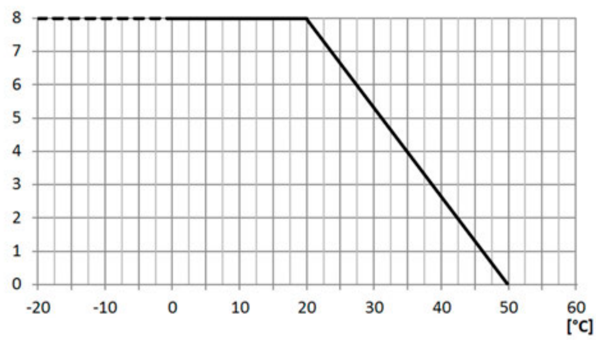
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
X20SLX402	
	
X20SLX806	
	
X20SLX842	
	

Table 91: Derating in relation to operating temperature and mounting orientation

Outputs

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

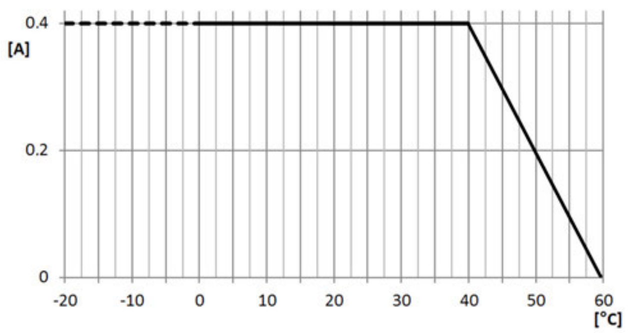
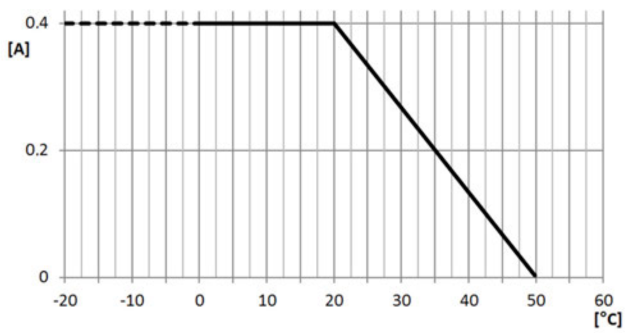
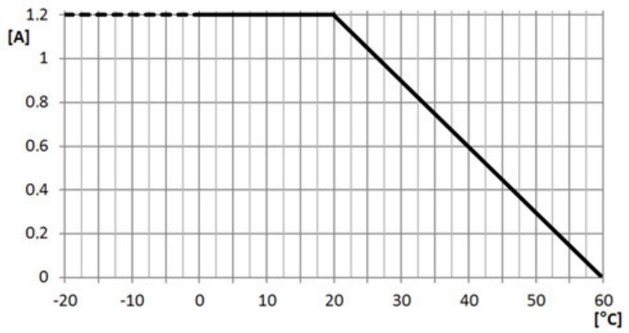
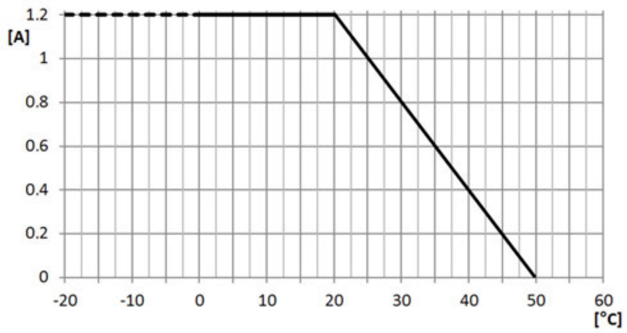
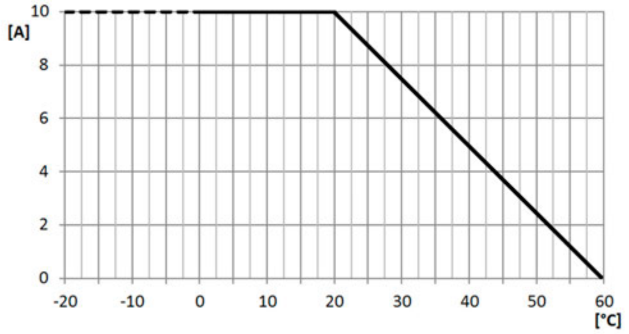
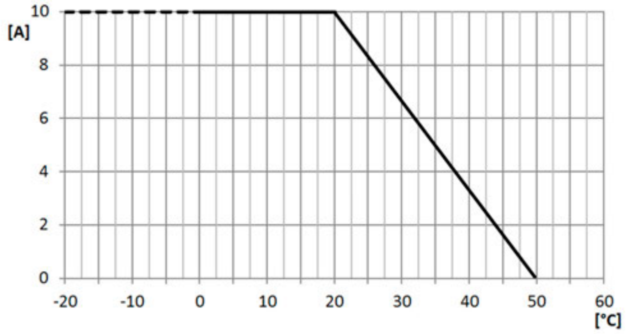
<table>
<tr><th>Horizontal (0 to 60°C, coated: -40 to 60°C)</th><th>Vertical (0 to 50°C, coated: -40 to 50°C)</th></tr>
<tr><td colspan="2">X20SLX402</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">X20SLX806</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">X20SLX842</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Table 92: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.8.3.5 LED status indicators

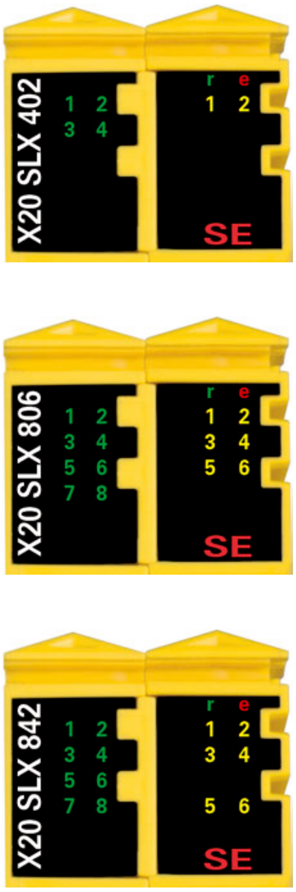
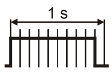
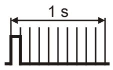
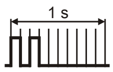
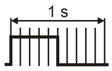
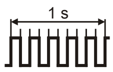
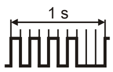
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Solid red / Single green flash		Invalid firmware
	1 to 8	Input state of the corresponding digital input The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an input channel
			Blinking	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels or startup not yet completed
		Green	On	Input set
	1 to 6	Output status of the corresponding digital output The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels or startup not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not supplied with voltage, safety firmware in OPERATIONAL state
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state or "SafeOSstate!=RUN"
				Safe communication channel not OK, openSAFETY connection valid problem or "SafeOSstate!=RUN"
				Boot phase, faulty firmware, setup mode active (hardware upgrade 1.10.2.x and later) For details about setup mode, see section "Setup mode".
				Test/Pilot firmware or safety application created with test/pilot version of SafeDESIGNER
				SafeDESIGNER in "Debug" mode
			On	Safety state active for the entire module (= state "FailSafe")
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 93: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.8.3.6 Pinouts

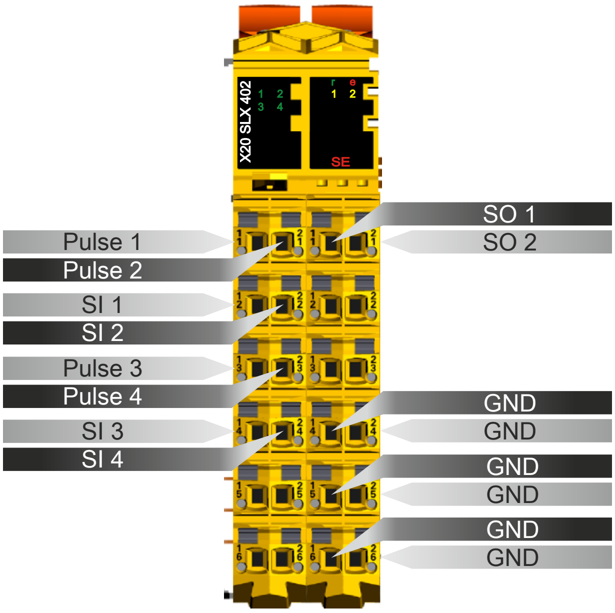


Figure 86: X20SLX402 - Pinout

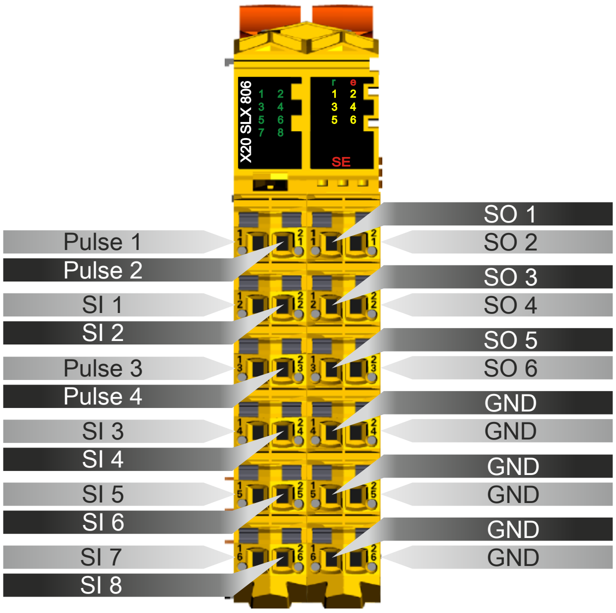


Figure 87: X20SLX806 - Pinout

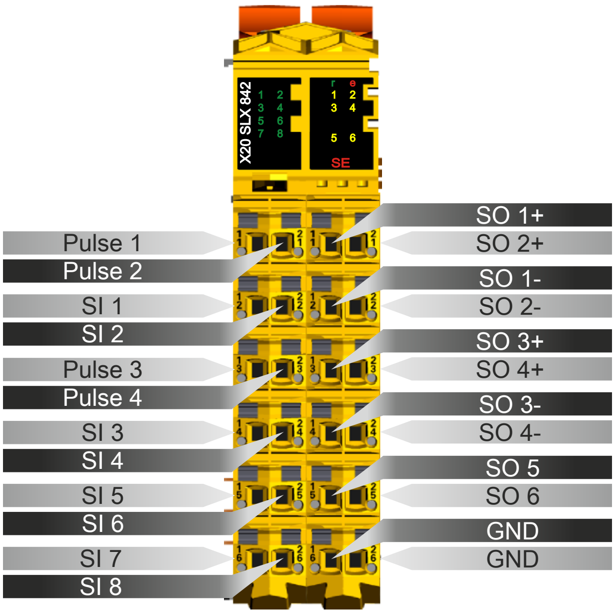


Figure 88: X20SLX842 - Pinout

2.6.8.3.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.8.3.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

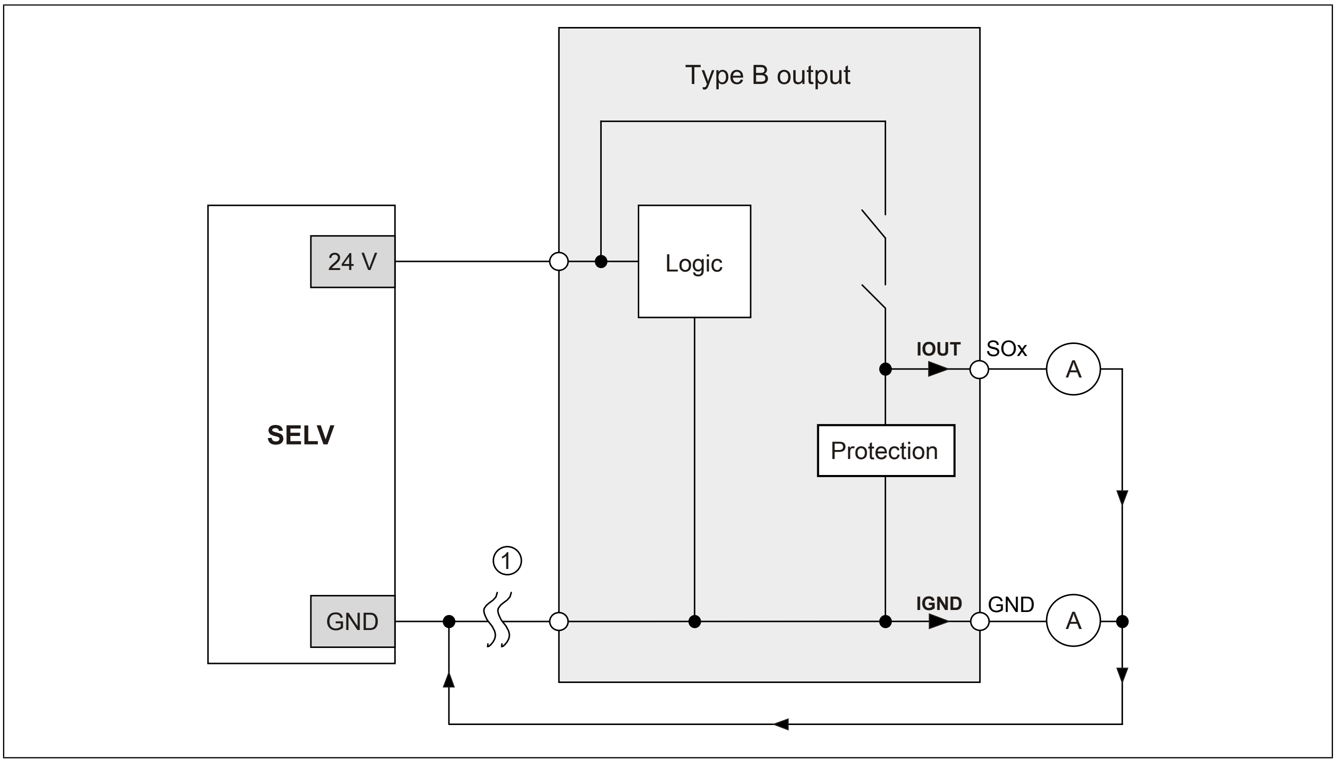


Figure 89: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

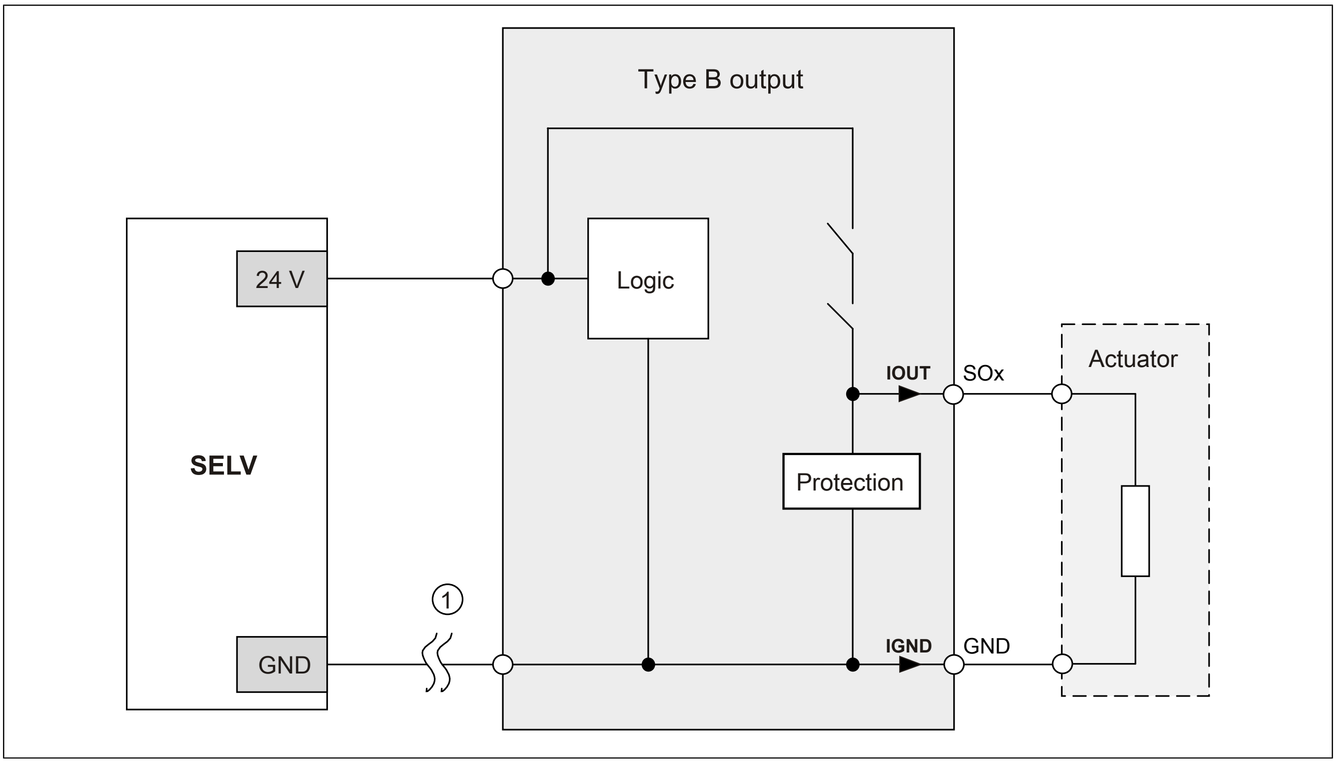


Figure 90: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

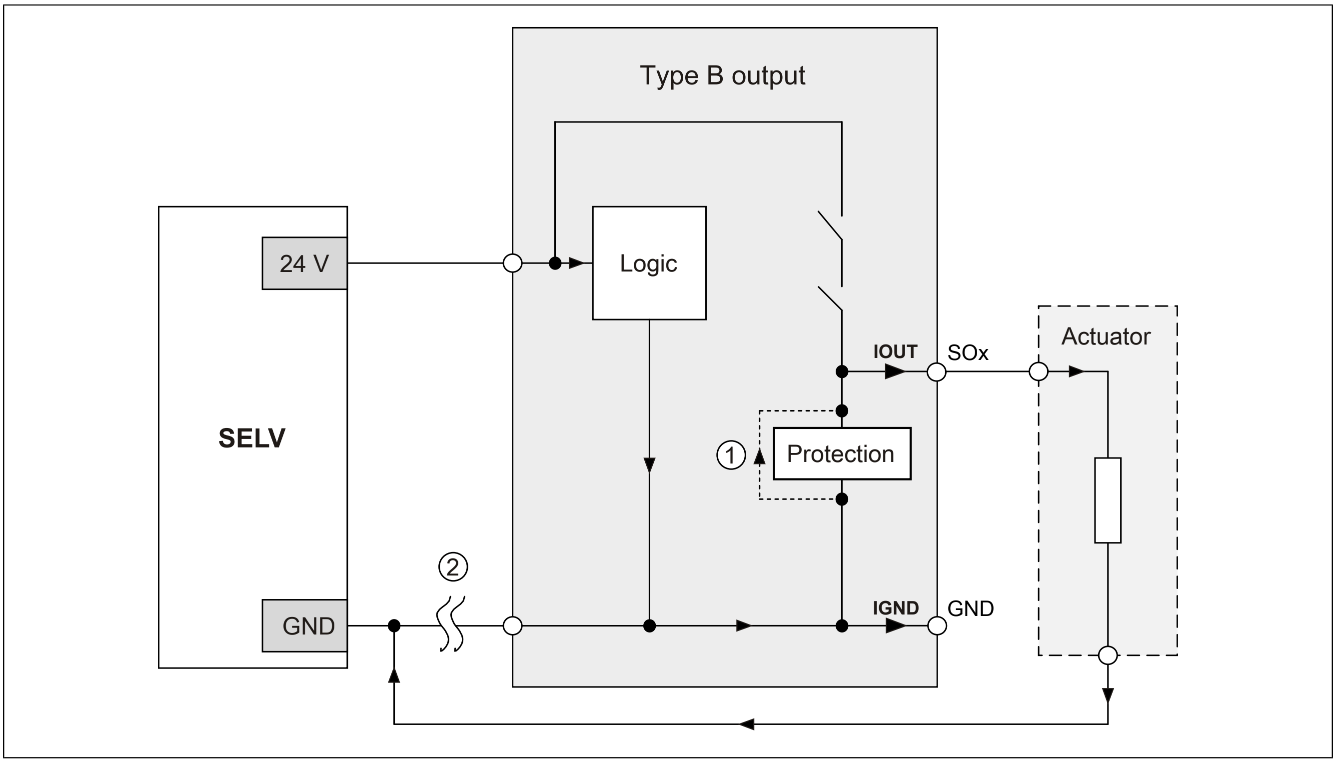


Figure 91: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

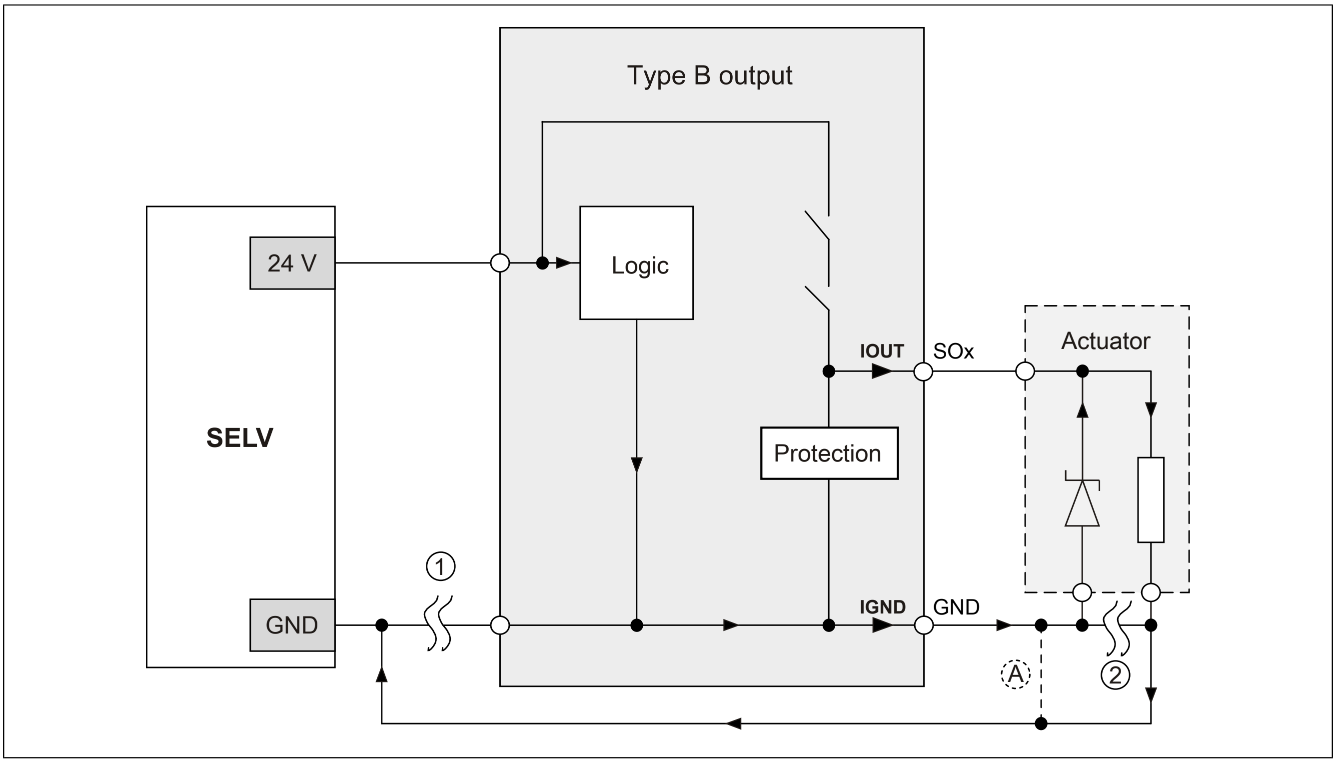


Figure 92: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

2.6.8.3.7.2 Connecting single-channel sensors with contacts

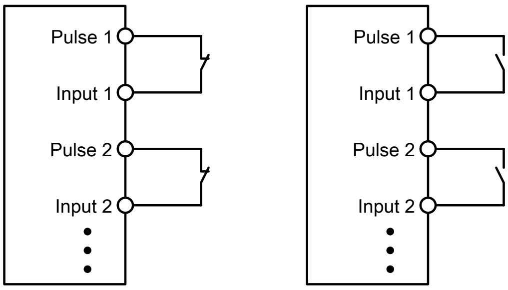


Figure 93: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.8.3.7.3 Connecting two-channel sensors with contacts

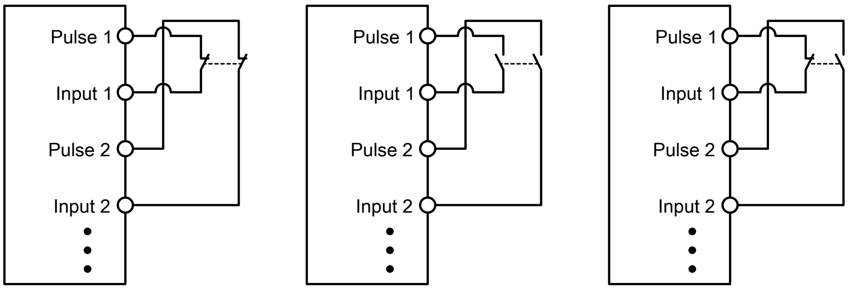


Figure 94: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.8.3.7.4 Connecting multi-channel sensors with contacts

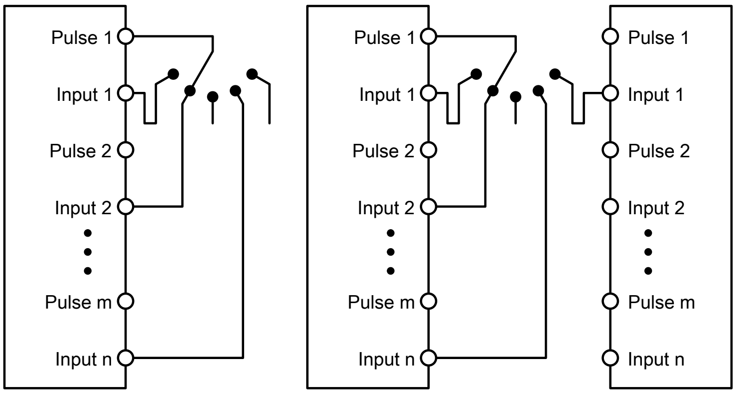


Figure 95: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.8.3.7.5 Connecting electronic sensors

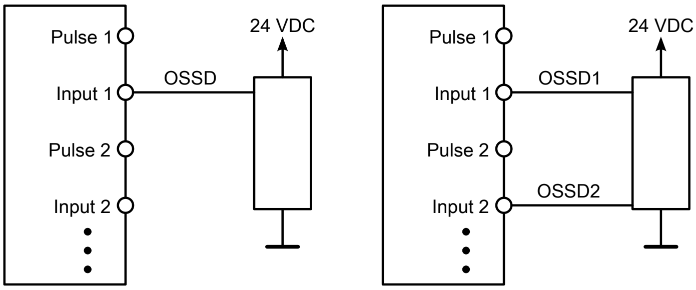


Figure 96: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.8.3.7.6 Using the same pulse signals

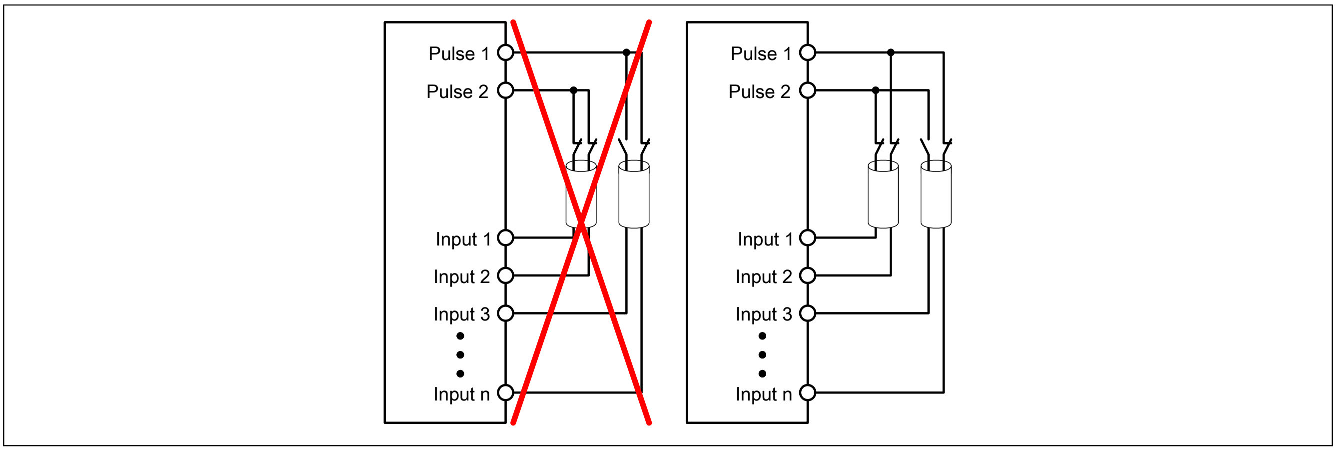


Figure 97: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.8.3.7.7 Connecting safety-oriented actuators for Type A outputs

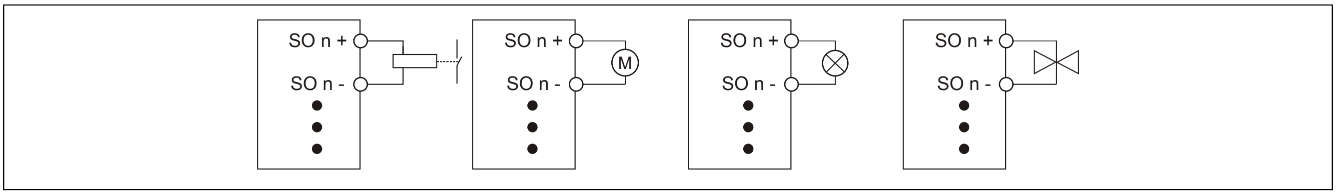


Figure 98: Connecting safety-oriented actuators for Type A outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

2.6.8.3.7.8 Connecting safety-oriented actuators for Type B outputs

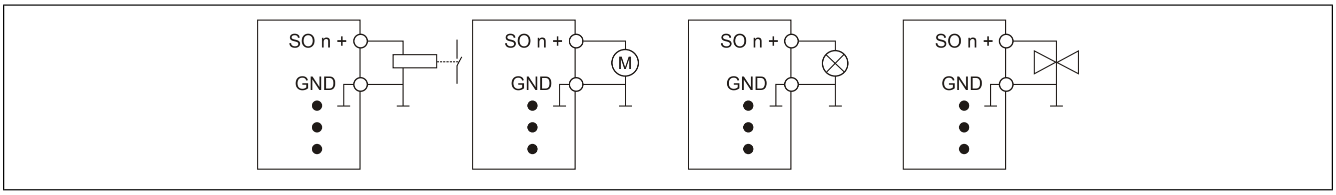


Figure 99: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.8.3.8 Error detection

2.6.8.3.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.8.3.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type A output channels

Danger!

Type A output channels also cut off the load on the GND side. Check whether the actuator you have connected permits a cutoff on the GND side. X20 and X67 systems do not support this type of cutoff, for example.

Danger!

Note that wiring SOx+ directly to GND via an actuator is not permitted; wiring 24 VDC directly to SOx- via an actuator is also not permitted.

These types of errors will not be detected by the module. The user must prevent these types of errors through careful wiring.

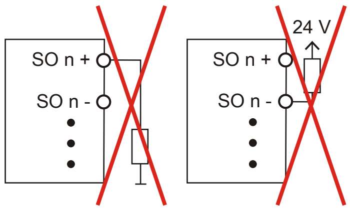


Figure 100: Invalid wiring

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

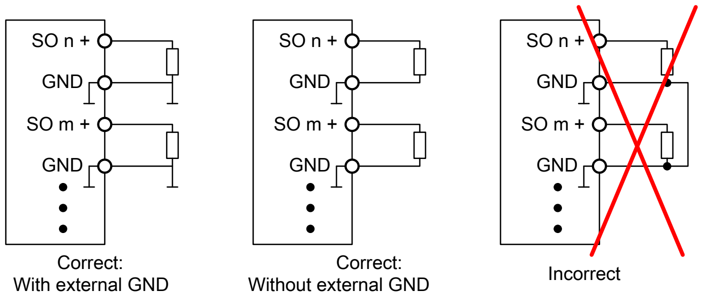


Figure 101: Invalid wiring

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 94: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 95: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 96: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td>Detected</td><td>Detected</td><td>Detected</td><td>Not detected</td></tr>
</table>

Table 97: SO error detection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td>X20SLXxxx</td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected [2]</td><td>Not detected [2]</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Short circuit between SOx+ (output type A) and SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
</table>

Table 97: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.8.3.9 Input circuit diagram

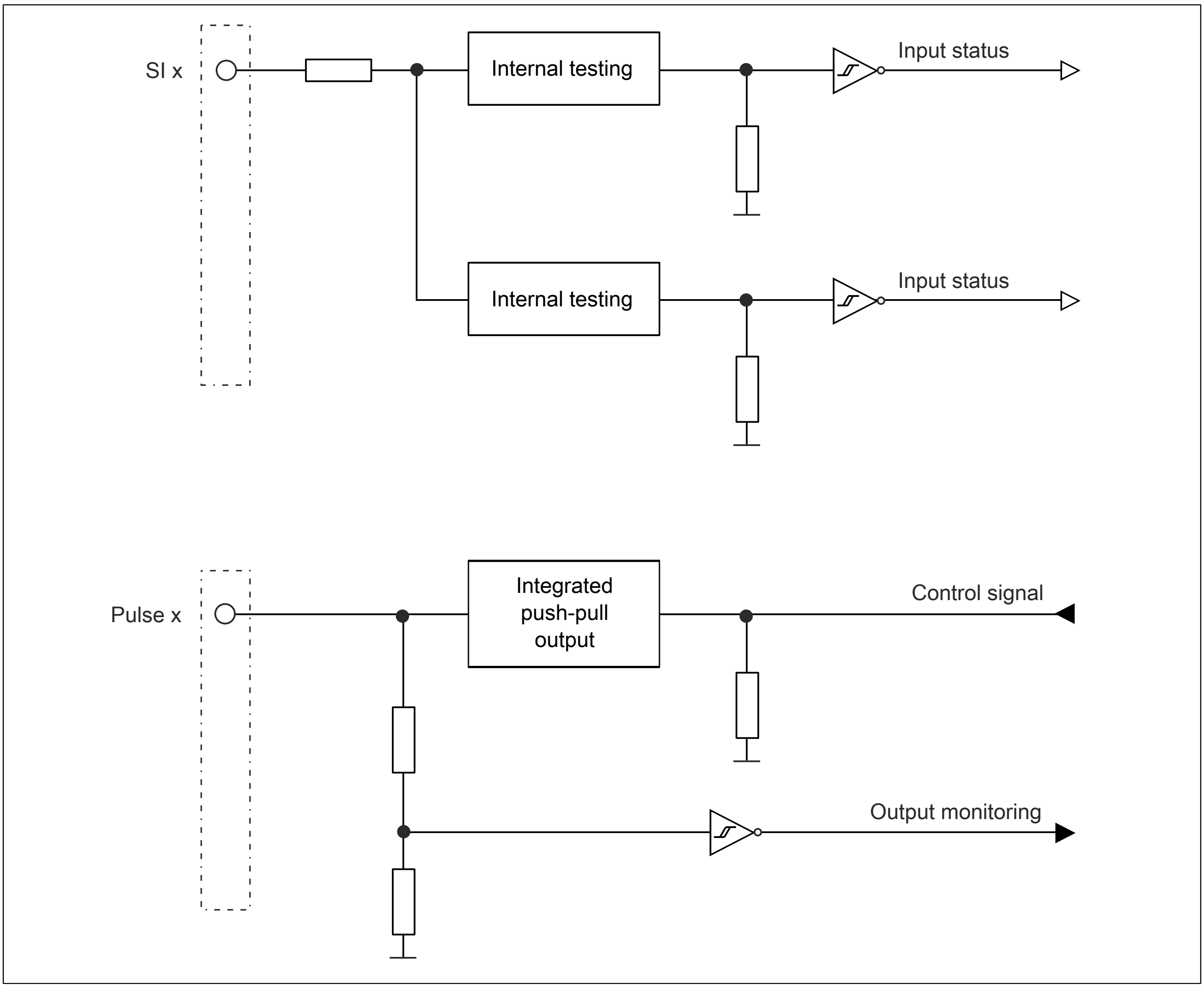


Figure 102: Input circuit diagram

2.6.8.3.10 Type A output circuit diagram

Type A digital output channels are designed for positive and GND switching inside the module.

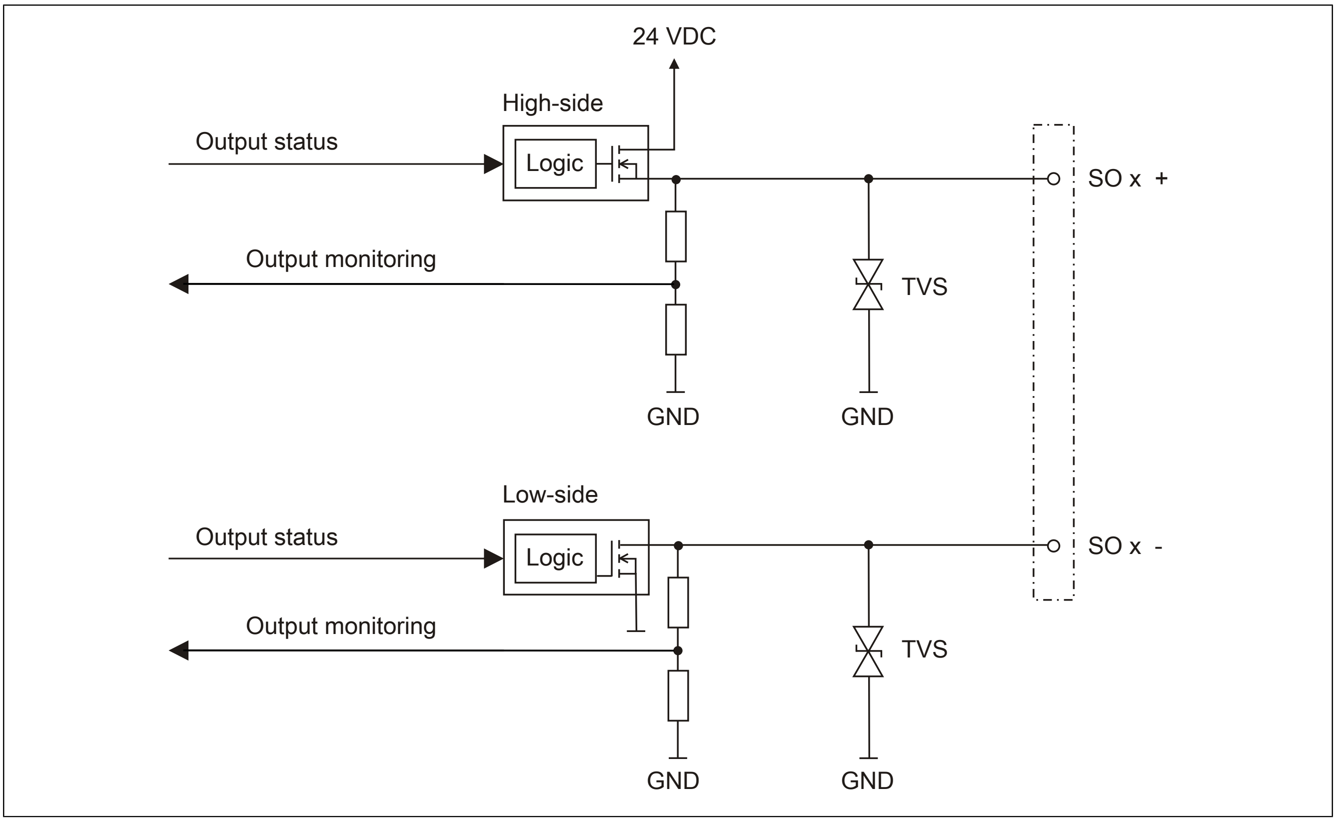


Figure 103: Type A output circuit diagram

2.6.8.3.11 Type B output circuit diagram

Type B digital output channels are designed for positive and positive switching inside the module.

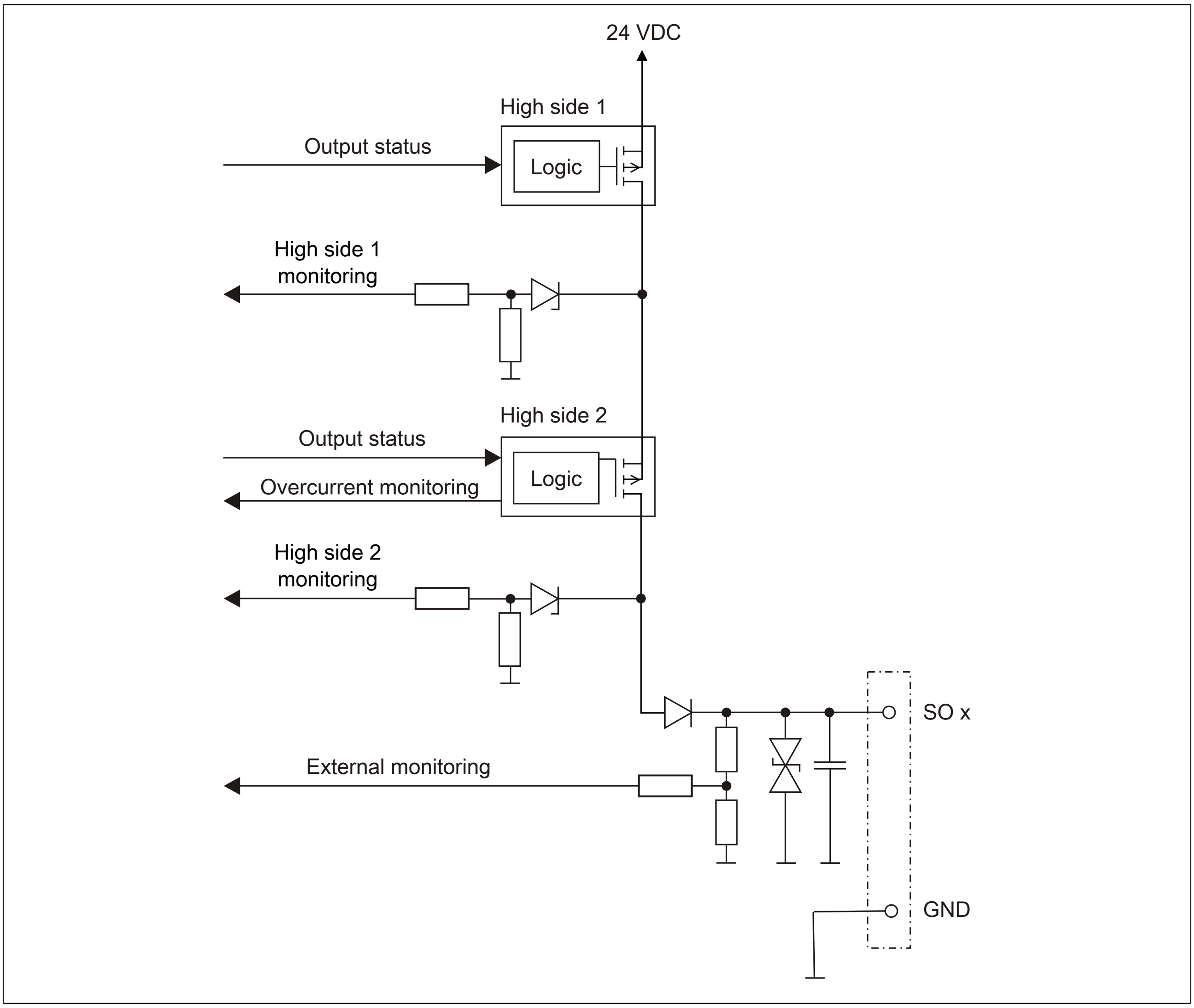


Figure 104: Type B output circuit diagram

2.6.8.3.12 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.8.3.13 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 µs

Maximum I/O update time for input channels
1150 µs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels
1300 µs

2.6.8.3.14 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

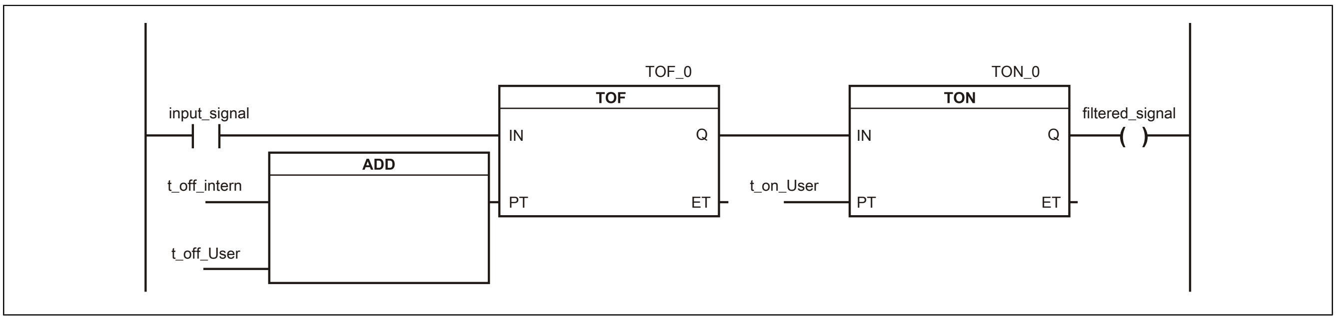


Figure 105: SI input filter - Diagram 1

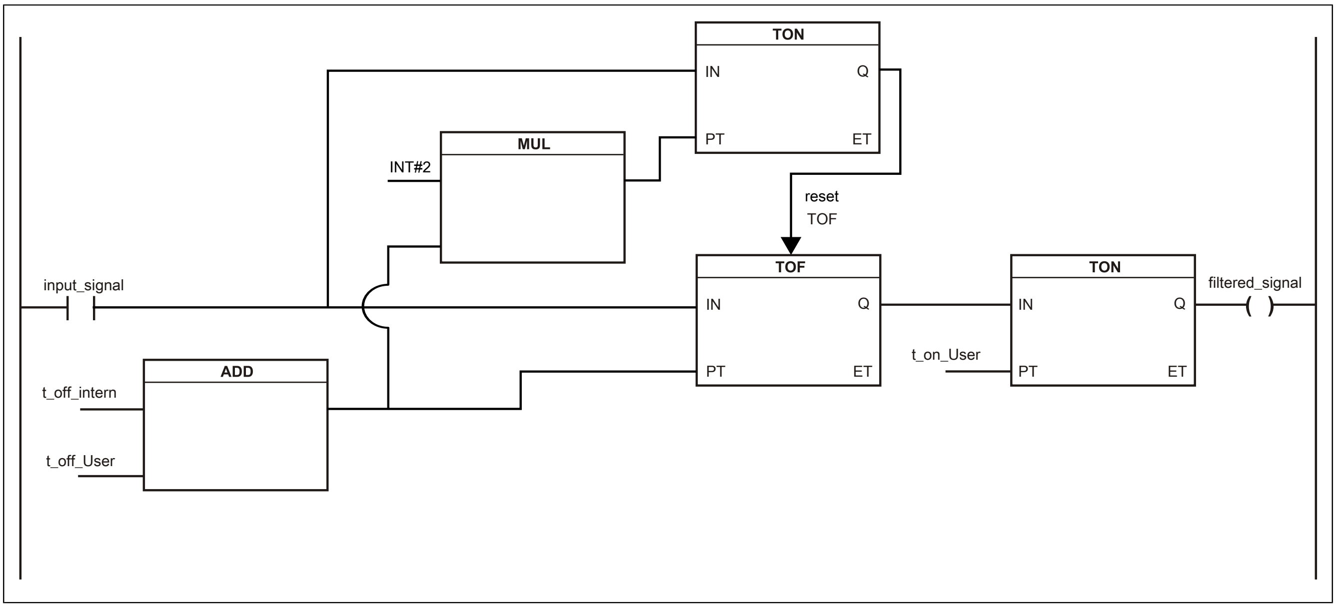


Figure 106: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.8.3.15 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.8.3.16 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.8.3.17 Register description

2.6.8.3.17.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 98: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 1</td><td>1</td><td>-</td></tr>
<tr><td>SafeDESIGNER project</td><td>Name of the safety project</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeDESIGNER version</td><td>SafeDESIGNER version for the safety project</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.5.x or later)</td><td>This parameter enables blackout or standalone mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Off</td><td>Both blackout mode and standalone mode are disabled.</td></tr><tr><td>Blackout mode</td><td>Blackout mode is enabled.</td></tr><tr><td>Standalone mode</td><td>Standalone mode is enabled. This makes it possible to start up the SafeLOGIC-X controller without an active communication connection.</td></tr></table></td></tr>
</table>

Table 99: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 100: I/O configuration parameters: Output signal path

Group: SafeDESIGNER to SafeLOGIC communication

Starting with SafeLOGIC V1.4.0.0 and Automation Runtime V3.04:

When SPROXY is enabled, the SafeLOGIC controller can be accessed via a TCP/IP port on the standard CPU.

This uses the SafeDESIGNER setting "SL communication via the CPU" (SafeDESIGNER V2.80 or higher).

Parameter	Description	Default value	Unit
Activate SPROXY	Enables the SafeDESIGNER online connection	On	-
Server communication port	TCP/IP port number used to access the SafeLOGIC controller • Recommended values: 50,000 to 50,100 **Note:** If multiple SafeLOGIC controllers are being used in the project, then a different port number must be configured for each one!	50000	-

Table 101: I/O configuration parameters: SafeDESIGNER to SafeLOGIC communication

Group: CPU to SafeLOGIC communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the CPU to the SafeLOGIC controller • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64.	8	-
Number of INT channels	Number of INT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 4.	0	-
Number of UINT channels	Number of UINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 4.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 2.	0	-
Number of UDINT channels	Number of UDINT channels from the CPU to the SafeLOGIC controller • Permissible values: 0 to 2.	0	-

Table 102: I/O configuration parameters: CPU to SafeLOGIC communication

Group: SafeLOGIC to CPU communication

Parameter	Description	Default value	Unit
Number of BOOL channels	Number of BOOL channels from the SafeLOGIC controller to the CPU • Permissible values: 0, 8, 16, 24, 32, 40, 48, 56, 64.	8	-
Number of INT channels	Number of INT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 4.	0	-
Number of UINT channels	Number of UINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 4.	0	-
Number of DINT channels (Safety Release 1.4 and Automation Runtime V3.08 required)	Number of DINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 2.	0	-
Number of UDINT channels	Number of UDINT channels from the SafeLOGIC controller to the CPU • Permissible values: 0 to 2.	0	-

Table 103: I/O configuration parameters: SafeLOGIC to CPU communication

Group: SafeLOGIC to SafeLOGIC communication

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Use as source SafeLOGIC</td><td>This parameter configures this SafeLOGIC controller as a data source for another SafeLOGIC controller.</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>This SafeLOGIC controller is available as a data source for another SafeLOGIC controller.</td></tr><tr><td>Off</td><td>This SafeLOGIC controller is not available as a data source for other SafeLOGIC controllers.</td></tr></table></td></tr>
<tr><td>Extended source SafeLOGIC communication
(Safety Release 1.4 and Automation Runtime V3.08 required)</td><td>This parameter enables the option of configuring the number of data points for "SafeLOGIC to SafeLOGIC communication" for connections where this SafeLOGIC controller serves as a data source for another SafeLOGIC controller.</td><td>Off</td><td>-</td></tr>
</table>

Table 104: I/O configuration parameters: SafeLOGIC to SafeLOGIC communication

2.6.8.3.17.2 Parameters in SafeDESIGNER

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic release</td><td>-</td></tr>
<tr><td>Node Guarding Timeout</td><td>Timeout for changing the safety modules to the PRE_OPERATIONAL state after the SafeLOGIC controller drops out or if there is a communication problem between the safety module and the SafeLOGIC controller.
This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module.
• Permissible values: 30 to 300 s
Notes
• The shorter the time, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>60</td><td>s</td></tr>
<tr><td rowspan="2">External Startup Flags</td><td>Enables external startup flags</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Enables external startup flags</td></tr><tr><td>No</td><td>Disables external startup flags</td></tr></table></td></tr>
<tr><td>Number of scans</td><td>This parameter defines the number of module search scans completed while booting.
This parameter is used to optimize the startup behavior of the system, especially if optional modules are configured but not available.
• Permissible values: 1 to 10</td><td>5.
Hardware upgrade 1.10.2.0 or later: 3</td><td>-</td></tr>
<tr><td rowspan="2">Activate Setup Mode on empty SafeKEY (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables setup mode after downloading a project to a blank SafeKEY / blank section of the CompactFlash card.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Setup mode is enabled.</td></tr><tr><td>No</td><td>Setup mode is disabled.</td></tr></table></td></tr>
<tr><td rowspan="2">Auto acknowledge firmware mismatch (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a firmware exchange (acknowledgment request "Firmware Acknowledge").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of firmware exchange is enabled.</td></tr><tr><td>No</td><td>Automatic acknowledgment of firmware exchange is not enabled.</td></tr></table></td></tr>
<tr><td rowspan="2">Auto acknowledge SafeKEY exchange (hardware upgrade 1.10.2.x or later)</td><td>This parameter enables automatic acknowledgment of a SafeKEY exchange (acknowledgment request "SafeKEY Exchange").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic acknowledgment of SafeKEY exchange is enabled.</td></tr><tr><td>No</td><td>Automatic acknowledgment of SafeKEY exchange is not enabled.</td></tr></table></td></tr>
</table>

Table 105: SafeDESIGNER parameters: Basic

Danger!

If parameter "External Startup Flags" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Information:

Startup time is also affected by the asynchronous bandwidth on the POWERLINK network. For optimization options, see Automation Help under Communication → POWERLINK → General information → Multiple asynchronous send.

Information:

The information in section "Setup mode" on page 368 must be observed when using parameter "Activate Setup Mode on empty SafeKEY". The information in section "Automatic acknowledgment" on page 351 must be observed when using parameters "Auto acknowledge firmware mismatch" and "Auto acknowledge SafeKEY exchange".

Group: Safety Response Time Defaults

The parameters for the safety response time are generally set in the same way for all stations involved in the application. This is why these parameters are configured for the SafeLOGIC controller in group "Safety Response Time Defaults" in SafeDESIGNER.

If "Manual Configuration = No" is set for the individual modules, then these default values are used.

Parameter	Description	Default value	Unit
Default Safe Data Duration	This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller and SafeIO module. For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added. • Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)	150000	µs
Default Additional Tolerated Packet Loss	This parameter specifies the number of additional tolerated lost packets during data transfer. • Permissible values: 0 to 10	0	Packets
Default Packets per Node Guarding	This parameter specifies the maximum number of packets used for node guarding. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.	5	Packets

Table 106: SafeDESIGNER parameters: Safety Response Time Defaults

Group: Module Configuration

Parameter	Description	Default value	Unit
External Machine Options	Enables external machine options **Parameter value** / **Description** Yes-ATTENTION: Enables external machine options No: Disables external machine options	No	-
Cycle Time max	Parameter for checking whether a maximum time between 2 SafeLOGIC cycles is exceeded. • Permissible values: 2100 to 41,000 µs (corresponds to 2.1 to 41 ms) **Important:** This value should not be the same as the actual cycle time; jitter must also be taken into account. The actual cycle time is influenced by the SafeDESIGNER application and the "SLXioCycle" data point. The actual cycle time of the safety application can be seen in the SafeLOGIC "Info" dialog box.	40000	µs
Disable OSSD	This parameter can be used to switch off automatic testing of the output driver for all of the module's channels. **Parameter value** / **Description** Yes-ATTENTION: Automatic testing of the output driver is switched off. No: Automatic testing of the output driver is enabled.	No	-

Table 107: SafeDESIGNER parameters: Module Configuration

Danger!

If parameter "External Machine Options" is set to "Yes-ATTENTION", thus enabling one of these functions to be used in SafeDESIGNER, then the associated notices in chapter "Operation via the AsSafety library" must be taken into account. Failure to do so can result in hazardous situations caused by malfunctions.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalInputxx

Parameter	Description	Default value	Unit
Pulse Source	This parameter can be used to specify the pulse source for the input channel.	See table	-
	All available pulse outputs can be specified as "Pulse Source". The default values can be determined using the following table: **Channel** / **Default "Pulse Source"** 1, 5 / Channel 1 2, 6 / Channel 2 3, 7 / Channel 3 4, 8 / Channel 4		
	Note: If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".		
Pulse Mode	This parameter can be used to specify the "Pulse Mode" for the input channel.	Internal	-
	Parameter value / **Description** Internal: The channel works exclusively with the pulse output that is configured for "Pulse Source". No Pulse: The pulse check on the channel is disabled. Potential "low phases" of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.		
Filter Off	Switch-off filter for the channel to remove potentially disruptive signal low phases. • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	0	µs
Filter On	Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short. • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	200000	µs
Discrepancy Time	Parameter only available for odd-numbered channels. This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can be undefined without triggering an error. • Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)	50000	µs
Two-Channel Processing Mode	Parameter only available for odd-numbered channels. This parameter specifies the type of dual-channel evaluation. Permissible values: • None • Equivalent • Antivalent	None	-

Table 108: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** / **Description** Yes-ATTENTION: "Automatic restart" function is activated. No: "Automatic restart" function is not activated.	No	-

Table 109: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.8.3.17.3 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>SLXioCycle</td><td>Read</td><td>-</td><td>UDINT</td><td>Exchanging cyclic data between the SafeLOGIC-X controller and CPU (time in µs).
This value is influenced by:
• Quantity and data width of SafeNODEs
• Cycle times set in Automation Studio (POWERLINK, X2X, Crosslink task)
• Automation Studio configuration (see items above)
The value must be <30 ms; otherwise, the max. SafeLOGIC-X cycle time (parameter "Cycle Time max") is exceeded.
In addition, values <15 ms are recommended since large values slow down the SafeDESIGNER online connection.</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWversionSCM</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - SCMar</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.5.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.5.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>ApplSDcrc</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>CRC of the SafeDESIGNER application on the module</td></tr>
<tr><td>ApplSDtime</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Timestamp of the SafeDESIGNER application on the module in UNIX format</td></tr>
<tr><td>ApplMOptCRC</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>CRC of the external machine options on the module</td></tr>
<tr><td>ApplMOptTime</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Timestamp of the external machine options on the module in UNIX format</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.5.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
• Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
• The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors or download of the SafeDESIGNER application to the safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table>
</td></tr>
</table>

Table 110: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>SLXbootState</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>Startup state of the SafeLOGIC-X system
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>0</td><td>Invalid - Firmware not yet running</td></tr>
<tr><td>1</td><td>Start - Waiting for synchronization of internal cyclic systems</td></tr>
<tr><td>4</td><td>Start OK - Application data valid</td></tr>
<tr><td>25</td><td>Safety PREOPERATIONAL state or "SafeOSstate!=RUN"</td></tr>
<tr><td>34</td><td>Waiting on X2X parameters from Automation Runtime</td></tr>
<tr><td>50[2]</td><td>Ready for RUN - Waiting on "SafeModuleOK" for the modules</td></tr>
<tr><td>52[2]</td><td>Waiting period for stable valid "SafeModuleOK" active</td></tr>
<tr><td>54[2]</td><td>Startup complete - SafeRUN</td></tr>
</table>
[2] Possible to establish connection to the SafeLOGIC-X controller via the SafePLC window in SafeDESIGNER (see dialog box "SafePLC" (control dialog box) in Automation Help).</td></tr>
<tr><td>SafeOsState</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>Status of the safety application. For details, see "SafeLOGIC "Info" dialog box in SafeDESIGNER".
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>0x00</td><td>Invalid (e.g. SafeKEY blank) or startup still active (BOOT_STATE!=0x12)</td></tr>
<tr><td>0x0F</td><td>ON (startup / internal initialization) or error (check logbook)</td></tr>
<tr><td>0x33</td><td>Loading (startup / internal initialization)</td></tr>
<tr><td>0x55</td><td>Stop [Safe]</td></tr>
<tr><td>0x66</td><td>Run [Safe]</td></tr>
<tr><td>0x99</td><td>Halt [Debug]</td></tr>
<tr><td>0xAA</td><td>Stop [Debug]</td></tr>
<tr><td>0xCC</td><td>Run [Debug]</td></tr>
<tr><td>0xF0</td><td>No execution</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelOkxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
</table>

Table 110: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UDINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table><tr><th>Bit 23 to 20</th><th>Bit 19 to 16</th><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr><tr><td>Channel 6</td><td>Channel 5</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table></td></tr>
<tr><td>BOOL1xx</td><td>Write</td><td>Read</td><td>BOOL</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>INT1xx</td><td>Write</td><td>Read</td><td>INT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>UINT1xx</td><td>Write</td><td>Read</td><td>UINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>DINT1xx</td><td>Write</td><td>Read</td><td>DINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>UDINT1xx</td><td>Write</td><td>Read</td><td>UDINT</td><td>CPU to SafeLOGIC communication channel</td></tr>
<tr><td>BOOL0xx</td><td>Read</td><td>Write</td><td>BOOL</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>INT0xx</td><td>Read</td><td>Write</td><td>INT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>UINT0xx</td><td>Read</td><td>Write</td><td>UINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>DINT0xx</td><td>Read</td><td>Write</td><td>DINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>UDINT0xx</td><td>Read</td><td>Write</td><td>UDINT</td><td>SafeLOGIC to CPU communication channel</td></tr>
<tr><td>SafeBOOLx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>SafeLOGIC to SafeLOGIC communication channel</td></tr>
<tr><td>SafeMachineOptionxx</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Internal channel for machine options</td></tr>
</table>

Table 110: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Information:

Channels for SafeLOGIC to SafeLOGIC communication: See section "Display in SafeDESIGNER"

PLCopen state diagrams

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

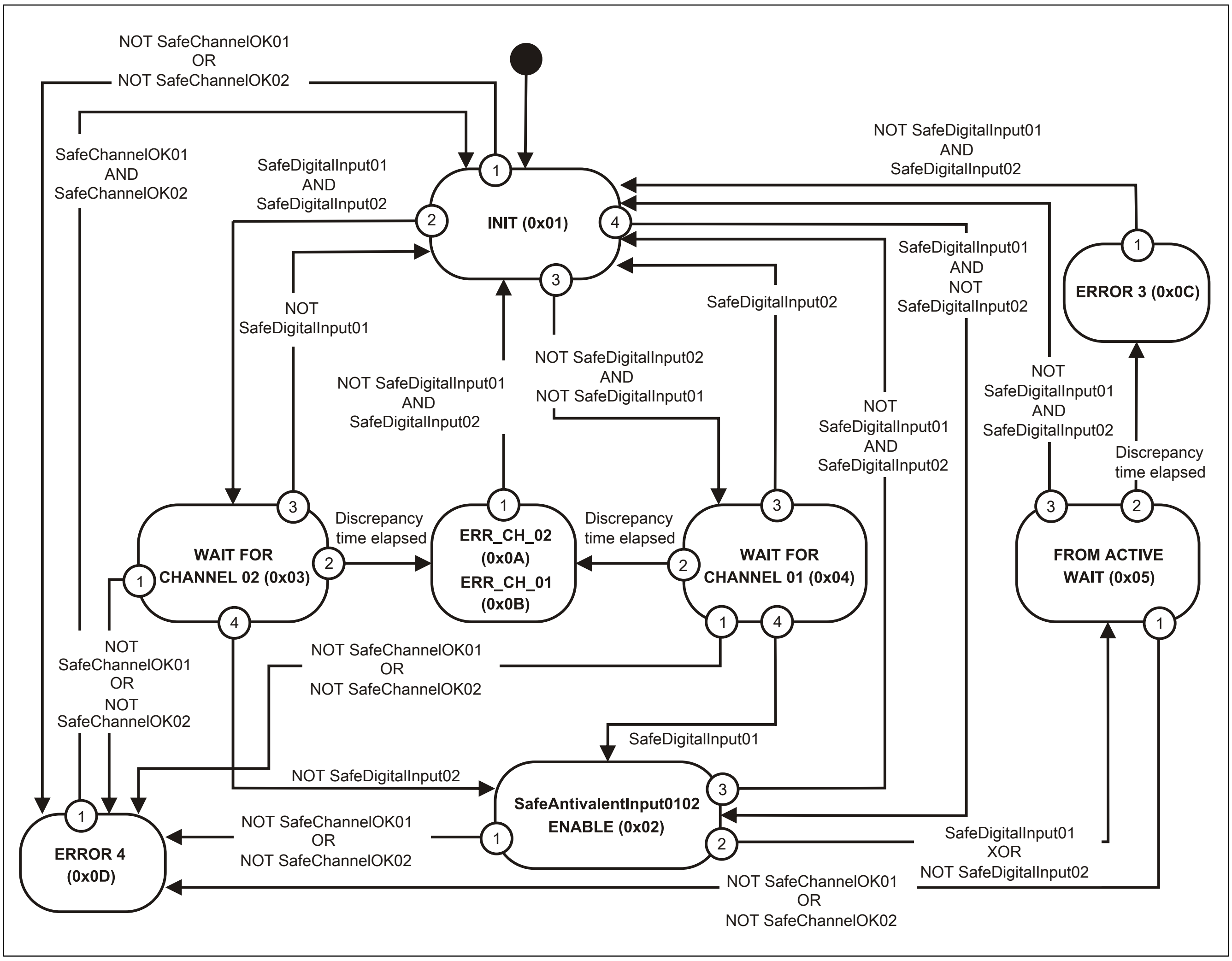


Figure 107: "Antivalent" function block - State diagram

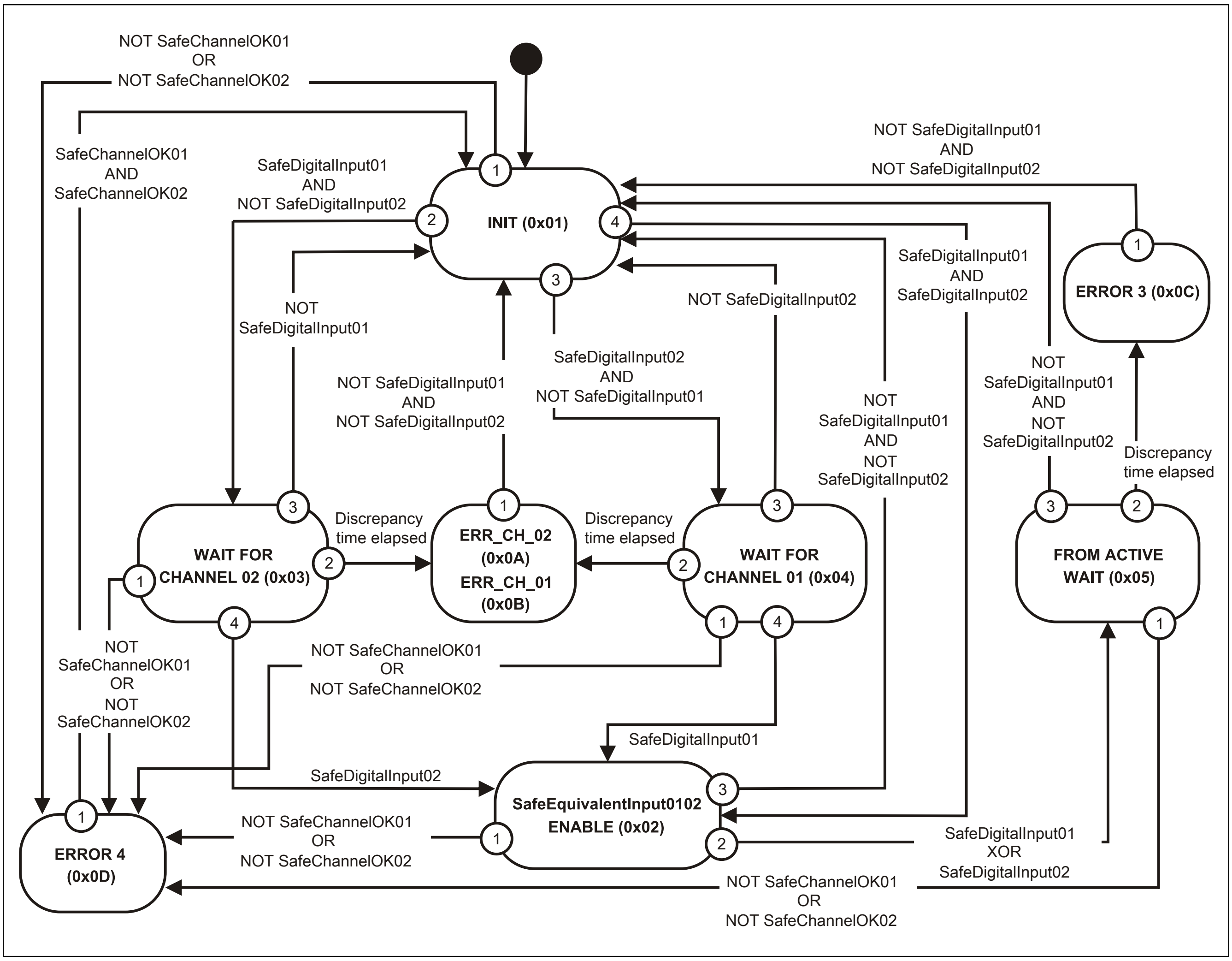


Figure 108: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

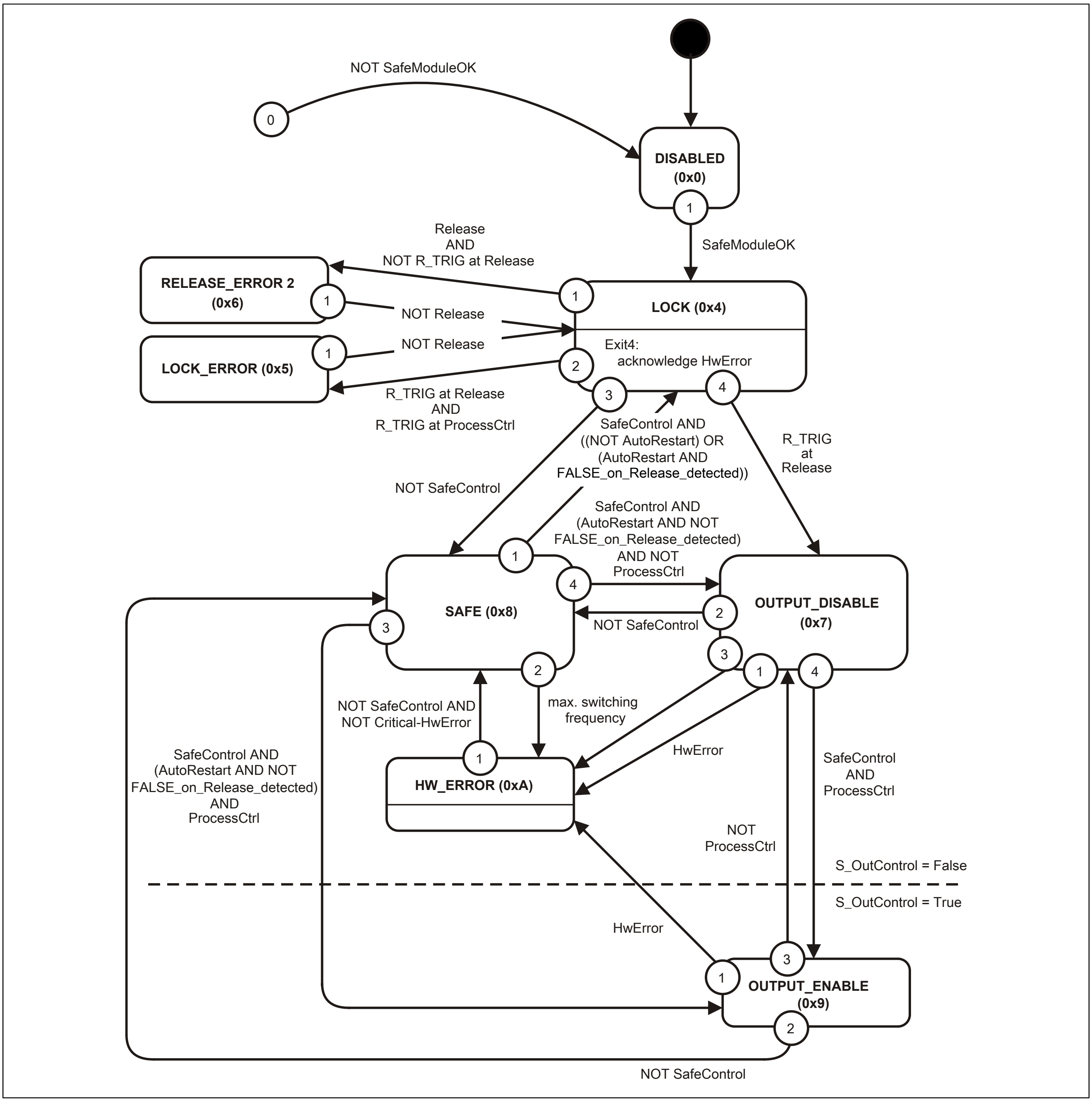


Figure 109: Restart interlock - State diagram

2.6.8.3.17.4 SafeLOGIC "Info" dialog box in SafeDESIGNER

Dialog box "SafePLC info" appears if the "Info" button in dialog box "SafePLC" (control dialog box) or in dialog box "Debug" is pressed.

The dialog box shows information about the current project in the safe programming system, the project stored/ running on the safety controller, the current status of the safety controller, debugging information, etc.

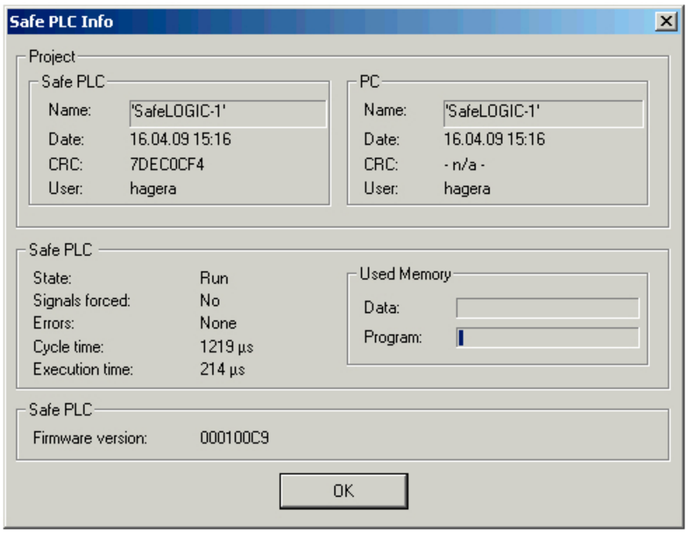


Figure 110: SafeLOGIC "Info" dialog box

Project	**Project-defining data**	
Safe PLC	Project data saved on the SafeKEY being used for the SafeLOGIC controller	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC
	User	User who made the last change
PC	SafeDESIGNER project data on the PC	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC, "- n/a -" if the project is not yet compiled
	User	User who made the last change
Safe PLC	**Status and information about the SafeLOGIC controller**	
State	Indicates the operating states of the safety controller.	
Signals forced	No	No variables are forced.
	Yes	Variables are forced.
Errors	Information regarding error messages present in the SafeDESIGNER message window	
Cycle time	Cycle time that is actually required, maximum value since the last power up This value is only relevant if "Safe PLC state = Run".	
Execution time	Actual application execution time This value corresponds to the "Safe PLC Cycle time" minus system and communication overhead.	
Used memory	Bar that shows the system resources being used	
	Data	Data memory for the safety application
	Program	Application memory for the safety application
Firmware version	Firmware version	

2.6.8.3.18 Maintenance scenarios

The operating elements on the SafeLOGIC controller (X20SL8xxx series) or the operating elements of the "Remote Control" in SafeDESIGNER (X20SL8xxx series and X20SLXxxx series) are available to handle the following maintenance scenarios.

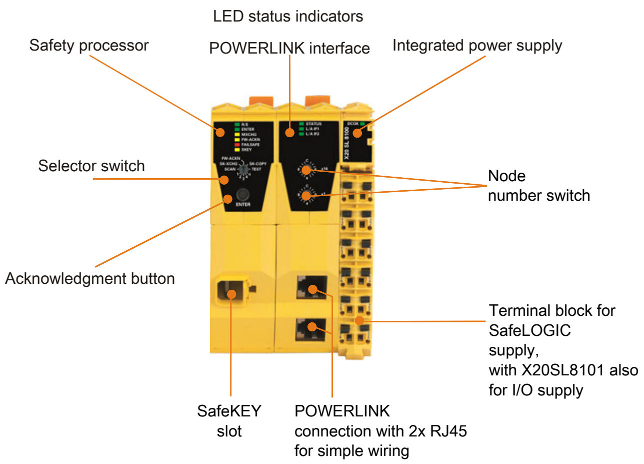


Figure 111: X20SL810x - Operating elements

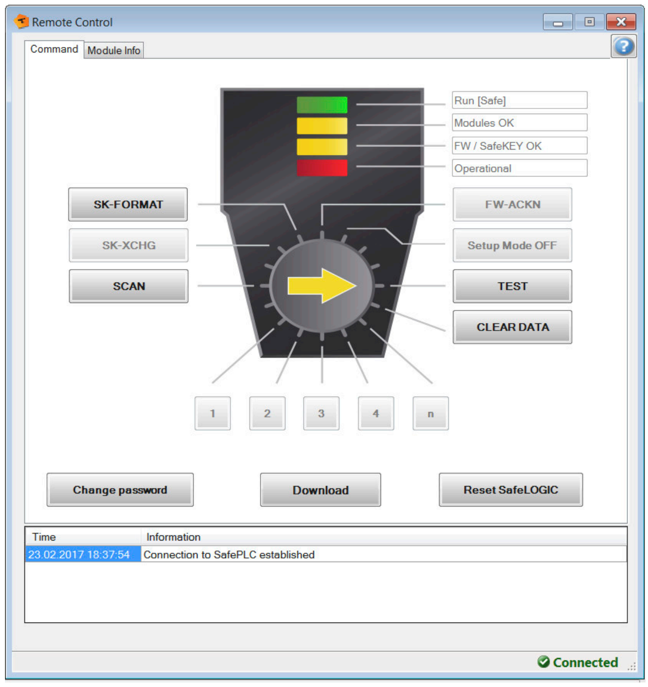


Figure 112: SafeDESIGNER - "Remote control" operating elements

For a detailed description of operating elements, see section Operating and connection elements of the technical data sheet for X20SL8xxx-series devices.

For a detailed description of operating elements, see SafeDESIGNER section Operating elements of the Remote Control in Automation Help.

2.6.8.3.18.1 Module replacement

The SafeLOGIC controller recognizes on its own when safe modules have been replaced. Following a module replacement, the complete system (SafeLOGIC, SafeLOGIC-X system components, openSAFETY) automatically ensures that the module operates again using the correct parameters and that incompatible modules are rejected. Nevertheless, the following errors are still possible after a module replacement:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

Terminals swapped between several modules

To determine whether terminals have been swapped between several modules, the user must test the safety function by performing a wiring test.

Danger!

The user must ensure that the wiring test can detect when terminals have been swapped.

Be sure to validate the entire safety function!

Wiring errors

A wiring error can occur if the wiring between the sensor or actuator and the X20 terminal is disconnected. To detect this sort of error in the wiring, the user must test the safety function by performing a wiring test.

Danger!

The user must make sure that the wiring test can detect wiring errors.

Be sure to validate the entire safety function!

SafeIO modules swapped with each other

Errors in the standard application can cause SafeIO modules to become swapped, which appears identical to a module replacement to the SafeLOGIC controller. To detect this error, the user must confirm the number of replaced modules. This links the number of modules replaced by the user and the replacements recognized by the system so that any additional replacements can be detected.

The user is informed of the number of detected module replacements via the MXCHG status. In the process, the module identifiers (UDIDs) on the SafeKEY or in the safety section of the CompactFlash card are compared to the UDIDs of the modules in the network.

If there are 1, 2, 3 or 4 different UDIDs, the user is provided information about the exact number of differences. The user must then check whether the number of replaced modules recognized by the SafeLOGIC controller corresponds to the actual number of replaced modules. If the values are the same, the user must confirm the number and perform a wiring test. This wiring test can be limited specifically to the modules that have been replaced.

If there are more than 4 different UDIDs, a standard message is provided indicating that there are differences on more than 4 modules. In this case, the user must perform a comprehensive wiring test for all modules.

If the number of modules indicated and the actual number of replaced modules do not match, the user must confirm the number of replacements determined by the SafeLOGIC controller and perform a comprehensive wire test for all modules.

Danger!

Be sure to validate the entire safety function!

Replacing an individual module

If only one module was replaced (MXCHG status indicates 1 module was replaced) and the wiring was not changed, the user can skip the wiring test because in this case the following errors can be ruled out:

- Terminals swapped between several modules
- Wiring errors
- SafeIO modules swapped with each other

Danger!

The wiring test can only be excluded if no additional changes are made when replacing an individual module (e.g. unplugging terminals, removing the wiring, etc.).

Confirming a module replacement

To confirm the number of the replaced modules, the correct number of modules must be selected:

- 1 - One module replaced
- 2 - Two modules replaced
- 3 - Three modules replaced
- 4 - Four modules replaced
- n - Five or more modules replaced

The replacement can be confirmed and the accompanying wiring test can be limited to the replaced modules when up to four modules are replaced. When more than four modules are replaced, a comprehensive wiring test must be performed for all modules.

Following confirmation of the module replacement, the SafeLOGIC controller immediately commences a module scan.

Danger!

The user must ensure that the wiring test can detect a wiring error or when terminals have been swapped.

Be sure to validate the entire safety function!

2.6.8.3.18.2 Other errors in module configuration

The aforementioned differences are limited exclusively to module replacements. An error – "Missing module" status – is reported if a device is missing (except when the device is defined as optional), has an incorrect hardware code or other problems are present on the module (e.g. incorrect parameters that may not be changed by the SafeLOGIC controller). This status is only indicated if a module or firmware replacement is not being indicated. This status cannot be acknowledged.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

2.6.8.3.18.3 Acknowledging a firmware modification

A change to the firmware is indicated by the FW-ACKN status and must be confirmed using the FW-ACKN action. A firmware modification must always be concluded with full functional testing.

> ## Danger!
>
> **Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions and trained in the procedure of exchanging firmware.**
>
> **Be sure to validate the entire safety function!**

> ## Danger!
>
> **Only use firmware versions listed in the FS certificates for B&R safety technology. These FS certificates are available for download from the B&R website at http://www.br-automation.com.**

2.6.8.3.18.4 Triggering a module scan

A module scan determines if all configured modules are present in the application and if they correspond to the project configuration. The module scan runs automatically but at large time intervals. To minimize the time it takes for the SafeLOGIC controller to recognize a newly replaced module, this function can also be triggered manually by the user. The result of the scan is described in the following sections:

- "Module replacement"
- "Other errors in module configuration"
- "Acknowledging a firmware modification"

The process itself is started using the SCAN function and indicated using the "Scanning" status. The results are reported after the "Scanning" status is completed (e.g. three modules replaced).

2.6.8.3.18.5 SafeKEY or safety section of the CompactFlash card

The following data is stored on the SafeKEY (X20SL8xxx series) or in the safety section of the CompactFlash card (X20SLXxxx series):

- SafeDESIGNER application (application and all SafeDESIGNER parameters for the modules)
- Configuration (unique module code (UDID), firmware versions of modules)
- Subsequently loadable data elements (machine options, tables, etc.)

Size of the SafeDESIGNER application on the SafeKEY

The size of the current application on the SafeKEY is calculated by SafeDESIGNER during compilation and displayed in the message window (e.g. "The safety application uses 0.688 MB (11 sectors) memory.").

Notes:

- The output only takes the size of the SafeDESIGNER application into account. Space on the data storage device used by firmware or subsequently loadable data (tables, machine options, etc.) is not taken into account.
- If the online project comparison is not needed (see Automation Help → SafeDESIGNER), the download size of the application can be reduced by disabling the following communication setting: Online → Communication settings → Download project source to SL.

Removing a SafeKEY (X20SL8xxx series only)

Removing a SafeKEY always results in a change to BOOT mode, and the safety application is completely shut down.

Information:

Removing a SafeKEY during operation causes the SafeLOGIC controller to be restarted and all safety-related actuators to be cut off.

Removing a SafeKEY during operation can destroy the data on the SafeKEY.

Removing a SafeKEY during operation must therefore be avoided at all cost.

The "Backing up the SafeKEY" sequence is not affected by this general rule.

Acknowledging a SafeKEY replacement

Replacing a SafeKEY or replacing a CompactFlash card with a CompactFlash that has a modified safety section is indicated by the "FW-ACKN" status and must be acknowledged with the SK-XCHG function. Complete functional testing is then required.

Information:

A SafeKEY replacement can only be acknowledged if a valid SafeDESIGNER project has already been transferred to the SafeKEY or CompactFlash card.

Danger!

Replacing a SafeKEY or CompactFlash card will enable the safety application stored on the SafeKEY or CompactFlash card. Always check the project CRC and date that the safety application project was saved on the SafeKEY or CompactFlash card.

Danger!

Be sure to validate the entire safety function!

Changing the application on the SafeLOGIC controller by replacing the SafeKEY (X20SL8xxx series only)

All relevant configuration data and all application data and parameters are stored on the SafeKEY. In order to transfer the previous configuration data to a new SafeKEY when changing the application, the following sequence must be carried out.

- Set the selector switch to the SK-COPY position.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- The SafeKEY configuration data is saved on the SafeLOGIC controller. The SKEY LED blinks with each access.
- The FW-ACKN LED will flash after the copying procedure. This SafeKEY can now be replaced by the SafeKEY with the new application. 30 seconds are provided to do this. The FW-ACKN LED blink frequency increases after 20 seconds to signal the end of the replacement phase.
- The acknowledgment button must be pressed again after the new SafeKEY has been inserted. The selector switch remains on the setting SK-COPY.
- The internal, temporarily saved configuration data is saved on the new SafeKEY. A reset is then triggered automatically, and the data from the new SafeKEY is applied.
- Following the reset, the SafeKEY replacement must be acknowledged. To do this, move the selector switch to the setting SK-XCHG.
- Press the acknowledgment button - Action confirmed by the ENTER LED.
- Perform complete functional testing.

Information:

If the new SafeKEY is not acknowledged after 30 seconds, the function will end, i.e. if the function is triggered inadvertently, the copy function ends automatically after 30 seconds. If a SafeKEY is not inserted after 30 seconds, the SafeLOGIC controller switches to BOOT mode.

Danger!

This procedure enables the safety application stored on the new SafeKEY. Always check the project CRC and date that the safety application project was saved on the SafeKEY.

Danger!

Be sure to validate the entire safety function!

Information:

This sequence can also be used to create a SafeKEY backup using a second SafeKEY with an identical safety application. After executing the sequence, two identical SafeKEYs are available (backup copy).

Information:

Only data relevant to the machine is copied, not all of the safety application data.

2.6.8.3.18.6 Replacing a SafeLOGIC controller

Replacing a SafeLOGIC controller involves the same procedures as a normal module replacement. When replacing a SafeLOGIC controller, the SafeKEY from the SafeLOGIC controller being replaced must be kept in order to avoid activating an old safety-related application.

Danger!

Be sure to validate the entire safety function!

2.6.8.3.18.7 Authorization (X20SL8xxx series only)

The following functions can be blocked by the standard CPU:

- Confirming a module replacement
- Acknowledging a firmware modification
- Acknowledging a SafeKEY replacement
- Backing up the SafeKEY
- Replacing a SafeLOGIC controller

This allows actions to be executed in accordance with an application-specific user concept. This option is not possible from a safety perspective, however, since these functions are executed on the standard CPU.

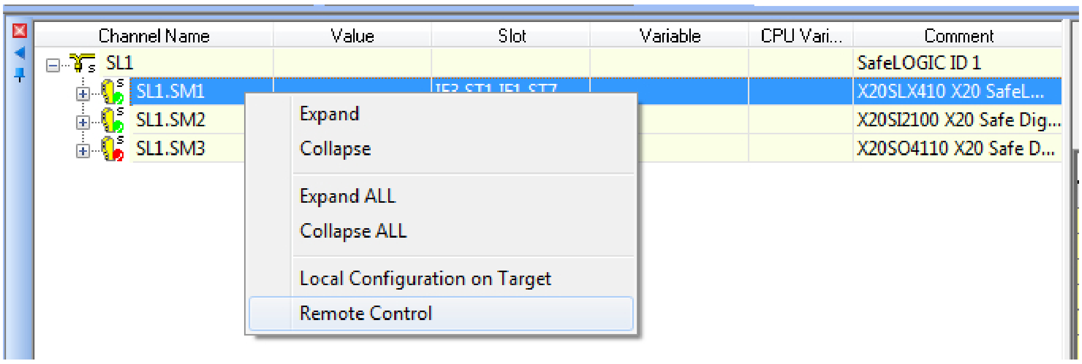
The following table lists the associated objects in Index "0x2402" that can be accessed using the POWERLINK library.

Index:Subindex	Object description	Data type	Access	Value	Description
0x2402:0x00	NumberOfEntries	USINT	R	0x22	Number of entries in this index
0x2402:0x01	EnableAuthorization	UDINT	RW	"AENA", 0x41454E41	Enables authorization
				"ADIS", 0x41444953	Disables authorization
0x2402:0x04	EnableModuleExchange	UDINT	RW	"UDID", 0x55444944	Provides authorization to acknowledge a module replacement
				All other values	Does not provide authorization to acknowledge a module replacement
0x2402:0x05	EnableFWMismatch	UDINT	RW	"FWAC", 0x46574143	Provides authorization to acknowledge a firmware replacement
				All other values	Does not provide authorization to acknowledge a firmware replacement
0x2402:0x06	EnableSKeyExchange	UDINT	RW	"SKEY", 0x534B4559	Provides authorization to acknowledge a SafeKEY replacement
				All other values	Does not provide authorization to acknowledge a SafeKEY replacement

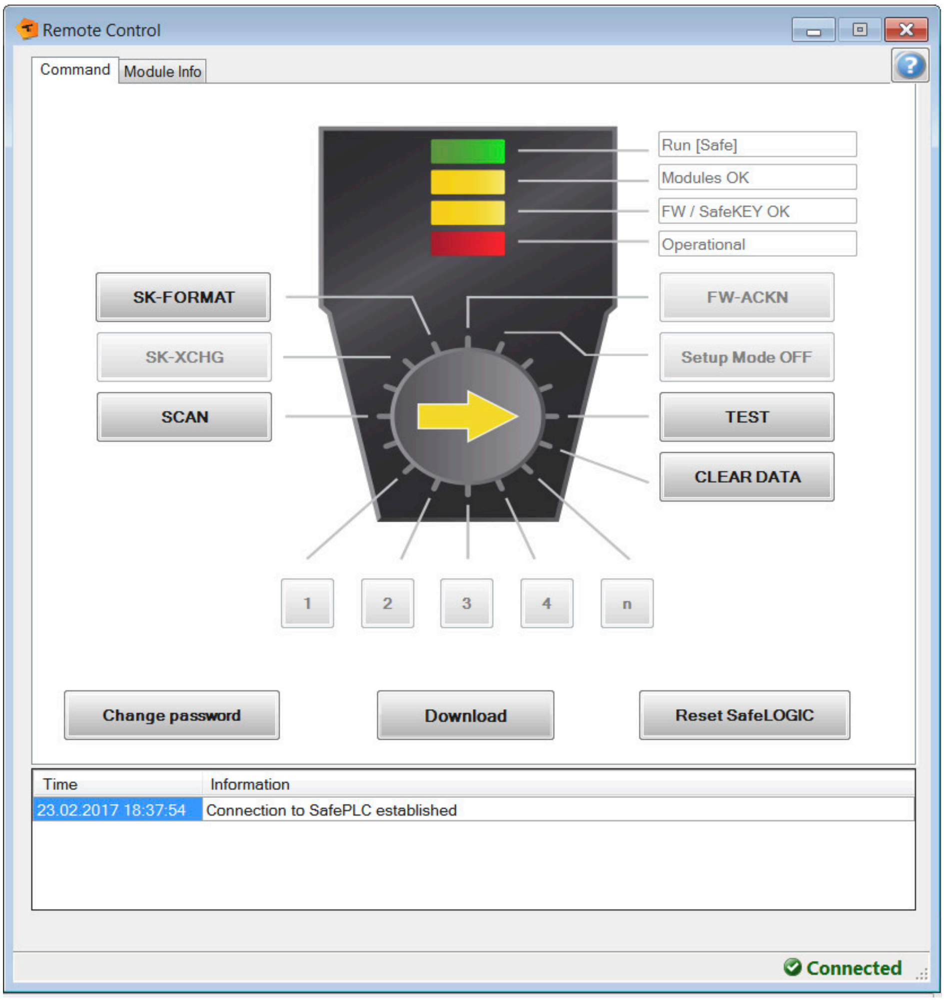
User requests made to the SafeLOGIC controller that are not authorized by the CPU are indicated by a steadily lit ENTER LED.

2.6.8.3.19 Quick start

With the X20SLXxxx series, it is not possible to perform a direct download via the SafePLC window in SafeDESIGNER. The application must be downloaded using the remote interface instead. The remote interface can be accessed from the Safety View.



The password must be entered in order to gain access (or a new password defined at the beginning). Startup must be carried out via the remote interface, as is the case with the X20SL8xxx series via its operating elements.



In addition, the AsSafety library can also be used for startup (see section "Operation via the AsSafety library").

Information:

The possibilities listed above are also available for the X20SL8xxx series starting with Safety Release 1.7.

2.6.8.3.19.1 Download mechanism

Downloading takes place in 2 steps – first to the CompactFlash card and then to the SafeLOGIC-X controller. "Download completed" indicates that the data has been applied during a download to the CompactFlash card.

Information:

The "Download completed" window in SafeDESIGNER is displayed already after downloading to the CompactFlash card. The download to the SafeLOGIC-X controller takes place afterward; it is completed by restarting the SafeLOGIC-X controller.

2.6.8.3.19.2 Visualization

In order to carry out maintenance scenarios, an HMI application must be created using library "AsSafety".

Information:

For details, see Solutions -> Technology Solutions in Automation Help.

2.6.8.3.19.3 Possible data loss

Data for the SafeLOGIC-X controller is stored on the CompactFlash card.

Information:

Note that this data can be lost when reformatting the CompactFlash card, for example.

2.6.8.3.19.4 Necessary resources

Automation Runtime resources are necessary for the safety system.

Information:

When converting from a SafeLOGIC controller to a SafeLOGIC-X controller, note that more Automation Runtime resources are needed for the SafeLOGIC-X controller.

2.6.8.3.20 Software functions

2.6.8.3.20.1 Operation via the AsSafety library

Information about using library "AsSafety" is available under Programming -> Libraries -> Safety -> AsSafety in Automation Help.

2.6.8.3.20.2 Automatic acknowledgment

As specified in previous chapters, automatic acknowledgment is usually not permitted. Provided that the user implements appropriate quality assurance measures and/or constraints, it is nevertheless possible to deviate from this to permit the following automatic acknowledgment.

> ## Danger!
>
> **The automatic acknowledgment of SafeLOGIC controller acknowledgment requests under improper circumstances is not permitted and can lead to dangerous states.**
>
> **It is the sole responsibility of the user to assess the requirements of the safety application in order to determine whether additional measures are necessary.**

"SafeKEY exchange" acknowledgment request

The SafeDESIGNER application and machine option are saved in the safety section of the CompactFlash card (X20SLXxxx series) or on the SafeKEY (X20SL8xxx series). Replacing the CompactFlash card or SafeKEY may result in the unintended exchange of this data. The "SafeKEY exchange" acknowledgment request is meant to prevent this unintentional exchange of data.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards or SafeKEYs:

- The SafeDESIGNER application must be completely validated on a reference machine.
- The machine options file must be completely validated on a reference machine.
- Sufficient measures must be implemented to prevent the SafeDESIGNER application or machine options file from being mixed up across different machine types.
- No test versions of the SafeDESIGNER application or machine options file are permitted.

Under the conditions specified, an automated update of the SafeDESIGNER application or machine options file is permitted to be implemented on the SafeLOGIC/SafeLOGIC-X controller.

"Firmware acknowledge" acknowledgment request

B&R Automation Runtime sees to it independently that the firmware versions stored on the CompactFlash card are transferred to the automation components in the network. This mechanism may cause other firmware versions to be enabled in the system than those that were active when the SafeDESIGNER application was validated. A change to the firmware of the safety modules always requires revalidation of the SafeDESIGNER application. The "Firmware acknowledge" acknowledgment request is meant to prevent an unintentional exchange of firmware versions.

It is important to ensure that the following criteria are met with regard to automatic acknowledgment that potentially involves CompactFlash cards:

- The firmware files installed on the safety modules must be completely validated together with the SafeDESIGNER application on a reference machine.

"UDID mismatch" acknowledgment request

The "UDID mismatch" request occurs in the following situations:

- When modules are exchanged by the user (e.g. during a service call). In this case, it is possible for the connection lines to be mixed up.
- When errors occur in the standard application that lead to a mix-up of modules.

To rule out these mix-ups, a wiring test must be performed after a "UDID mismatch" request is acknowledged.

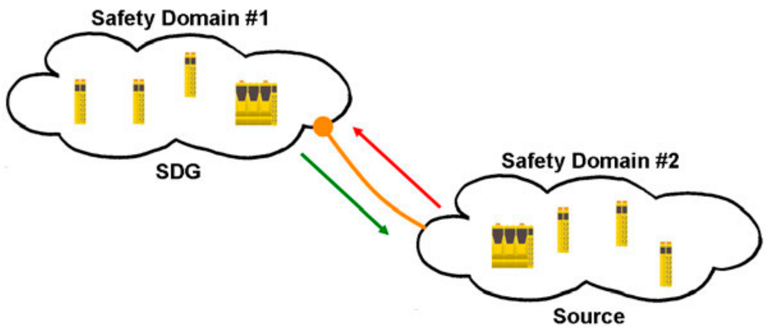
The "UDID mismatch" acknowledgment request is meant to prevent the unintentional mix-up of signals caused by exchanging a module or errors in the standard application.

- Service personnel are to be informed that the mandatory wiring test when exchanging modules must be performed independently of the automatic acknowledgment of the "UDID mismatch" request.
- It is not permitted to use more than 1 module per module type in the Automation Studio application or SafeDESIGNER application.

If the last requirement cannot be met, a "UDID mismatch" acknowledgment request is not permitted to be acknowledged automatically since it would not cover the possible mix-up of signals caused by errors in the standard application.

2.6.8.3.20.3 SafeLOGIC to SafeLOGIC communication

The safety system makes it possible to exchange safety-related information between two safety controllers (SafeLOGIC). SafeLOGIC to SafeLOGIC communication can be used to implement functions such as a global E-stop across a machine network or if a dependency exists between the safety applications on two or more machines. This makes it possible to establish a central collection point for safety information that will be responsible for distributing current values to all relevant locations.



Information:

The safety domain number is taken from the SafeLOGIC ID. In order to use SafeLOGIC to SafeLOGIC communication, the SafeLOGIC IDs must be unique. This uniqueness should be taken into consideration from the very beginning.

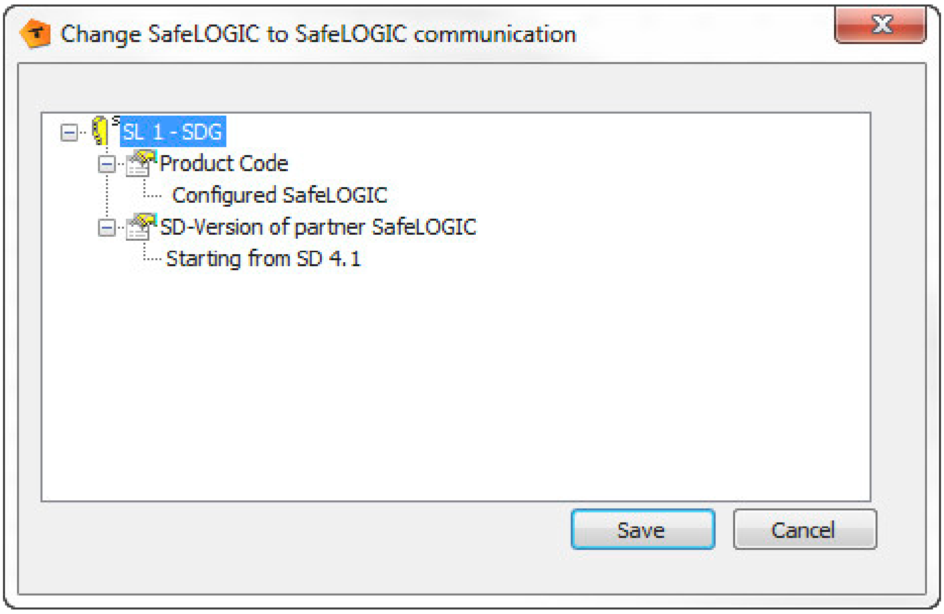
To help with this, a SafeLOGIC controller provides a Safety Domain Gateway (SDG) that can be used to connect additional SafeLOGIC controllers (source controllers). This gateway functionality ensures communication between several safety domains. The connection between source SafeLOGIC controllers and SDG SafeLOGIC controller is indicated in the source SafeLOGIC controller's project as an additional safety module that provides additional communication channels. An SDG SL controller itself can also be used as a source controller and connected to another SDG SL controller. This can be done to achieve cascading communication relationships.

A source SL controller can also be connected several times to the same SDG SL controller, just as it is possible for the source SL controller to communicate with several SDG SL controllers. This results in several ways for SafeLOGIC to SafeLOGIC communication to take place.

System requirements

The following points must be taken into account for safe data exchange between at least 2 SafeLOGIC controllers:

- SafeDESIGNER <4.1: The same SafeDESIGNER versions must be used.
- SafeDESIGNER 4.1 to 4.2.1: The SafeDESIGNER versions must be within this version range.
- SafeDESIGNER 4.2.2 and later: SafeDESIGNER 3.0 or later is permitted to be used.
  The corresponding parameters in the following dialog box must be configured in order to establish a connection to the remote station.



  – Configured SafeLOGIC: Remote station with which communication takes place (e.g. X20SL8100)
  – SD-Version of partner SafeLOGIC: Version with which the application on the remote station was created

Possibilities

The system supports various communication options. The corresponding communication type is defined via parameters in Automation Studio (see "Group: SafeLOGIC to SafeLOGIC communication").

Fixed communication

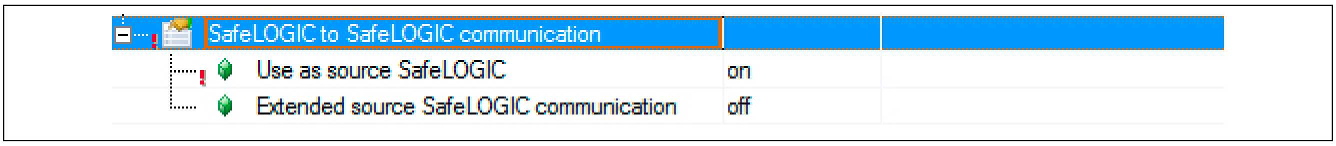
- 8 BOOL channels (1 byte) per communication direction
- One source SL controller can only communicate with one SDG SL controller
- No "any to any" constellation
- Cannot be used with SafeLOGIC-X

Extended communication (Release 1.4 or later and Automation Studio 3.0.90 or later)

- Freely configurable communication channels
- Limited to 16 channels (where 8 BOOLs count as 1 channel; other data types are calculated 1:1).
- One source SL controller can communicate with several SDG SL controllers
- "Any to any" constellation possible

Configuration in Automation Studio

To use SafeLOGIC to SafeLOGIC communication, a SafeLOGIC controller first needs to be configured as a source SL controller. This is done in the I/O configuration.

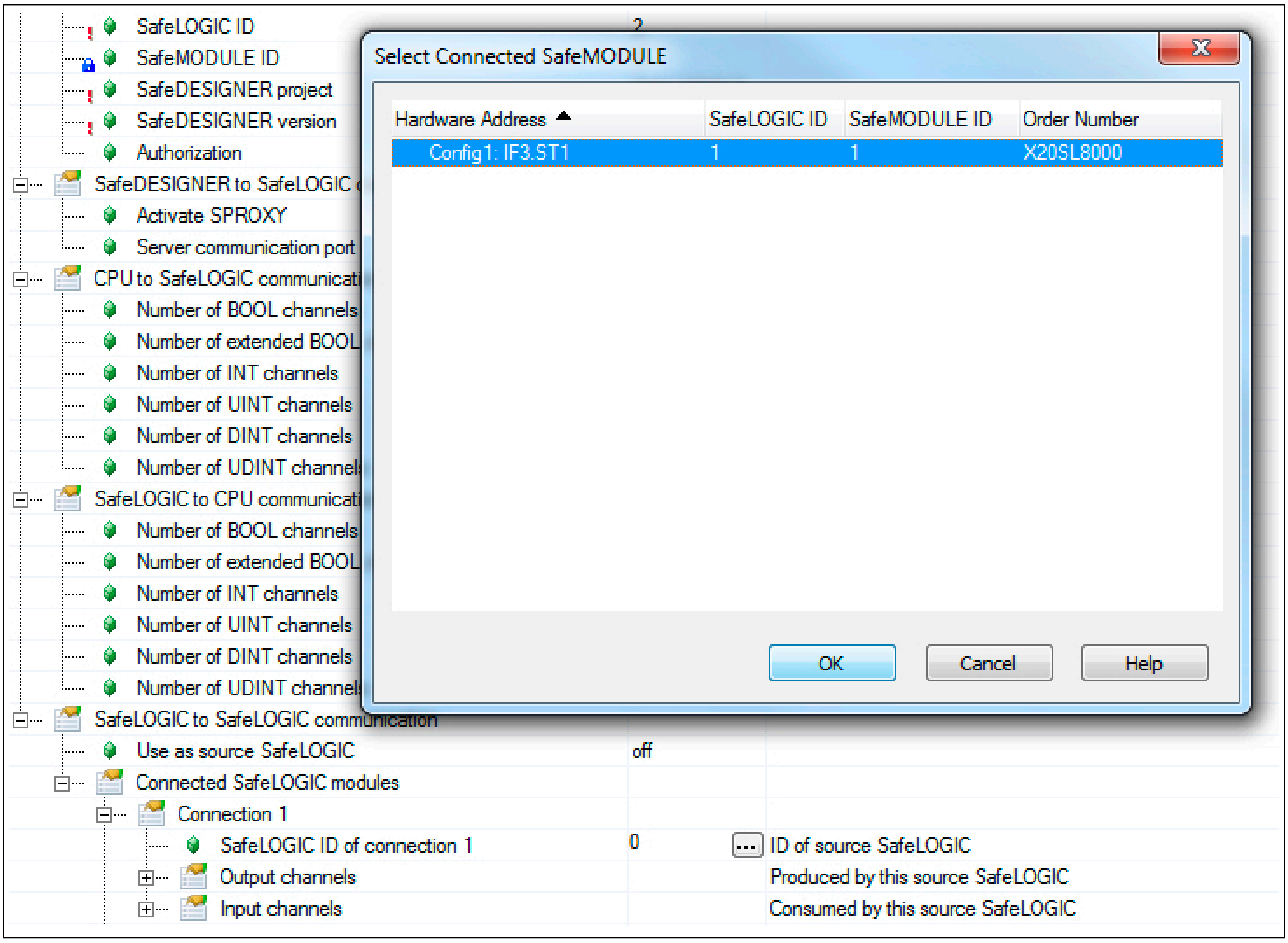


After the "Use as source SafeLOGIC" parameter has been enabled, it is possible to define the type of SafeLOGIC to SafeLOGIC communication as fixed or extended. If the "Extended source SafeLOGIC communication" parameter is not enabled, then fixed communication is used.

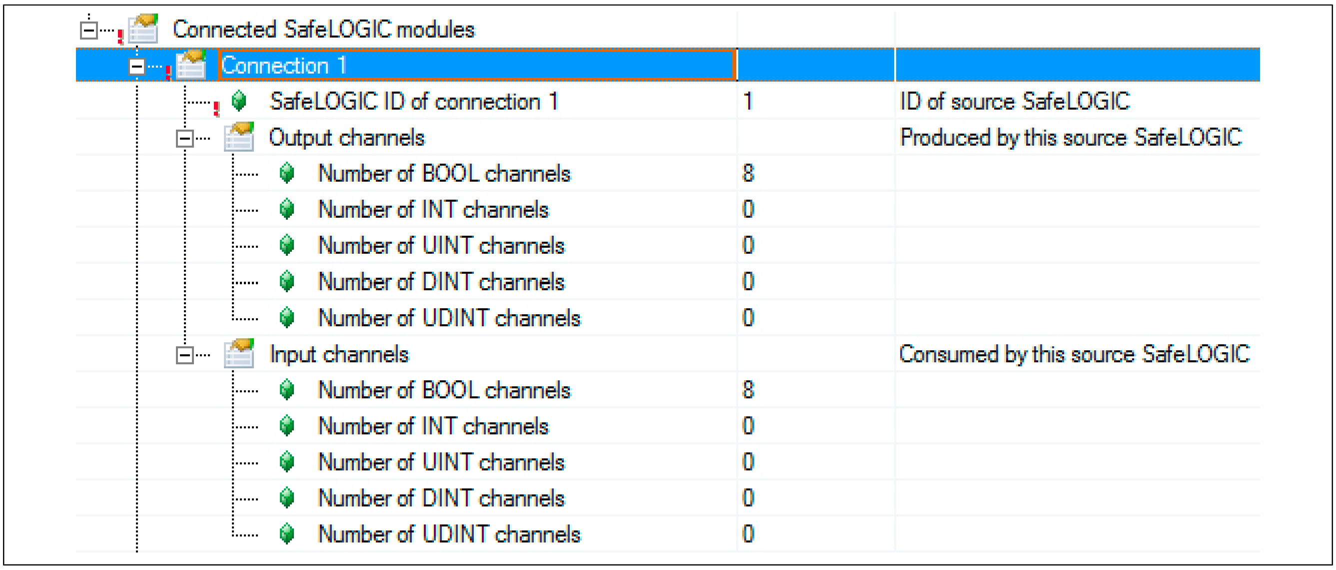
Information:

Changing the type of communication (fixed or extended) at a later time may result in channel overlaps in SafeDESIGNER; the communication channels must therefore be reconnected.

The source SL controller is then connected to the SDG SL controller in the next step. This is done using the connection points in Automation Studio under the I/O configuration of a SafeLOGIC controller (X20SL80x1 and X20SL81xx). Each SafeLOGIC ID (safety domain) is specified from the connection sections using the wizard in Automation Studio.



The necessary communication channels must be defined under each connection. With fixed communication, they are limited to 8 BOOL channels in each direction.



If SafeLOGIC to SafeLOGIC communication should be established between existing or separate Automation Studio projects, several things must be taken into consideration:

- SafeLOGIC IDs must be unique.
- A dummy configuration that includes all safety components must be created on the peer station.
- The dummy configuration must match the real configuration - the SafeMODULE IDs are important here.
- If the projects have multiple iCNs (intelligent controlled nodes), all iCNs must always be taken into account in the iCN project.

Display in SafeDESIGNER

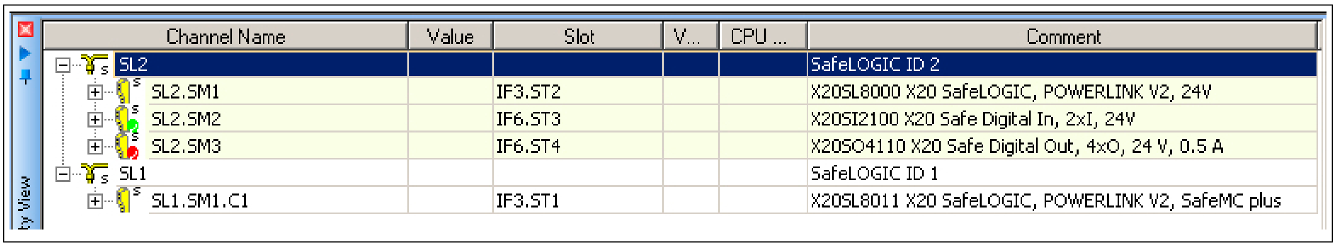
The communication channels are also shown in the SafeDESIGNER project for the respective SafeLOGIC controller (source or SDG).

Danger!

All of the communication channels being used in the project must be mapped in both SafeDESIGNER projects using the same variable names. Channels and variable names are used to calculate a checksum that is then checked at runtime. If the checksum does not match, then the system issues a corresponding logger message in the Safety Logger and communication does not take place.

SafeDESIGNER project – Source SL controller

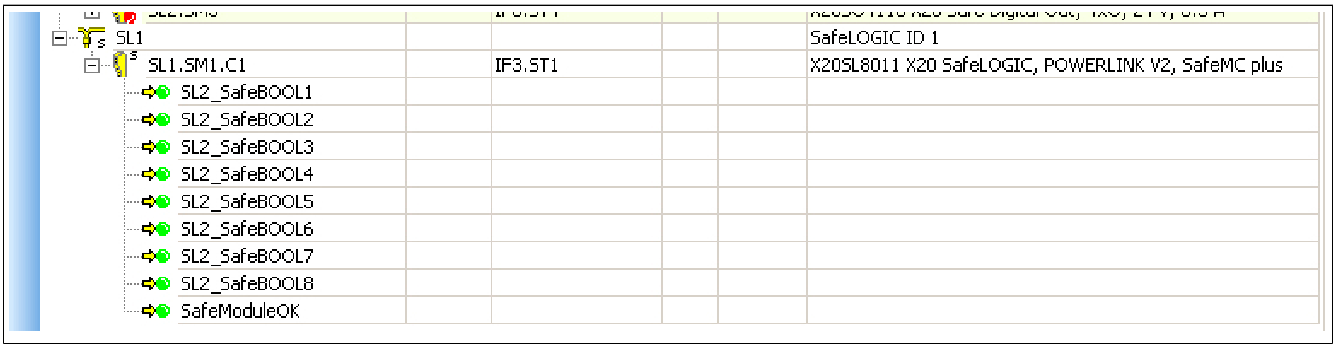
In the source SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.



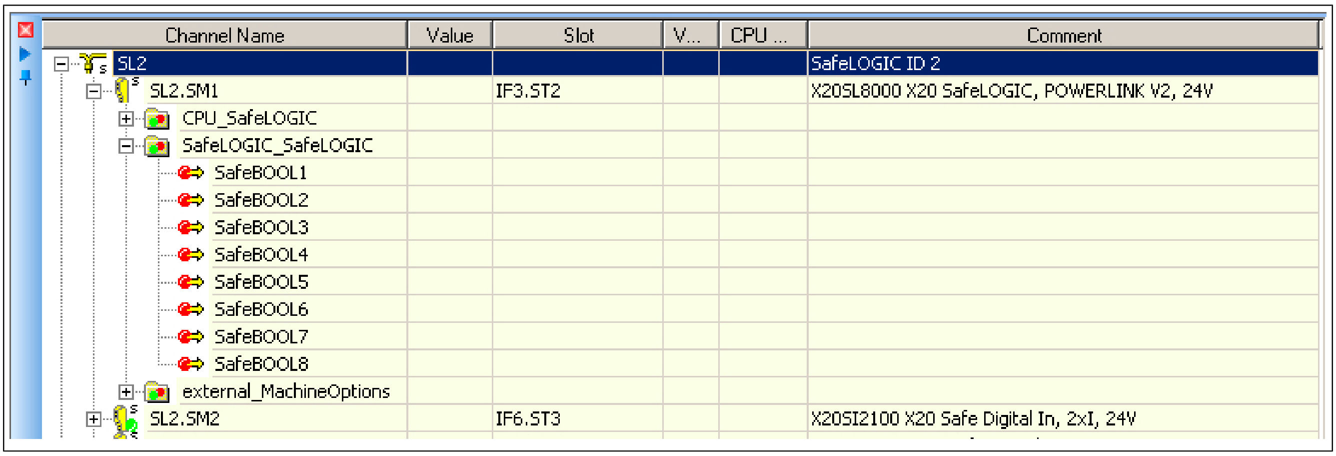
If this module is selected, it is possible to configure its safety-related parameters (see section "Parameters for connection - Release 1.10 and later").

Fixed communication

The input channels sent from the SDG SL controller to the source SL controller and bit information about the status of the connection are listed under the module.

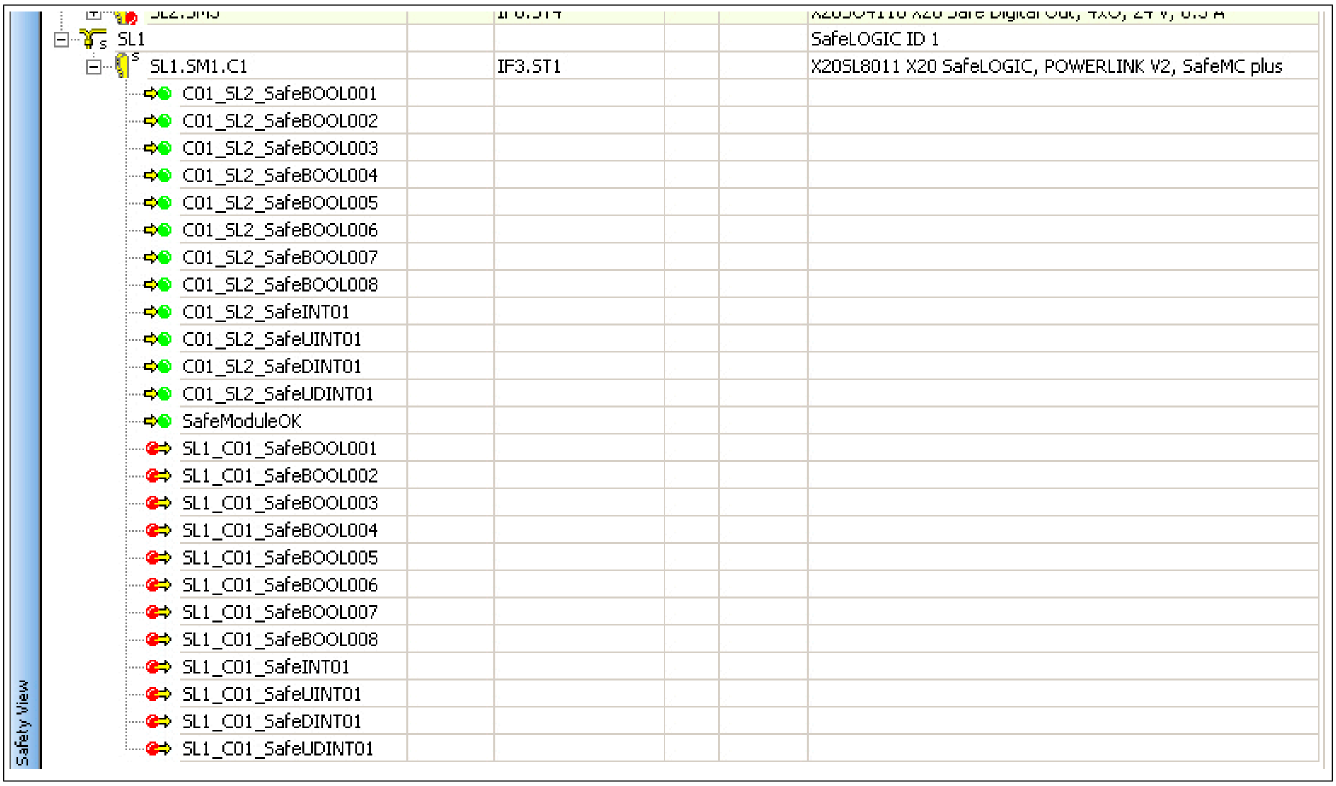


The output channels sent from the source SL controller to the SDG SL controller are listed under the actual SL controller in the project in section "SafeLOGIC_SafeLOGIC".



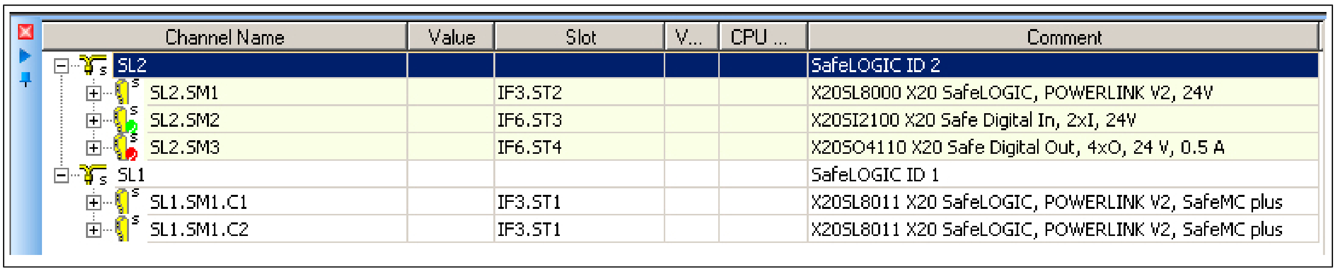
Extended communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

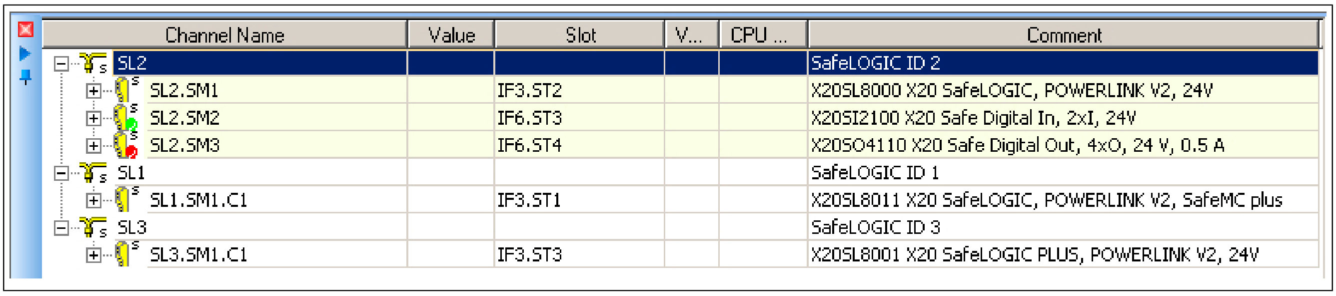


Additional connection

If the source SL controller should be connected once again to the same SDG SL controller, an additional module underneath the same node is available with the necessary parameters and communication channels.



If the source SL controller should be connected to another SDG SL controller, an additional node for the safety domain as well as a module with the necessary parameters and communication channels is available.



SafeDESIGNER project – SDG SL controller

In the SDG SL controller's SafeDESIGNER project, communication is indicated by an additional module. This module has its own node that represents the connection to this safety domain.

Channel Name	Value	Slot	V...	CPU ...	Comment
SL1					SafeLOGIC ID 1
SL1.SM1		IF3.ST1			X20SL8011 X20 SafeLOGIC, POWERLINK V2, SafeMC plus
SL1.SM2		IF6.ST1			X20SI4100 X20 Safe Digital In, 4xI, 24V
SL1.SM3		IF6.ST2			X20SO2120 X20 Safe Digital Out, 2xO, 24 V, 2A
SL2					SafeLOGIC ID 2
SL2.SM1.C1		IF3.ST2			X20SL8000

Information:

No connection parameters are available in the SDG SL controller's project. They must be configured in the source SL controller's project.

Fixed communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

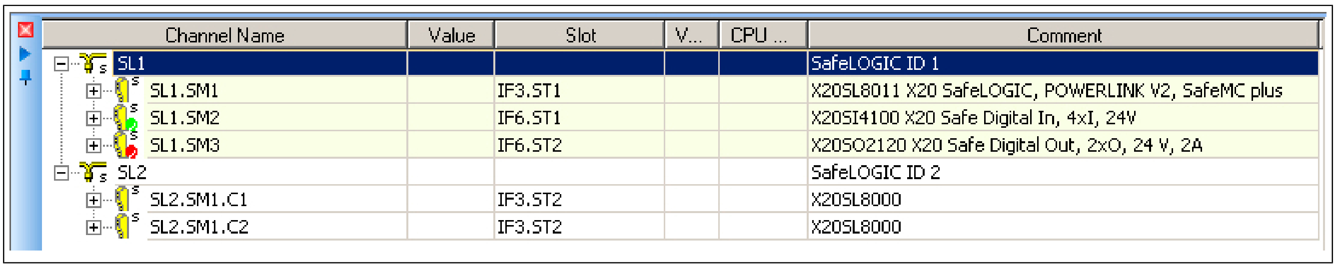
SL1.SM3		IF6.ST2			X20SO2120 X20 Safe Digital Out, 2xO, 24 V, 2A
SL2					SafeLOGIC ID 2
SL2.SM1.C1		IF3.ST2			X20SL8000
SafeBOOL1					
SafeBOOL2					
SafeBOOL3					
SafeBOOL4					
SafeBOOL5					
SafeBOOL6					
SafeBOOL7					
SafeBOOL8					
SafeModuleOK					
SL2_SafeBOOL1					
SL2_SafeBOOL2					
SL2_SafeBOOL3					
SL2_SafeBOOL4					
SL2_SafeBOOL5					
SL2_SafeBOOL6					
SL2_SafeBOOL7					
SL2_SafeBOOL8					

Extended communication

The input channels, output channels and bit information regarding the status of the connection are listed under the module.

Additional connection

If the source SL controller should be connected once again to the SDG SL controller, an additional module underneath the same node is available with the necessary communication channels.



Parameters for connection - up to Release 1.9

Safety Release 1.4 or higher:

Cycle time parameters are also available for communication in order to define the "Worst_Case_Response_Time_us". As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Parameter	Value
Basic	
Min_required_FW_Rev	Basic Release
Optional	No
External_UDID	No
Safety_Response_Time	
Synchronous_Network_Only	Yes
Max_SDG_Powerlink_CycleTime_us	5000
Max_Powerlink_CycleTime_us	5000
Max_CPU_CrossLinkTask_CycleTime_us	5000
Min_SDG_Powerlink_CycleTime_us	200
Min_Powerlink_CycleTime_us	200
Min_CPU_CrossLinkTask_CycleTime_us	0
Worst_Case_Response_Time_us	100000
Max_SDG_Cycle_Time_us	5000
Min_SDG_Cycle_Time_us	1600
Slow_Connection	No

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 111: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks.</td></tr></table></td></tr>
<tr><td>Max_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the maximum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum cycle time of the POWERLINK network in which the other SafeLOGIC controller is operated.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 30,000 µs (corresponds to 0.2 to 30 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_CycleTime_us</td><td>This parameter specifies the minimum cycle time for copying data between the two POWERLINK networks. The value 0 means that both SafeLOGIC controllers are in the same POWERLINK network.
Permissible values: 0 to 3,000,000 µs (corresponds to 0 to 3 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 12,500,000 µs (corresponds to 3 ms to 12.5 s)
Note:
Keep parameter "Slow_Connection" in mind when entering large values here!</td><td>100000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
<tr><td>Max_SDG_Cycle_Time_us</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_SDG_Cycle_Time_us</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
<tr><td rowspan="2">Slow_Connection</td><td>This parameter specifies whether this connection is a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr><tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr></table></td></tr>
</table>

Table 112: SafeDESIGNER parameters: Safety_Response_Time

Information:

Parameter "CPU_CrossLinkTask_CycleTime_us" is needed if the source SL and SDG SL controllers are in different networks or located on different controllers. If this is not the case, the minimum and maximum value must be set to "0".
For this parameter, the entire connection distance between the controllers must be taken into account – including copy times between the interfaces involved.

Information:

Parameter "Slow_Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow_Connection = Yes").

Parameters for connection - Release 1.10 and later

Cycle time parameters are also available for communication in order to define the maximum data transmission time. As with communication that takes place with other safety modules, this is a timeout value that elapses whenever an error occurs (e.g. lost network connection).

Information:

Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module to the source SafeLOGIC controller, the parameters for the connection are available and must be configured in the source SL controller's project.

Materialnummer: **X20SL8100**
Description: **X20 SafeLOGIC, POWERLINK V2, 24V, univ.**
SafeMODULE ID: **3**
Import file: -

Parameter	Value	Unit
Basic		
Min required FW Rev	Basic Release	
Optional	No	
External UDID	No	
Safety Response Time		
Synchronous Network Only	Yes	
Safe Data Duration	20000	us
Additional Tolerated Packed Loss	0	packets
Slow Connection	No	
Node Guarding Lifetime	5	iterations
Max SDG Cycle Time	5000	us
Min SDG Cycle Time	1600	us

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 113: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permitted data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td rowspan="2">Slow Connection</td><td>This parameter specifies whether this connection is classified as a slow connection.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally).
Rule of thumb: "Yes" from ratio 50:1 (telegram runtime: SafeLOGIC cycle time)</td></tr><tr><td>No</td><td>Default connection, parameter calculation unchanged</td></tr></table></td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
<tr><td>Max SDG Cycletime</td><td>This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min SDG Cycletime</td><td>This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safety response time.
• Permissible values: 800 to 20,000 µs (corresponds to 0.8 to 20 ms)</td><td>1600</td><td>µs</td></tr>
</table>

Table 114: SafeDESIGNER parameters: Safety Response Time

Information:

Parameter "Slow Connection" can also be used to specify that the connection between the source SL and SDG SL controllers is slow. If a value of just a few seconds is needed for the connection timeout, then this parameter must be enabled ("Slow Connection = Yes").

2.6.8.3.20.4 Setup mode

Setup mode supports the user during commissioning.

Setup mode is supported in hardware upgrade 1.10.2.x and later.
Automation Runtime B4.26 or higher is required to use setup mode.

Active setup mode is indicated by both the FAILSAFE LED (X20SL81xx series) or SE LED (X20SLXxxx series) as well as an entry in the logbook.

When setup mode is active, acknowledgment requests "SafeKEY exchange", "Firmware acknowledge" and "UDID mismatch" are no longer necessary.

Setup mode can be enabled and disabled using the operating elements of the "Remote Control" in SafeDESIGNER (X20SL81xx and X20SLXxxx series) or using the selector switch and acknowledgment button (X20SL81xx series).

Danger!

Setup mode is only permitted to be enabled during the commissioning of the machine/system.
Setup mode must be disabled during operation.

Danger!

After setup mode is ended, functional testing including a wiring test must be carried out.
If a SafeKEY or SafeLOGIC controller is replaced while setup mode is active, then setup mode will be disabled.
Functional testing must also be carried out in this case.

Functional testing is only permitted to be performed by personnel familiar with the safety application and its functions.

Be sure to validate the entire safety function!

2.6.9 Digital input modules

2.6.9.1 Overview

Model number	Short description	Page
X20SI2100	X20 safe digital input module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC	372
X20SI4100	X20 safe digital input module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20cSI4100	X20 safe digital input module, coated, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20SI8110	X20 safe digital input module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width	372
X20SI9100	X20 safe digital input module, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372
X20cSI9100	X20 safe digital input module, coated, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC	372

2.6.9.2 X20(c)SIx1x0

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 115: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 116: Organization of general notices

2.6.9.2.1 General information

The modules are equipped with 2 to 20 safe digital inputs. They are designed for a nominal voltage of 24 VDC.

The modules can be used to read in digital signals in safety-related applications up to PL e or SIL 3.

The modules are equipped with filters that are individually configurable for switch-on and switch-off behavior. The modules also provide pulse signals for diagnosing the sensor line.

These modules are designed for X20 12-pin terminal blocks.

- 2 to 20 safe digital inputs
- 2 to 4 pulse outputs
- Sink circuit
- Software input filter configurable for each channel

2.6.9.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.9.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.9.2.2 Overview

Module	X20SI2100	X20SI4100	X20SI8110	X20SI9100
Safe digital inputs				
Number of inputs	2	4	8	20
Nominal voltage	24 VDC			
Input filter				
Hardware	≤150 µs			
Software	Default 0 ms, configurable between 0 and 500 ms			
Input circuit	Sink			
Pulse outputs				
Design	Push-Pull			
Switching voltage	I/O power supply minus residual voltage			

Table 117: Digital input modules

2.6.9.2.3 Order data

X20SI2100 / X20SI4100 X20SI8110 X20SI9100

Model number	Short description
	Digital input modules
X20SI2100	X20 safe digital input module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC
X20SI4100	X20 safe digital input module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20cSI4100	X20 safe digital input module, coated, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20SI8110	X20 safe digital input module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, single-width
X20SI9100	X20 safe digital input module, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
X20cSI9100	X20 safe digital input module, coated, 20 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC
	Required accessories
	Bus modules
X20BM13	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous, single-width
X20BM16	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous, single-width
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 118: X20SI2100, X20SI4100, X20cSI4100, X20SI8110, X20SI9100, X20cSI9100 - Order data

2.6.9.2.4 Technical data

Model number	X20SI2100	X20SI4100	X20cSI4100	X20SI8110	X20SI9100	X20cSI9100
Short description						
I/O module	2 safe digital inputs, 2 pulse outputs, 24 VDC	4 safe digital inputs, 4 pulse outputs, 24 VDC		8 safe digital inputs, 4 pulse outputs, 24 VDC	20 safe digital inputs, 4 pulse outputs, 24 VDC	
General information						
B&R ID code	0x1F15	0x1DBD	0xDD5A	0xE742	0xAEC8	0xDD5B
System requirements						
Automation Studio	3.0.71 or later		4.0.16 or later	4.0 or later	3.0.81.15 or later	4.0.16 or later
Automation Runtime	2.95 or later		V3.08 or later	4.0 or later	3.00 or later	V3.08 or later
SafeDESIGNER	2.58 or later		3.1.0 or later	3.4.0 or later	2.71 or later	3.1.0 or later
Safety Release	1.1 or later		1.7 or later		1.3 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status					
Diagnostics						
Module run/error	Yes, using status LED and software					
Inputs	Yes, using status LED and software					
Blackout mode						
Scope	Module					
Function	Module function					
Standalone mode	No					
Max. I/O cycle time	800 µs			1 ms	1600 µs	
Power consumption						
Bus	0.25 W	0.32 W		0.4 W		
Internal I/O	1 W	1.25 W		2.5 W	1.6 W	
Electrical isolation						
Channel - Bus	Yes					
Channel - Channel	No					
Certifications						
CE	Yes					
KC	Yes		-		Yes	-
EAC	Yes					
UL	cULus E115267 Industrial control equipment			cULus E115267 Industrial con- trol equipment	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5			-	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X					
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)			In preparation	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
LR	ENV1			-		
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013					
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3					
Functional safety	EN 50156-1:2004					
Safety characteristics						
EN ISO 13849-1:2015						
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1)]					
PL	PL e					
DC	>94%					
MTTFD	2500 years					
Mission time	Max. 20 years					
IEC 61508:2010, IEC 61511:2004, EN 62061:2013						
SIL CL	SIL 3					
SFF	>90%					
PFH / PFH_d						
Module	$<1*10^{-10}$					
openSAFETY wired	Negligible					
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour					
PFD	$<2*10^{-5}$					
Proof test interval (PT)	20 years					

Table 119: X20SI2100, X20SI4100, X20cSI4100, X20SI8110, X20SI9100, X20cSI9100 - Technical data

Model number	X20SI2100	X20SI4100	X20cSI4100	X20SI8110	X20SI9100	X20cSI9100
I/O power supply						
Nominal voltage	24 VDC					
Voltage range	24 VDC -15% / +20%					
Integrated protection	Reverse polarity protection					
Safe digital inputs						
Nominal voltage	24 VDC					
Input characteristics per EN 61131-2	Type 1					
Input filter						
Hardware	≤150 µs					
Software	Configurable between 0 and 500 ms					
Input circuit	Sink					
Input voltage	24 VDC -15% / +20%					
Input current at 24 VDC	Max. 4.59 mA, hardware revision J0 and later: Max. 3.28 mA			Max. 3.28 mA		
Input resistance	Min. 5.23 kΩ, hardware revision J0 and later: Min. 7.33 kΩ			Min. 7.33 kΩ		
Error detection time	100 ms				200 ms	
Isolation voltage between channel and bus	500 V_{eff}					
Switching threshold						
Low	<5 VDC					
High	>15 VDC					
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line					
Pulse outputs						
Variant	Push-Pull					
Nominal output current	100 mA, hardware revision J0 and later: 50 mA			50 mA		
Output protection	Thermal shutdown of all channels in the event of overload or short circuit Hardware revision J0 and later: Shutdown of individual channels in the event of overload or short circuit [2]			Shutdown of individual channels in the event of overload or short circuit [2]		
Peak short-circuit current	300 mA, hardware revision J0 and later: 25 A for 15 µs			0.5 A for 120 µs	25 A for 5 ms, hardware revision D0 and later: 25 A for 15 µs	
Short-circuit current	100 mA_{eff}			15 mA_{eff}	100 mA_{eff}	
Leakage current when switched off	0.1 mA					
Residual voltage	Max. 0.6 VDC at 100 mA, hardware revision J0 and later: 2 VDC			≤4 VDC	0.3 VDC, hardware revision D0 and later: 3 VDC	
Switching voltage	I/O power supply minus residual voltage					
Total nominal current	200 mA, hardware revision J0 and later: 100 mA	400 mA, hardware revision J0 and later: 200 mA		200 mA		
Operating conditions						
Mounting orientation						
Horizontal	Yes					
Vertical	Yes					
Installation elevation above sea level	0 to 2000 m, no limitation					
Degree of protection per EN 60529	IP20					
Ambient conditions						
Temperature						
Operation						
Horizontal mounting orientation	0 to 60°C		-40 to 60°C [3]	0 to 60°C		-40 to 60°C [3]
Vertical mounting orientation	0 to 50°C		-40 to 50°C [4]	0 to 50°C		-40 to 50°C [4]
Derating	See section "Derating".					
Storage	-40 to 85°C					
Transport	-40 to 85°C					
Relative humidity						
Operation	5 to 95%, non-condensing		Up to 100%, condensing	5 to 95%, non-condensing		Up to 100%, condensing
Storage	5 to 95%, non-condensing					
Transport	5 to 95%, non-condensing					
Mechanical properties						
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.			Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module (single-width) separately.	Order 2x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm			$12.5^{+0.2}$ mm	$25^{+0.2}$ mm	

Table 119: X20SI2100, X20SI4100, X20cSI4100, X20SI8110, X20SI9100, X20cSI9100 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Up to hardware upgrade <1.10.1.0: -25 to 60°C
4) Up to hardware upgrade <1.10.1.0: -25 to 50°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SI2100	X20SI4100	X20SI8110	X20SI9100
Derating bonus				
At 24 VDC	+2.5°C			+5°C
At 20.4 VDC	+2.5°C		+5°C	+5°C
Dummy module on the left	+0°C		+2.5°C	+0°C
Dummy module on the right	+2.5°C			
Dummy module on the left and right	+5°C			
Pulse outputs	+0°C		+10°C [1]	+0°C
4 safe inputs (SI)	+0°C		+2.5°C [2]	+0°C
With double PFH / PFH_d	+0°C		+15°C [3]	+0°C

Table 120: Derating bonus

1) Pulse output loaded with maximum 2 safe inputs (SI)
2) Only 4 safe inputs (SI) in use
3) Hardware revision E0 or later and hardware upgrade 1.10.1.0 or later

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

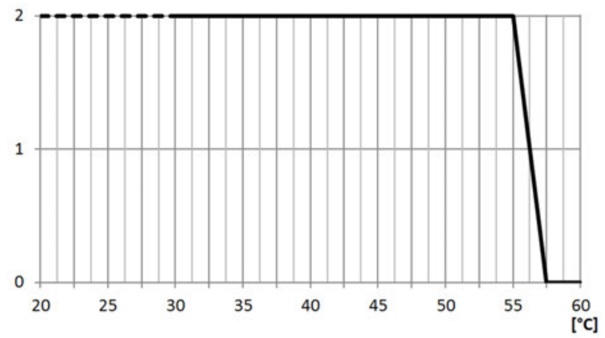
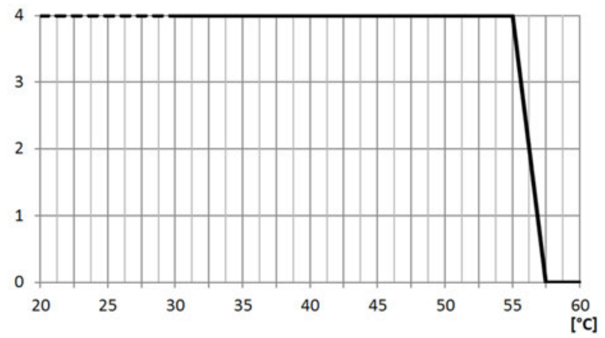
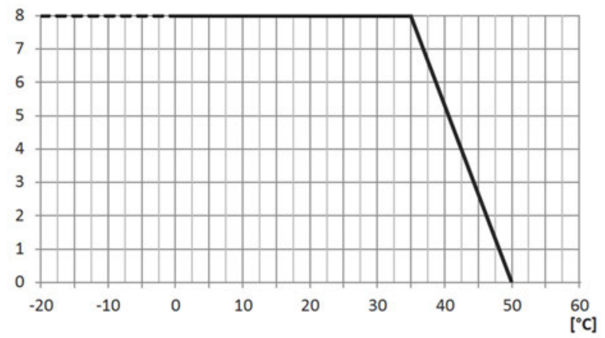
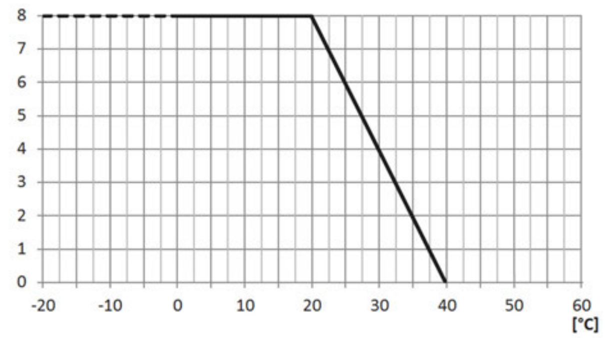
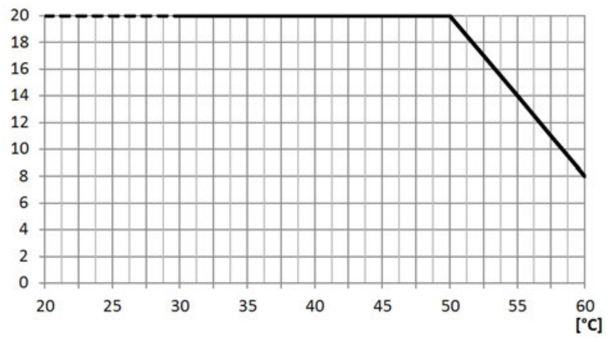
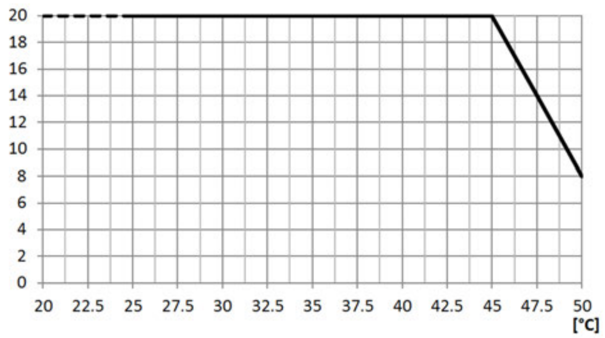
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
X20SI2100	
	No derating
X20SI4100	
	No derating
X20SI8110	
	
X20SI9100	
	

Table 121: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.9.2.5 LED status indicators

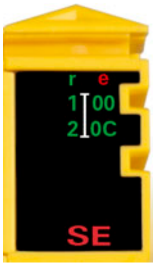
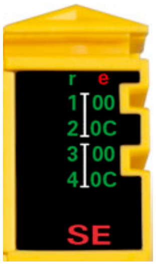
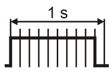
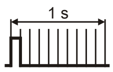
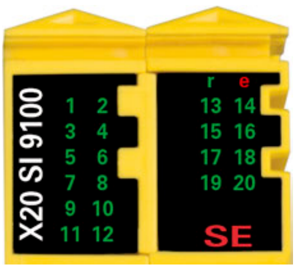
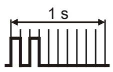
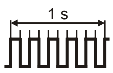
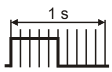
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 20	Input state of the corresponding digital input The number of channel LEDs varies depending on the number of channels on the module type.		
X20SI2100		Red	On	Warning/Error on an input channel
			Blinking (only for X20SI9100 and X20SI8110)	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Input set
	OO	These LEDs may not be available depending on the module type. Errors in dual-channel evaluation are indicated by channel LEDs 1 to 20 in this case.		Open - Open: Dual-channel evaluation on channels 1 and 2 using the "Equivalent" function block
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
X20SI4100		Green	On	Evaluation channel set
	OC	These LEDs may not be available depending on the module type. Errors in dual-channel evaluation are indicated by channel LEDs 1 to 20 in this case.		Open - Closed: Dual-channel evaluation on channels 1 and 2 using the "Antivalent" function block
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Evaluation channel set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
X20SI8110				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
X20SI9100			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 122: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.9.2.6 Pinouts

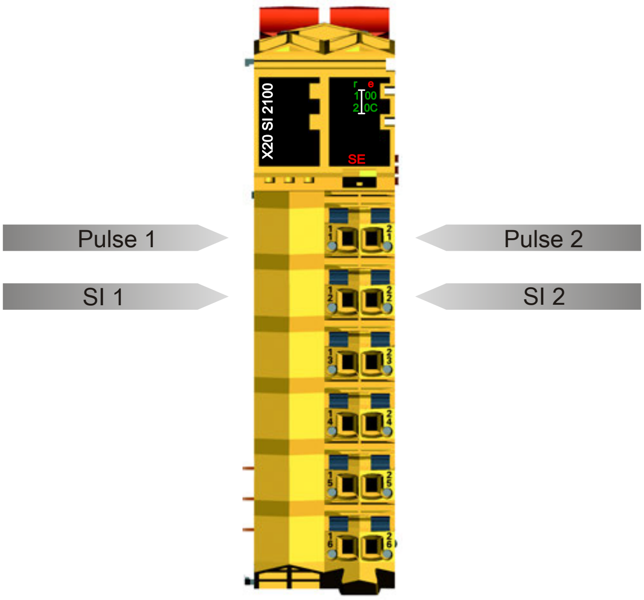


Figure 113: X20SI2100 - Pinout

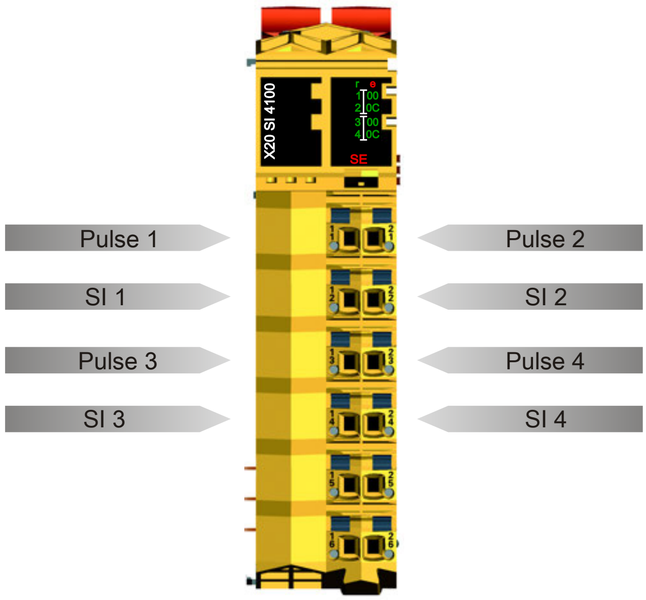


Figure 114: X20SI4100 - Pinout

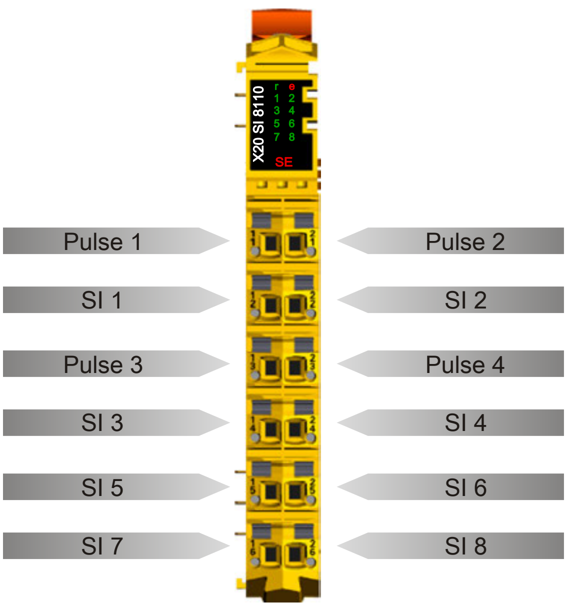


Figure 115: X20SI8110 - Pinout

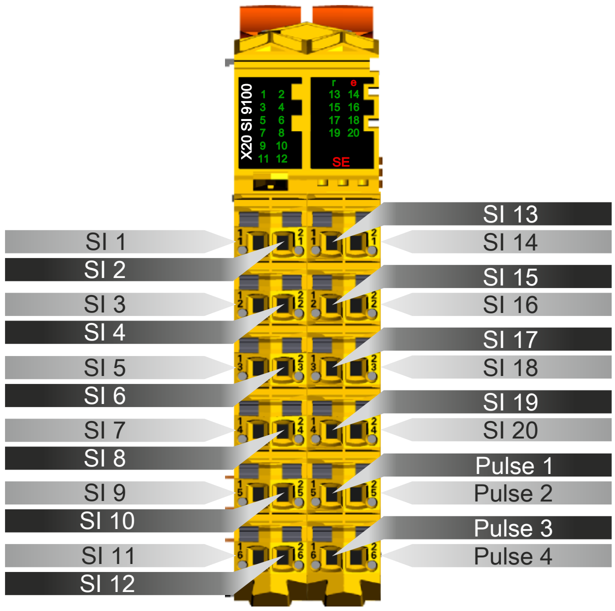


Figure 116: X20SI9100 - Pinout

2.6.9.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

> **Information:**
>
> **For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).**

2.6.9.2.7.1 Connecting single-channel sensors with contacts

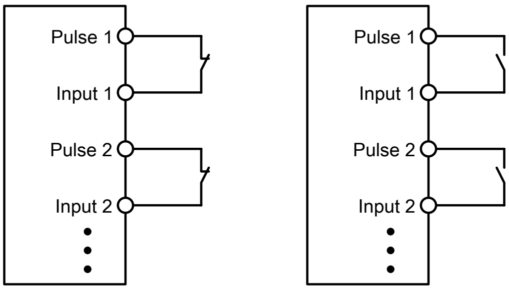


Figure 117: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.9.2.7.2 Connecting two-channel sensors with contacts

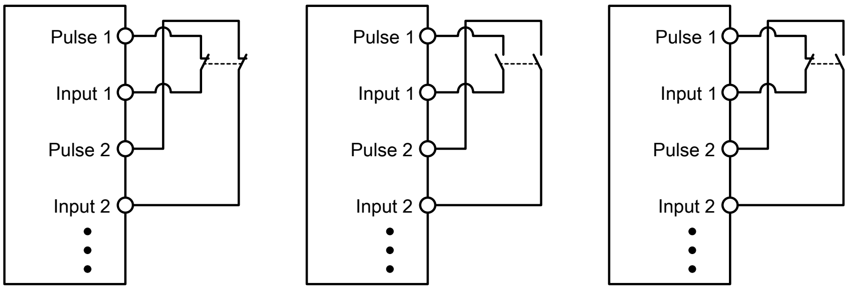


Figure 118: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.9.2.7.3 Connecting multi-channel sensors with contacts

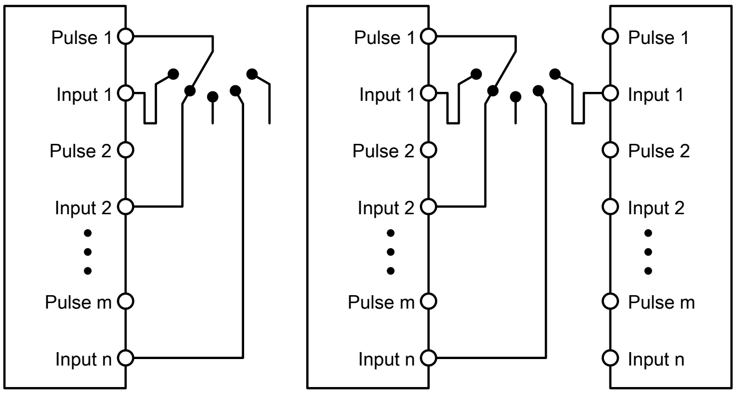


Figure 119: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.9.2.7.4 Connecting electronic sensors

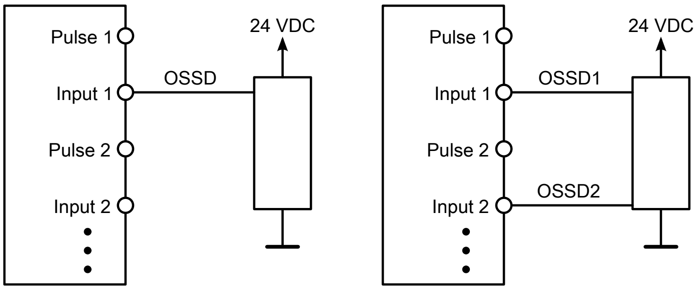


Figure 120: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.9.2.7.5 Using the same pulse signals

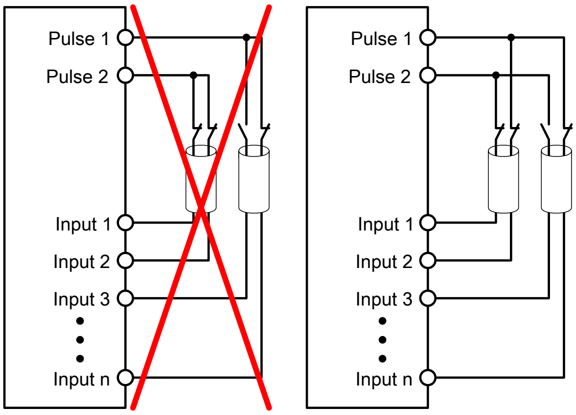


Figure 121: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.9.2.8 Error detection

2.6.9.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

2.6.9.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 123: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 124: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 125: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

2.6.9.2.9 Input circuit diagram

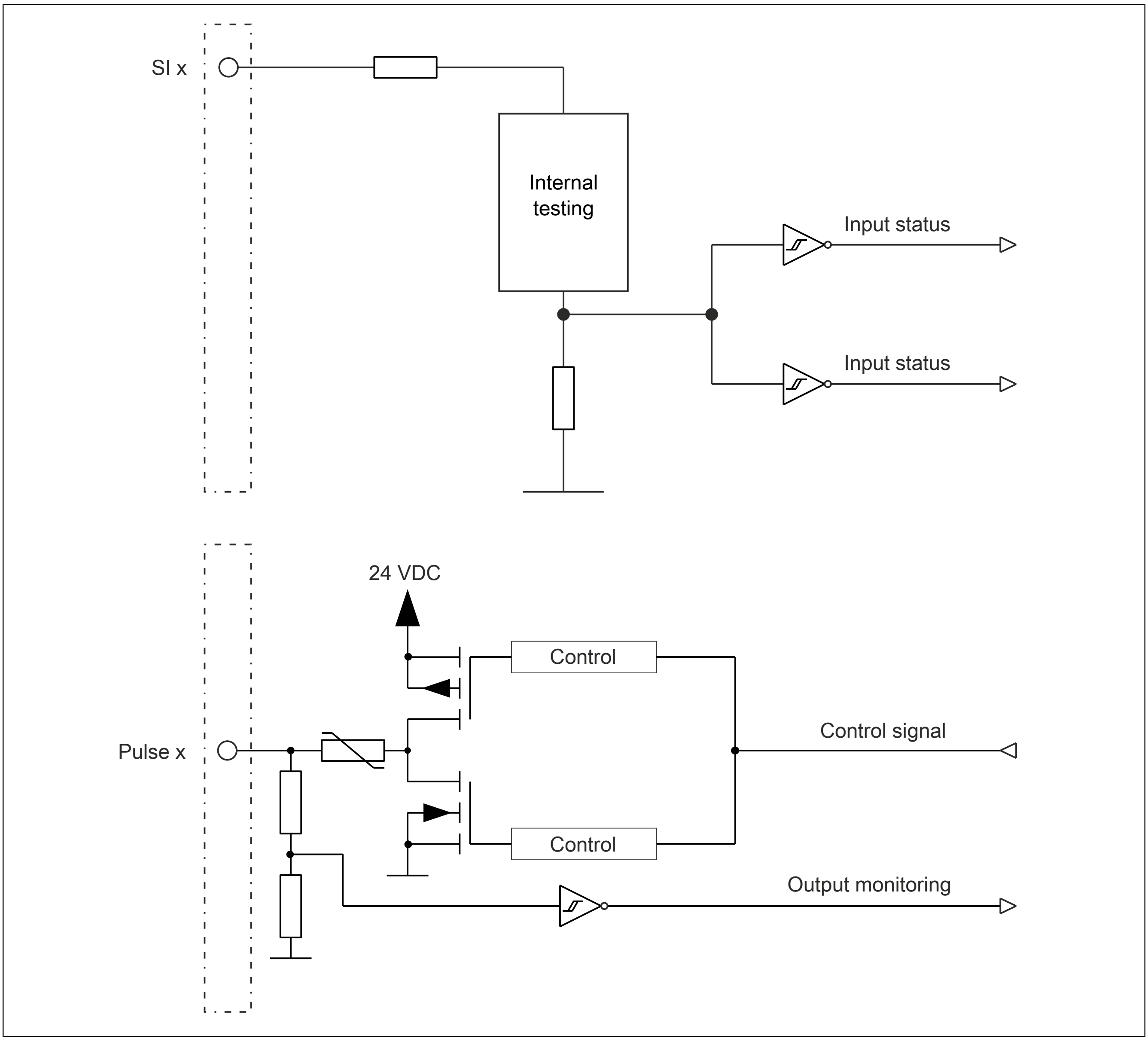


Figure 122: Input circuit diagram

2.6.9.2.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.9.2.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time			
X20SI2100	X20SI4100	X20SI8110	X20SI9100
400 µs	400 µs	500 µs	800 µs

Maximum I/O update time			
X20SI2100	X20SI4100	X20SI8110	X20SI9100
1750 µs + Filter time (see chapter "Filter")	1750 µs + Filter time (see chapter "Filter")	1150 µs + Filter time (see chapter "Filter")	3350 µs + Filter time (see chapter "Filter")

2.6.9.2.12 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

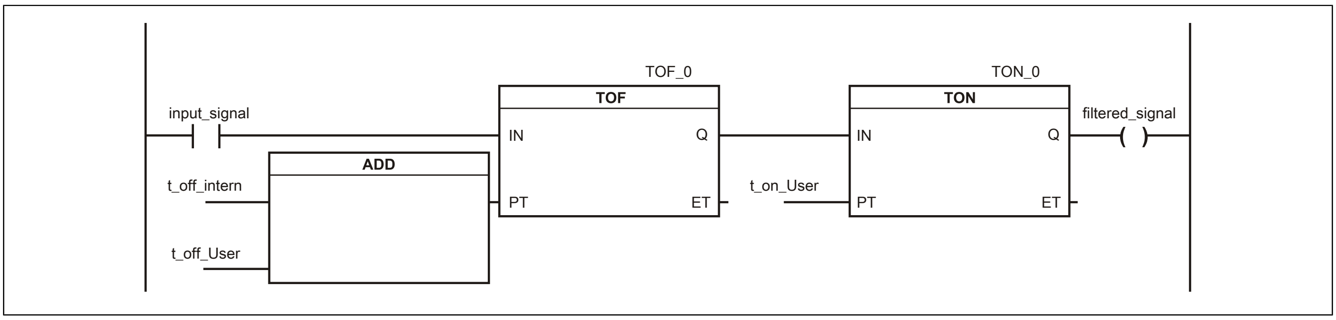


Figure 123: SI input filter - Diagram 1

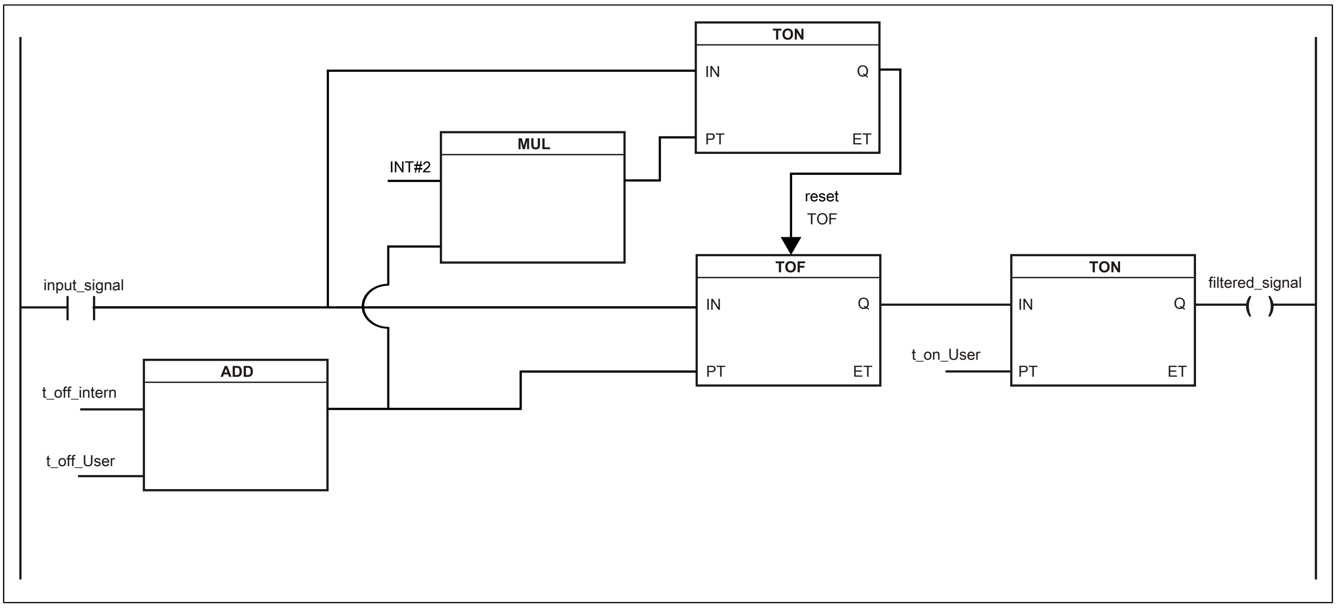


Figure 124: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.9.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.9.2.14 Register description

2.6.9.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 126: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Module supervised</td><td>System behavior when a module is missing
<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td><td>On</td><td>-</td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td>Blackout mode
(hardware upgrade 1.10.0.5 or later, X20SI8110: hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).
<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td><td>Off</td><td>-</td></tr>
<tr><td>Input status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.
This parameter may not be available depending on the module type.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 127: I/O configuration parameters: General

2.6.9.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 128: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 μs (corresponds to 0 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>μs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>μs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 μs (corresponds to 0 to 25 ms)</td><td>0</td><td>μs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 μs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>μs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 129: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Pulse_Source
(Release 1.4 and later)</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>See table.</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="4">Possible "Pulse_Source" for X20SI2100 and X20SI4100</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td></tr>
</table>
All available pulse outputs on the X20SI8110 and X20SI9100 can be specified as "Pulse_Source".
The default values can be determined using the following tables.
<table>
<tr><th>Channel</th><th>Default "Pulse_Source" for X20SI8110</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table>
<table>
<tr><th>Channel</th><th>Default "Pulse_Source" for X20SI9100</th></tr>
<tr><td>1, 3, 5, 7, 9, 11</td><td>Channel 1</td></tr>
<tr><td>2, 4, 6, 8, 10, 12</td><td>Channel 2</td></tr>
<tr><td>13, 15, 17, 19</td><td>Channel 3</td></tr>
<tr><td>14, 16, 18, 20</td><td>Channel 4</td></tr>
</table>
Note:
If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".
</td></tr>
<tr><td rowspan="2">Pulse_Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.
Release 1.4 and later:
The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external" (X20SI2100 and X20SI4100 only).</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter_On_us</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>X20SI2100 and X20SI4100: 100000
X20SI8110 and X20SI9100: 200000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)
(up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>TwoChannelProcessingMode
(only for X20SI8110 and X20SI9100)</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 130: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

2.6.9.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 131: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 μs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>μs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 132: SafeDESIGNER parameters: Safety Response Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>See table.</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="4">Possible "Pulse Source" for X20SI2100 and X20SI4100</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td></tr>
</table>
All available pulse outputs on the X20SI8110 and X20SI9100 can be specified as "Pulse Source".
The default values can be determined using the following tables.
<table>
<tr><th>Channel</th><th>Default "Pulse Source" for X20SI8110</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table>
<table>
<tr><th>Channel</th><th>Default "Pulse Source" for X20SI9100</th></tr>
<tr><td>1, 3, 5, 7, 9, 11</td><td>Channel 1</td></tr>
<tr><td>2, 4, 6, 8, 10, 12</td><td>Channel 2</td></tr>
<tr><td>13, 15, 17, 19</td><td>Channel 3</td></tr>
<tr><td>14, 16, 18, 20</td><td>Channel 4</td></tr>
</table>
Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".
</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.
Release 1.4 and later:
The channel works exclusively with the pulse output that is set for "Pulse Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external" (X20SI2100 and X20SI4100 only).</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)
(up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Two-Channel Processing Mode
(only for X20SI8110 and X20SI9100)</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 133: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

2.6.9.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>SI2100
SI4100</th><th>SI8110</th><th>SI9100</th><th>Access via Automation Studio</th><th>Access via Safe-DESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>●</td><td>●</td><td>●</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
• Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
• The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>●</td><td>●</td><td>●</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxy_state</td><td>●</td><td>-</td><td>-</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>-</td><td>●</td><td>-</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>-</td><td>-</td><td>●</td><td>(Read) [1]</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>●</td><td>-</td><td>-</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="3">Type of error</th></tr>
<tr><th>Inputs</th><th colspan="2">Pulse outputs</th></tr>
<tr><th>Input stuck at high</th><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 4 to 7 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
</table>

Table 134: Channel list

Channel name	SI2100 SI4100	SI8110	SI9100	Access via Automation Studio	Access via Safe-DESIGNER	Data type	Description
InputErrorStates	-	●	●	(Read) 1)	-	UDINT	Channel status, additional information for channel error **Type of error** **Inputs** **Input stuck at high** Bit no. 0 to 19 = Channel 1 to 20 If a bit is set, the corresponding error has been detected on the respective channel.
PulseoutputErrors	-	●	●	(Read) 1)	-	UDINT	Channel status, additional information for channel error **Type of error** **Pulse outputs** **Feedback stuck at high (shorted to 24 VDC)**: Bit no. 8 to 11 = Channel 1 to 4 **Feedback stuck at low (ground fault)**: Bit no. 0 to 3 = Channel 1 to 4 If a bit is set, the corresponding error has been detected on the respective channel.
SafeModuleOK	●	●	●	-	Read	SAFEBOOL	Indicates if the safe communication channel is OK
SafeDigitalInputxx	●	●	●	Read	Read	SAFEBOOL	Physical channel SI xx
SafeEquivalentInputxxyy	●	-	-	Read	Read	SAFEBOOL	Dual-channel evaluation of equivalent channel SI xx/yy
SafeAntivalentInputxxyy	●	-	-	Read	Read	SAFEBOOL	Dual-channel evaluation of antivalent channel SI xx/yy
SafeTwoChannelInputxxyy	-	●	●	Read	Read	SAFEBOOL	Dual-channel evaluation of channel SI xx/yy
SafeChannelOKxx	●	-	●	Read	Read	SAFEBOOL	Status of physical channel SI xx
SafeInputOKxx	-	●	-	Read	Read	SAFEBOOL	Status of physical channel SI xx
SafeEquivalentOKxxyy	●	-	-	Read	Read	SAFEBOOL	Status of dual-channel evaluation of equivalent channel SI xx/yy
SafeAntivalentOKxxyy	●	-	-	Read	Read	SAFEBOOL	Status of dual-channel evaluation of antivalent channel SI xx/yy
SafeTwoChannelOkxxyy	-	●	●	Read	Read	SAFEBOOL	Status of dual-channel evaluation of channel SI xx/yy

Table 134: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

PLCopen state diagrams

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

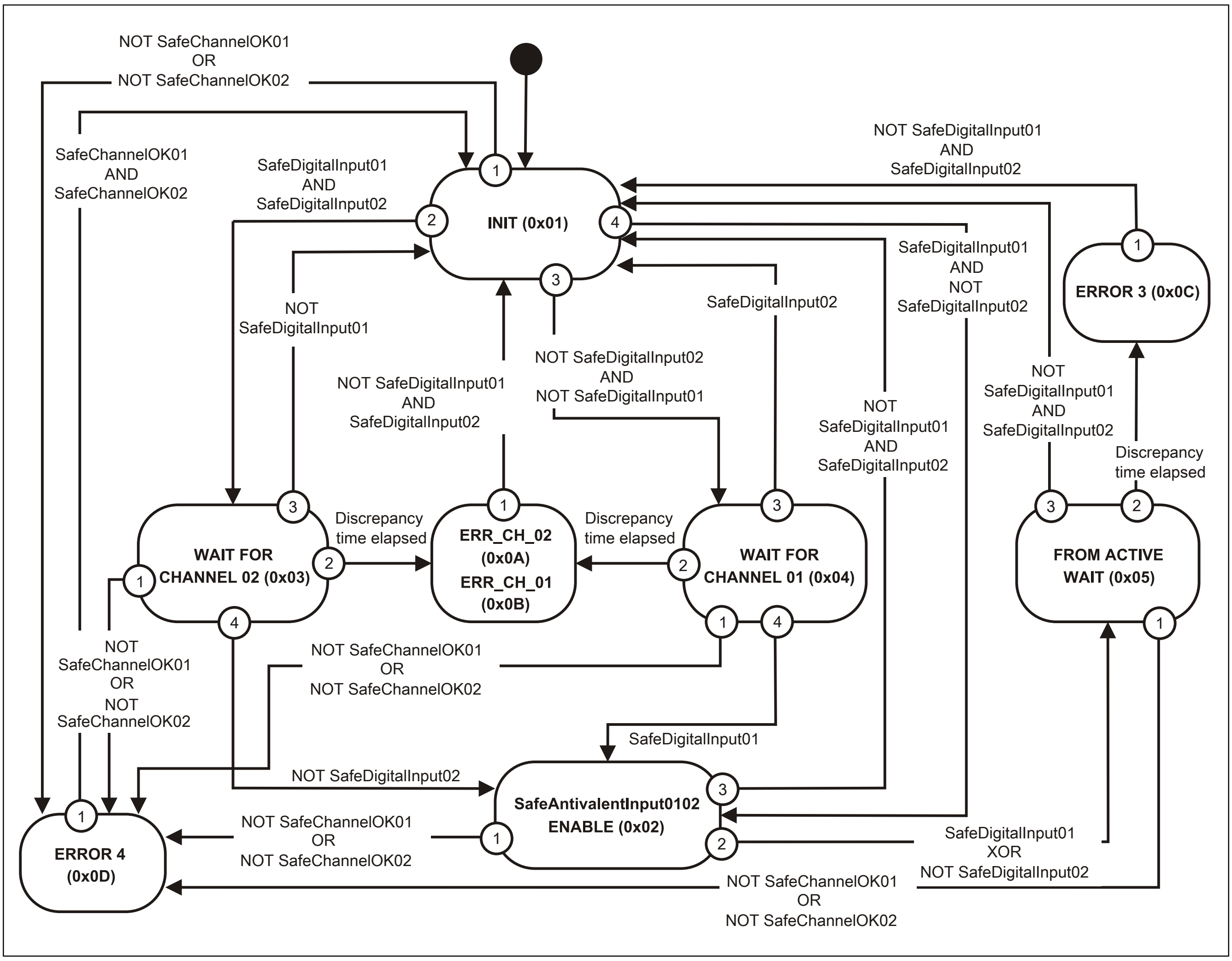


Figure 125: "Antivalent" function block - State diagram

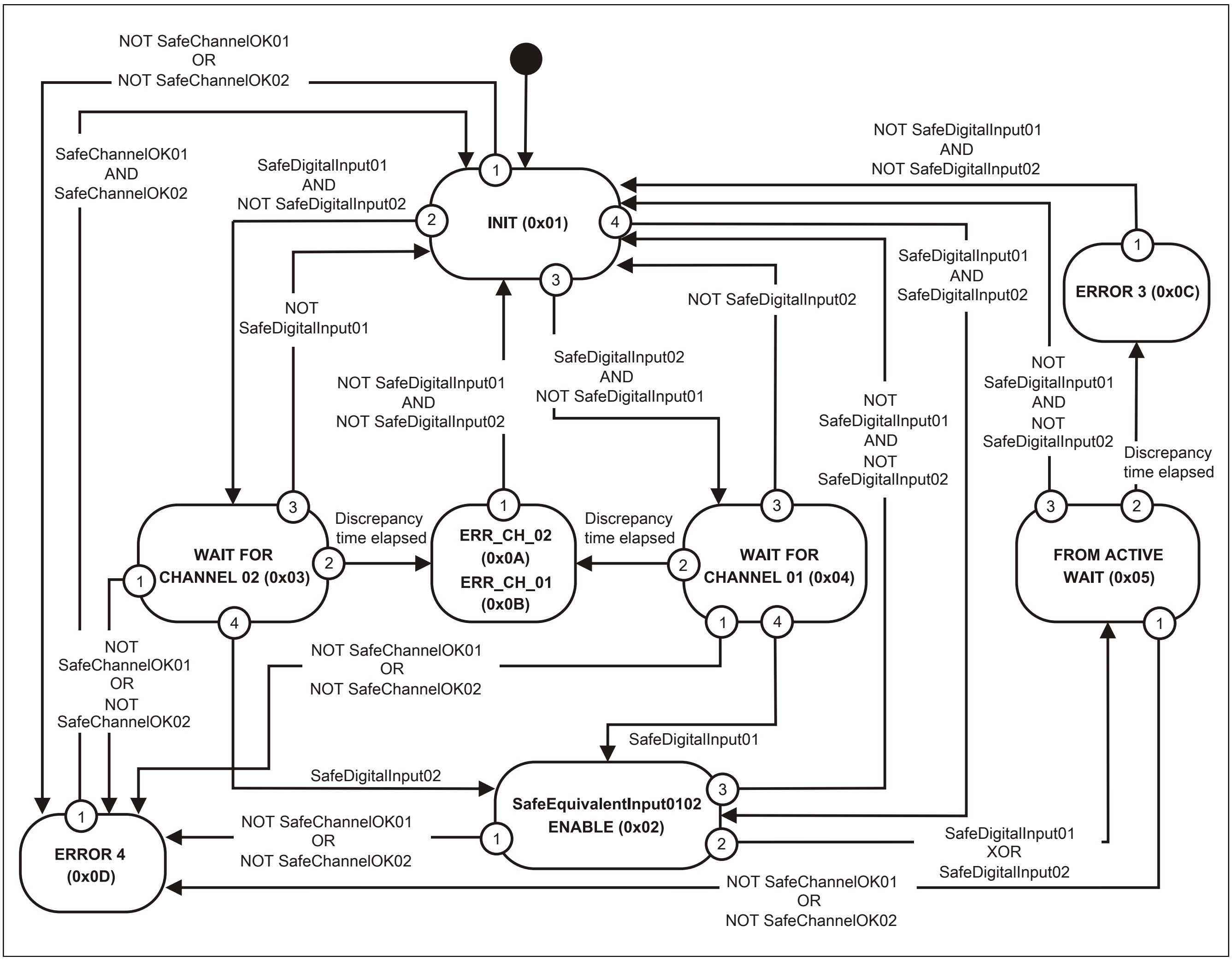


Figure 126: "Equivalent" function block - State diagram

2.6.10 Digital output modules

2.6.10.1 Overview

Model number	Short description	Page
X20SO2110	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs	436
X20SO2120	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	436
X20SO4110	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs	436
X20cSO4110	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs	436
X20SO4120	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	436
X20cSO4120	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	436
X20SO6300	X20 safe digital output module, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	405
X20cSO6300	X20 safe digital output module, coated, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	405

2.6.10.2 X20(c)SO6300

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 135: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 136: Organization of general notices

2.6.10.2.1 General information

The modules are equipped with 6 safe digital outputs. The nominal output current is 0.2 A.

The modules can be used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- 6 safe digital outputs with 0.2 A
- Source circuit
- Output type B
- Integrated output protection

2.6.10.2.1.1 Function

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

2.6.10.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.10.2.2 Overview

Module	X20SO6300
Number of outputs	6
Nominal voltage	24 VDC
Nominal output current	0.2 A
Total nominal current	1.2 A
Output protection	Active shutdown in the event of overcurrent or short circuit Integrated protection for switching inductive loads

Table 137: Digital output modules

2.6.10.2.3 Order data

Model number	Short description	Figure
	Digital output modules	
X20SO6300	X20 safe digital output module, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	
X20cSO6300	X20 safe digital output module, coated, 6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	
	Terminal blocks	
X20TB52	X20 terminal block, 12-pin, safety-keyed	

Table 138: X20SO6300, X20cSO6300 - Order data

2.6.10.2.4 Technical data

Model number	X20SO6300	X20cSO6300
Short description		
I/O module	6 safe type B1 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	
General information		
B&R ID code	0xB815	0xDD88
System requirements		
Automation Studio	3.0.81.15 or later	4.0.16 or later
Automation Runtime	3.00 or later	V3.08 or later
SafeDESIGNER	2.70 or later	3.1.0 or later
Safety Release	1.2 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using status LED and software	
Outputs	Yes, using status LED and software	
Blackout mode		
Scope	Module	
Function	Module function	
Standalone mode	No	
Max. I/O cycle time	1 ms	
Power consumption		
Bus	0.32 W	
Internal I/O	1.4 W	
Electrical isolation		
Channel - Bus	Yes	
Channel - Channel	No	
Certifications		
CE	Yes	
KC	Yes	-
EAC	Yes	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	In preparation	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013	
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3	
Functional safety	EN 50156-1:2004	
Safety characteristics		
EN ISO 13849-1:2015		
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" 1)	
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" 1)	
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" 1)	
MTTFD	2500 years	
Mission time	Max. 20 years	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" 1)	
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" 1)	
PFH / PFH_d		
Module	$<1*10^{-10}$	
openSAFETY wired	Negligible	
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour	
PFD	$<2*10^{-5}$	
Proof test interval (PT)	20 years	
I/O power supply		
Nominal voltage	24 VDC	
Voltage range	24 VDC -15% / +20%	

Table 139: X20SO6300, X20cSO6300 - Technical data

Model number	X20SO6300	X20cSO6300
Integrated protection	Reverse polarity protection	
Safe digital outputs		
Variant	FET, 2x positive switching, type B1, output level readable	
Nominal voltage	24 VDC	
Nominal output current	0.2 A	
Total nominal current	1.2 A	
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads [2]	
Braking voltage when switching off inductive loads	Max. 45 VDC	
Error detection time	1 s	
Isolation voltage between channel and bus	500 V_{eff}	
Peak short-circuit current	Max. 10 A	
Leakage current when switched off	<100 µA	
Residual voltage	<800 mVDC at nominal current	
Switching voltage	I/O power supply minus residual voltage	
Max. switching frequency	100 Hz	
Test pulse length	Max. 10 µs	
Max. capacitive load	100 nF	
Current on loss of ground		
I_{OUT}	<100 µA	
I_{GND}	<70 mA	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level	0 to 2000 m, no limitation	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	0 to 60°C	-25 to 60°C
Vertical mounting orientation	0 to 50°C	-25 to 50°C
Derating	See section "Derating".	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm	

Table 139: X20SO6300, X20cSO6300 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SO6300
Derating bonus	
At 24 VDC	+0°C
Dummy module to the left	+2.5°C
Dummy module to the right	+0°C
Dummy module to the left and right	+5°C
With double PFH / PFH_d	+0°C

Table 140: Derating bonus

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

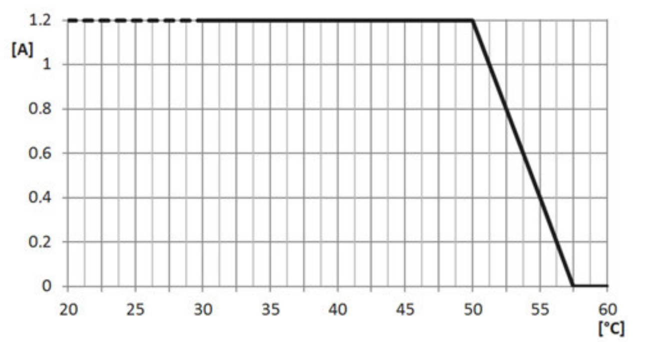
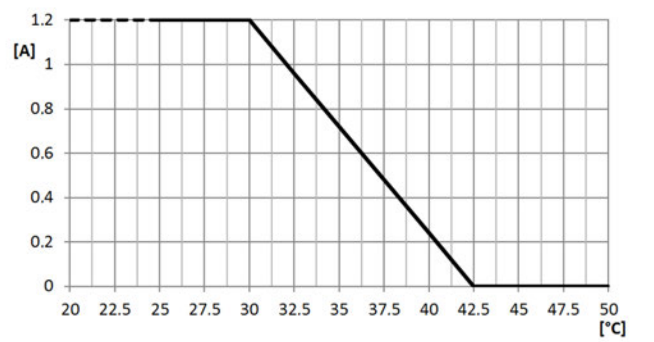
Horizontal (0 to 60°C, coated: -25 to 60°C)	Vertical (0 to 50°C, coated: -25 to 50°C)
	

Table 141: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.10.2.5 LED status indicators

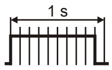
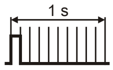
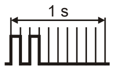
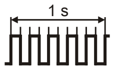
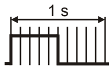
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 6	Output status of the corresponding digital output		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 142: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.10.2.6 Pinout

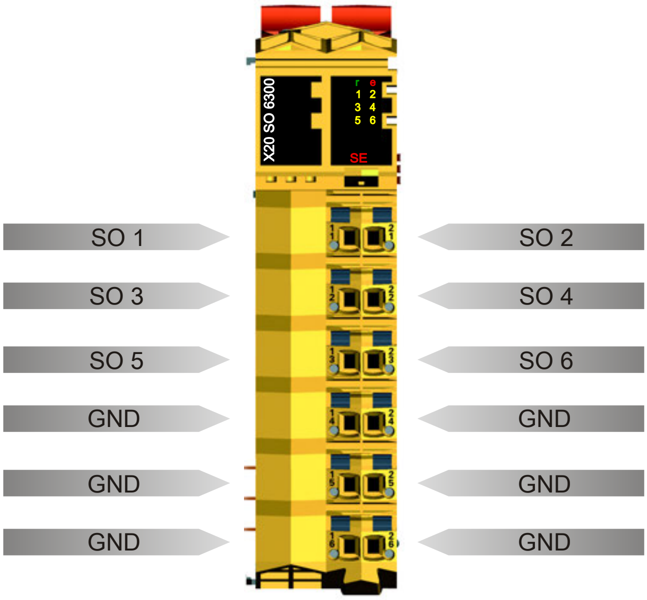


Figure 127: X20SO6300 - Pinout

2.6.10.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.10.2.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

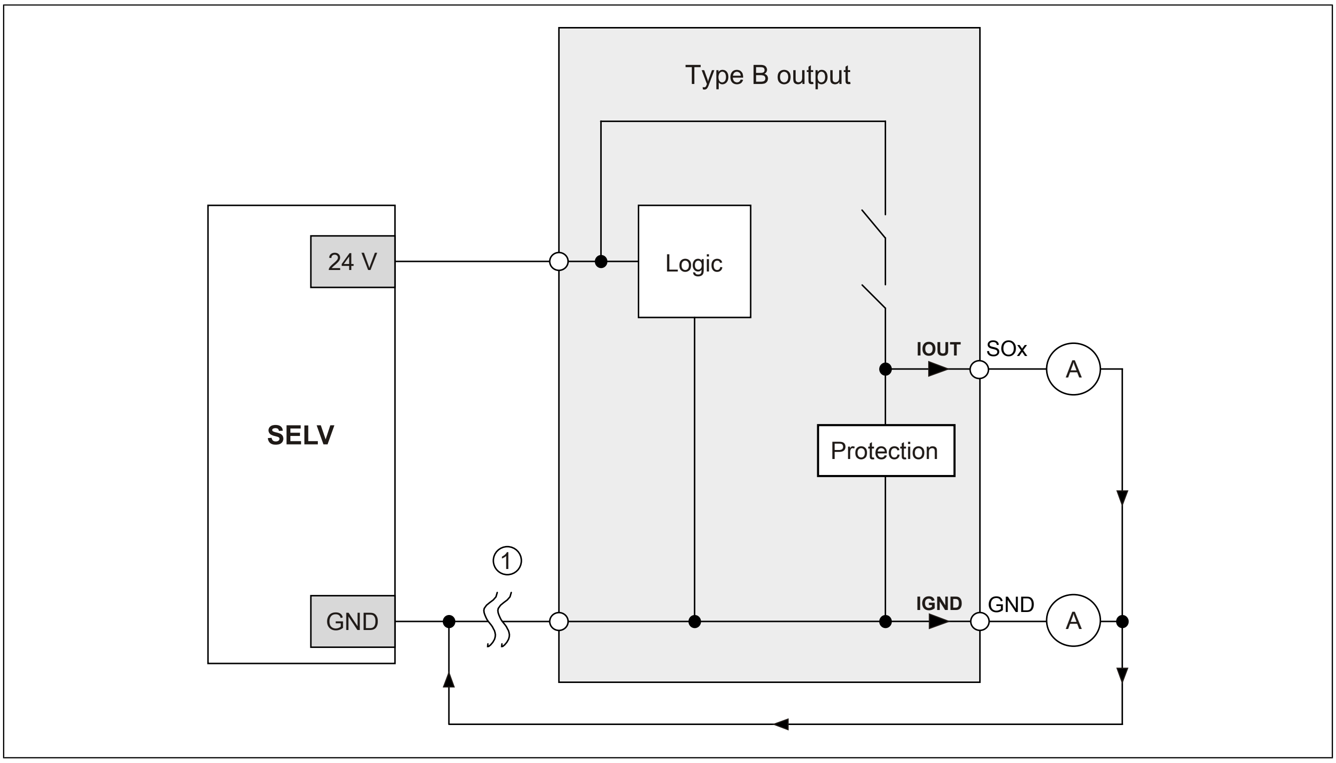


Figure 128: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

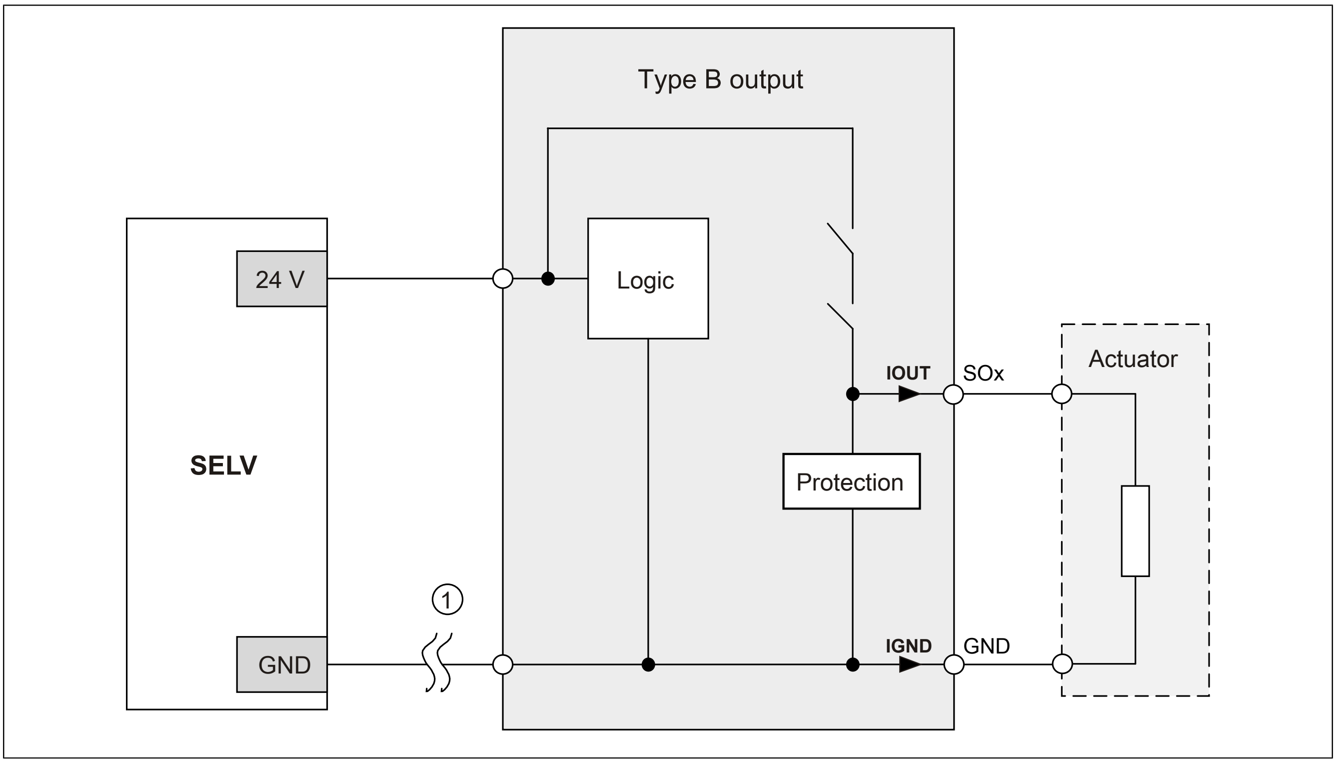


Figure 129: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

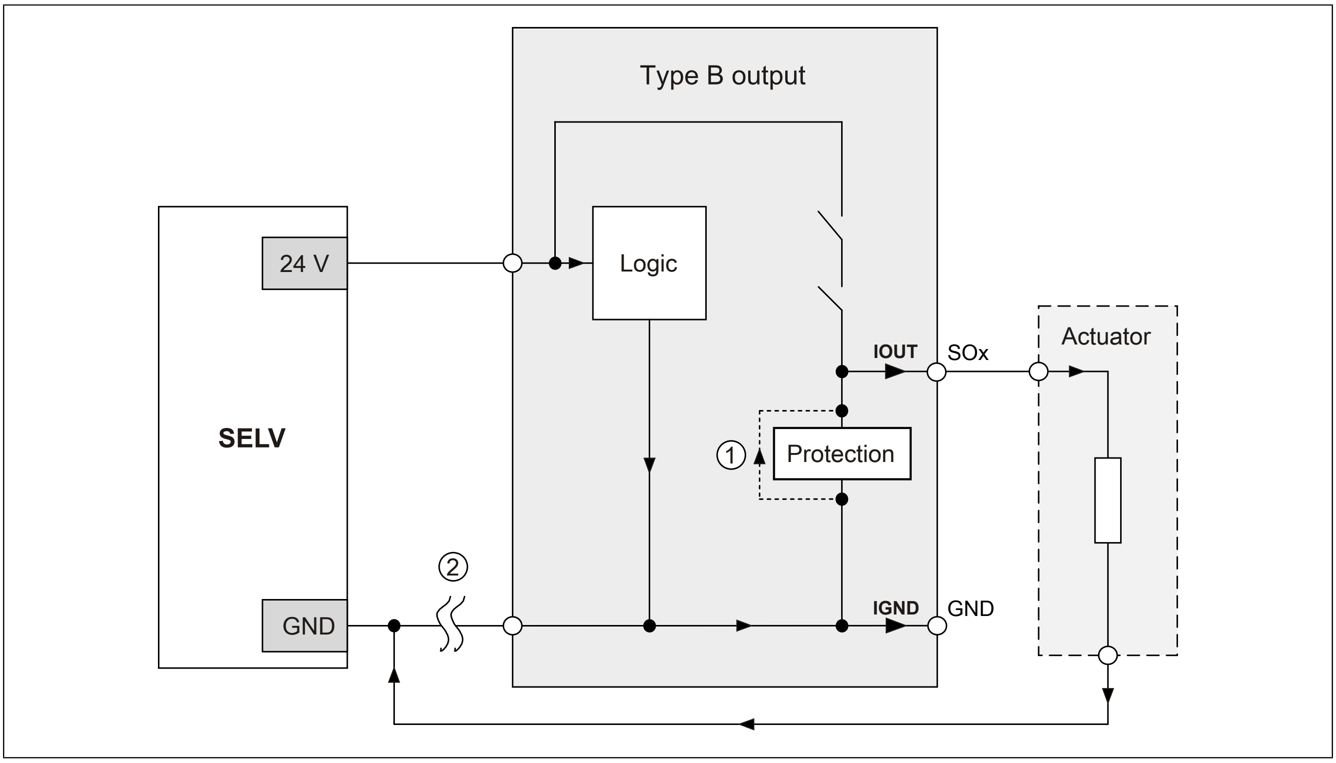


Figure 130: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

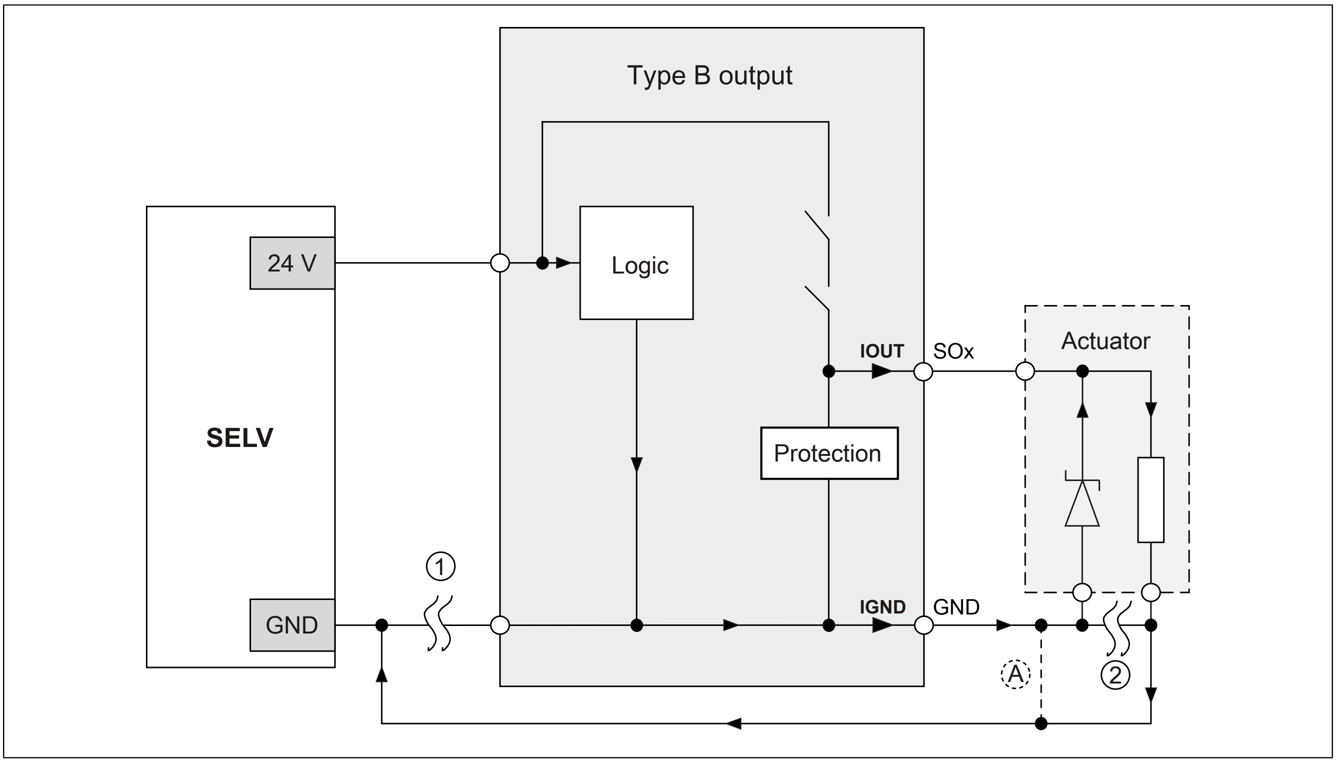


Figure 131: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

2.6.10.2.7.2 Connecting safety-oriented actuators for Type B outputs

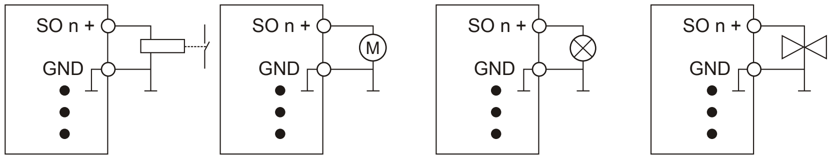


Figure 132: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.10.2.7.3 ACOPOS / ACOPOSmulti connection

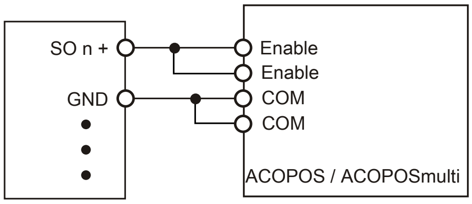


Figure 133: ACOPOS / ACOPOSmulti connection

The SO module can be directly connected to the ACOPOS or ACOPOSmulti safe inputs.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the ACOPOS or ACOPOSmulti. With this connection, the ACOPOS drive satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With this connection, the ACOPOSmulti drive satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015.

Information:

Detailed information about the connection/function of ACOPOS and ACOPOSmulti drives can be found in the corresponding user manuals.

2.6.10.2.7.4 Electronic actuator connection

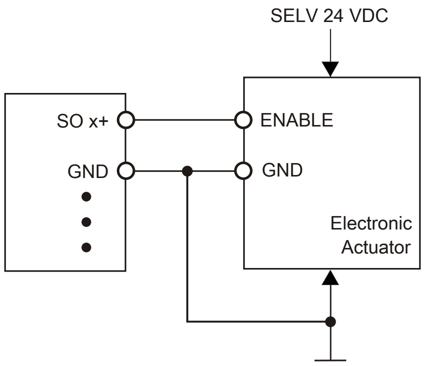


Figure 134: Electronic actuator connection

The X20SO6300 module can be directly connected with the safe inputs of conventional electronic actuators. To prevent possible faults caused by the loss of the GND connection, additional GND connections must be implemented on the output module as well as on the actuator.

Danger!

Due to the "only-plus-switching" design used for the outputs, any short circuits on SOx to high potentials will result in active actuators that cannot be cut off. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

Information:

Detailed information concerning the safety guidelines and the connection/function of the electronic actuator can be found in the corresponding user's manuals.

2.6.10.2.8 Error detection

2.6.10.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

2.6.10.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

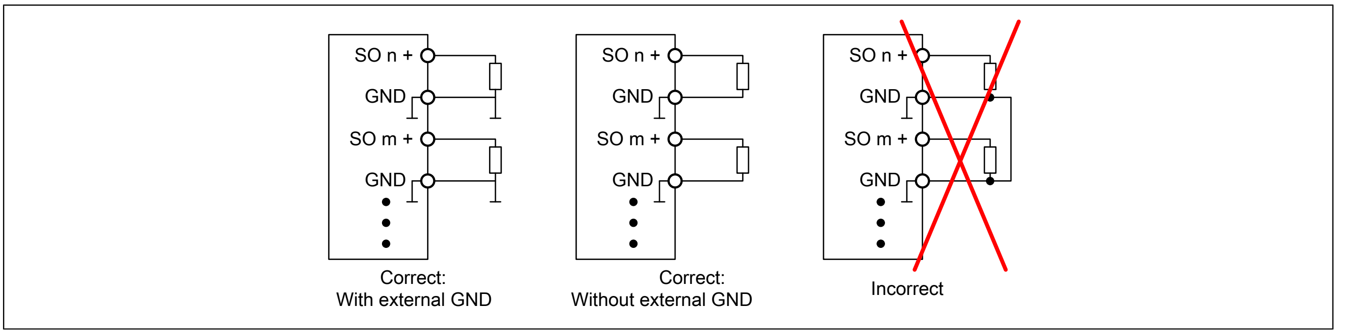


Figure 135: Invalid wiring

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected [2]</td><td>Not detected [2]</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
</table>

Table 143: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx	**Not detected**	Detected	**Not detected**	Detected
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				

Table 143: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.10.2.9 Output circuit diagram

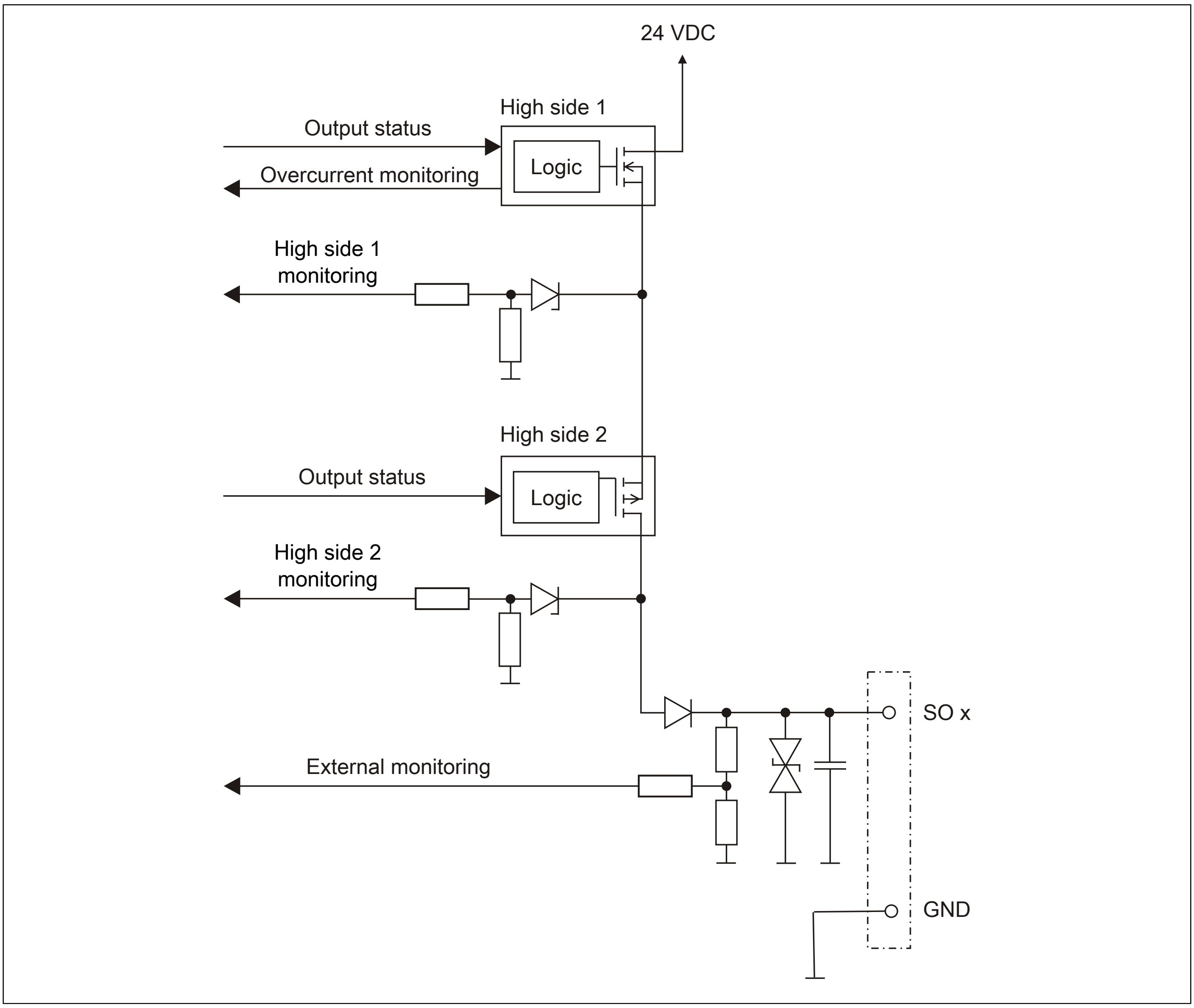


Figure 136: Output circuit diagram

2.6.10.2.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 μs

2.6.10.2.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 μs

Maximum I/O update time
1800 μs

2.6.10.2.12 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.10.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.2.14 Register description

2.6.10.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 144: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Output status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>Restart inhibit state information</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>Max switching frequency channel x
(up to firmware version < 300)</td><td>Maximum switching frequency of the output channel.
• Permissible values: 1 Hz, 10 Hz, 100 Hz
This value specifies the max. switching frequency of the actuator connected to the output. It is especially important to adjust this parameter to the actual conditions for inductive or capacitive loads because the internal delay for checking the voltage to see if it is 0 V after a cutoff signal occurs is calculated using this parameter. Therefore, if this value is too high (e.g. 1000 Hz) and the voltage does not go to 0 within the corresponding time (in this example 500 µs) after a cutoff signal because of the connected actuator, then a channel error occurs.

If the output is controlled by the application using a higher switching frequency than configured, a channel-specific error may erroneously be detected on the module, which causes the channel to be cut off.</td><td>1</td><td>Hz</td></tr>
</table>

Table 145: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 146: I/O configuration parameters: Output signal path

2.6.10.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Disable_OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 147: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2010 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2010 or PL d per EN ISO 13849-1:2015, a daily check of the safety function by the user is necessary.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 148: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** / **Description** Yes-ATTENTION: "Automatic restart" function is activated. No: "Automatic restart" function is not activated.	No	-

Table 149: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 150: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 151: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Disable OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr><tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr></table></td></tr>
</table>

Table 152: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** / **Description** Yes-ATTENTION: "Automatic restart" function is activated. No: "Automatic restart" function is not activated.	No	-

Table 153: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeChannelOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UDINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table>
<tr><th>Bit 23 to 20</th><th>Bit 19 to 16</th><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr>
<tr><td>Channel 6</td><td>Channel 5</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table></td></tr>
</table>

Table 154: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

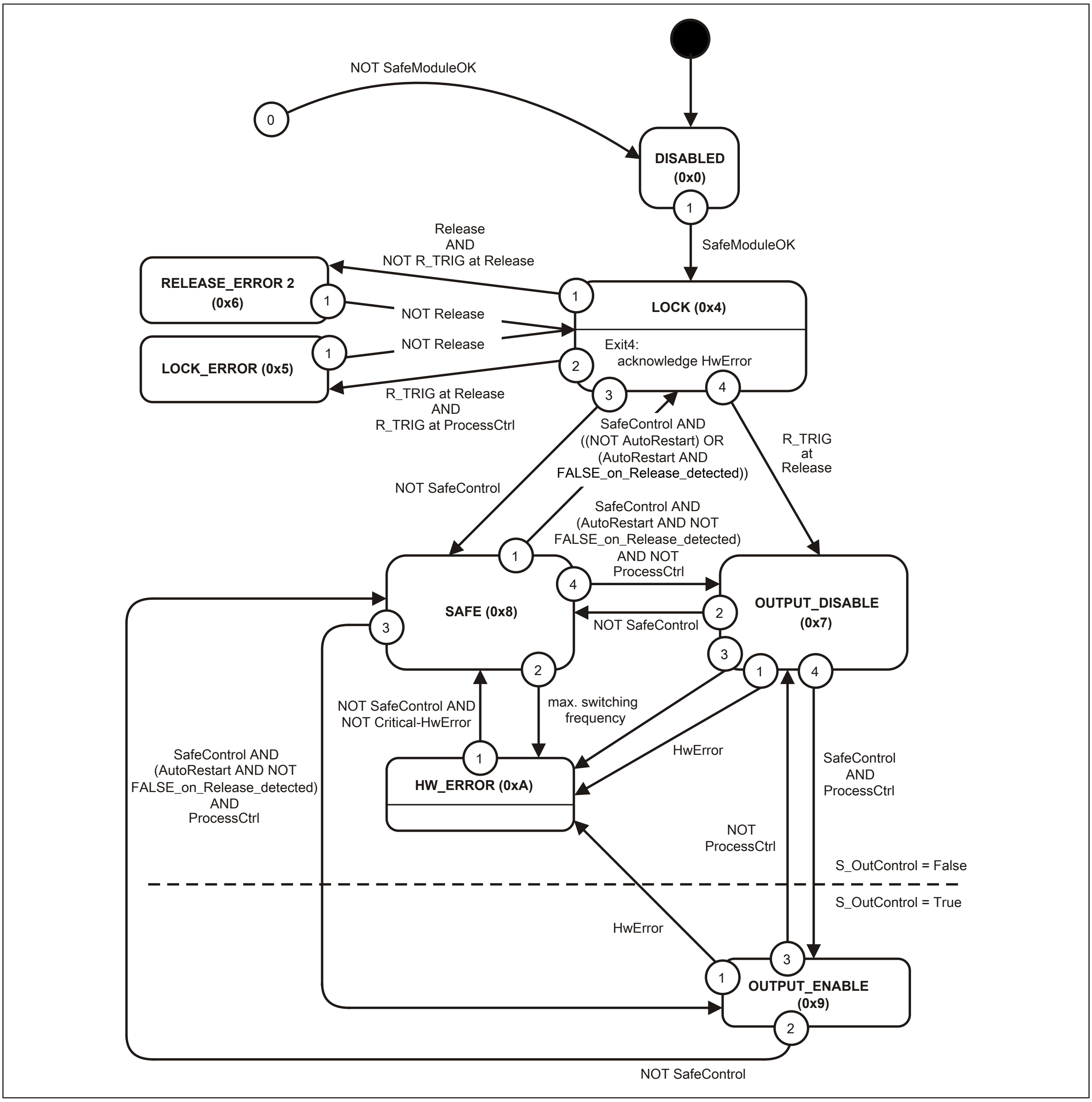


Figure 137: Restart interlock - State diagram

2.6.10.3 X20(c)SOx1x0

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 155: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 156: Organization of general notices

2.6.10.3.1 General information

The modules are equipped with 2 or 4 safe digital outputs. The nominal output current is 0.5 or 2 A.

The modules can be used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side low-side" variant (output type A) is limited to actuators without reference potential (e.g. relays, valves). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Safe digital output modules are equipped with protection against automatic restart in the event of network errors as well as current measurement for open-circuit detection.

These modules are designed for X20 12-pin terminal blocks.

- 2 or 4 safe digital outputs with 0.5 or 2 A
- Source circuit
- Output type A
- Current monitoring
- Open-circuit detection
- Integrated output protection

2.6.10.3.1.1 Function

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

2.6.10.3.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

Information:

For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.10.3.2 Overview

Module	X20SO2110	X20SO2120	X20SO4110	X20SO4120
Number of outputs	2	2	4	4
Nominal voltage	24 VDC			
Nominal output current	0.5 A	2 A	0.5 A	2 A
Total nominal current	1 A	4 A	2 A	5 A
Output protection	Thermal shutdown in the event of overcurrent or short circuit, Integrated protection for switching inductive loads			

Table 157: Digital output modules

2.6.10.3.3 Order data

X20SO21x0

X20SO41x0

Model number	Short description
	Digital output modules
X20SO2110	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs
X20SO2120	X20 safe digital output module, 2 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs
X20SO4110	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs
X20cSO4110	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs
X20SO4120	X20 safe digital output module, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs
X20cSO4120	X20 safe digital output module, coated, 4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs
	Required accessories
	Bus modules
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 158: X20SO2110, X20SO2120, X20SO4110, X20cSO4110, X20SO4120, X20cSO4120 - Order data

2.6.10.3.4 Technical data

Model number	X20SO2110	X20SO2120	X20SO4110	X20cSO4110	X20SO4120	X20cSO4120
Short description						
I/O module	2 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs	2 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	4 safe type A digital outputs, with current monitoring, 24 VDC, 0.5 A, OSSD <500 µs		4 safe type A digital outputs, with current monitoring, 24 VDC, 2 A, OSSD <500 µs	
General information						
B&R ID code	0x1F16	0x2009	0x1DBE	0xDD84	0x2007	0xDD5C
System requirements						
Automation Studio	3.0.71 or later			4.0.16 or later	3.0.71 or later	4.0.16 or later
Automation Runtime	2.95 or later			V3.08 or later	2.95 or later	V3.08 or later
SafeDESIGNER	2.58 or later			3.1.0 or later	2.58 or later	3.1.0 or later
Safety Release	1.1 or later			1.7 or later	1.1 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status					
Diagnostics						
Module run/error	Yes, using status LED and software					
Outputs	Yes, using status LED and software					
Blackout mode						
Scope	Module					
Function	Module function					
Standalone mode	No					
Max. I/O cycle time	800 µs					
Power consumption						
Bus	0.25 W					
Internal I/O	0.98 W		1.3 W			
Electrical isolation						
Channel - Bus	Yes					
Channel - Channel	No					
Certifications						
CE	Yes					
KC	Yes			-	Yes	-
EAC	Yes					
UL	cULus E115267 Industrial control equipment					
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5					
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X					
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)					
LR	ENV1					
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013					
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3					
Functional safety	EN 50156-1:2004					
Safety characteristics						
EN ISO 13849-1:2015						
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" 1)					
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" 1)					
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" 1)					
MTTFD	2500 years					
Mission time	Max. 20 years					

Table 159: X20SO2110, X20SO2120, X20SO4110, X20cSO4110, X20SO4120, X20cSO4120 - Technical data

Model number	X20SO2110	X20SO2120	X20SO4110	X20cSO4110	X20SO4120	X20cSO4120
IEC 61508:2010, IEC 61511:2004, EN 62061:2013						
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" 1)					
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" 1)					
PFH / PFH_d						
Module	$<1*10^{-10}$					
openSAFETY wired	Negligible					
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour					
PFD	$<2*10^{-5}$					
Proof test interval (PT)	20 years					
I/O power supply						
Nominal voltage	24 VDC					
Voltage range	24 VDC -15% / +20%					
Integrated protection	Reverse polarity protection					
Safe digital outputs						
Variant	FET, 1x positive switching, 1x negative switching, type A, output level readable, open-circuit detection					
Nominal voltage	24 VDC					
Nominal output current	0.5 A	2 A	0.5 A		2 A	
Total nominal current	1 A	4 A	2 A		5 A	
Output protection	Thermal shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads 2)					
Braking voltage when switching off inductive loads	Max. 90 VDC 3)					
Open-circuit detection	Via internal current measurement, output current <10 mA: Signal "CurrentOK" = FALSE, output current 10 to 50 mA: Signal "CurrentOK" = Undefined, output current >50 mA: Signal "CurrentOK" = TRUE					
Error detection time	1 s					
Isolation voltage between channel and bus	500 V_{eff}					
Peak short-circuit current	Max. 12 A					
Leakage current when switched off	<10 µA					
Residual voltage	<120 mVDC at 0.5 A nominal current without OSSD	<480 mVDC at 2 A nominal current without OSSD	<120 mVDC at 0.5 A nominal current without OSSD		<480 mVDC at 2 A nominal current without OSSD	
Switching voltage	I/O power supply minus residual voltage					
Max. switching frequency	1000 Hz					
Test pulse length	Max. 500 µs					
Time between two test pulses	Min. 49.5 ms					
Max. capacitive load	100 nF					
Operating conditions						
Mounting orientation						
Horizontal	Yes					
Vertical	Yes					
Installation elevation above sea level	0 to 2000 m, no limitation					
Degree of protection per EN 60529	IP20					
Ambient conditions						
Temperature						
Operation						
Horizontal mounting orientation	0 to 60°C			-40 to 60°C 4)	0 to 60°C	-40 to 60°C 4)
Vertical mounting orientation	0 to 50°C			-40 to 50°C 5)	0 to 50°C	-40 to 50°C 5)
Derating	See section "Derating".					
Storage	-40 to 85°C					
Transport	-40 to 85°C					
Relative humidity						
Operation	5 to 95%, non-condensing			Up to 100%, condensing	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing					
Transport	5 to 95%, non-condensing					
Mechanical properties						
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.					
Spacing	$25^{+0.2}$ mm					

Table 159: X20SO2110, X20SO2120, X20SO4110, X20cSO4110, X20SO4120, X20cSO4120 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Due to the internal protective circuit, this braking voltage only takes effect starting at a load of typ. 250 mA.
4) Up to hardware upgrade <1.10.1.0 and hardware revision <L0: -25 to 60°C
5) Up to hardware upgrade <1.10.1.0 and hardware revision <L0: -25 to 50°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SO2110	X20SO2120	X20SO4110	X20SO4120
Derating bonus				
At 24 VDC	+0°C			
Dummy module to the left	+2.5°C			
Dummy module to the right	+0°C			
Dummy module to the left and right	+5°C			
With double PFH / PFH_d	+0°C			

Table 160: Derating bonus

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

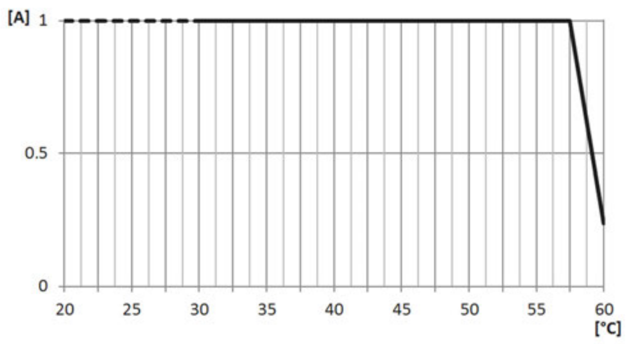
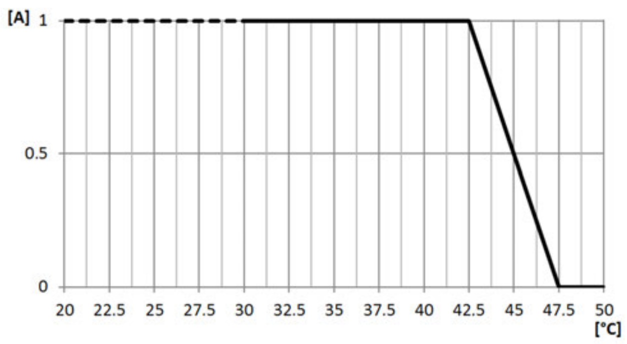
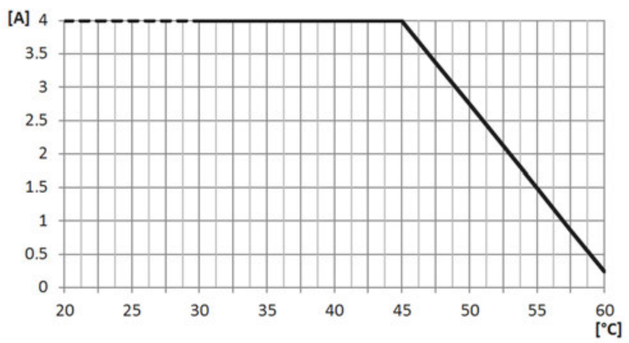
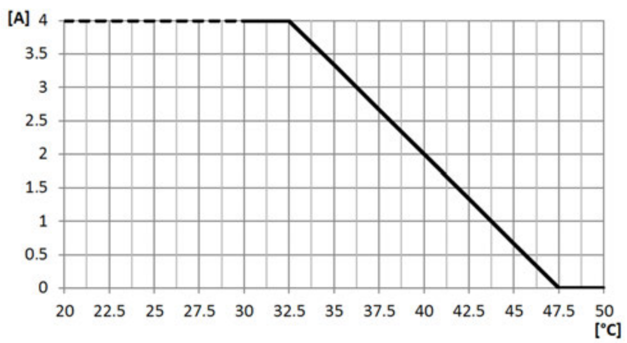
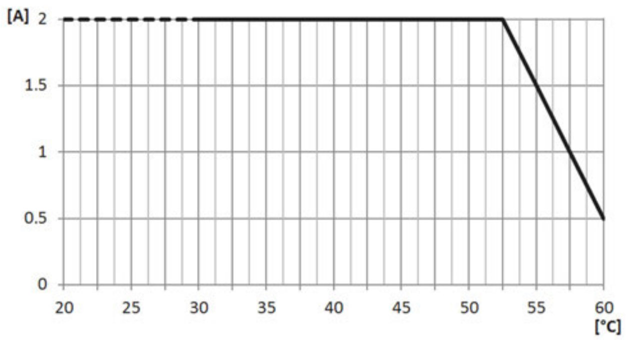
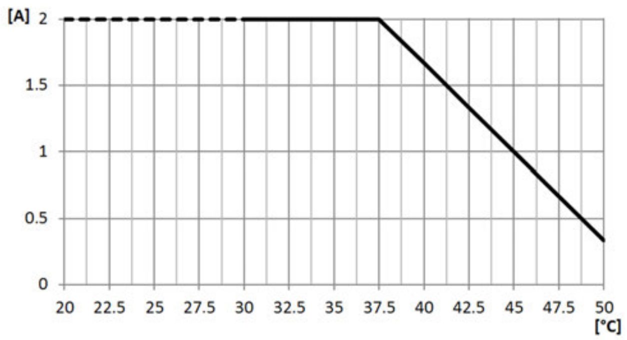
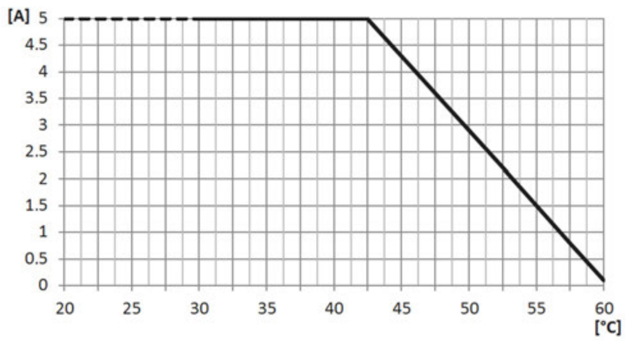
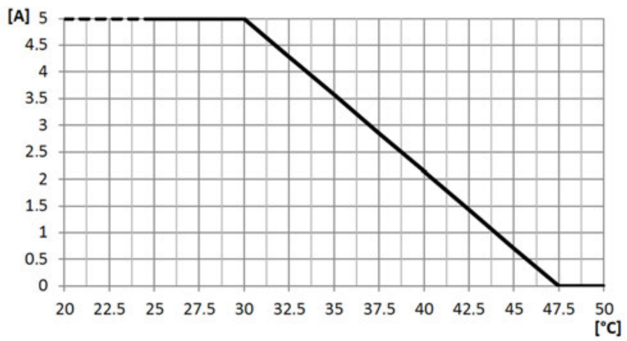
<table>
<tr><th>Horizontal (0 to 60°C, coated: -40 to 60°C)</th><th>Vertical (0 to 50°C, coated: -40 to 50°C)</th></tr>
<tr><td colspan="2">X20SO2110</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">X20SO2120</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">X20SO4110</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">X20SO4120</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Table 161: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.10.3.5 LED status indicators

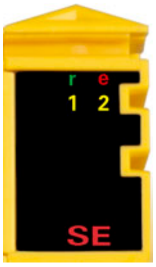
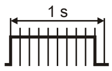
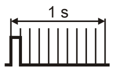
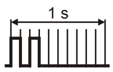
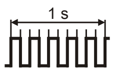
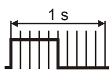
Figure	LED	Color	Status	Description
 X20SO21x0  X20SO41x0	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 4	Output status of the corresponding digital output The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 162: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.10.3.6 Pinouts

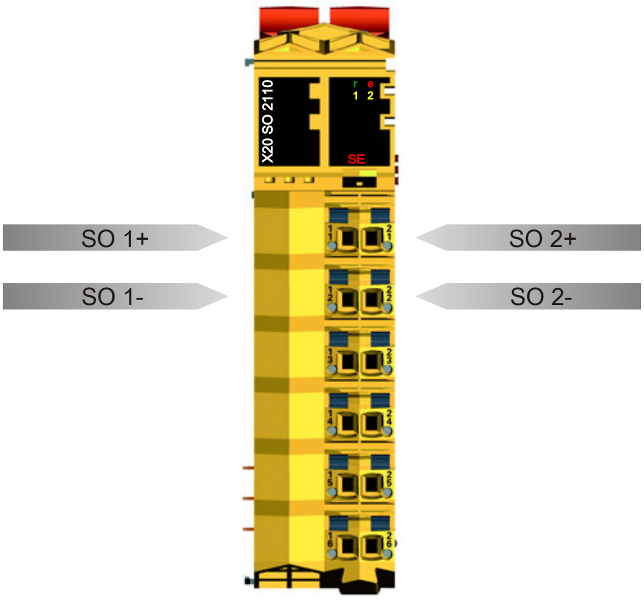


Figure 138: X20SO21x0 - Pinout

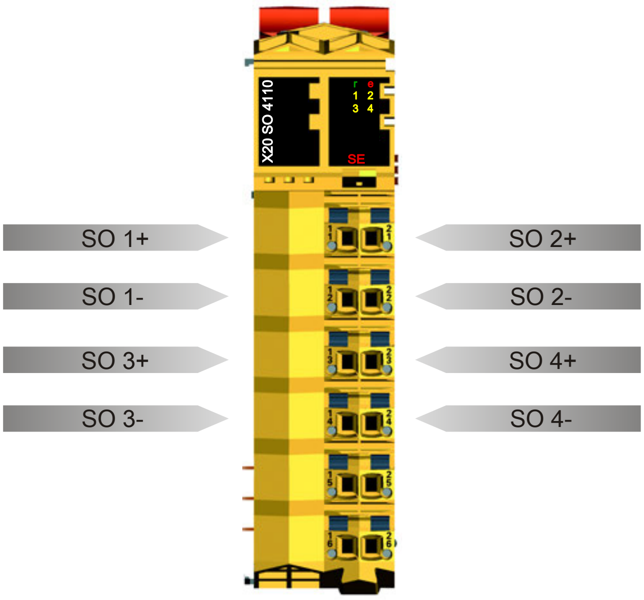


Figure 139: X20SO41x0 - Pinout

2.6.10.3.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.10.3.7.1 Connecting safety-oriented actuators for Type A outputs

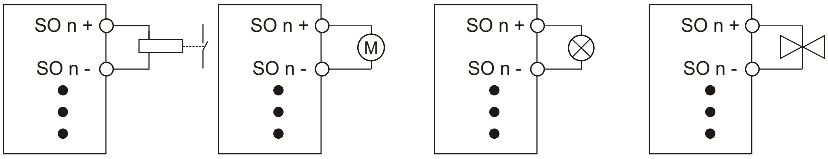


Figure 140: Connecting safety-oriented actuators for Type A outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

2.6.10.3.7.2 ACOPOS / ACOPOSmulti connection

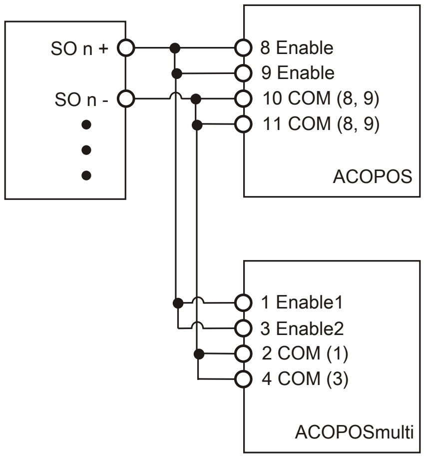


Figure 141: ACOPOS / ACOPOSmulti connection

The SO module can be directly connected to the ACOPOS or ACOPOSmulti safe inputs.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the ACOPOS or ACOPOSmulti. With this connection, the ACOPOS drive satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With this connection, the ACOPOSmulti drive satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015.

Information:

When connecting the SO module to the ACOPOS drive, the module-internal test of the output circuit must be disabled using module parameter "Disable OSSD = Yes-ATTENTION"; otherwise, the OSSD gaps may cause the ACOPOS drive to switch off unintentionally.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Information:

Detailed information about the connection/function of ACOPOS and ACOPOSmulti drives can be found in the corresponding user manuals.

2.6.10.3.8 Error detection

2.6.10.3.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.10.3.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type A output channels

Danger!

Type A output channels also cut off the load on the GND side. Check whether the actuator you have connected permits a cutoff on the GND side. X20 and X67 systems do not support this type of cutoff, for example.

Danger!

Note that wiring SOx+ directly to GND via an actuator is not permitted; wiring 24 VDC directly to SOx- via an actuator is also not permitted.

These types of errors will not be detected by the module. The user must prevent these types of errors through careful wiring.

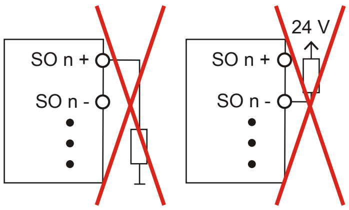


Figure 142: Invalid wiring

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected [2)]</td><td>Not detected [2)]</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
</table>

Table 163: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx	**Not detected**	Detected	**Not detected**	Detected
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				

Table 163: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.10.3.9 Type A output circuit diagram

Type A digital output channels are designed for positive and GND switching inside the module.

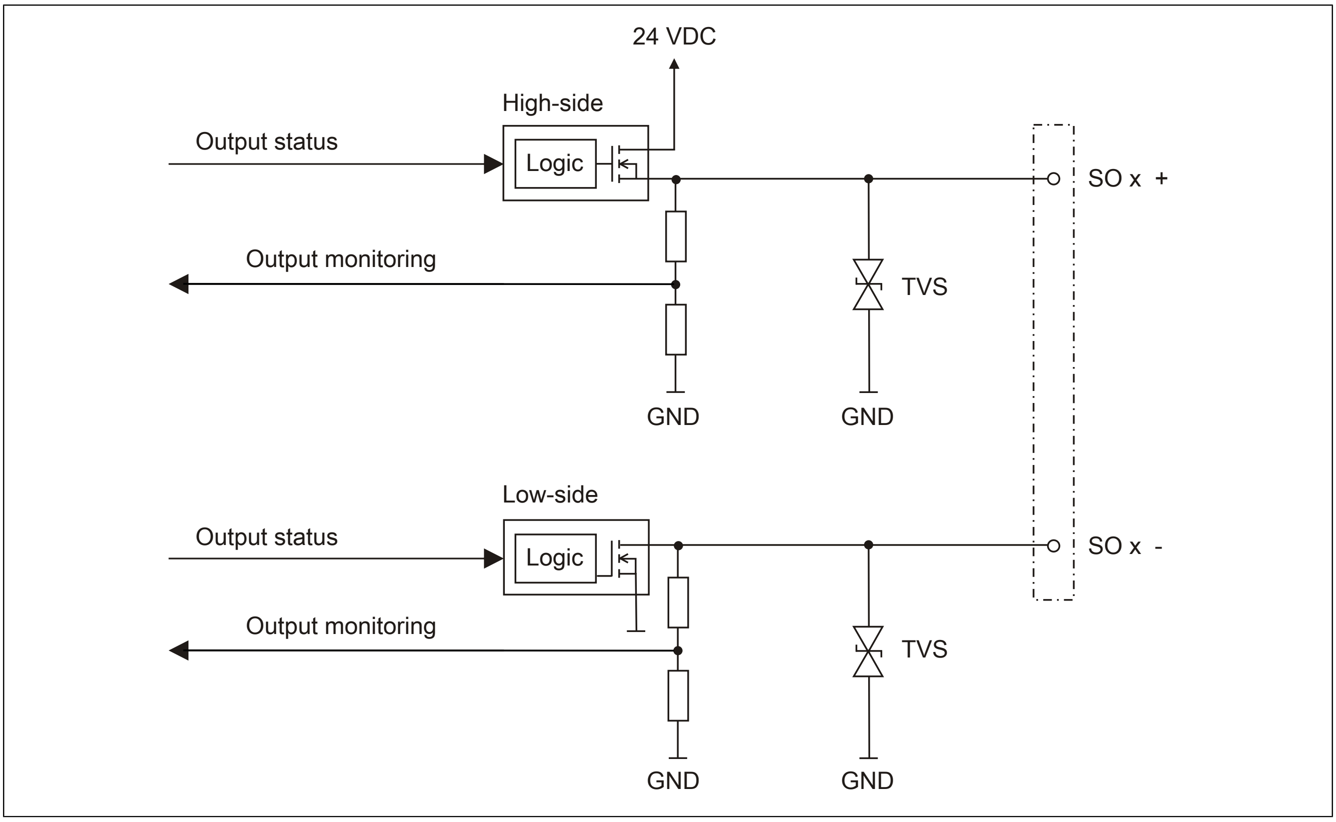


Figure 143: Type A output circuit diagram

2.6.10.3.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 μs

2.6.10.3.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
400 μs

Maximum I/O update time
1600 μs

2.6.10.3.12 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.10.3.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.3.14 Register description

2.6.10.3.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 164: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Output status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>Restart inhibit state information</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>Max switching frequency channel x
(up to firmware version < 300)</td><td>Maximum switching frequency of the output channel.
• Permissible values: 1 Hz, 10 Hz, 100 Hz, 1000 Hz
This value specifies the max. switching frequency of the actuator connected to the output. It is especially important to adjust this parameter to the actual conditions for inductive or capacitive loads because the internal delay for checking the voltage to see if it is 0 V after a cutoff signal occurs is calculated using this parameter. Therefore, if this value is too high (e.g. 1000 Hz) and the voltage does not go to 0 within the corresponding time (in this example 500 µs) after a cutoff signal because of the connected actuator, then a channel error occurs.

If the output is controlled by the application using a higher switching frequency than configured, a channel-specific error may erroneously be detected on the module, which causes the channel to be cut off.</td><td>1</td><td>Hz</td></tr>
</table>

Table 165: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 166: I/O configuration parameters: Output signal path

2.6.10.3.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Disable_OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 167: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2010 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2010 or PL d per EN ISO 13849-1:2015, a daily check of the safety function by the user is necessary.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.
<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td><td>No</td><td>-</td></tr>
<tr><td>Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.
<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td><td>Yes</td><td>-</td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_ CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 168: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** / **Description** Yes-ATTENTION: "Automatic restart" function is activated. No: "Automatic restart" function is not activated.	No	-

Table 169: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.3.14.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 170: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 171: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Disable OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 172: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** / **Description** Yes-ATTENTION: "Automatic restart" function is activated. No: "Automatic restart" function is not activated.	No	-

Table 173: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.10.3.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeChannelOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>CurrentOKxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Status of current measurement of channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table>
<tr><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr>
<tr><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table></td></tr>
</table>

Table 174: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

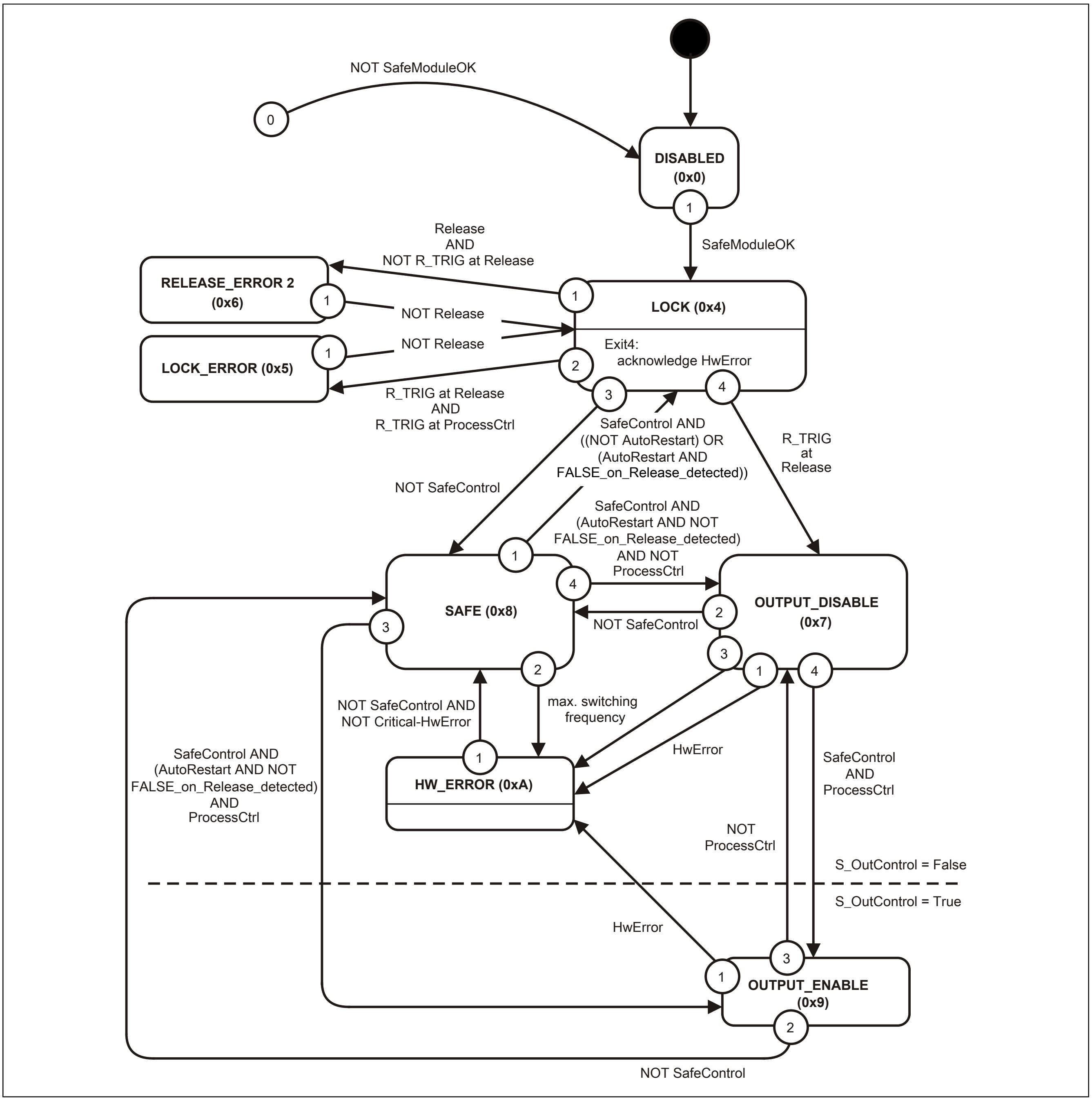


Figure 144: Restart interlock - State diagram

2.6.11 Digital mixed modules

2.6.11.1 Overview

Model number	Short description	Page
X20SC0402	X20 safe digital mixed module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	465
X20SC0806	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	465
X20SC0842	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs	465
X20SC2212	X20 safe digital mixed module, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	510
X20cSC2212	X20 safe digital mixed module, coated, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	510

2.6.11.2 X20SC0xxx

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 175: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 176: Organization of general notices

2.6.11.2.1 General information

The modules are equipped with 4 to 8 safe digital inputs and 2 to 6 safe digital outputs. They are designed for a nominal voltage of 24 VDC.

The modules can be used to read in digital signals and control actuators in safety-related applications up to PL e or SIL 3.

The modules are equipped with filters that are individually configurable for switch-on and switch-off behavior. The modules also provide pulse signals for diagnosing the sensor line.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side low-side" variant (output type A) is limited to actuators without reference potential (e.g. relays, valves). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- 4 to 8 safe digital inputs, sink circuit
- 4 pulse outputs
- Software input filter configurable for each channel
- 4 safe digital outputs, output type A with 3 A, source circuit
- 2 or 6 safe digital outputs, output type B with 50 mA or 0.2 A, source circuit
- Integrated output protection

2.6.11.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

2.6.11.2.2 Overview

Module	X20SC0402	X20SC0806	X20SC0842
Safe digital inputs			
Number of inputs	4	8	8
Nominal voltage	24 VDC		
Input filter			
Hardware	≤150 µs		
Software	Configurable between 0 and 500 ms		
Input circuit	Sink		
Pulse outputs			
Design	Push-Pull		
Switching voltage	I/O power supply minus residual voltage		
Safe digital HS-LS outputs			
Number of outputs	-		4
Nominal voltage	-		24 VDC
Nominal output current	-		3 A
Total nominal current	-		10 A 1)
Output protection	-		Thermal short-circuit shutdown, integrated protection for switching inductive loads
Safe digital HS-HS outputs			
Number of outputs	2	6	2
Nominal voltage	24 VDC		
Nominal output current	0.2 A		50 mA
Total nominal current	0.4 A	1.2 A	100 mA
Output protection	Active cutoff if overcurrent or short circuit occurs, integrated protection for switching inductances		

Table 177: Digital mixed modules

1) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.

2.6.11.2.3 Order data

X20SC0402 X20SC0806 X20SC0842

Model number	Short description
	Digital mixed modules
X20SC0402	X20 safe digital mixed module, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SC0806	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SC0842	X20 safe digital mixed module, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs
	Required accessories
	Bus modules
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 178: X20SC0402, X20SC0806, X20SC0842 - Order data

2.6.11.2.4 Technical data

Model number	X20SC0402	X20SC0806	X20SC0842
Short description			
I/O module	4 safe digital inputs, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	8 safe digital inputs, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	8 safe digital inputs, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs
General information			
B&R ID code	0xE7F8	0xE75A	0xE7F9
System requirements			
Automation Studio	4.0 or later		
Automation Runtime	4.0 or later		
SafeDESIGNER	3.4.0 or later		
Safety Release	1.7 or later		
Status indicators	I/O function per channel, operating state, module status		
Diagnostics			
Module run/error	Yes, using status LED and software		
Outputs	Yes, using status LED and software		
Inputs	Yes, using status LED and software		
Blackout mode			
Scope	Module		
Function	Module function		
Standalone mode	No		
Max. I/O cycle time	1 ms		
Power consumption			
Bus	0.4 W		
Internal I/O	2.5 W		
Electrical isolation			
Channel - Bus	Yes		
Channel - Channel	No		
Certifications			
CE	Yes		
EAC	Yes		
UL	cULus E115267 Industrial control equipment		
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X		
DNV GL	In preparation		
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013		
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3		
Functional safety	EN 50156-1:2004		
Safety characteristics			
EN ISO 13849-1:2015			
MTTFD	2500 years		
Mission time	Max. 20 years		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
PFH / PFH_d			
Module	$<1*10^{-10}$		
openSAFETY wired	Negligible		
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour		
PFD	$<2*10^{-5}$		
Proof test interval (PT)	20 years		
Safe digital inputs			
EN ISO 13849-1:2015			
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1]		
PL	PL e		
DC	>94%		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
SIL CL	SIL 3		
SFF	>90%		

Table 179: X20SC0402, X20SC0806, X20SC0842 - Technical data

Model number	X20SC0402	X20SC0806	X20SC0842
Safe digital outputs			
EN ISO 13849-1:2015			
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" 1)		
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" 1)		
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" 1)		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" 1)		
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" 1)		
I/O power supply			
Nominal voltage	24 VDC		
Voltage range	24 VDC -15% / +20%		
Integrated protection	Reverse polarity protection		
Safe digital inputs			
Nominal voltage	24 VDC		
Input characteristics per EN 61131-2	Type 1		
Input filter			
Hardware	≤150 µs		
Software	Configurable between 0 and 500 ms		
Input circuit	Sink		
Input voltage	24 VDC -15% / +20%		
Input current at 24 VDC	Max. 3.28 mA		
Input resistance	Min. 7.33 kΩ		
Error detection time	100 ms		
Isolation voltage between channel and bus	500 V_{eff}		
Switching threshold			
Low	<5 VDC		
High	>15 VDC		
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line		
Safe digital HS-LS outputs			
Variant	-		FET, 1x positive switching, 1x negative switching, type A, output level readable
Nominal voltage	-		24 VDC
Nominal output current	-		3 A
Total nominal current	-		10 A 2)
Output protection	-		Thermal short-circuit shutdown, integrated protection for switching inductive loads 3)
Braking voltage when switching off inductive loads	-		Max. 90 VDC 4)
Error detection	-		1 s
Isolation voltage between channel and bus	-		500 V_{eff}
Peak short-circuit current	-		Max. 100 A
Leakage current when switched off	-		<1 mA
Residual voltage	-		≤1 VDC at nominal current
Switching voltage	-		I/O power supply minus residual voltage
Max. switching frequency	-		1000 Hz
Test pulse length	-		Max. 500 µs
Max. capacitive load	-		100 nF
Safe digital HS-HS outputs			
Variant	FET, 2x positive switching, type B2, output level readable		
Nominal voltage	24 VDC		
Nominal output current	0.2 A		50 mA
Total nominal current	0.4 A	1.2 A	100 mA
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads 3)		
Braking voltage when switching off inductive loads	Max. 45 VDC		
Error detection time	1 s		
Isolation voltage between channel and bus	500 V_{eff}		
Peak short-circuit current	Max. 10 A		500 mA
Leakage current when switched off	<100 µA		<1 mA
Residual voltage	≤1.2 VDC at nominal current		≤3 VDC at nominal current
Switching voltage	I/O power supply minus residual voltage		
Max. switching frequency	100 Hz		
Test pulse length	Max. 10 µs		Max. 500 µs
Max. capacitive load	100 nF		

Table 179: X20SC0402, X20SC0806, X20SC0842 - Technical data

Model number	X20SC0402	X20SC0806	X20SC0842
Current on loss of ground			
I_{OUT}	<100 µA		
I_{GND}	<200 mA		<50 mA 5)
Pulse outputs			
Variant	Push-Pull		
Nominal output current	50 mA		
Output protection	Shutdown of individual channels in the event of overload or short circuit 3)		
Peak short-circuit current	0.5 A for 120 µs		
Short-circuit current	15 mA_{eff}		
Leakage current when switched off	0.1 mA		
Residual voltage	≤4 VDC		
Switching voltage	I/O power supply minus residual voltage		
Total nominal current	200 mA		
Operating conditions			
Mounting orientation			
Horizontal	Yes		
Vertical	Yes		
Installation elevation above sea level	0 to 2000 m, no limitation		
Degree of protection per EN 60529	IP20		
Ambient conditions			
Temperature			
Operation			
Horizontal mounting orientation	0 to 60°C		
Vertical mounting orientation	0 to 50°C		
Derating	See section "Derating".		
Storage	-40 to 85°C		
Transport	-40 to 85°C		
Relative humidity			
Operation	5 to 95%, non-condensing		
Storage	5 to 95%, non-condensing		
Transport	5 to 95%, non-condensing		
Mechanical properties			
Note	Order 2x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.		
Spacing	$25^{+0.2}$ mm		

Table 179: X20SC0402, X20SC0806, X20SC0842 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.
3) The protective function is provided for max. 30 minutes for a continuous short circuit.
4) Due to the internal protective circuit, this braking voltage only takes effect starting at a load of typ. 250 mA.
5) The value for this module is limited to 50 mA by the nominal output current of the HS-HS outputs.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter **"Installation notes for X20 modules" on page 23****.**

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SC0402	X20SC0806	X20SC0842
Derating bonus			
At 24 VDC	+2.5°C		+5°C
At 20.4 VDC	+7.5°C		+10°C
Dummy module on the left	+2.5°C		
Dummy module on the right	+0°C		
Dummy module on the left and right	+2.5°C		+5°C
Pulse output	+7.5°C 1)		+5°C 1)
4 safe inputs (SI)	+0°C	+2.5°C 2)	+0°C
With double PFH / PFH_d	+15°C 3)		

Table 180: Derating bonus

1) Pulse output loaded with maximum 2 safe inputs (SI)
2) Only 4 safe inputs (SI) in use
3) Hardware revision E0 or later and hardware upgrade 1.10.1.0 or later

Inputs

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

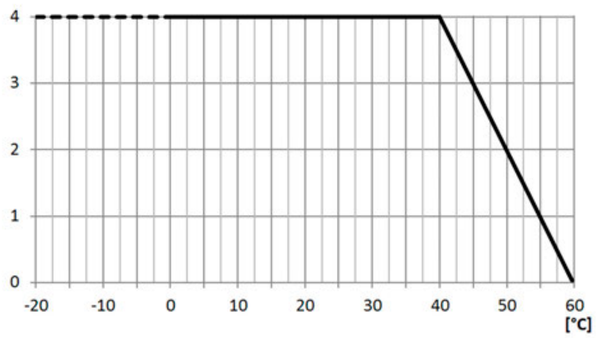
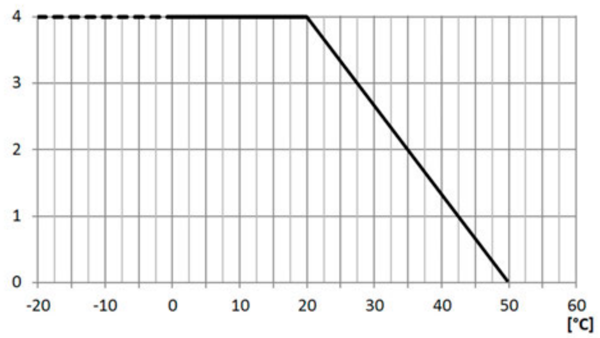
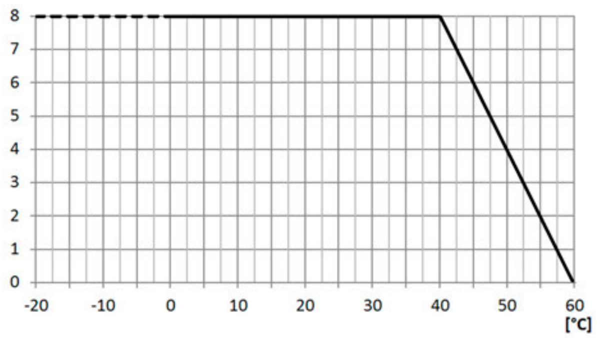
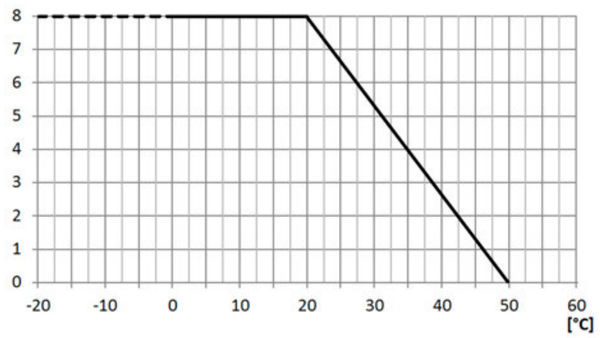
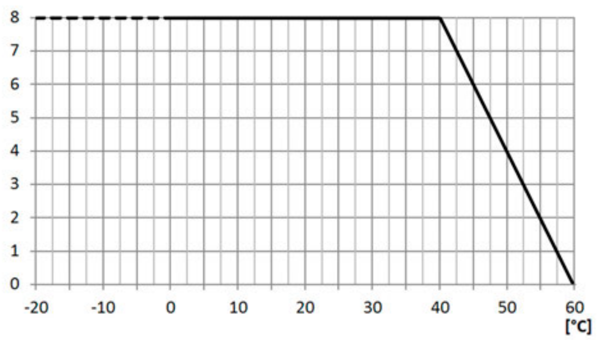
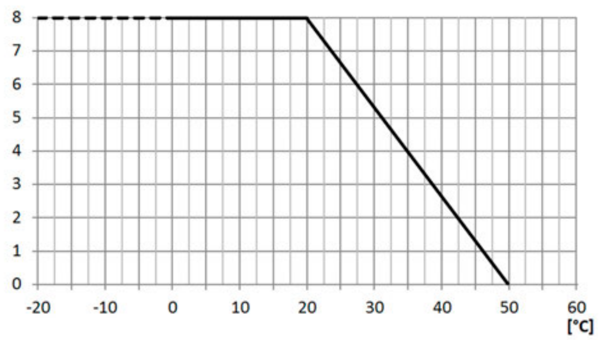
Horizontal (0 to 60°C)	Vertical (0 to 50°C)
X20SC0402	
	
X20SC0806	
	
X20SC0842	
	

Table 181: Derating in relation to operating temperature and mounting orientation

Outputs

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

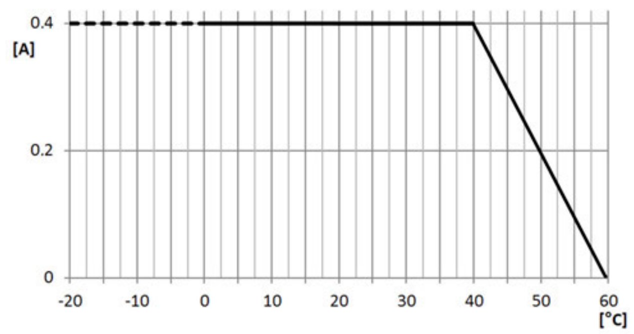
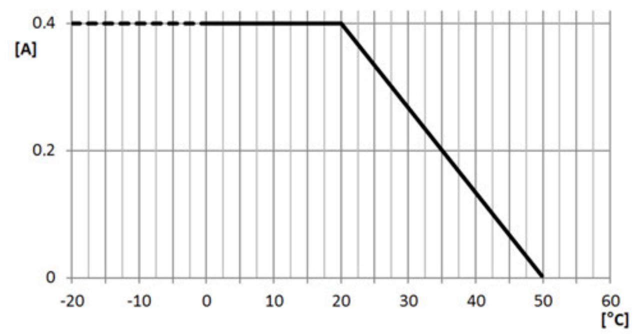
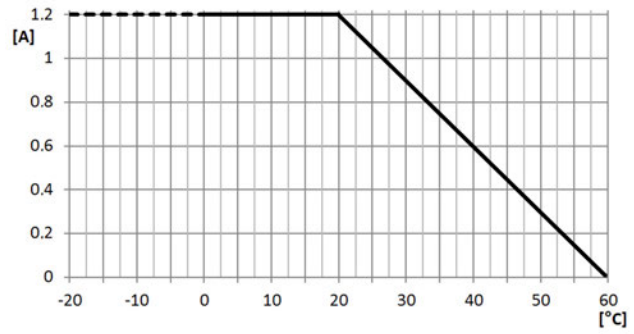
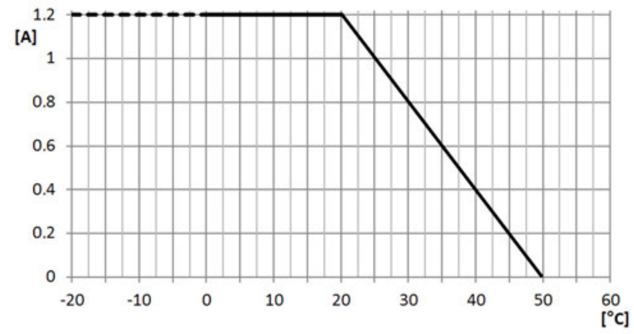
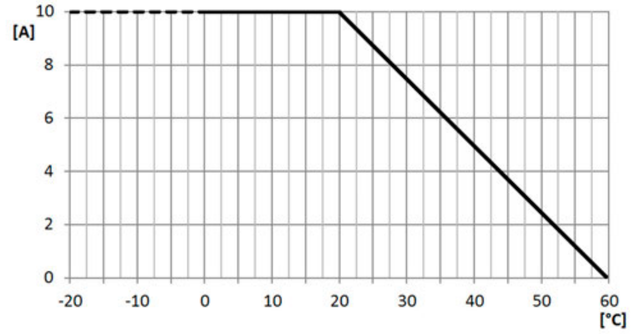
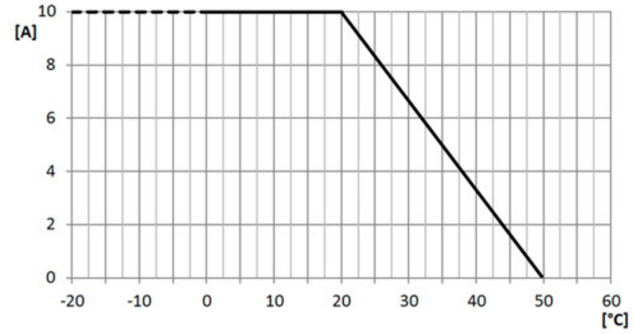
<table>
<tr><th>Horizontal (0 to 60°C)</th><th>Vertical (0 to 50°C)</th></tr>
<tr><td colspan="2">X20SC0402</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">X20SC0806</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">X20SC0842</td></tr>
<tr><td></td><td></td></tr>
</table>

Table 182: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.11.2.5 LED status indicators

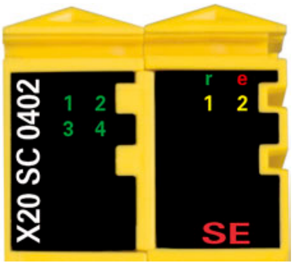
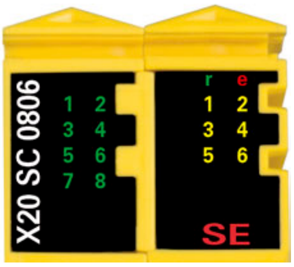
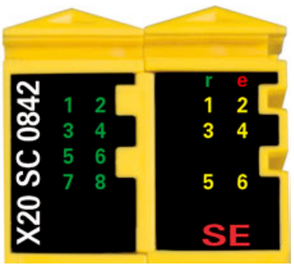
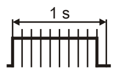
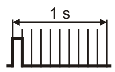
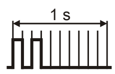
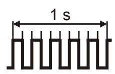
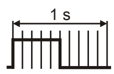
Figure	LED	Color	Status	Description
  	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 8	Input state of the corresponding digital input The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an input channel
			Blinking	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Input set
	1 to 6	Output status of the corresponding digital output The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 183: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.11.2.6 Pinouts

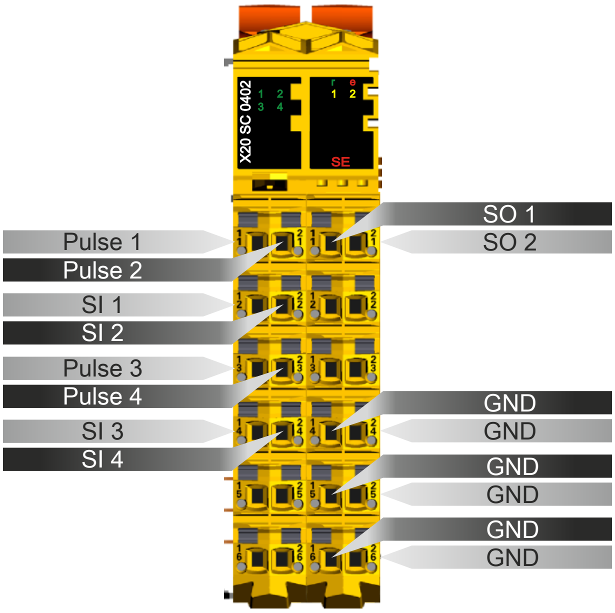


Figure 145: X20SC0402 - Pinout

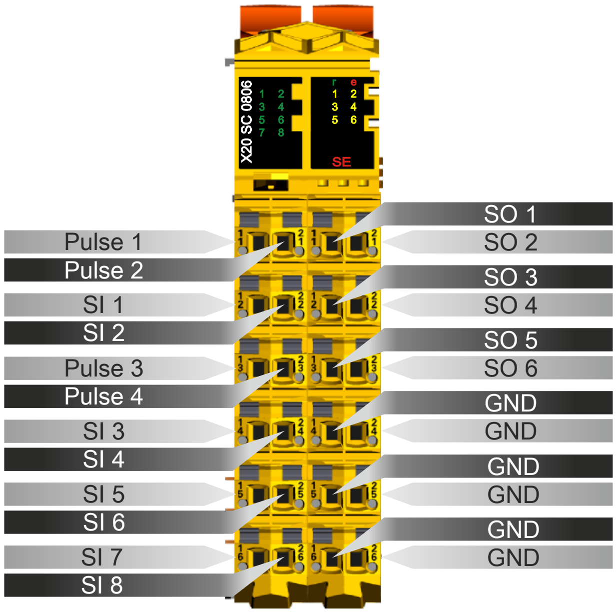


Figure 146: X20SC0806 - Pinout

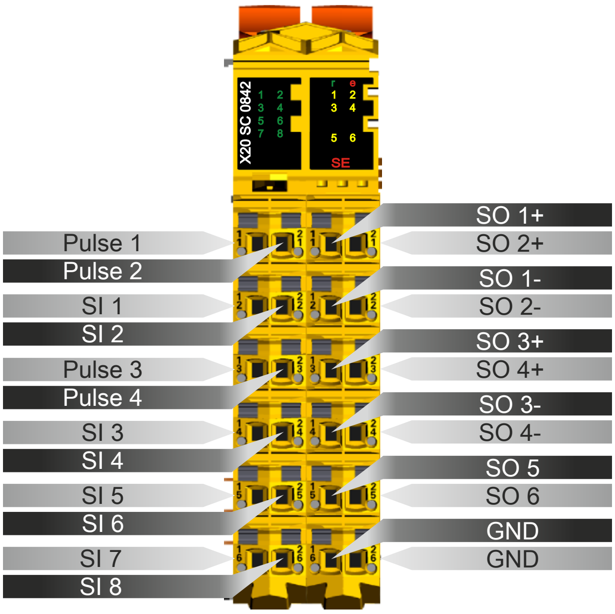


Figure 147: X20SC0842 - Pinout

2.6.11.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

> **Information:**
>
> **For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).**

2.6.11.2.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

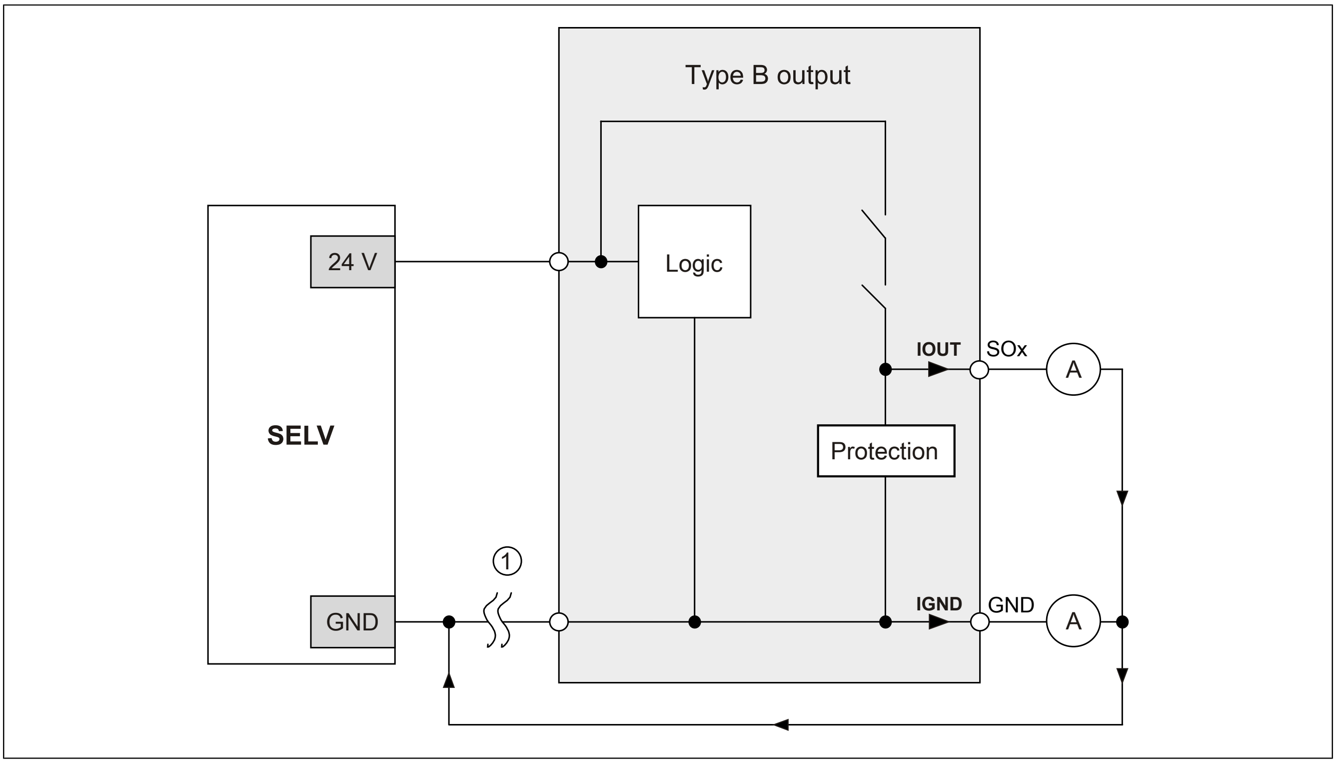


Figure 148: Module behavior when GND connection is lost

> **Danger!**
>
> **The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.**

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

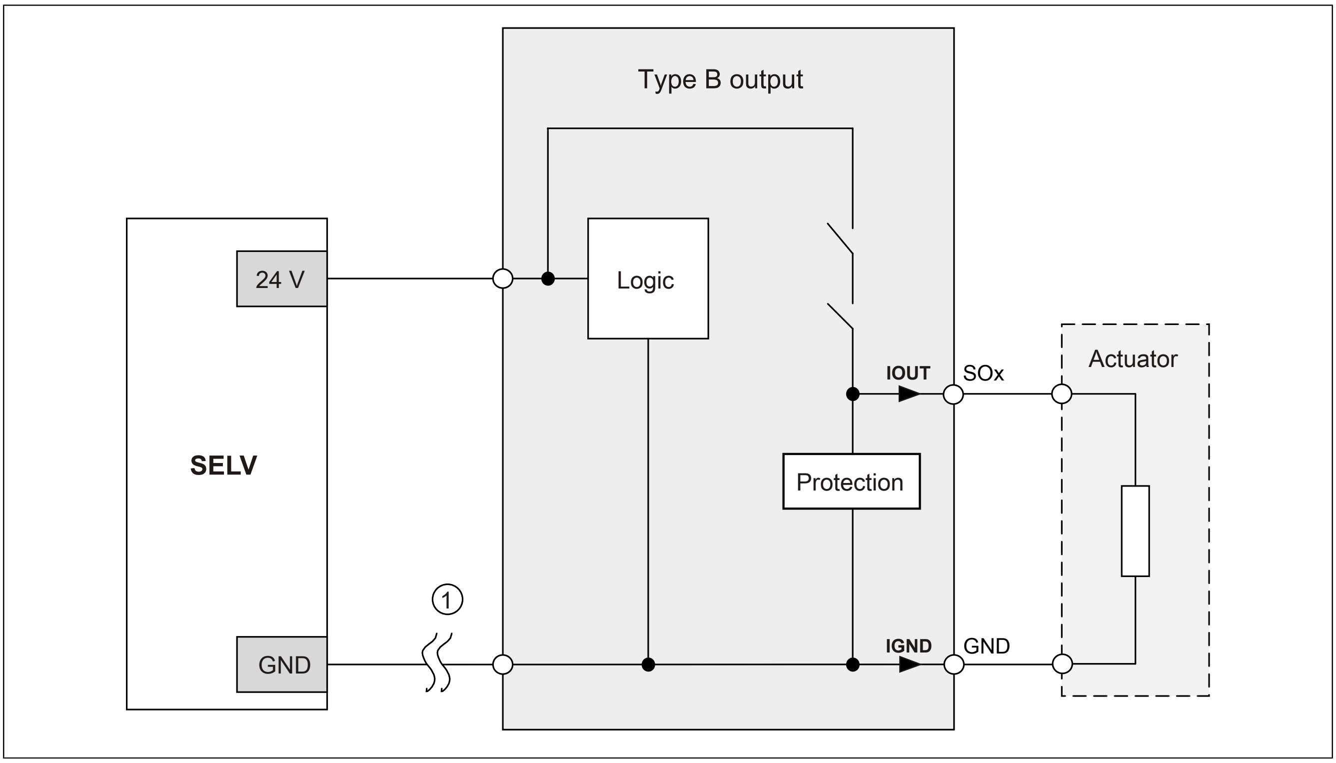


Figure 149: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

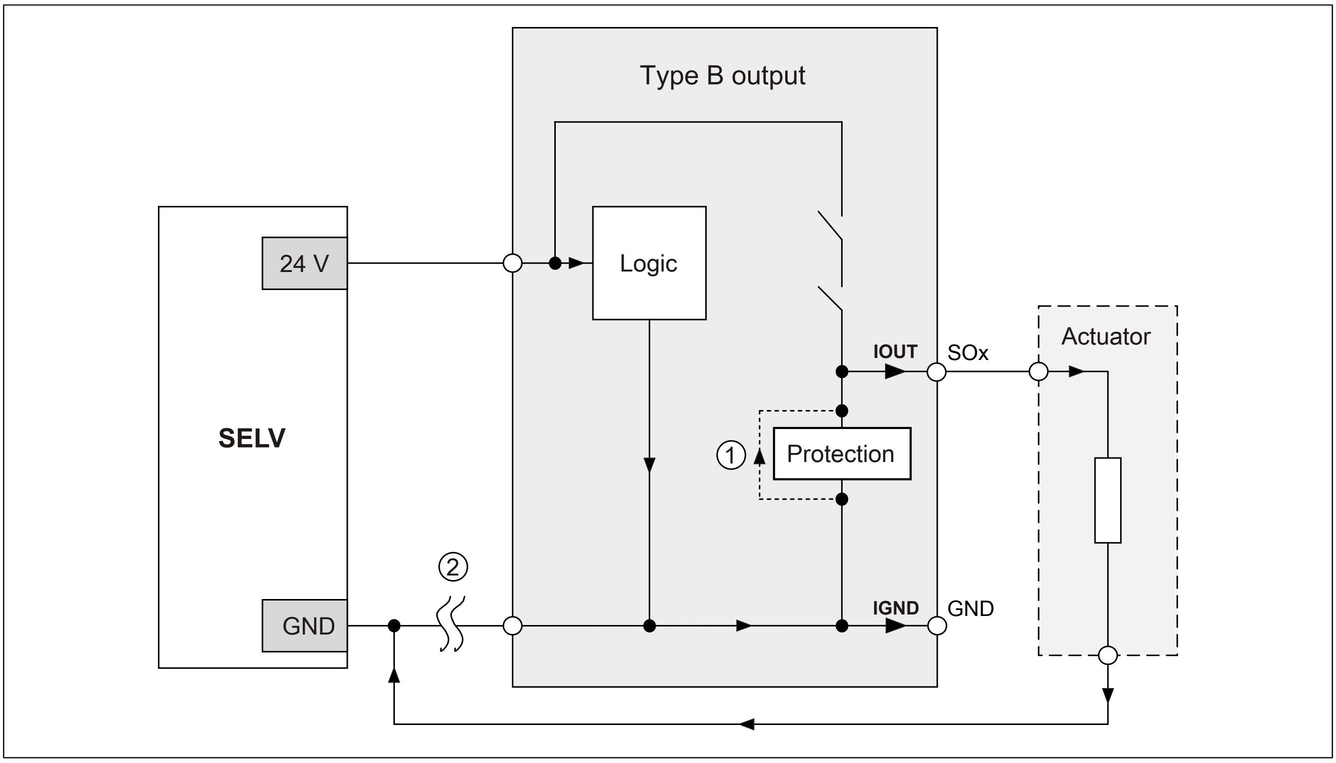


Figure 150: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

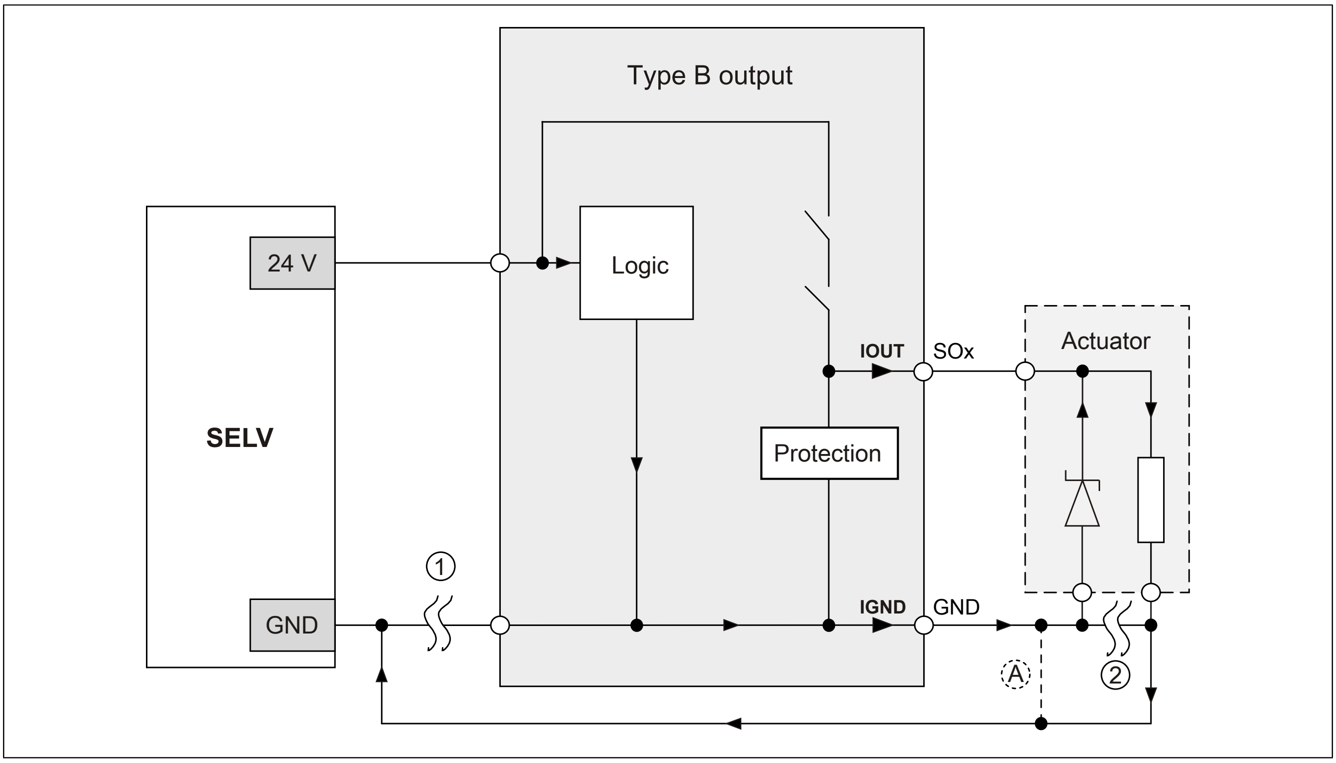


Figure 151: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

2.6.11.2.7.2 Connecting single-channel sensors with contacts

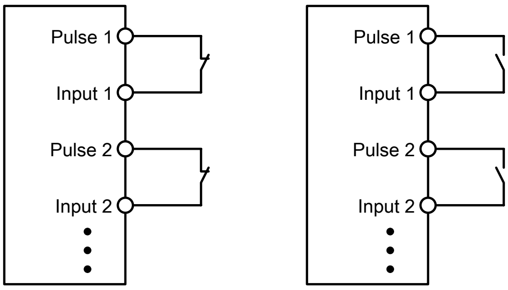


Figure 152: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.11.2.7.3 Connecting two-channel sensors with contacts

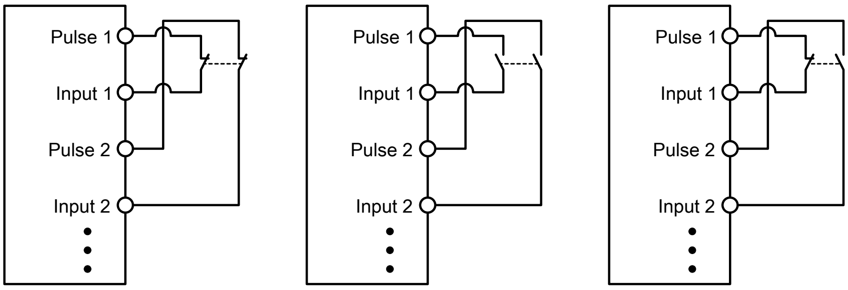


Figure 153: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.11.2.7.4 Connecting multi-channel sensors with contacts

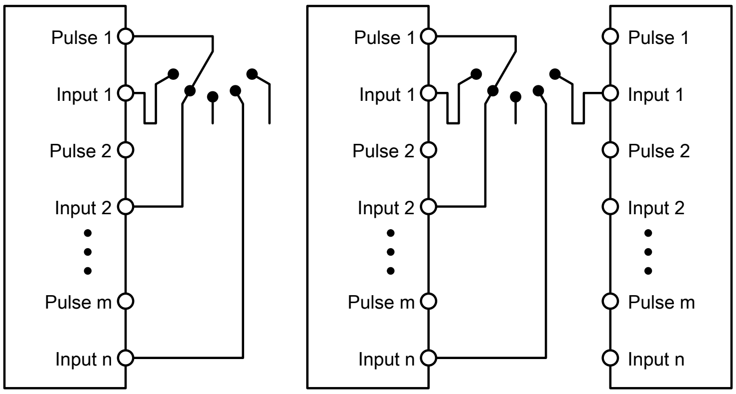


Figure 154: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.11.2.7.5 Connecting electronic sensors

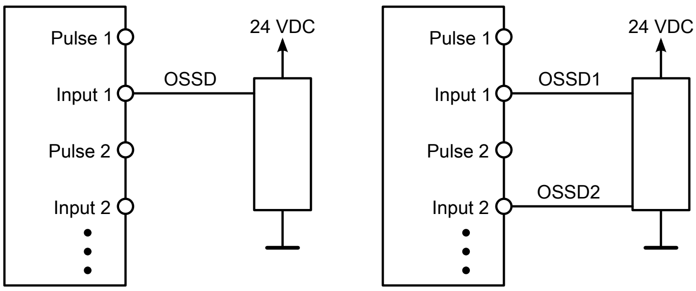


Figure 155: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.11.2.7.6 Using the same pulse signals

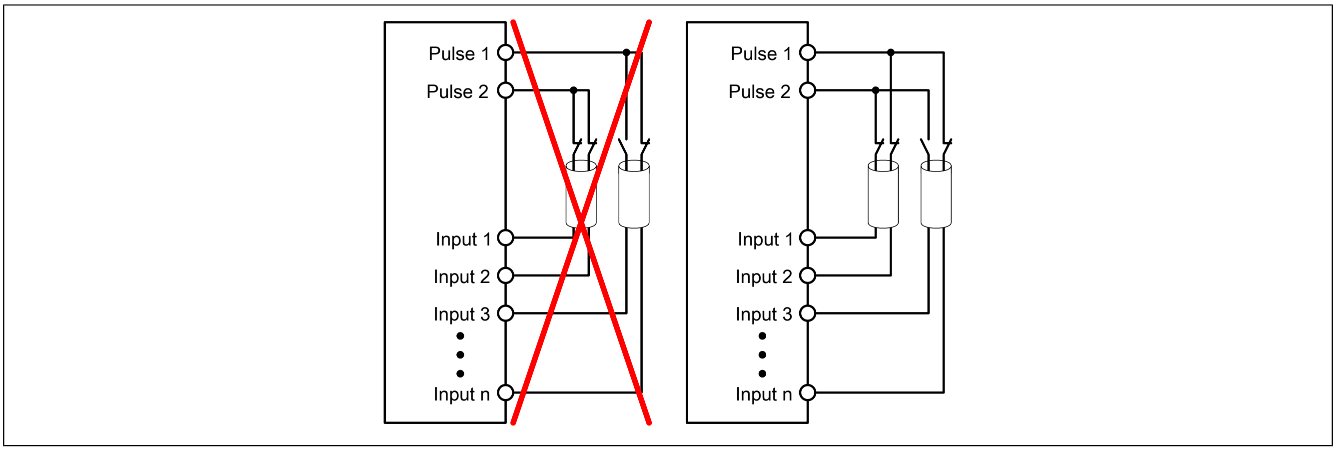


Figure 156: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.11.2.7.7 Connecting safety-oriented actuators for Type A outputs

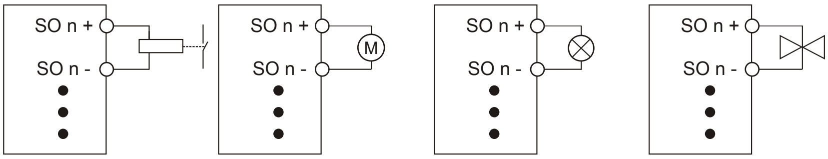


Figure 157: Connecting safety-oriented actuators for Type A outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

2.6.11.2.7.8 Connecting safety-oriented actuators for Type B outputs

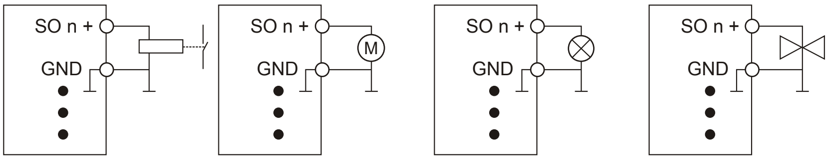


Figure 158: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.11.2.8 Error detection

2.6.11.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.11.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type A output channels

Danger!

Type A output channels also cut off the load on the GND side. Check whether the actuator you have connected permits a cutoff on the GND side. X20 and X67 systems do not support this type of cutoff, for example.

Danger!

Note that wiring SOx+ directly to GND via an actuator is not permitted; wiring 24 VDC directly to SOx- via an actuator is also not permitted.

These types of errors will not be detected by the module. The user must prevent these types of errors through careful wiring.

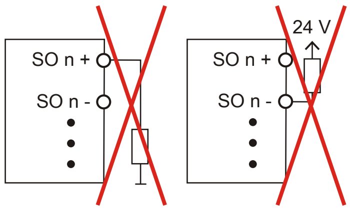


Figure 159: Invalid wiring

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

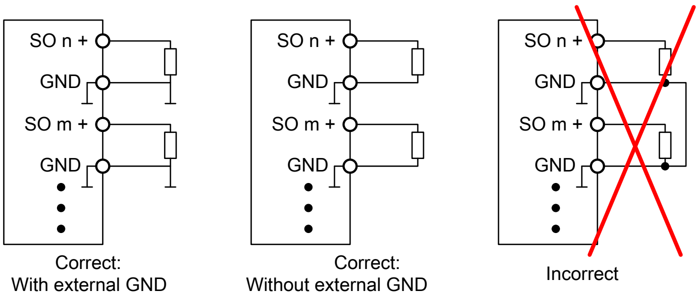


Figure 160: Invalid wiring

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 184: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 185: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 186: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td>Detected</td><td>Detected</td><td>Detected</td><td>Not detected</td></tr>
</table>

Table 187: SO error detection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td>X20SLXxxx</td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td><td rowspan="3"></td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected 2)</td><td>Not detected 2)</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Short circuit between SOx+ (output type A) and SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
</table>

Table 187: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.

2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.11.2.9 Input circuit diagram

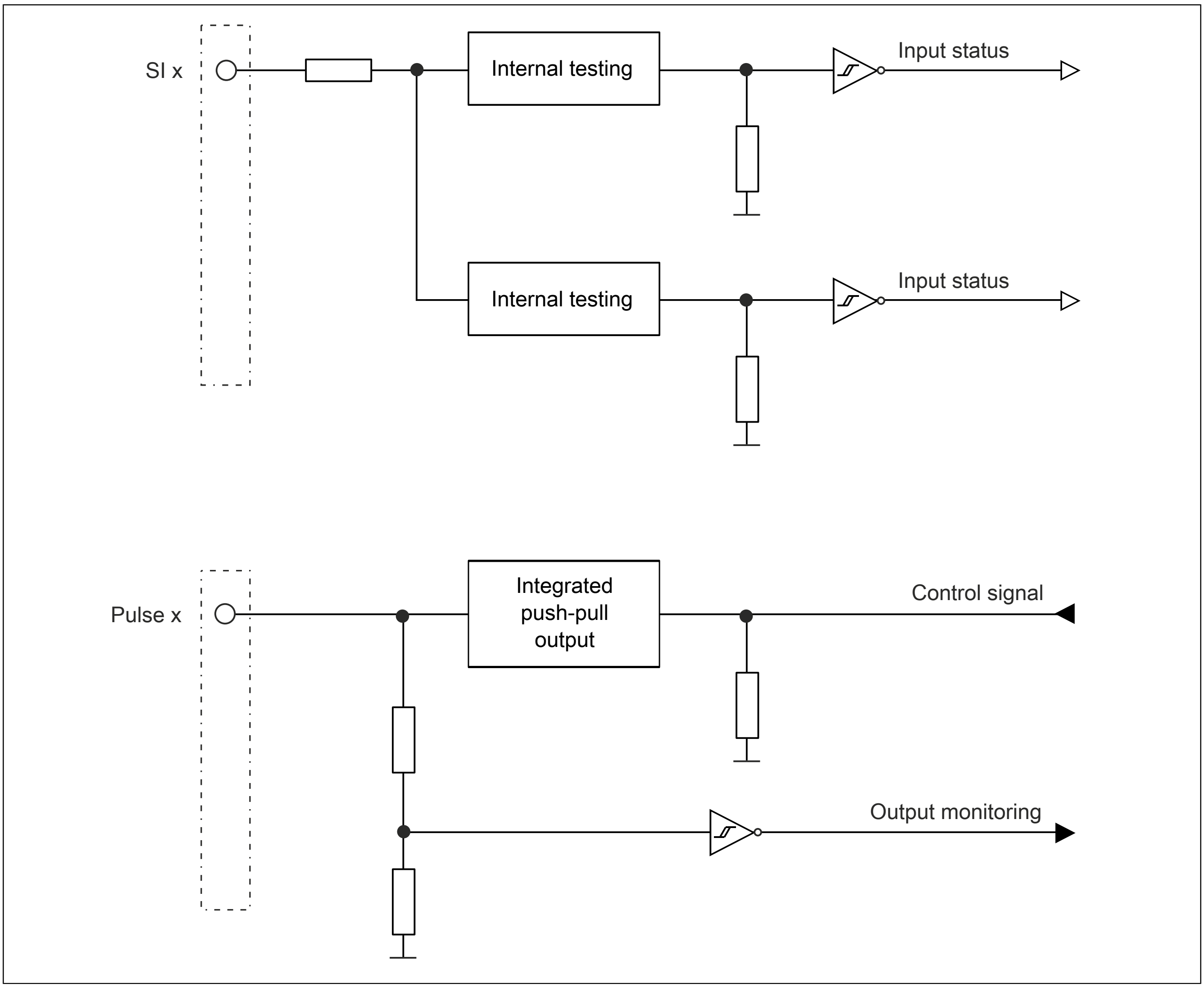


Figure 161: Input circuit diagram

2.6.11.2.10 Type A output circuit diagram

Type A digital output channels are designed for positive and GND switching inside the module.

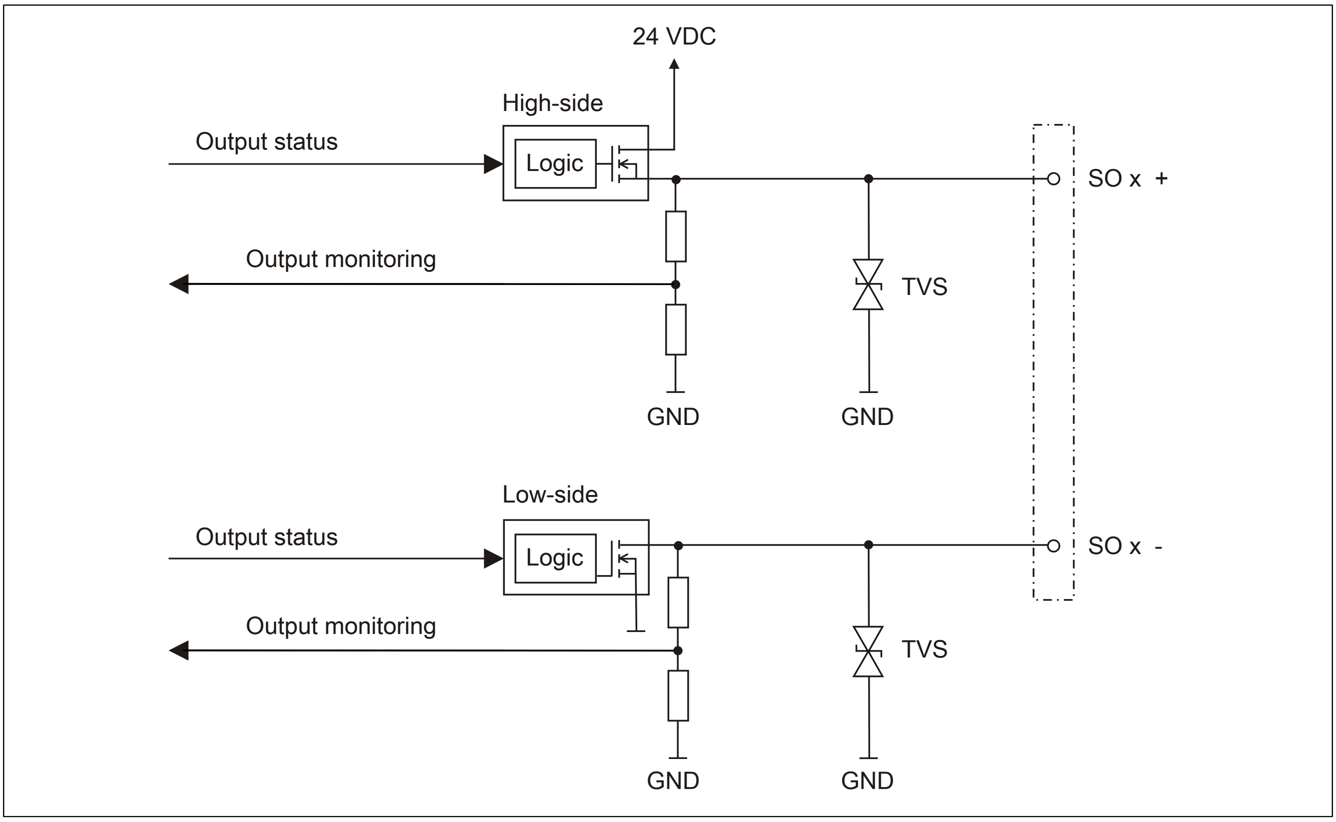


Figure 162: Type A output circuit diagram

2.6.11.2.11 Type B output circuit diagram

Type B digital output channels are designed for positive and positive switching inside the module.

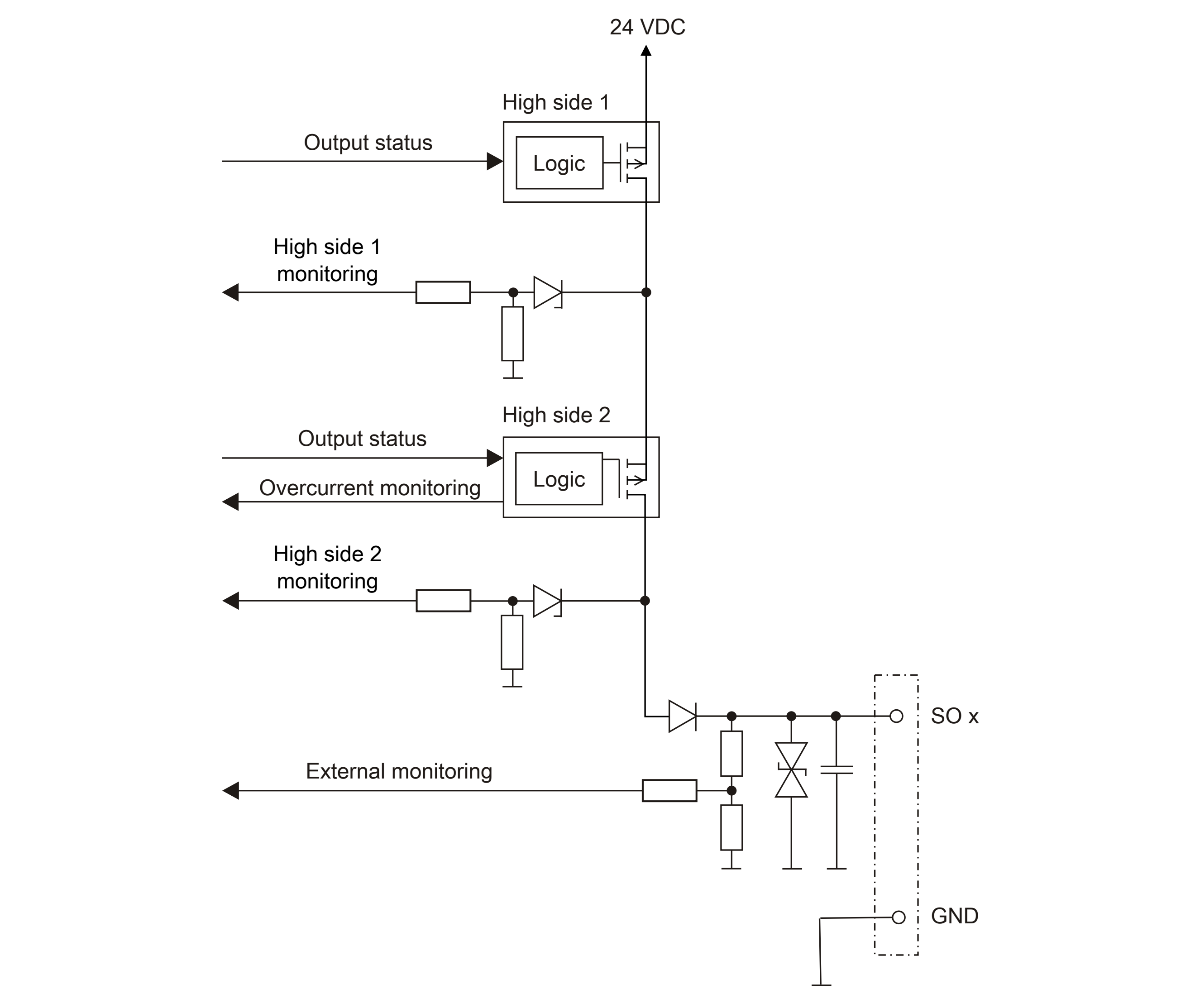


Figure 163: Type B output circuit diagram

2.6.11.2.12 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.11.2.13 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 µs

Maximum I/O update time for input channels
1150 µs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels
1300 µs

2.6.11.2.14 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

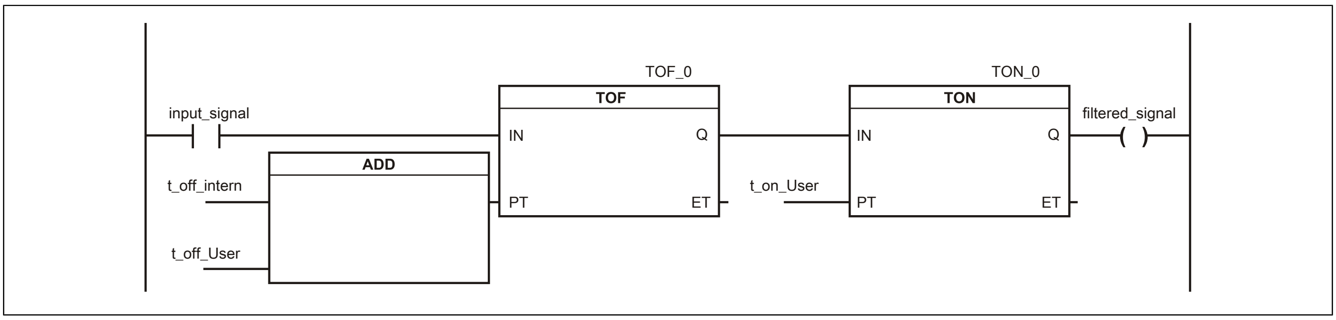


Figure 164: SI input filter - Diagram 1

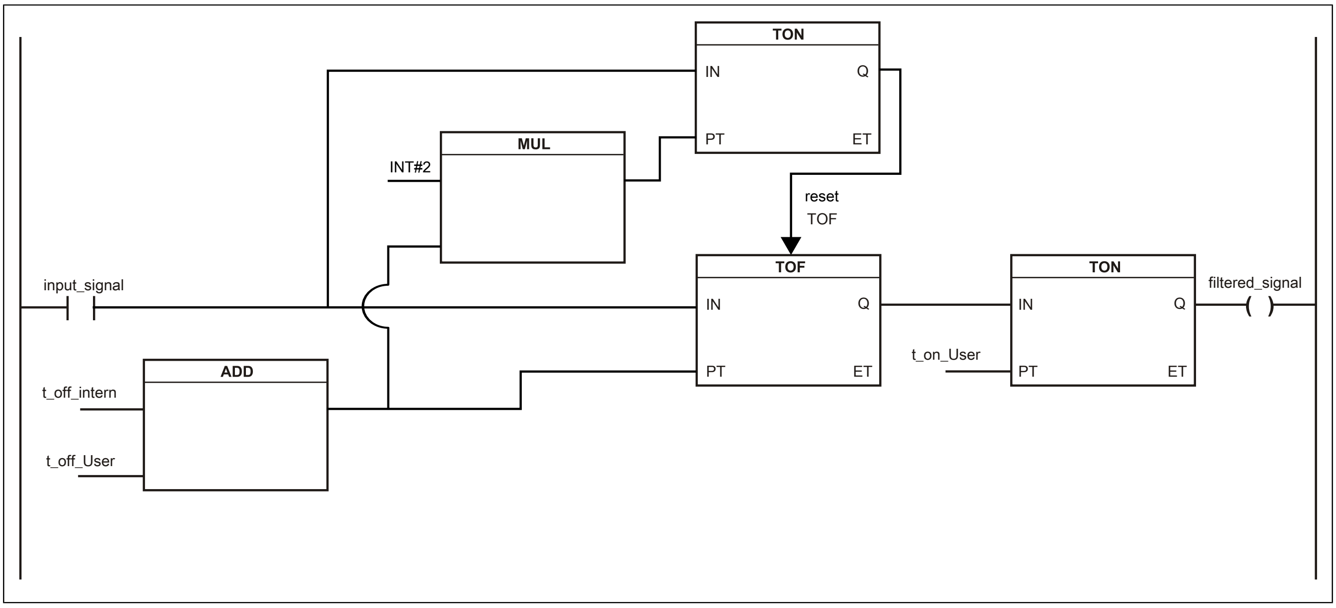


Figure 165: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.11.2.15 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.11.2.16 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.2.17 Register description

2.6.11.2.17.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 188: I/O configuration parameters: Function model

Group: General

Parameter	Description	Default value	Unit
Module supervised	System behavior when a module is missing **Parameter value** / **Description** On: A missing module triggers service mode. Off: A missing module is ignored.	On	-
Blackout mode (hardware upgrade 1.10.0.6 or later)	This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode). **Parameter value** / **Description** On: Blackout mode is enabled. Off: Blackout mode is disabled.	Off	-
Channel status information	This parameter enables/disables the channel-specific status information in the I/O mapping.	On	-
State number of 2-channel evaluation	This parameter enables/disables the status information of dual-channel evaluation.	Off	-
Restart inhibit state numbers	This parameter enables/disables restart interlock status information.	Off	-
SafeLOGIC ID	In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller. • Permissible values: 1 to 1024	Assigned automatically	-
SafeMODULE ID	Unique safety address of the module • Permissible values: 2 to 1023	Assigned automatically	-

Table 189: I/O configuration parameters: General

Group: Output signal path

Parameter	Description	Default value	Unit
DigitalOutputxx	This parameter specifies the mode that can be used by the standard application to access the output channel. **Parameter value** / **Description** Direct: The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly. Via SafeLOGIC: The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.	Direct	-

Table 190: I/O configuration parameters: Output signal path

2.6.11.2.17.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Disable_OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 191: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2010 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2010 or PL d per EN ISO 13849-1:2015, a daily check of the safety function by the user is necessary in which all of the module's output channels are in a switched-off state simultaneously for at least 1 second.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr>
<tr><td>No</td><td>No requirement for synchronization of the networks</td></tr>
</table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 192: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalInputxx

Parameter	Description	Default value	Unit
Pulse_Source	This parameter can be used to specify the pulse source for the input channel.	Default	-
	All available pulse outputs can be specified as "Pulse_Source". The default values can be determined using the following table: <table><tr><th>Channel</th><th>Default "Pulse_Source"</th></tr><tr><td>1, 5</td><td>Channel 1</td></tr><tr><td>2, 6</td><td>Channel 2</td></tr><tr><td>3, 7</td><td>Channel 3</td></tr><tr><td>4, 8</td><td>Channel 4</td></tr></table>		
	Note: If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".		
Pulse_Mode	This parameter can be used to specify the pulse mode for the input channel.	Internal	-
	<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Internal</td><td>The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr><tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr></table>		
Filter_Off_us	Switch-off filter for the channel to remove potentially disruptive signal "low phases". • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	0	µs
Filter_On_us	Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short. • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	200000	µs
Discrepancy_Time_us	Parameter only available for odd-numbered channels. This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error. • Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)	0	µs
TwoChannelProcessingMode	Parameter only available for odd-numbered channels. This parameter specifies the type of dual-channel evaluation. Permissible values: • None • Equivalent • Antivalent	None	-

Table 193: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto_Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr><tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr></table></td></tr>
</table>

Table 194: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.2.17.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 195: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 196: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Disable OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr><tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr></table></td></tr>
</table>

Table 197: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">All available pulse outputs can be specified as "Pulse Source". The default values can be determined using the following table:
<table>
<tr><th>Channel</th><th>Default "Pulse Source"</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table></td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table></td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal "low phases".
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Two-Channel Processing Mode</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 198: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr><tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr></table></td></tr>
</table>

Table 199: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.2.17.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1)]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1)]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>(Read) [1)]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
</table>

Table 200: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelOkxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UDINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table>
<tr><th>Bit 23 to 20</th><th>Bit 19 to 16</th><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr>
<tr><td>Channel 6</td><td>Channel 5</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table>
</td></tr>
</table>

Table 200: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

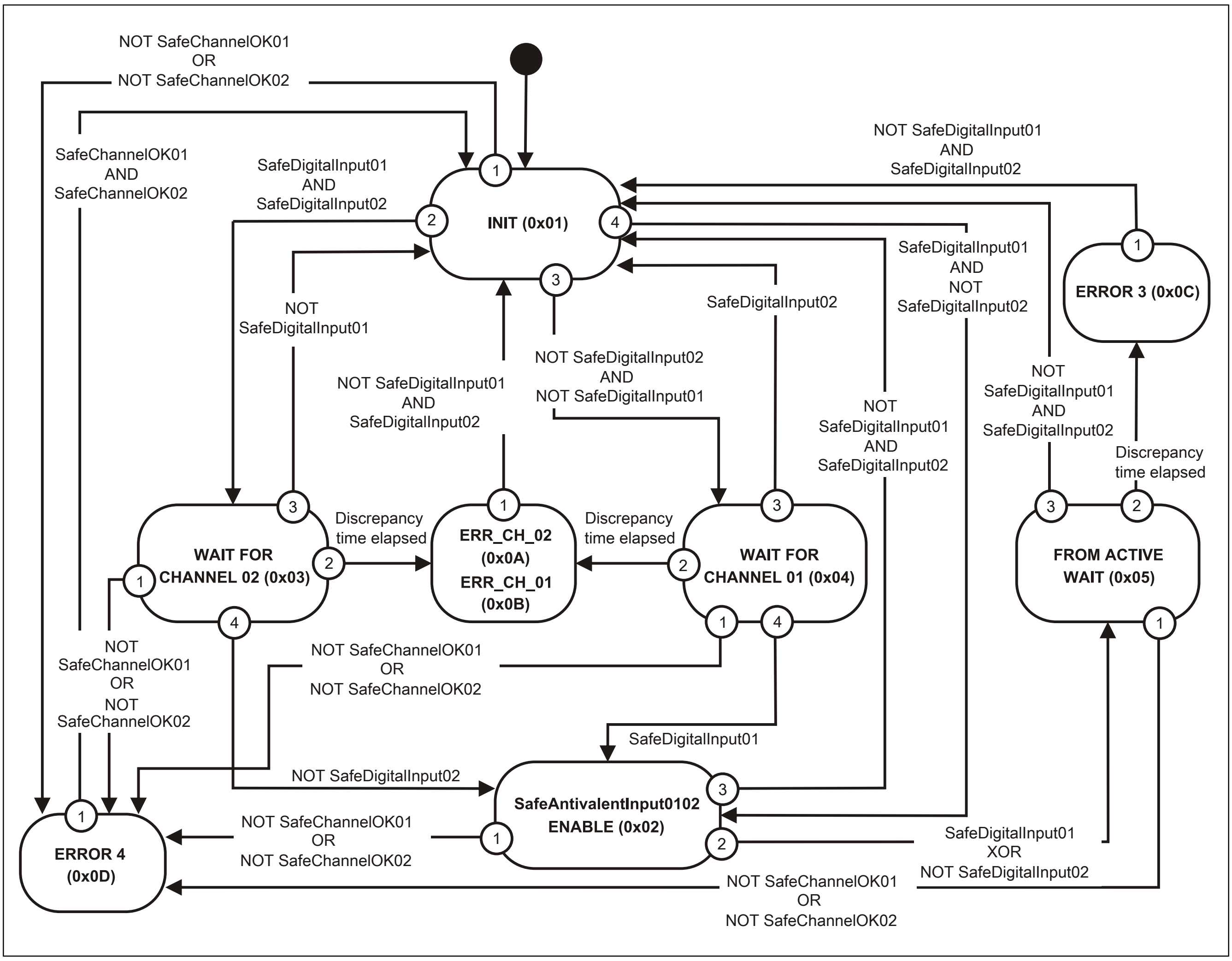


Figure 166: "Antivalent" function block - State diagram

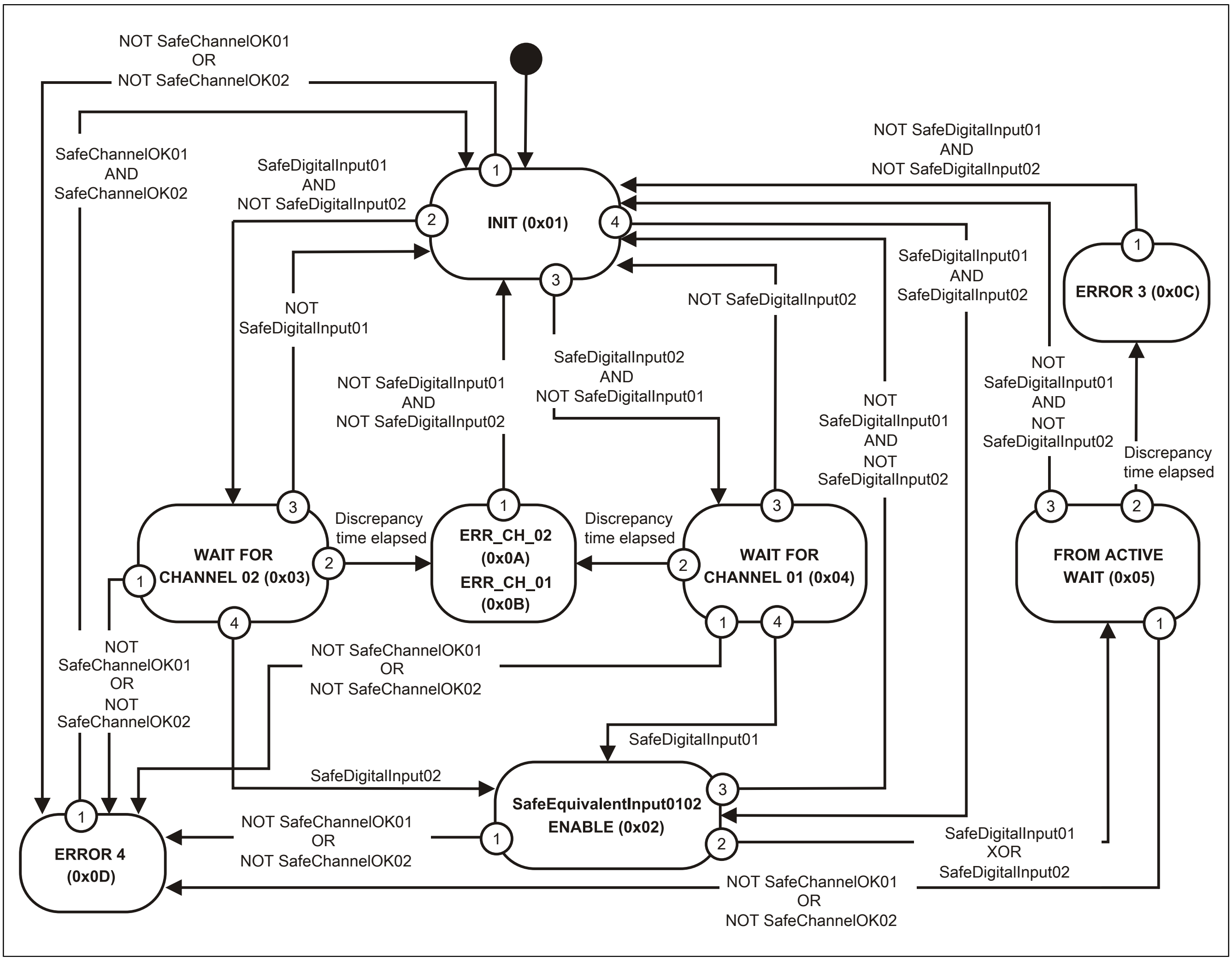


Figure 167: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

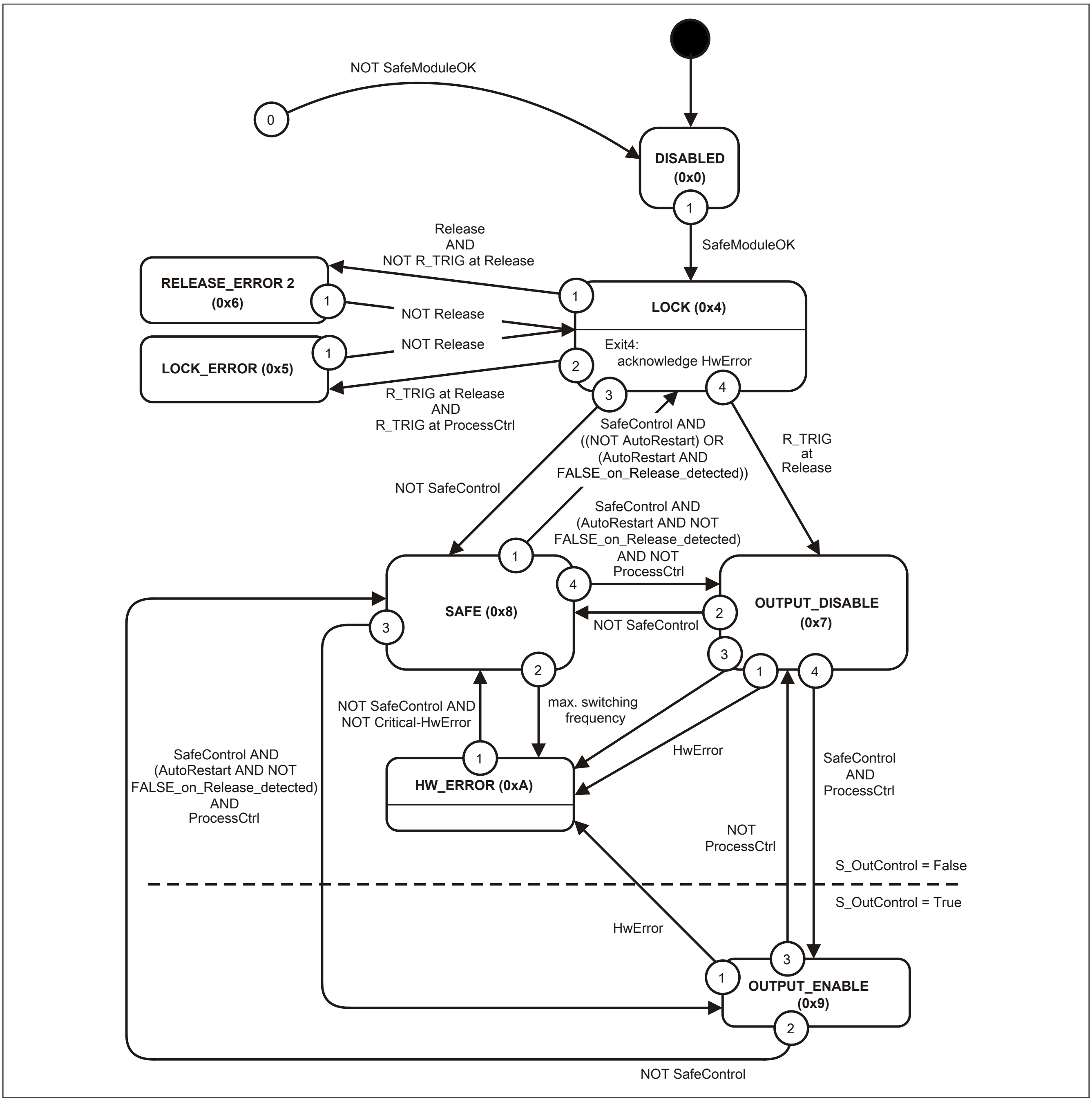


Figure 168: Restart interlock - State diagram

2.6.11.3 X20(c)SC2212

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 201: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 202: Organization of general notices

2.6.11.3.1 General information

The modules are equipped with 6 safe digital inputs and 2 safe digital outputs. They are designed for a nominal voltage of 24 VDC.

The modules can be used to read in digital signals and control actuators in safety-related applications up to PL e or SIL 3.

The modules are equipped with filters that are individually configurable for switch-on and switch-off behavior. The modules also provide pulse signals for diagnosing the sensor line.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 16-pin terminal blocks.

- 6 safe digital inputs, sink circuit
- 6 pulse outputs
- Software input filter configurable for each channel
- 2 safe digital outputs, output type B with 0.5 A, source circuit
- Integrated output protection

2.6.11.3.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.11.3.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

Information:

For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.11.3.2 Overview

Module	X20SC2212
Safe digital inputs	
Number of inputs	6
Nominal voltage	24 VDC
Input filter Hardware Software	 ≤150 µs Configurable between 0 and 500 ms
Input circuit	Sink
Pulse outputs	
Design	Push-Pull
Switching voltage	I/O power supply minus residual voltage
Safe digital outputs	
Number of outputs	2
Nominal voltage	24 VDC
Nominal output current	0.5 A
Total nominal current	1 A
Output protection	Thermal short circuit shutdown, integrated protection for switching inductive loads

Table 203: Digital mixed modules

2.6.11.3.3 Order data

Model number	Short description	Figure
	Digital mixed modules	
X20SC2212	X20 safe digital mixed module, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	
X20cSC2212	X20 safe digital mixed module, coated, 6 safe digital inputs, configurable input filter, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	
	Terminal blocks	
X20TB5F	X20 terminal block, 16-pin, safety-keyed	

Table 204: X20SC2212, X20cSC2212 - Order data

2.6.11.3.4 Technical data

Model number	X20SC2212	X20cSC2212
Short description		
I/O module	6 safe digital inputs, 6 pulse outputs, 24 VDC, 2 safe type B1 digital outputs, 24 VDC, 0.5 A, OSSD <500 µs	
General information		
B&R ID code	0xBDA5	0xDD9D
System requirements		
Automation Studio	3.0.81.15 or later	4.0.16 or later
Automation Runtime	3.00 or later	V3.08 or later
SafeDESIGNER	2.70 or later	3.1.0 or later
Safety Release	1.2 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using status LED and software	
Outputs	Yes, using status LED and software	
Inputs	Yes, using status LED and software	
Blackout mode		
Scope	Module	
Function	Module function	
Standalone mode	No	
Max. I/O cycle time	1 ms	
Power consumption		
Bus	0.25 W	
Internal I/O	1.4 W	
Electrical isolation		
Channel - Bus	Yes	
Channel - Channel	No	
Certifications		
CE	Yes	
KC	Yes	-
EAC	Yes	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013	
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3	
Functional safety	EN 50156-1:2004	
Safety characteristics		
EN ISO 13849-1:2015		
MTTFD	2500 years	
Mission time	Max. 20 years	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
PFH / PFH_d		
Module	$<1*10^{-10}$	
openSAFETY wired	Negligible	
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour	
PFD	$<2*10^{-5}$	
Proof test interval (PT)	20 years	

Table 205: X20SC2212, X20cSC2212 - Technical data

Model number	X20SC2212	X20cSC2212
Safe digital inputs		
EN ISO 13849-1:2015		
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1]	
PL	PL e	
DC	>94%	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 3	
SFF	>90%	
Safe digital outputs		
EN ISO 13849-1:2015		
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" [1]	
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" [1]	
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" [1]	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" [1]	
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" [1]	
I/O power supply		
Nominal voltage	24 VDC	
Voltage range	24 VDC -15% / +20%	
Integrated protection	Reverse polarity protection	
Safe digital inputs		
Nominal voltage	24 VDC	
Input characteristics per EN 61131-2	Type 1	
Input filter		
Hardware	≤150 µs	
Software	Configurable between 0 and 500 ms	
Input circuit	Sink	
Input voltage	24 VDC -15% / +20%	
Input current at 24 VDC	Max. 3.28 mA	
Input resistance	Min. 7.33 kΩ	
Error detection time	100 ms	
Isolation voltage between channel and bus	500 V_{eff}	
Switching threshold		
Low	<5 VDC	
High	>15 VDC	
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line	
Safe digital outputs		
Variant	FET, 2x positive switching, type B1, output level readable	
Nominal voltage	24 VDC	
Nominal output current	0.5 A	
Total nominal current	1 A	
Output protection	Thermal short-circuit shutdown, integrated protection for switching inductive loads [2]	
Braking voltage when switching off inductive loads	Max. 45 VDC	
Error detection time	1 s	
Isolation voltage between channel and bus	500 V_{eff}	
Peak short-circuit current	Max. 12 A	
Leakage current when switched off	<500 µA	
Residual voltage	≤300 mVDC at nominal current	
Switching voltage	I/O power supply minus residual voltage	
Max. switching frequency	1000 Hz	
Test pulse length	Max. 500 µs	
Max. capacitive load	100 nF	
Current on loss of ground		
I_{OUT}	<1 mA	
I_{GND}	<180 mA	
Pulse outputs		
Variant	Push-Pull	
Nominal output current	20 mA	
Output protection	Shutdown of individual channels in the event of overload or short circuit [2]	
Peak short-circuit current	25 A for 15 µs	
Short-circuit current	100 mA_{eff}	
Leakage current when switched off	0.1 mA	
Residual voltage	3 VDC	

Table 205: X20SC2212, X20cSC2212 - Technical data

Model number	X20SC2212	X20cSC2212
Switching voltage	I/O power supply minus residual voltage	
Total nominal current	120 mA	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level	0 to 2000 m, no limitation	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	0 to 60°C	-40 to 60°C 3)
Vertical mounting orientation	0 to 50°C	-40 to 50°C 4)
Derating	See section "Derating".	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm	

Table 205: X20SC2212, X20cSC2212 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Up to hardware upgrade <1.10.1.0 and hardware revision <E0: -25 to 60°C
4) Up to hardware upgrade <1.10.1.0 and hardware revision <E0: -25 to 50°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SC2212
Derating bonus	
At 24 VDC	+5°C
Dummy module on the left	+2.5°C
Dummy module on the right	+0°C
Dummy module on the left and right	+5°C
With double PFH / PFH_d	+0°C

Table 206: Derating bonus

Inputs

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

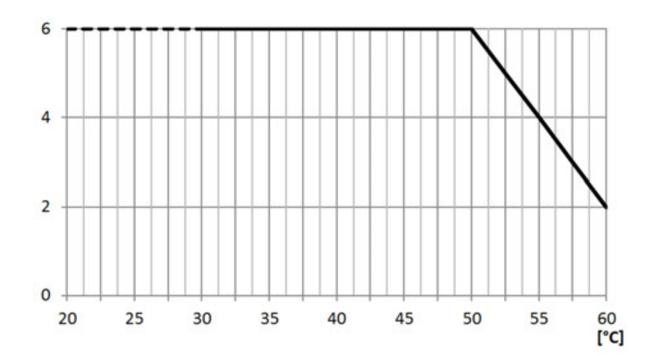
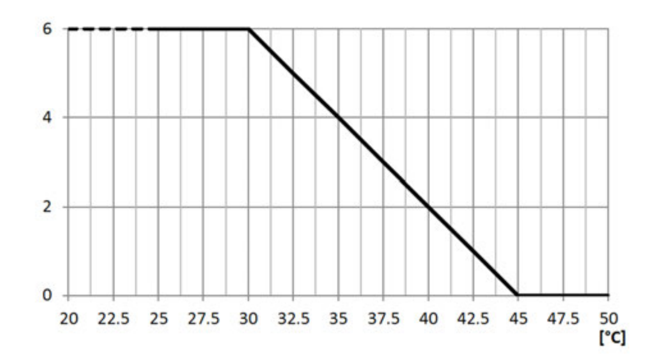
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
	

Table 207: Derating in relation to operating temperature and mounting orientation

Outputs

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

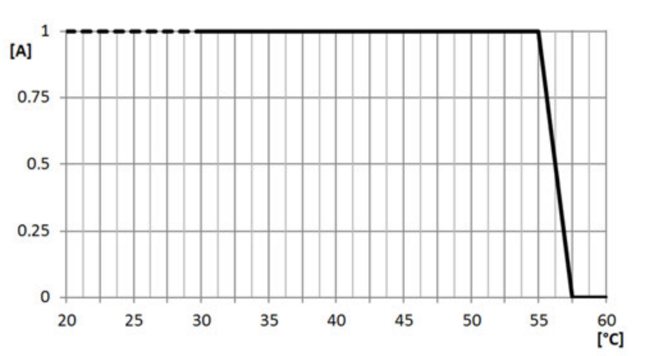
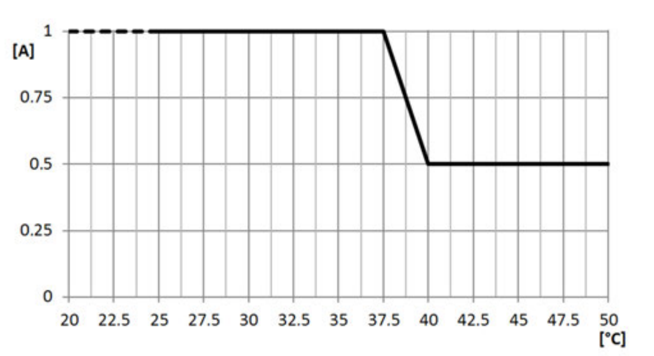
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
	

Table 208: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.11.3.5 LED status indicators

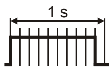
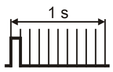
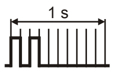
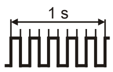
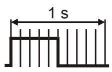
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 6	Input state of the corresponding digital input		
		Red	On	Warning/Error on an input channel
			Blinking	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Input set
	1 to 2	Output status of the corresponding digital output		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 209: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.11.3.6 Pinout

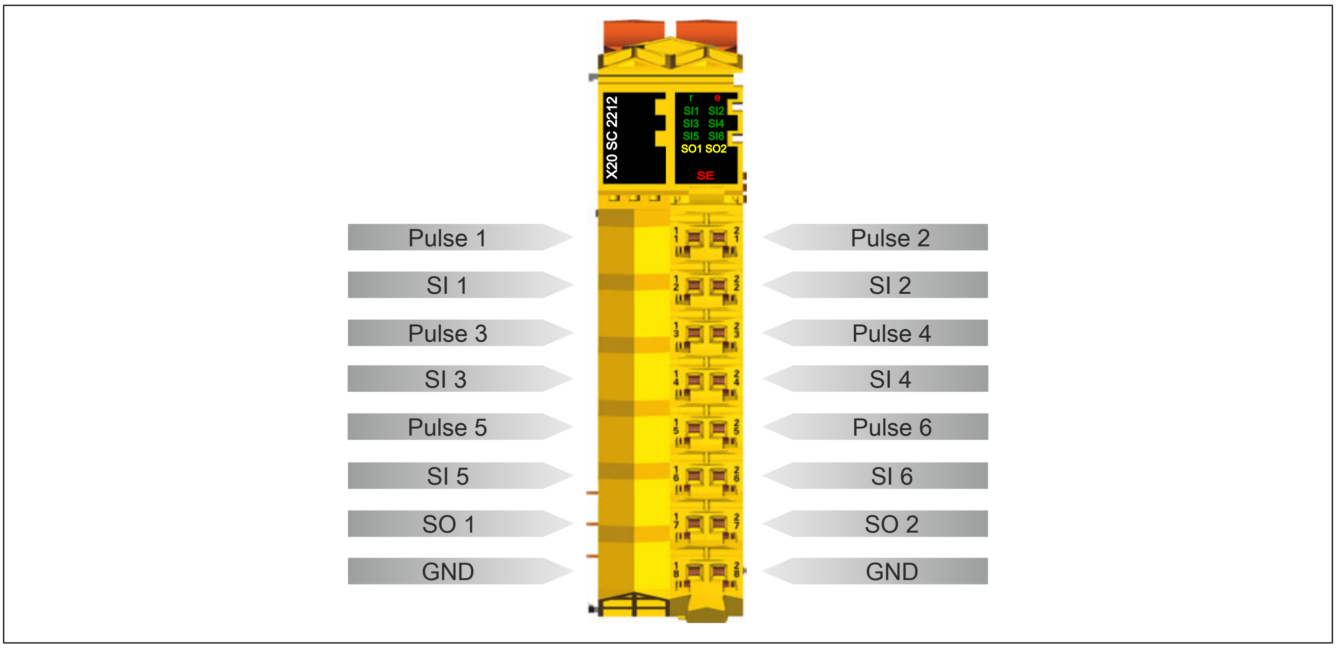


Figure 169: X20SC2212 - Pinout

2.6.11.3.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.11.3.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

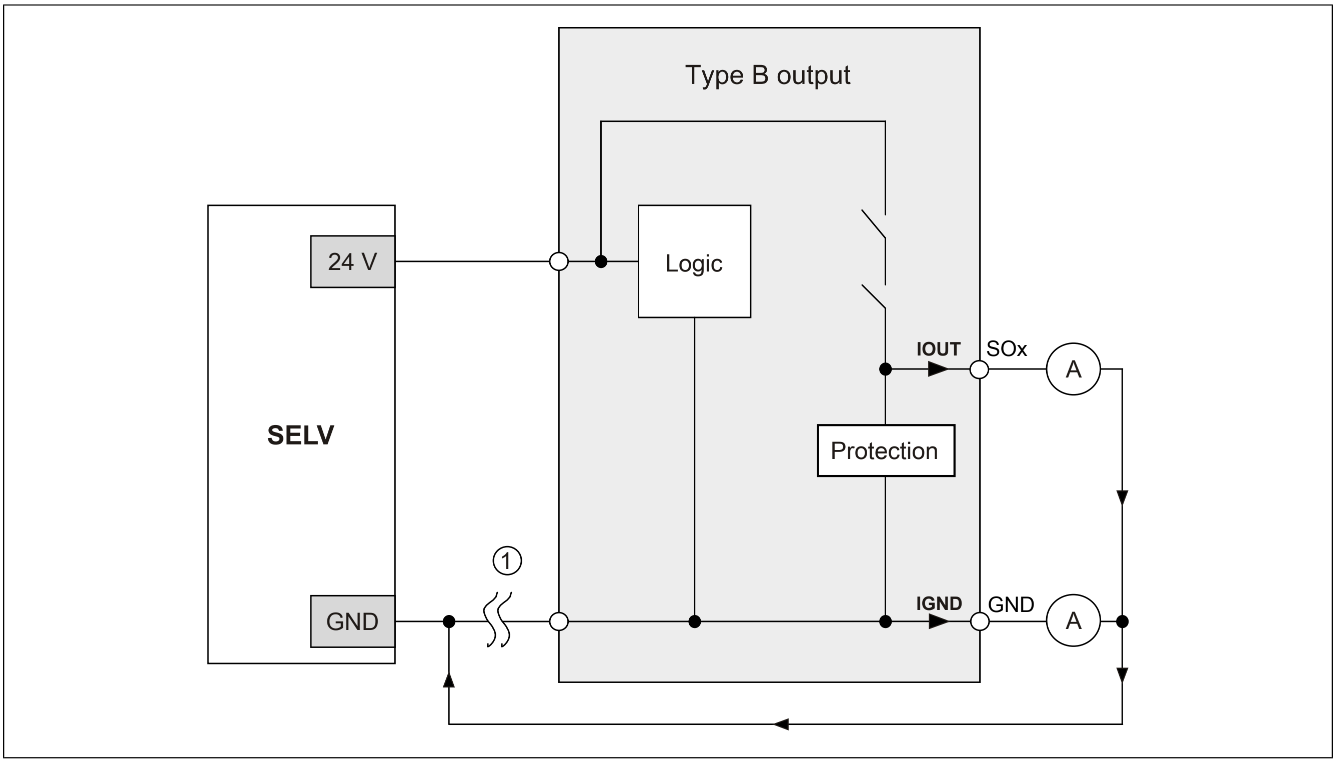


Figure 170: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

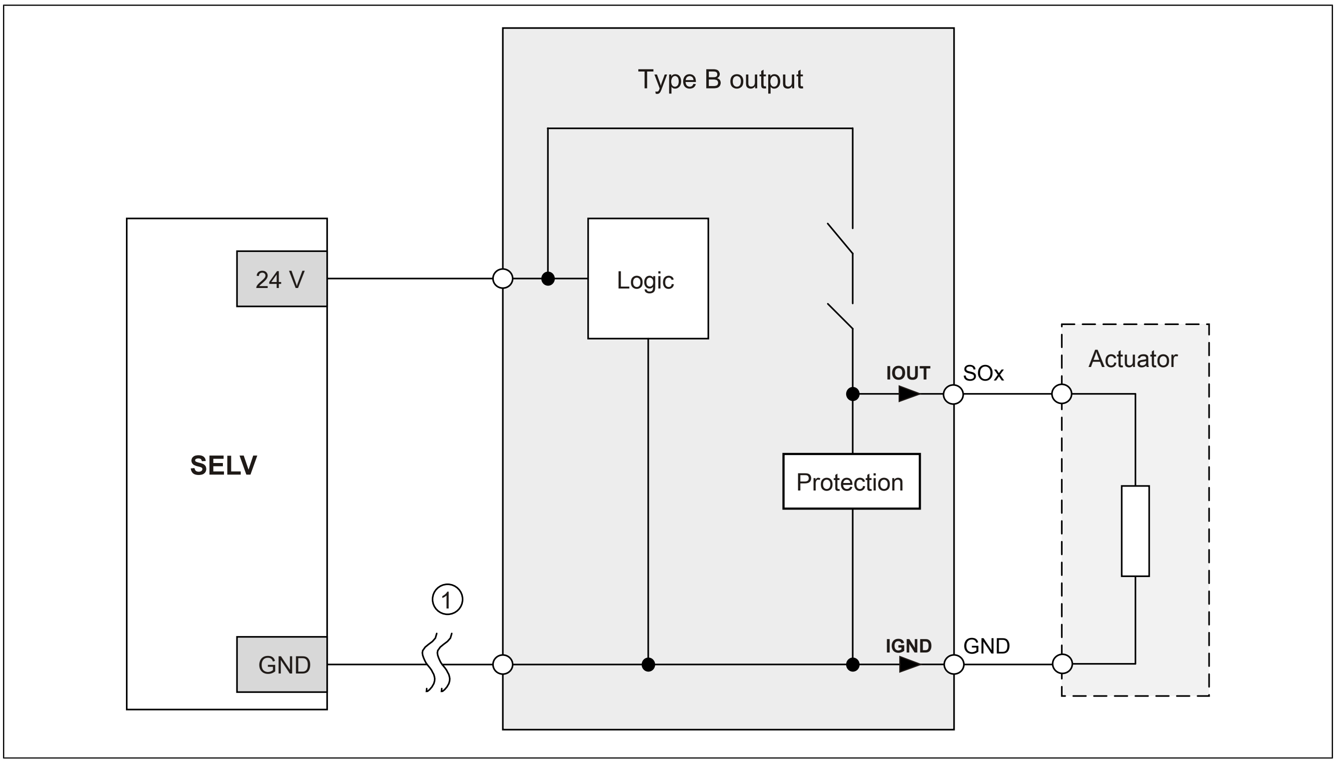


Figure 171: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

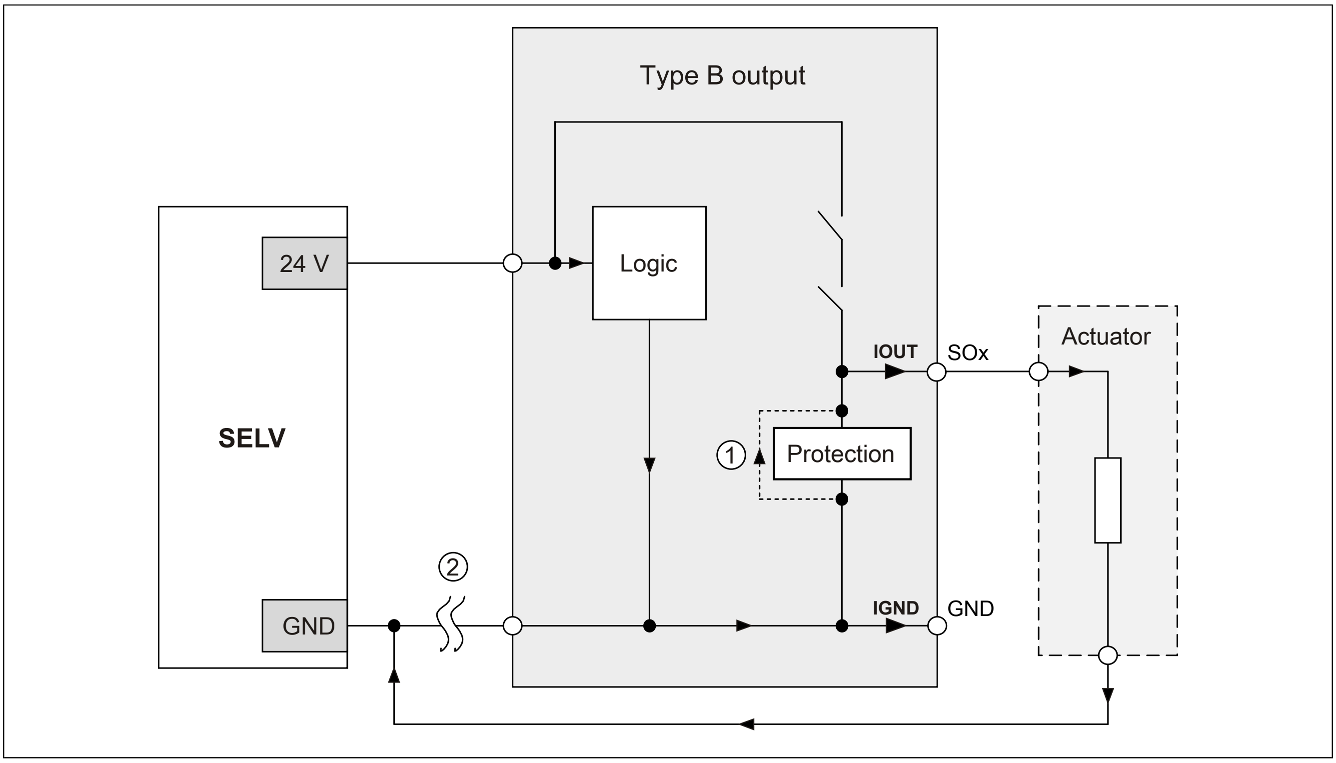


Figure 172: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

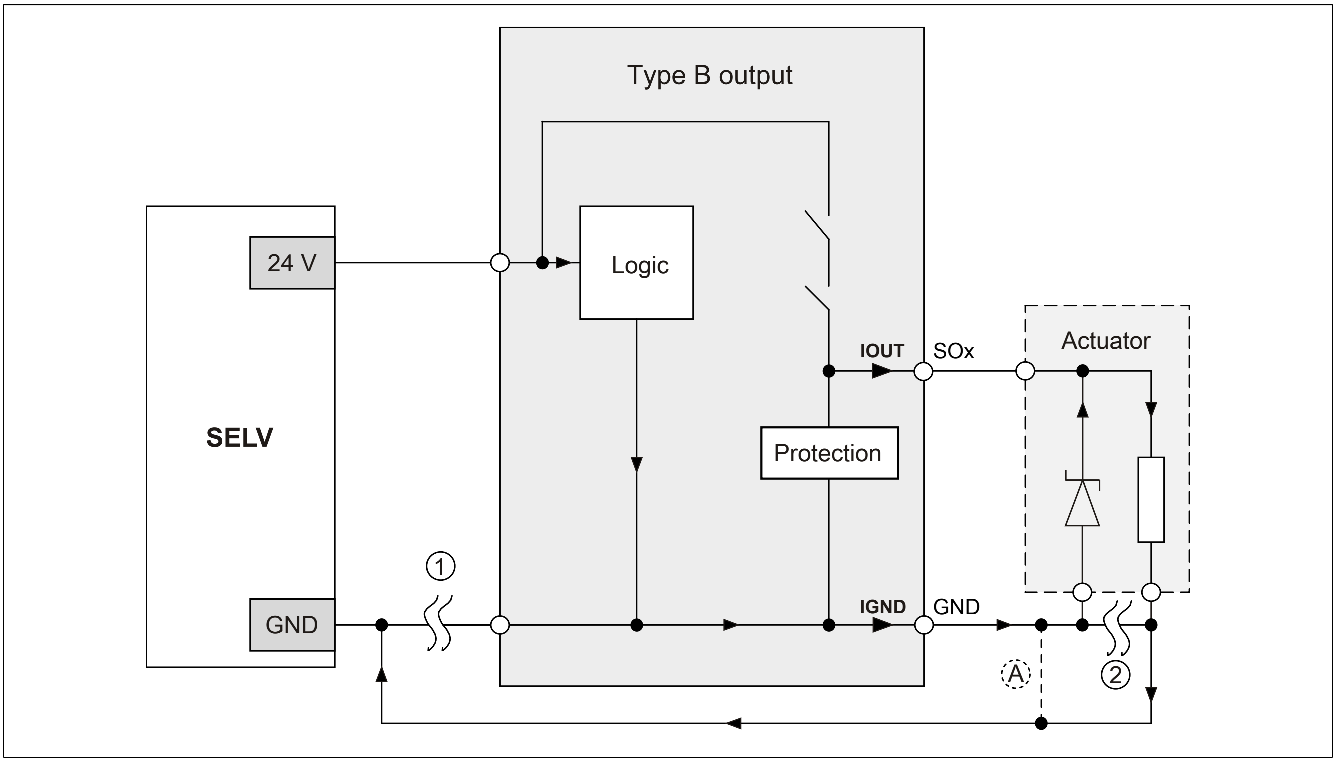


Figure 173: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

2.6.11.3.7.2 Connecting single-channel sensors with contacts

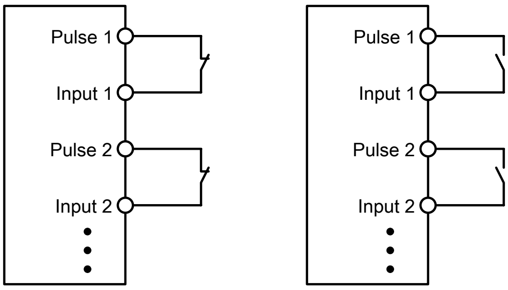


Figure 174: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.11.3.7.3 Connecting two-channel sensors with contacts

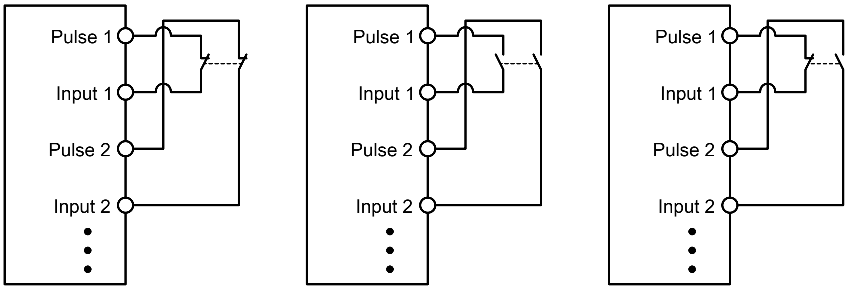


Figure 175: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.11.3.7.4 Connecting multi-channel sensors with contacts

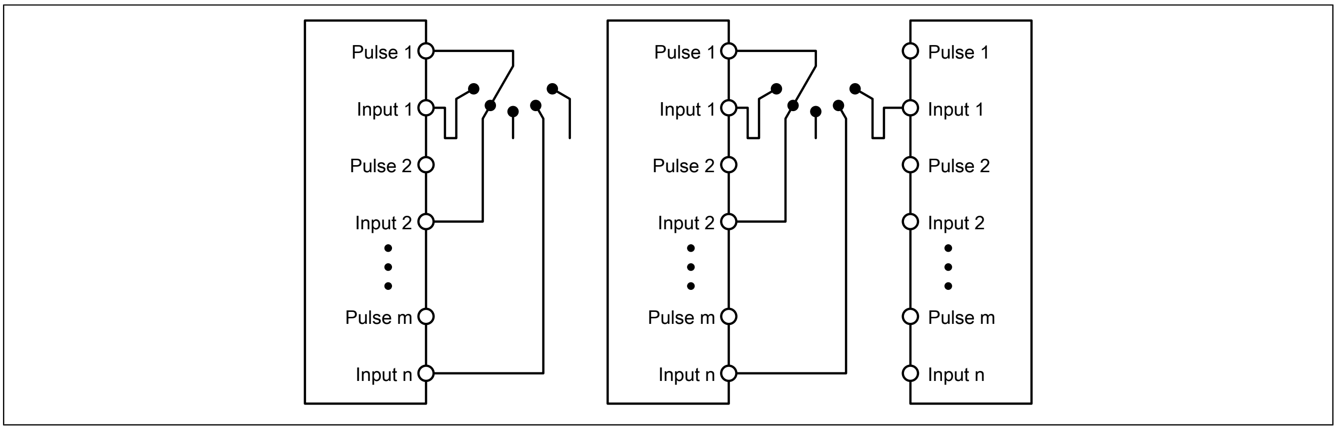


Figure 176: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.11.3.7.5 Connecting electronic sensors

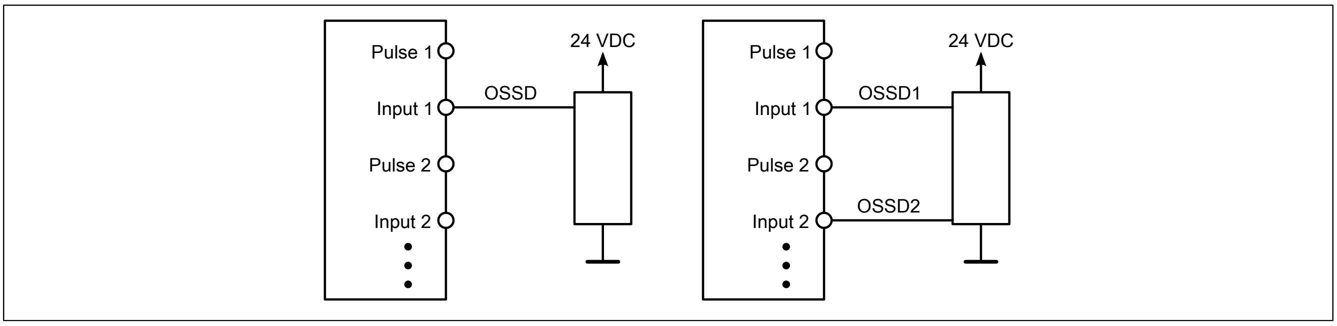


Figure 177: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.11.3.7.6 Using the same pulse signals

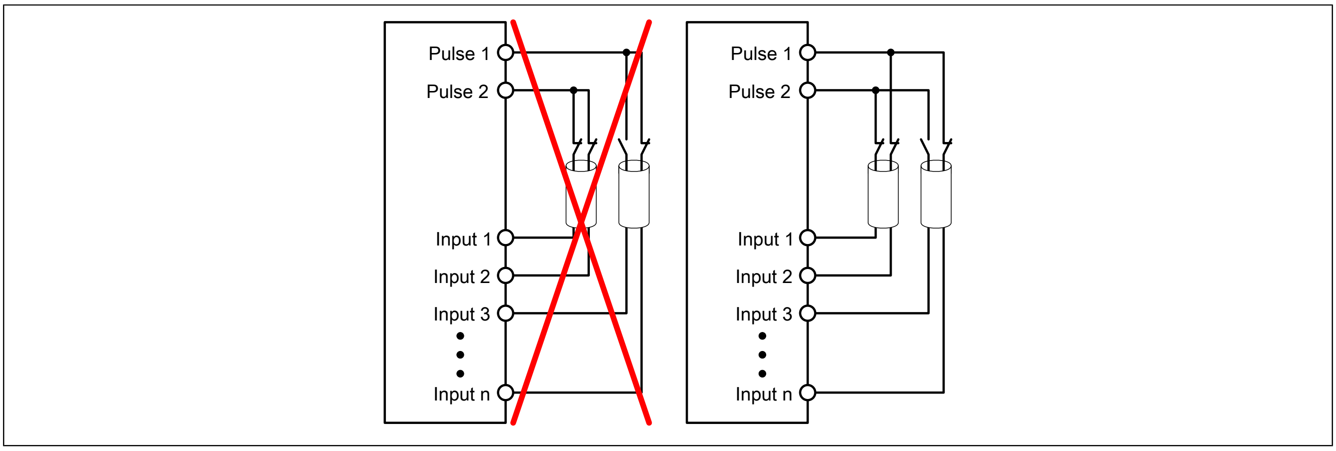


Figure 178: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.11.3.7.7 Connecting safety-oriented actuators for Type B outputs

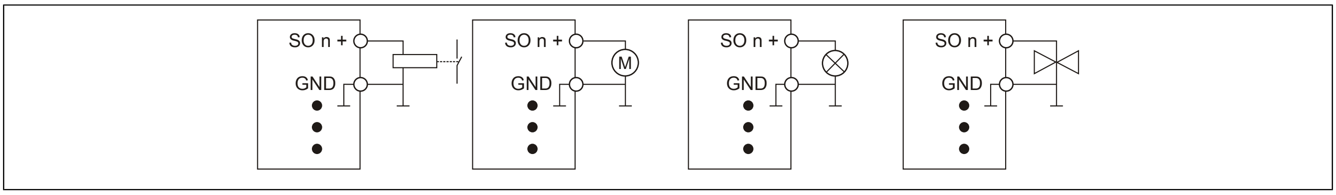


Figure 179: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.11.3.8 Error detection

2.6.11.3.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.11.3.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

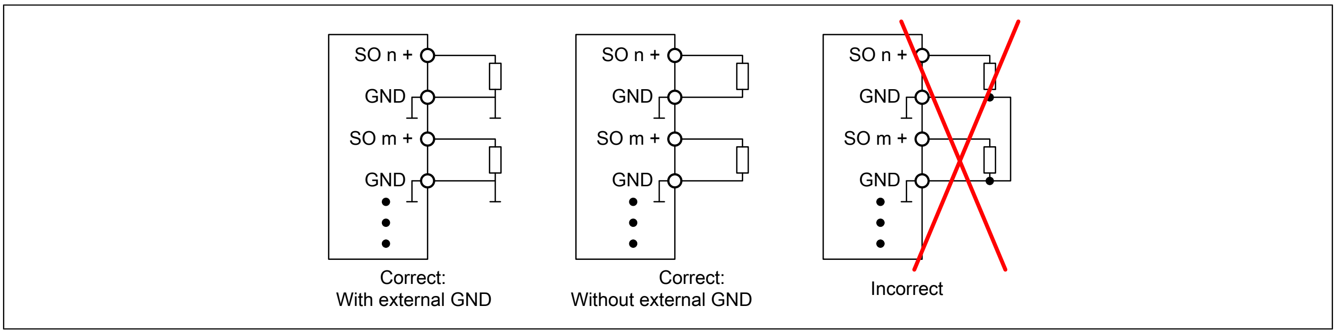


Figure 180: Invalid wiring

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 210: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 211: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 212: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected [2)]</td><td>Not detected [2)]</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
</table>

Table 213: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx X20SLXxxx X20SRTxxx X20SOx1x0	**Not detected**	Detected	**Not detected**	Detected

Table 213: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.11.3.9 Input circuit diagram

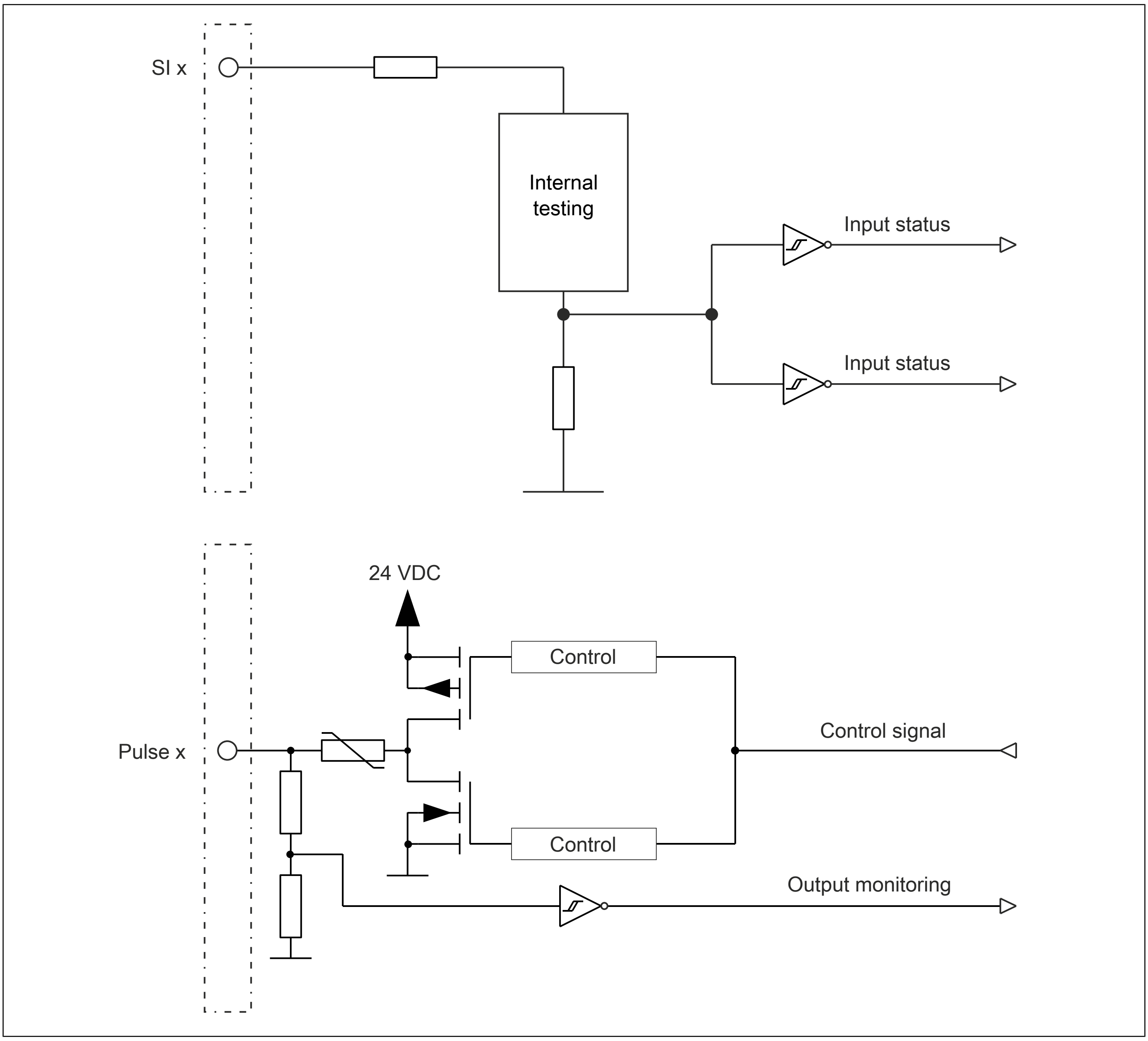


Figure 181: Input circuit diagram

2.6.11.3.10 Type B output circuit diagram

Type B digital output channels are designed for positive and positive switching inside the module.

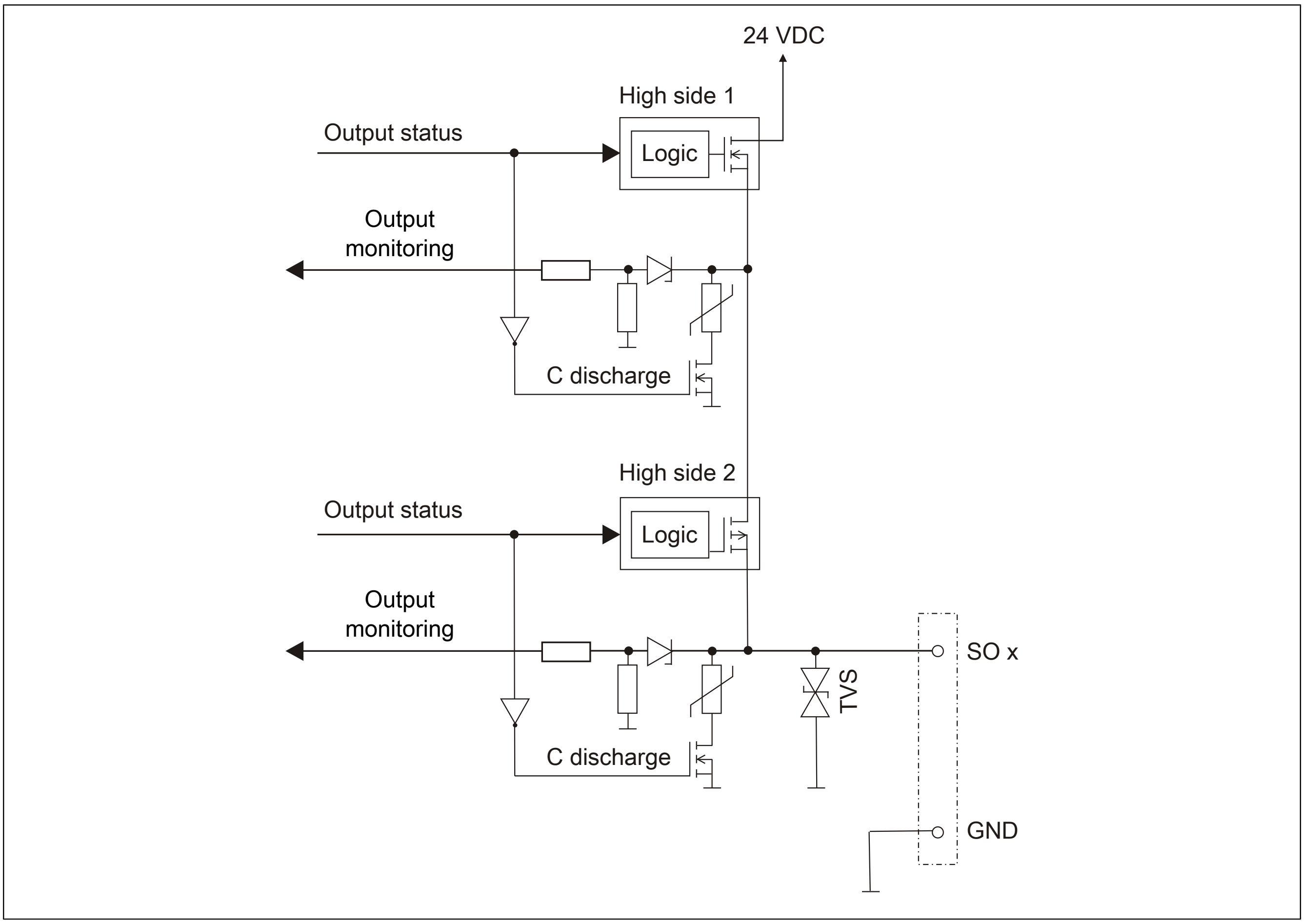


Figure 182: Type B output circuit diagram

2.6.11.3.11 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.11.3.12 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 µs

Maximum I/O update time for input channels
2150 µs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels
1800 µs

2.6.11.3.13 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

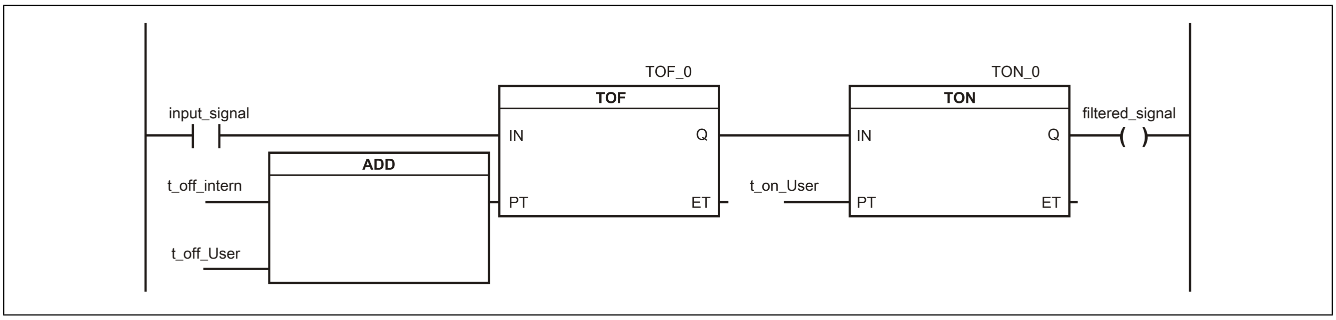


Figure 183: SI input filter - Diagram 1

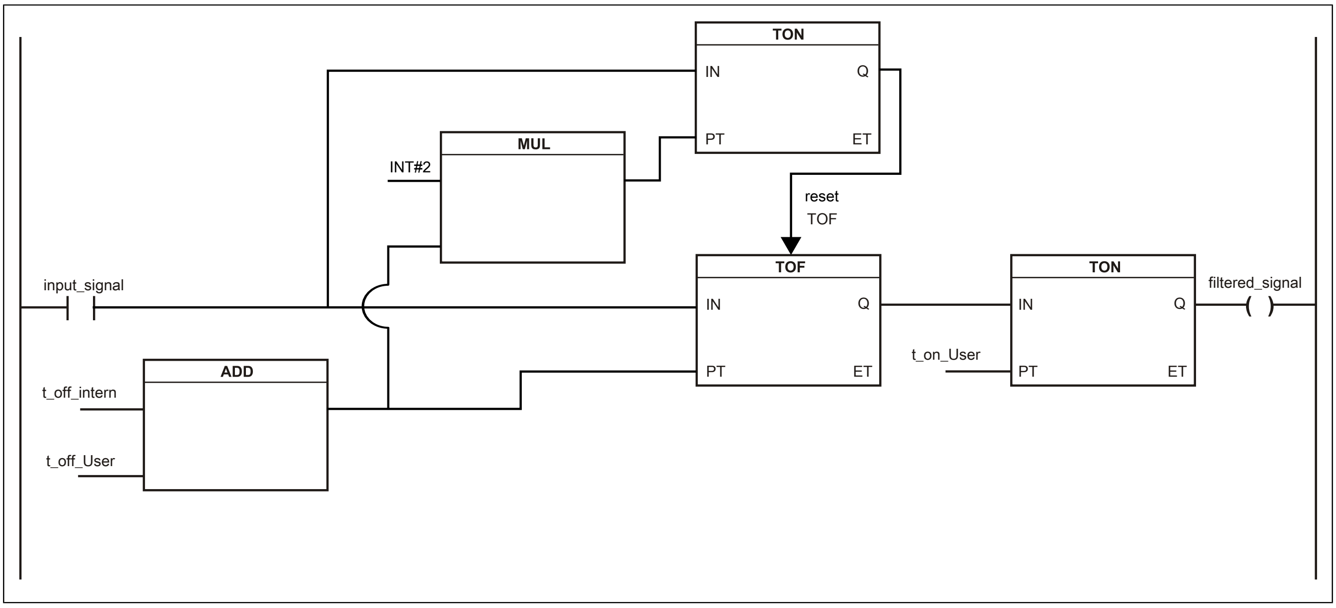


Figure 184: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.11.3.14 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.11.3.15 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.3.16 Register description

2.6.11.3.16.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 214: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>Max switching frequency channel x
(up to firmware version < 300)</td><td>Maximum switching frequency of the output channel.
• Permissible values: 1 Hz, 10 Hz, 100 Hz, 1000 Hz

This value specifies the max. switching frequency of the actuator connected to the output. It is especially important to adjust this parameter to the actual conditions for inductive or capacitive loads because the internal delay for checking the voltage to see if it is 0 V after a cutoff signal occurs is calculated using this parameter. Therefore, if this value is too high (e.g. 1000 Hz) and the voltage does not go to 0 within the corresponding time (in this example 500 μs) after a cutoff signal because of the connected actuator, then a channel error occurs.

If the output is controlled by the application using a higher switching frequency than configured, a channel-specific error may erroneously be detected on the module, which causes the channel to be cut off.</td><td>1</td><td>Hz</td></tr>
</table>

Table 215: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 216: I/O configuration parameters: Output signal path

2.6.11.3.16.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Disable_OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 217: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2010 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2010 or PL d per EN ISO 13849-1:2015, a daily check of the safety function by the user is necessary.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 218: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse_Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="6">Possible "Pulse_Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>5</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>6</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Channel 5</td><td>Default</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".</td></tr>
<tr><td rowspan="2">Pulse_Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal "low phases".
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter_On_us</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>0</td><td>µs</td></tr>
</table>

Table 219: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto_Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr><tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr></table></td></tr>
</table>

Table 220: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.3.16.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
</table>

Table 221: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 222: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Disable OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table></td></tr>
</table>

Table 223: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="6">Possible "Pulse Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>5</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>6</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Channel 5</td><td>Default</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal "low phases".
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>50000</td><td>µs</td></tr>
</table>

Table 224: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 225: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.11.3.16.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 5 = Channel 1 to 6</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 13 = Channel 1 to 6</td><td>Bit no. 0 to 5 = Channel 1 to 6</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
</table>

Table 226: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeEquivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeEquivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateOutputxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table><tr><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr><tr><td>Reserved</td><td>Reserved</td><td>Channel 2</td><td>Channel 1</td></tr></table></td></tr>
</table>

Table 226: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

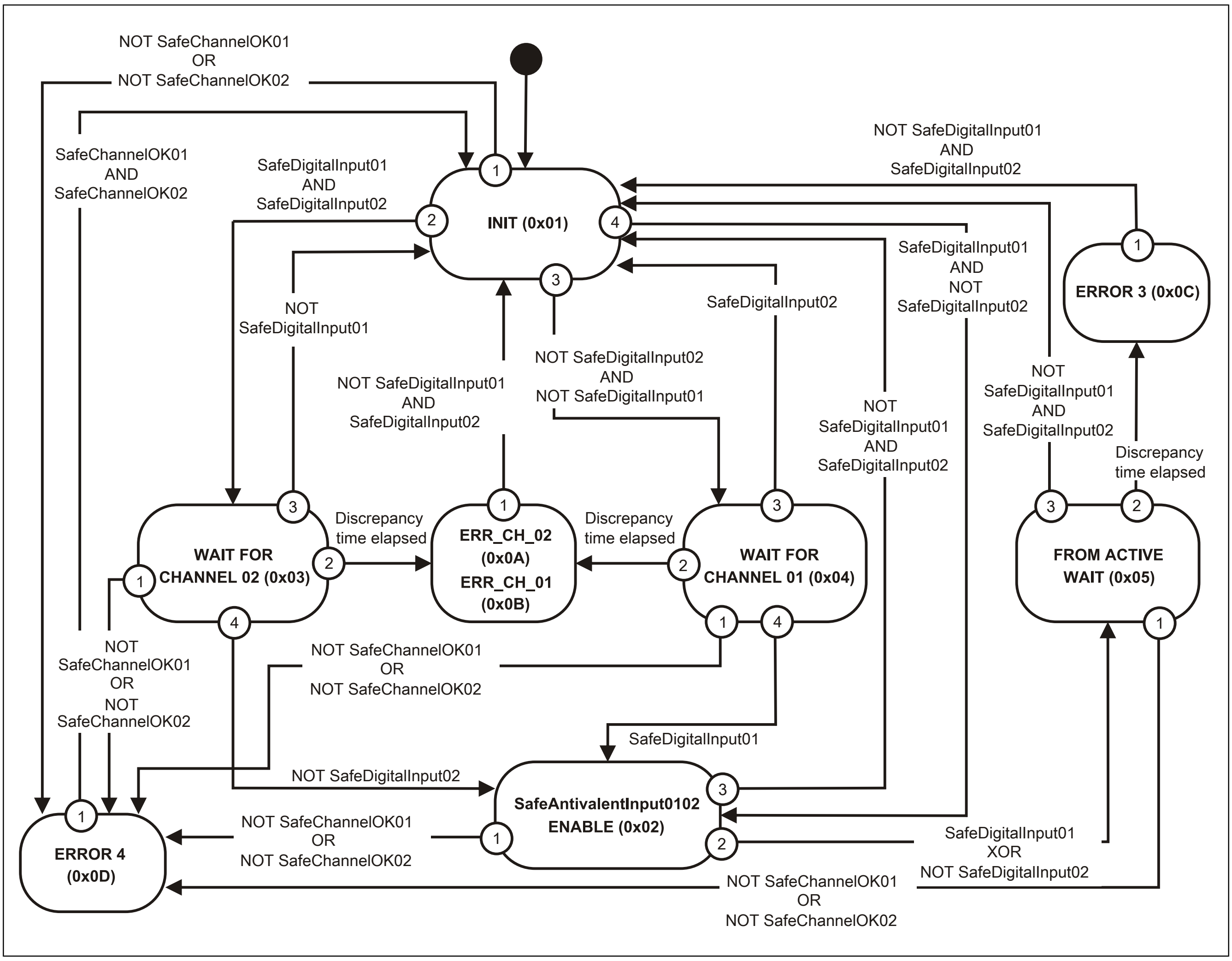


Figure 185: "Antivalent" function block - State diagram

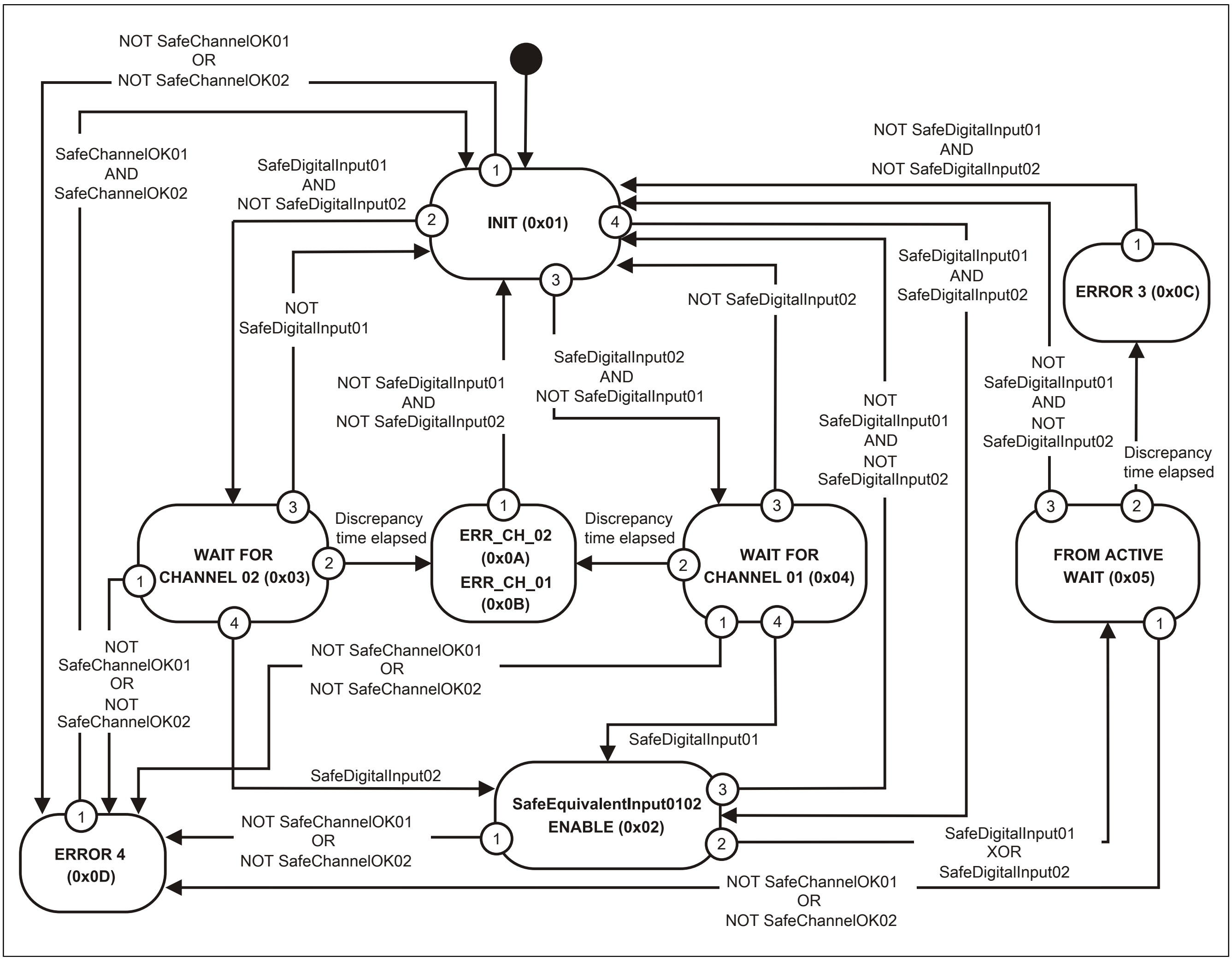


Figure 186: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

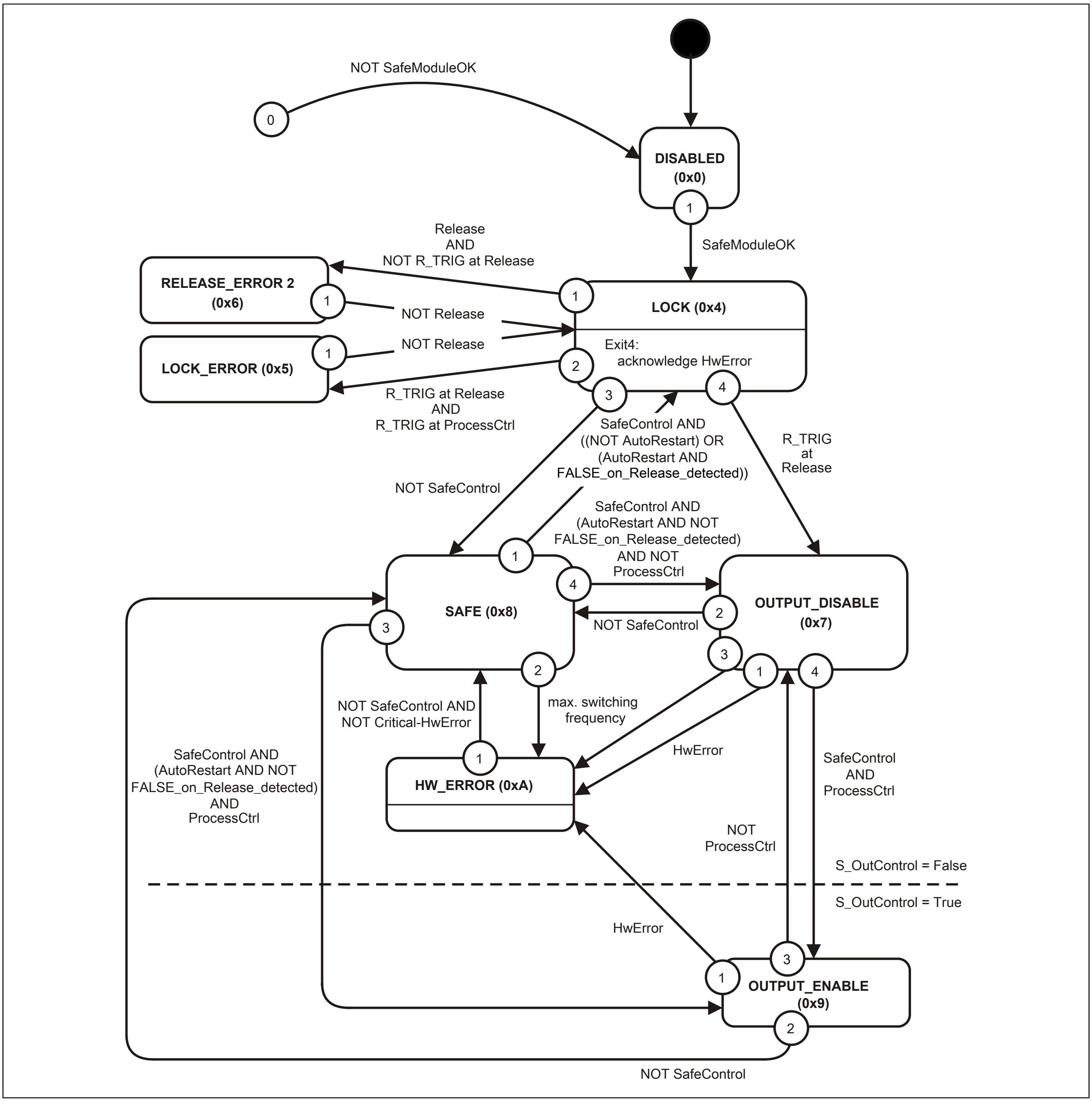


Figure 187: Restart interlock - State diagram

2.6.12 Relay modules

2.6.12.1 Overview

Model number	Short description	Page
X20SC2432	X20 safe digital mixed module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555
X20cSC2432	X20 safe digital mixed module, coated, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	555
X20SO2530	X20 safe digital output module, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20cSO2530	X20 safe digital output module, coated, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591
X20SO6530	X20 safe digital output module, 6 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A	591

2.6.12.2 X20(c)SC2432

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 227: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 228: Organization of general notices

2.6.12.2.1 General information

The modules are equipped with 2 safe digital inputs and 2 safe relay outputs. They are designed for a nominal voltage of 24 VDC.

The modules can be used to read in digital signals and for floating control of actuators in safety applications up to PL e or SIL 3.

The modules are equipped with filters that are individually configurable for switch-on and switch-off behavior. The modules also provide pulse signals for diagnosing the sensor line.

Safety relays are installed in the module. The positively driven feedback contacts are evaluated internally by the module. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- 2 safe digital inputs, sink circuit
- 2 pulse outputs
- Software input filter configurable for each channel
- 2 safe relay outputs
- Output type "Relay"
- Relay module for 48 VAC / 24 VDC
- Switching current 6 A
- Normally open contact
- Single-channel isolated outputs

2.6.12.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe relay outputs

The module is equipped with safe relay outputs for floating control of actuators in safety-related applications up to PL e or SIL 3.

2 safety relays are installed in the module. The positively driven feedback contacts are evaluated internally by the module. The B10d values are specified in the technical data for the safety-related perspective of the relay contacts. These values apply up to the specified maximum contact service life.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

2.6.12.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.12.2.2 Overview

Module	X20SC2432
Safe digital inputs	
Number of inputs	2
Nominal voltage	24 VDC
Input filter Hardware Software	 ≤150 µs Default 0 ms, configurable between 0 and 500 ms
Input circuit	Sink
Pulse outputs	
Design	Push-Pull
Switching voltage	I/O power supply minus residual voltage
Relay outputs	
Number of outputs	2
Switching voltage range	5 to 24 VDC, 5 to 48 VAC
Switching current range	5 mA to 6 A
Overload protection and short circuit protection	External 6 A gL/gG fuse (blow-out fuse), LS automat C characteristic 1.6 A

Table 229: Digital mixed modules

2.6.12.2.3 Order data

Model number	Short description	Figure
	Digital mixed modules	
X20SC2432	X20 safe digital mixed module, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	
X20cSC2432	X20 safe digital mixed module, coated, 2 safe digital inputs, configurable input filter, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	
	Terminal blocks	
X20TB52	X20 terminal block, 12-pin, safety-keyed	

Table 230: X20SC2432, X20cSC2432 - Order data

2.6.12.2.4 Technical data

Model number	X20SC2432	X20cSC2432
Short description		
I/O module	2 safe digital inputs, 2 pulse outputs, 24 VDC, 2 relays with 1 normally open contact each, 48 VAC / 6 A, 24 VDC / 6 A	
General information		
B&R ID code	0xA7A4	0xDD5D
System requirements		
Automation Studio	3.0.80 or later	4.0.16 or later
Automation Runtime	3.00 or later	V3.08 or later
SafeDESIGNER	2.70 or later	3.1.0 or later
Safety Release	1.2 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using status LED and software	
Outputs	Yes, using status LED and software	
Inputs	Yes, using status LED and software	
Blackout mode		
Scope	Module	
Function	Module function	
Standalone mode	No	
Max. I/O cycle time	1 ms	
Power consumption		
Bus	0.26 W	
Internal I/O	1.15 W	
Electrical isolation		
Channel - Bus	Yes	
Channel - Channel	Yes	
Certifications		
CE	Yes	
KC	Yes	-
EAC	Yes	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013	
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3	
Functional safety	EN 50156-1:2004	
Relays		
EN 50155	Yes	
EN 50205	Yes	
Safety characteristics		
EN ISO 13849-1:2015		
MTTFD	2500 years	
Mission time	Max. 20 years	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
PFH / PFH_d		
Module	$<1*10^{-10}$	
openSAFETY wired	Negligible	
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour	
PFD	$<2*10^{-5}$	
Proof test interval (PT)	20 years	

Table 231: X20SC2432, X20cSC2432 - Technical data

Model number	X20SC2432	X20cSC2432
Safe digital inputs		
EN ISO 13849-1:2015		
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1]	
PL	PL e	
DC	>94%	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 3	
SFF	>90%	
Safe relay channels		
EN ISO 13849-1:2015		
Category	Cat. 1 if the relay channel is used individually, Cat. 4 if both relay channels are connected in series [1]	
PL	PL c if the relay channel is used individually, PL e if both relay channels are connected in series [1]	
B10d		
DC1, 24 VDC	6 A / 780,000	
AC1, 48 VAC	6 A / 780,000	
AC15, 48 VAC	3 A / 1,960,000	
DC13, 24 VDC	5 A / 780,000	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 1 if the relay channel is used individually, SIL 3 if both relay channels are connected in series [1]	
I/O power supply		
Nominal voltage	24 VDC	
Voltage range	24 VDC -15% / +20%	
Integrated protection	Reverse polarity protection	
Safe digital inputs		
Nominal voltage	24 VDC	
Input characteristics per EN 61131-2	Type 1	
Input filter		
Hardware	≤150 µs	
Software	Configurable between 0 and 500 ms	
Input circuit	Sink	
Input voltage	24 VDC -15% / +20%	
Input current at 24 VDC	Max. 4.59 mA	
Input resistance	Min. 5.23 kΩ	
Error detection time	100 ms	
Isolation voltage between channel and bus	500 V_{eff}	
Switching threshold		
Low	<5 VDC	
High	>15 VDC	
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line	
Relay outputs		
Variant	2 relays, each with 1 normally open contact, internal high-side and low-side control	
Diagnostic status	Contact position determined by positively driven contacts	
Max. switching frequency	10 Hz	
Switching delay		
0 → 1	<50 ms	
1 → 0	<50 ms	
Isolation voltage between channel and bus	Safe disconnection of 300 VAC per EN 50178	
Isolation voltage between channel and channel	48 VAC	
Contact resistance (without terminal block)	20 mΩ	
Contact service life	See "Contact service life".	
Short-circuit protection, overload protection	External 6 A gL/gG fuse (blow-out fuse), LS automat C characteristic 1.6 A	
Switching voltage range	5 to 24 VDC, 5 to 48 VAC	
Switching current range	5 mA to 6 A	
Coil voltage	24 VDC -15% / +20%	
Short-circuit proof	Yes, 1000 A (with specified short-circuit / overload protection)	
Max. inrush current	30 A for 20 ms	
Overvoltage category per EN 60664-1	II	
Max. switching capacity		
AC1	48 VAC / 6 A	
AC15	48 VAC / 3 A	
DC1	24 VDC / 6 A	
DC13	24 VDC / 5 A / 0.1 Hz	
Pulse outputs		
Variant	Push-Pull	
Nominal output current	50 mA	

Table 231: X20SC2432, X20cSC2432 - Technical data

Model number	X20SC2432	X20cSC2432
Output protection	Shutdown of individual channels in the event of overload or short circuit [2]	
Peak short-circuit current	25 A for 5 ms	
Short-circuit current	1.4 A_{eff}	
Leakage current when switched off	0.1 mA	
Residual voltage	0.3 VDC	
Switching voltage	I/O power supply minus residual voltage	
Total nominal current	100 mA	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level	0 to 2000 m, no limitation	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation [3]	0 to 60°C	-25 to 60°C
Vertical mounting orientation	0 to 50°C	-25 to 50°C
Derating	See section "Derating".	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm	

Table 231: X20SC2432, X20cSC2432 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Compared to the specification in the X20 system user's manual, in which the angle of the horizontal mounting orientation is 70°, this applies only up to an angle of 85° on the X20(c)SC2432. Below this, the derating for face-up installation must be applied.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SC2432
Derating bonus	
At 24 VDC	+0°C
Dummy module to the left	+0°C
Dummy module to the right	+2.5°C
Dummy module to the left and right	+2.5°C
With double PFH / PFH_d	+0°C

Table 232: Derating bonus

The maximum nominal current per channel depends on the operating temperature and the mounting orientation. The resulting nominal current per channel can be found in the following table.

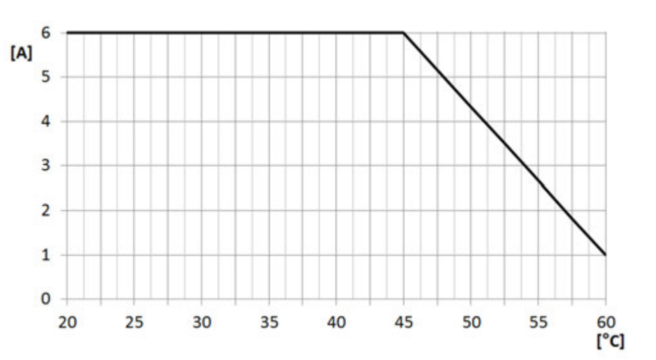
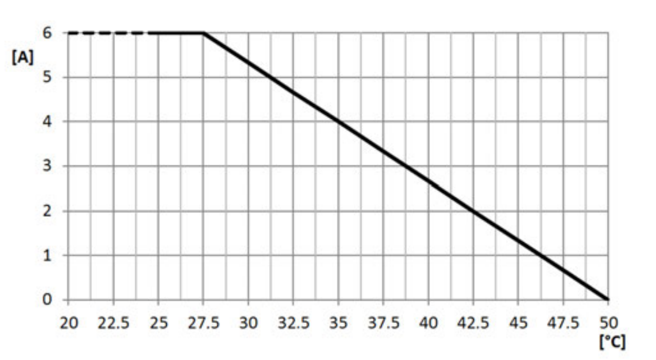
Horizontal (0 to 60°C, coated: -25 to 60°C)	Vertical (0 to 50°C, coated: -25 to 50°C)
	

Table 233: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

Contact service life of relay outputs

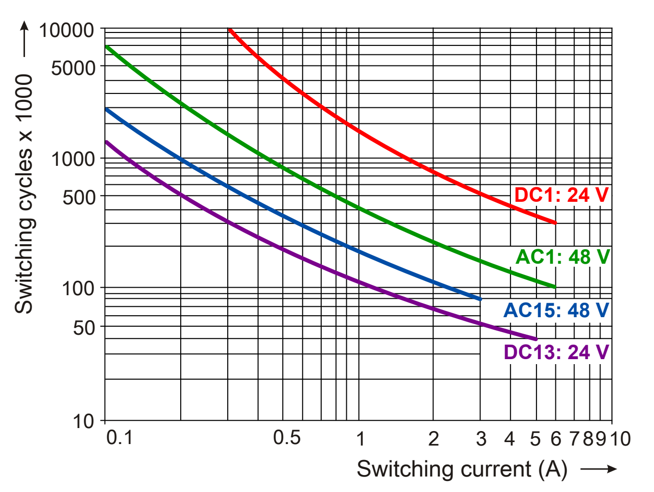
Contact service life of relay outputs


Table 234: Contact service life of relay outputs

2.6.12.2.5 LED status indicators

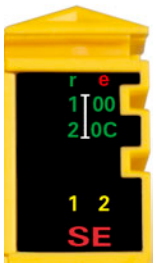
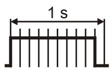
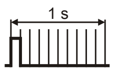
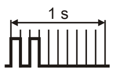
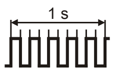
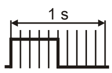
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 2	Input state of the corresponding digital input		
		Red	On	Warning/Error on an input channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Input set
	OO	Open - Open: Dual-channel evaluation on channels 1 and 2 using the "Equivalent" function block		
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Evaluation channel set
	OC	Open - Closed: Dual-channel evaluation on channels 1 and 2 using the "Antivalent" function block		
		Red	On	Warning/Error on this evaluation channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Evaluation channel set
	1 to 2	Output status of the corresponding digital output		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 235: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.12.2.6 Pinout

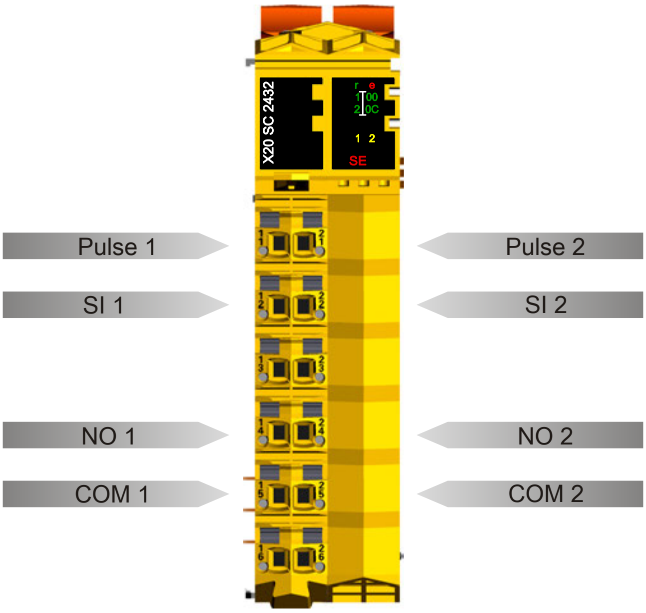


Figure 188: X20SC2432 - Pinout

2.6.12.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

> **Information:**
>
> **For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).**

2.6.12.2.7.1 Connecting single-channel sensors with contacts

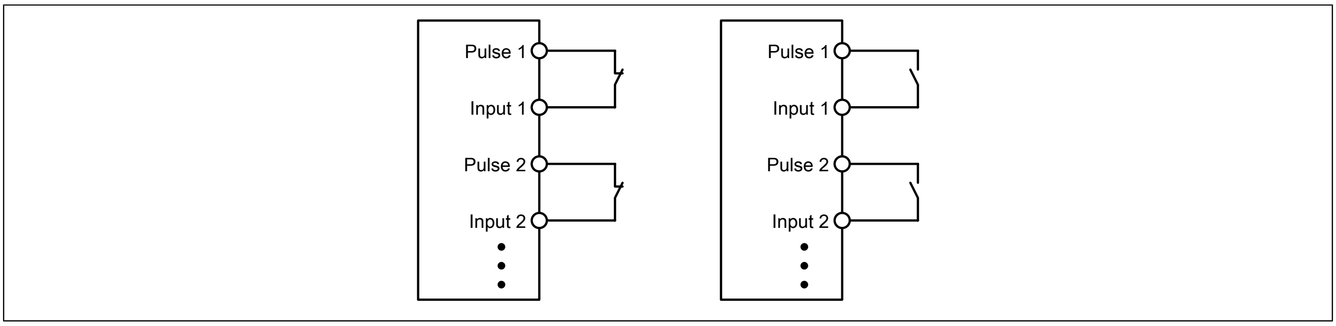


Figure 189: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.12.2.7.2 Connecting two-channel sensors with contacts

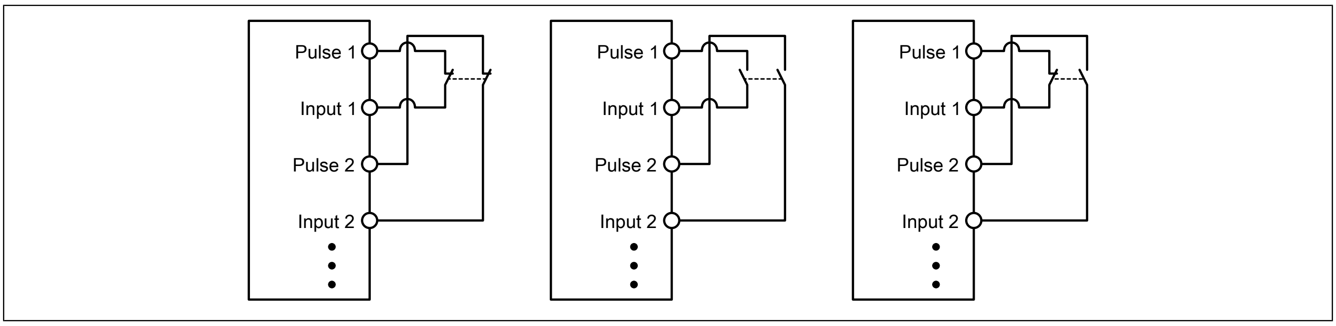


Figure 190: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.12.2.7.3 Connecting multi-channel sensors with contacts

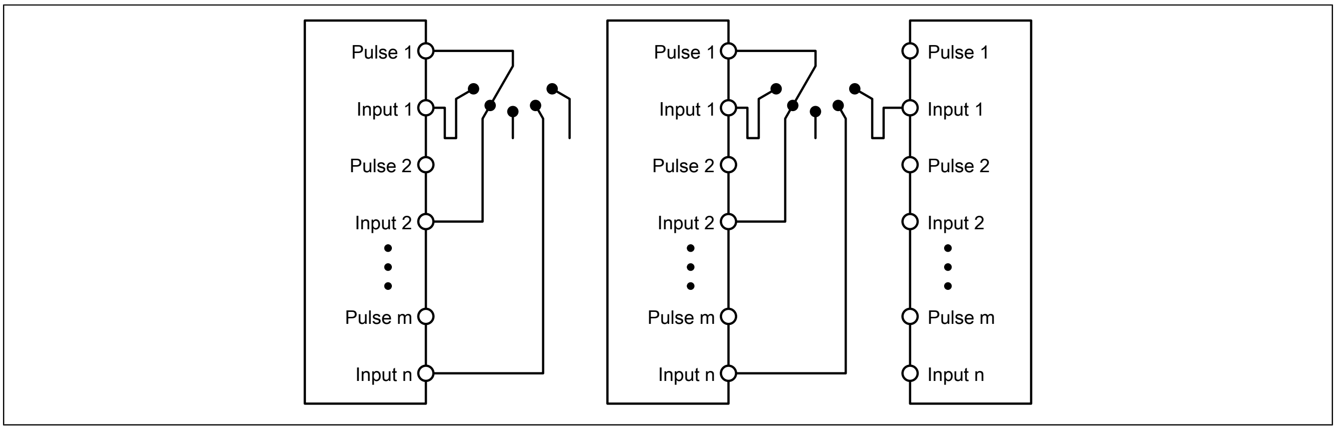


Figure 191: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.12.2.7.4 Connecting electronic sensors

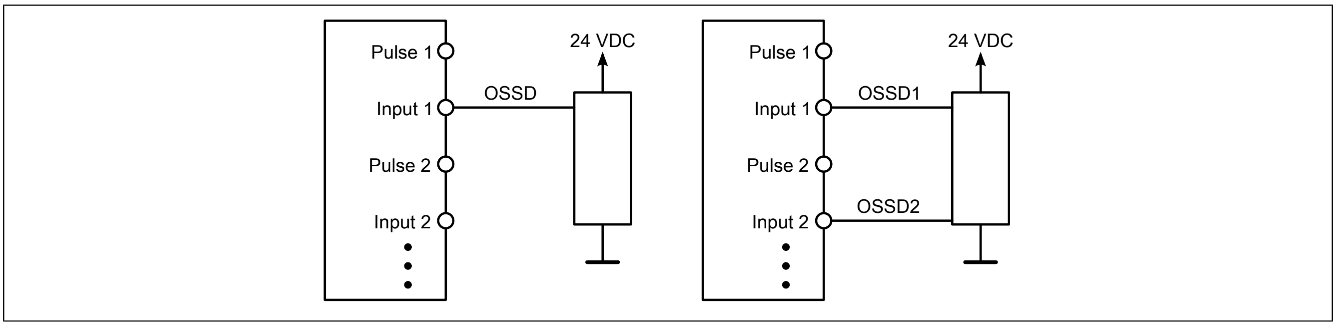


Figure 192: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.12.2.7.5 Using the same pulse signals

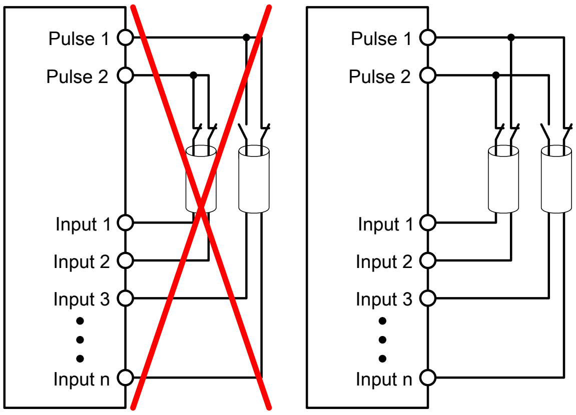


Figure 193: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.12.2.7.6 Connecting safety-oriented actuators for relay outputs

The connection example shown here only represents a selection of the possible wiring methods. However, the following must always be taken into consideration:

- Two relay channels must be connected in series for applications that correspond to EN ISO 13849-1:2015 above category 1.
- Relay contacts must be protected with a fuse (see technical data for the module).

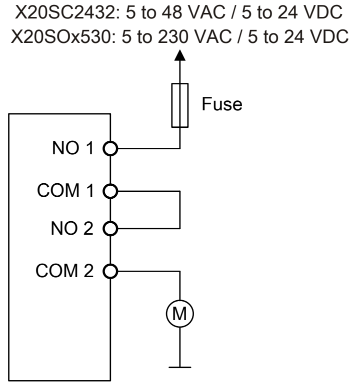


Figure 194: Connecting safety-oriented actuators for relay outputs

Danger!

Make sure that a proper protective circuit is used for the relay contacts (see technical data for the module). Also consider that operation outside of the specification is not permitted.

Operating outside of the specification or not using a protective circuit can cause the relay contacts to melt simultaneously, resulting in a loss of safety functionality.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Information:

Using signal "SafeDigitalOutputxxyy" and "SafeDigitalOutputxx" at the same time is not permitted and prevented by the system.

Using signal "SafeDigitalOutputxxyy" causes a switch-on sequence to be activated that switches on relay 2 with a 20 ms delay. This behavior is necessary to prevent simultaneous melting of the two relay contacts in certain operating states. Controlling two independent EN ISO 13849-1:2015 Category 1 actuators using signal "SafeDigitalOutputxxyy" must therefore be avoided since this causes delayed activation of the actuator on channel 2.

2.6.12.2.8 Error detection

2.6.12.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

2.6.12.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 236: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 237: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 238: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Connecting safety-oriented actuators for relay outputs

Danger!

A relay channel does not have error detection for wiring problems. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures or by the connected device.

Danger!

The user is responsible for ensuring that each relay channel is cut off at least 1x per week so that the appropriate internal tests can be completed.

2.6.12.2.9 Input circuit diagram

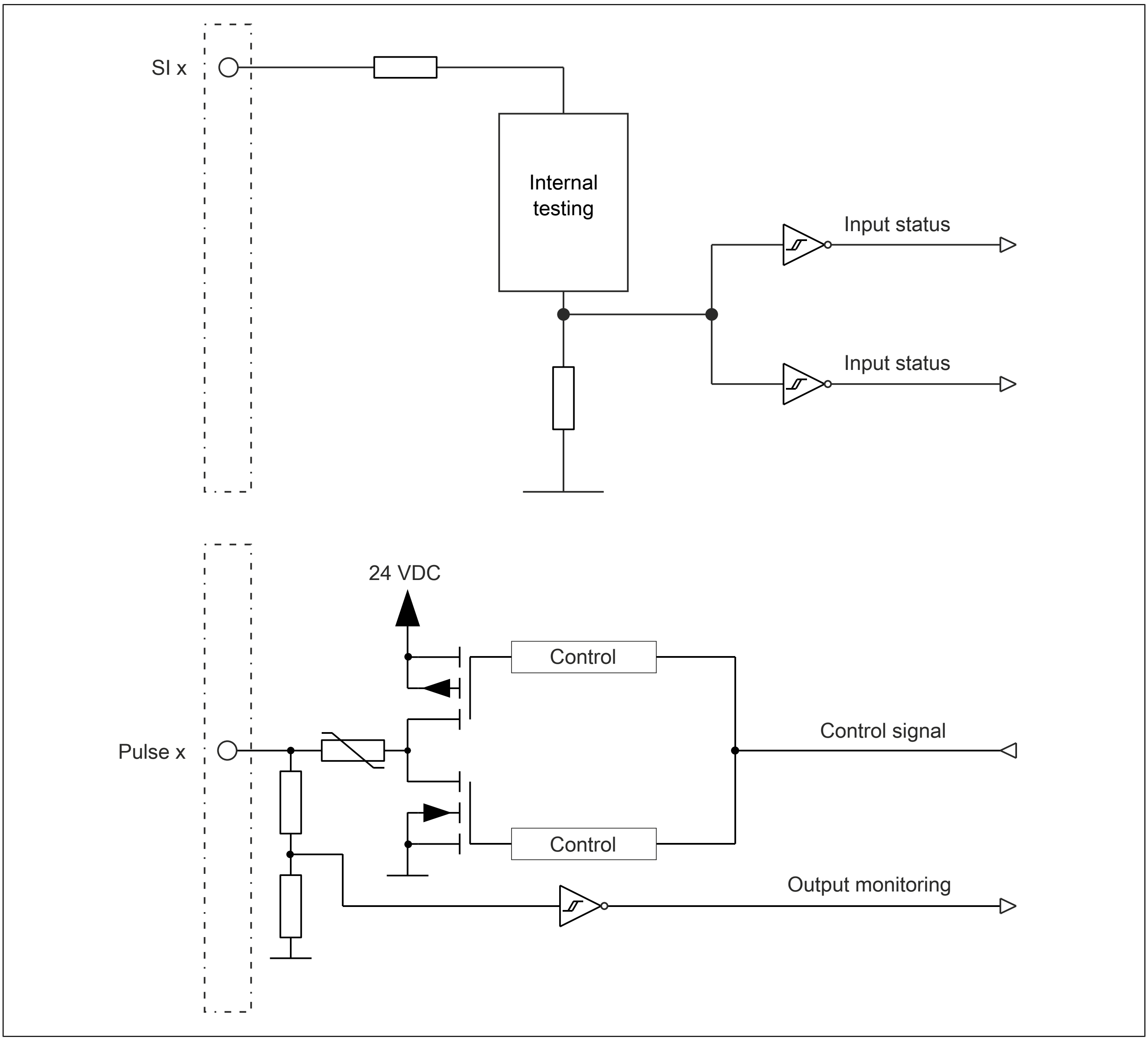


Figure 195: Input circuit diagram

2.6.12.2.10 Output circuit diagram

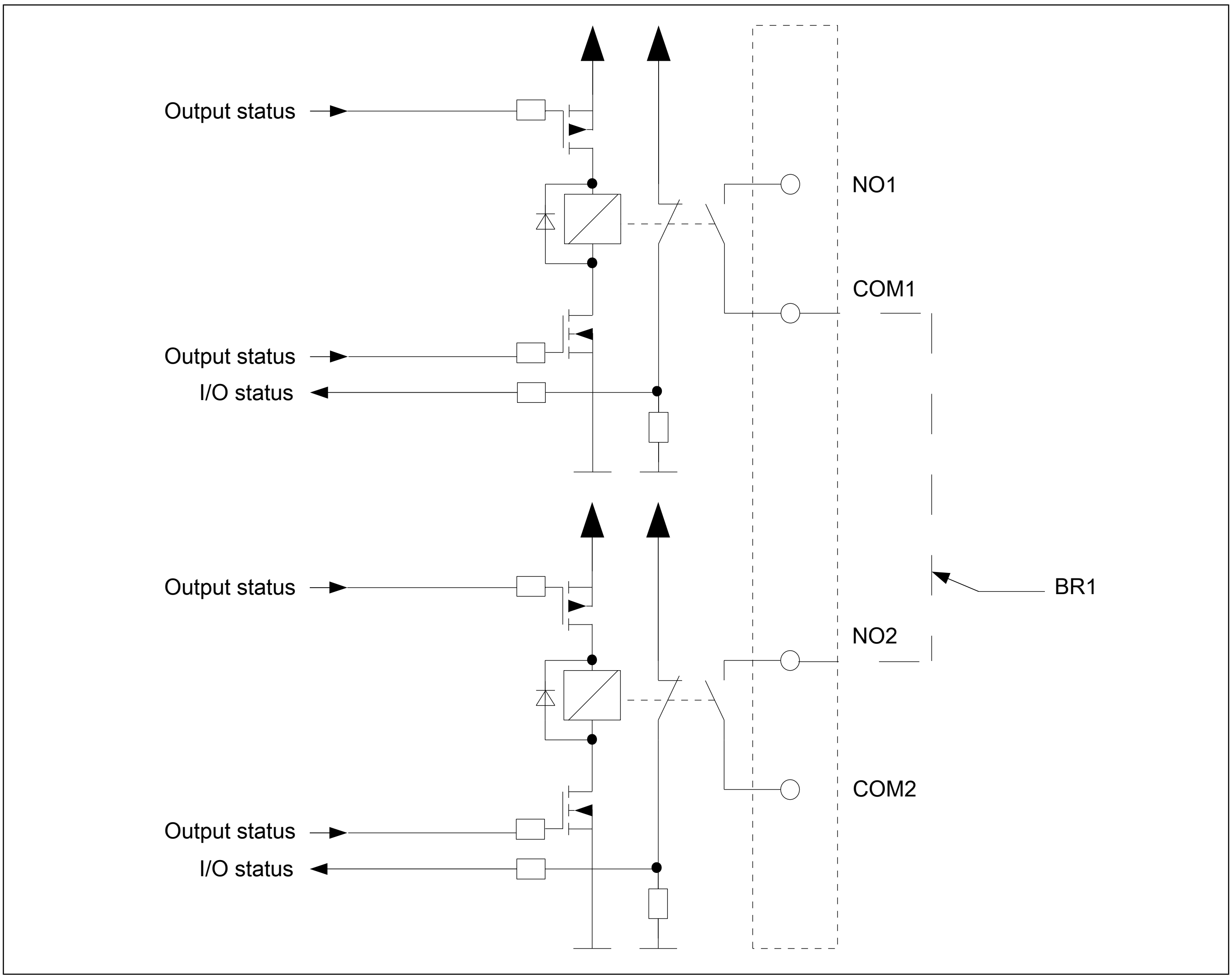


Figure 196: Output circuit diagram

2.6.12.2.11 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.12.2.12 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 µs

Maximum I/O update time for input channels
2150 µs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels
1000 µs + 50 ms

2.6.12.2.13 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

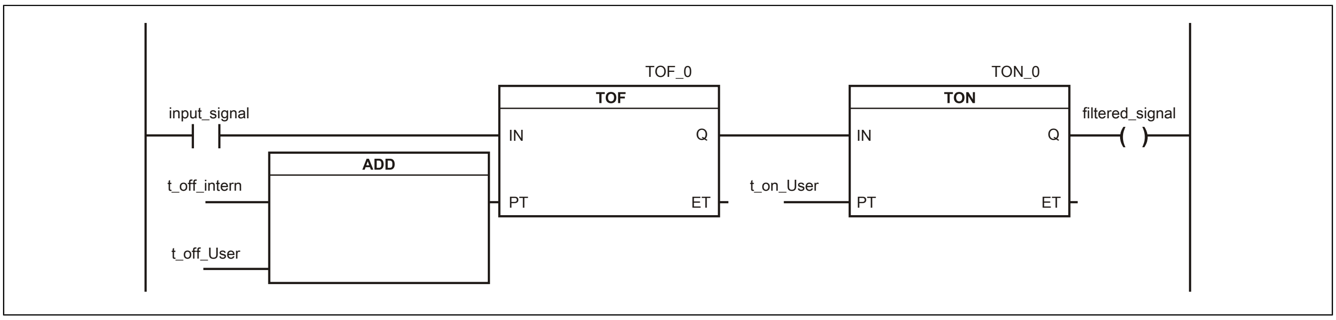


Figure 197: SI input filter - Diagram 1

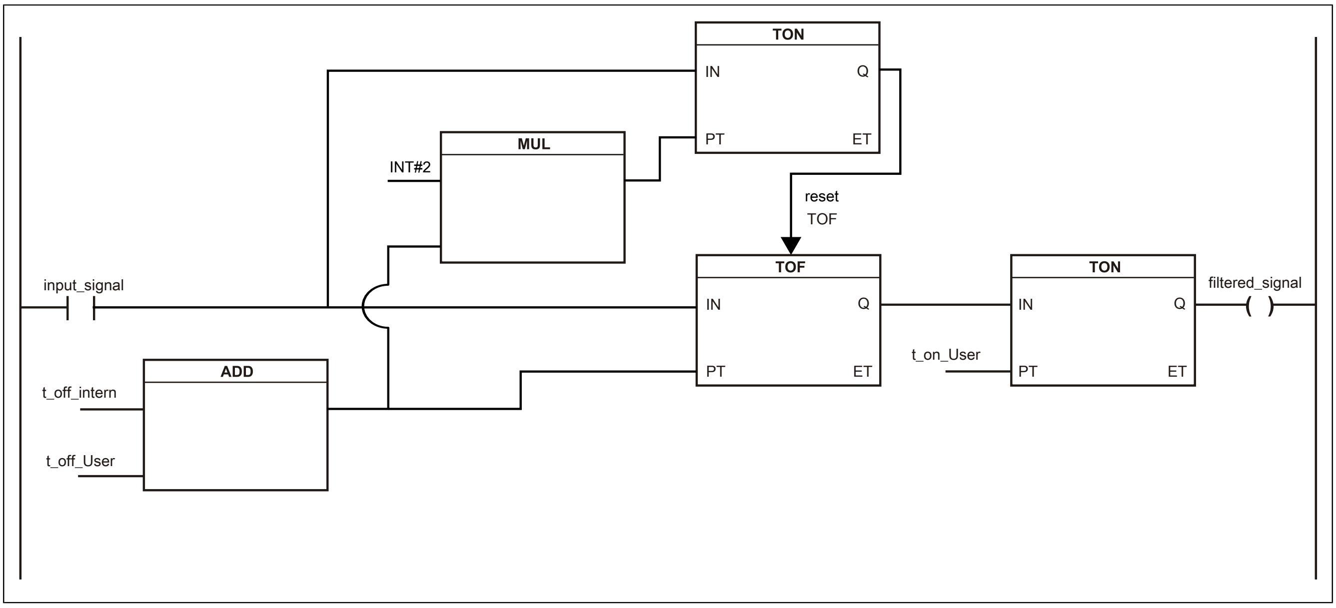


Figure 198: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.12.2.14 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.12.2.15 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.12.2.16 Register description

2.6.12.2.16.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 239: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 240: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx
DigitalOutputxxyy</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 241: I/O configuration parameters: Output signal path

2.6.12.2.16.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
</table>

Table 242: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.

<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td><td>No</td><td>-</td></tr>
<tr><td>Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.

<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td><td>Yes</td><td>-</td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 μs (corresponds to 0 to 25 ms)</td><td>5000</td><td>μs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>μs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 μs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>μs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 μs (corresponds to 0 to 25 ms)</td><td>0</td><td>μs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 μs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>μs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 243: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse_Source
(Release 1.4 and later)</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th></th><th colspan="2">Possible "Pulse_Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th></tr>
<tr><td>1</td><td>Default</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td></tr>
</table></td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".</td></tr>
<tr><td rowspan="2">Pulse_Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.
Release 1.4 and later:
The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table></td></tr>
<tr><td>Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter_On_us</td><td>Switch-on filter for the channel.
Signals can be "debounced" with the switch-on filter. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>150000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)
(up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
</table>

Table 244: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 245: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.12.2.16.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 246: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 247: SafeDESIGNER parameters: Safety Response Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="2">Possible "Pulse Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th></tr>
<tr><th>1</th><td>Default</td><td>-</td></tr>
<tr><th>2</th><td>Channel 1</td><td>Default</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.
Beginning with Release 1.4:
The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel
Signals can be "debounced" with the switch-on filter. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)
(up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>50000</td><td>µs</td></tr>
</table>

Table 248: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr>
<tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr>
</table></td></tr>
</table>

Table 249: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.12.2.16.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="3">Type of error</th></tr>
<tr><th>Inputs</th><th colspan="2">Pulse outputs</th></tr>
<tr><th>Input stuck at high</th><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 9 = Channel 1 to 2</td><td>Bit no. 4 to 5 = Channel 1 to 2</td><td>Bit no. 0 to 1 = Channel 1 to 2</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeEquivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeEquivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
</table>

Table 250: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>DigitalOutputxxyy</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal for combined channel SO xx/yy</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxxyy</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe combined channel SO xx/yy</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>ReleaseOutputxxyy</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of combined channel SO xx/yy</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table><tr><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr><tr><td>Reserved</td><td>Reserved</td><td>Channel 2</td><td>Channel 1</td></tr></table></td></tr>
</table>

Table 250: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Information:

Using signal "SafeDigitalOutputxxyy" and "SafeDigitalOutputxx" at the same time is not permitted and prevented by the system.

Using signal "SafeDigitalOutputxxyy" causes a switch-on sequence to be activated that switches on relay 2 with a 20 ms delay. This behavior is necessary to prevent simultaneous melting of the two relay contacts in certain operating states. Controlling two independent EN ISO 13849-1:2015 Category 1 actuators using signal "SafeDigitalOutputxxyy" must therefore be avoided since this causes delayed activation of the actuator on channel 2.

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

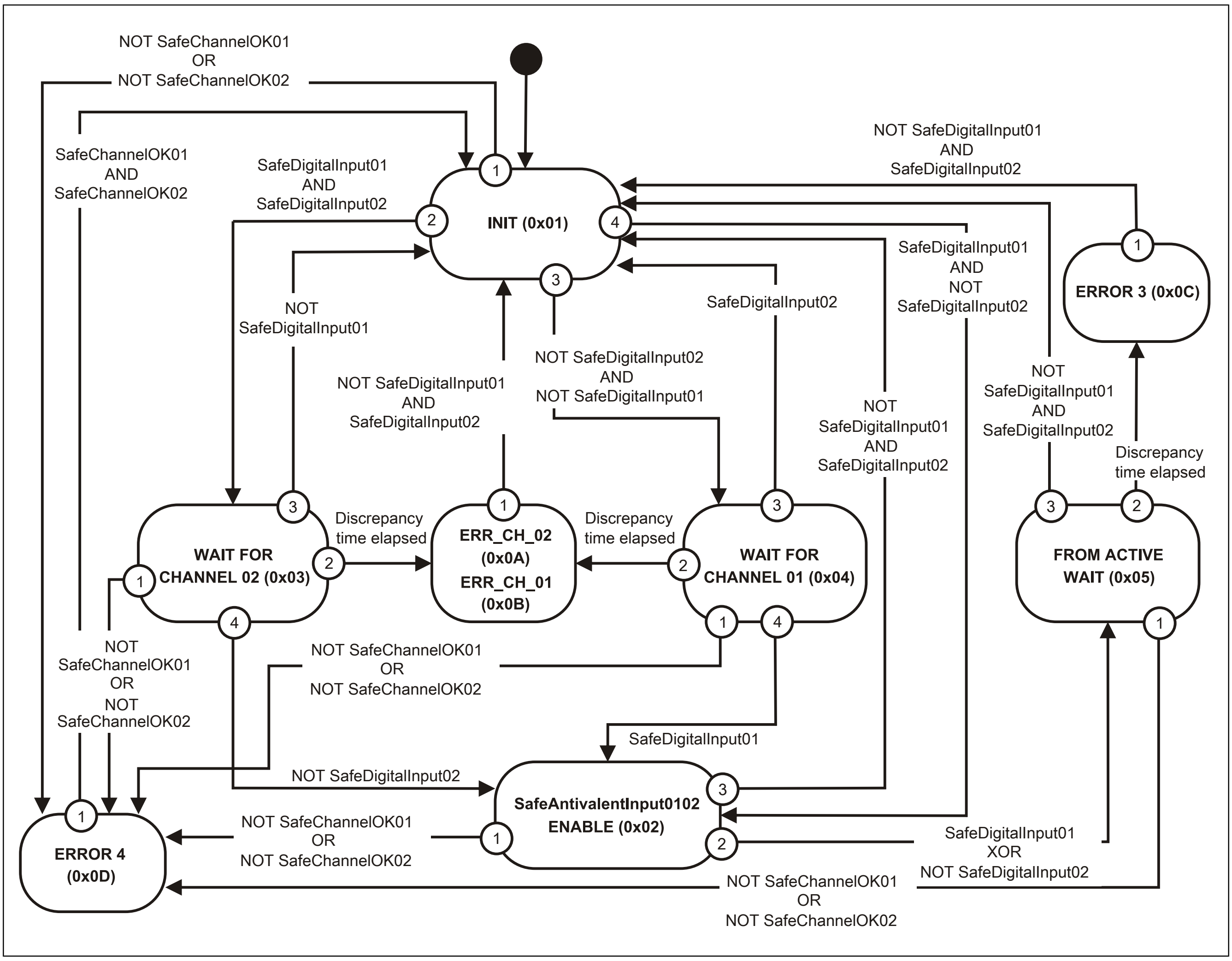


Figure 199: "Antivalent" function block - State diagram

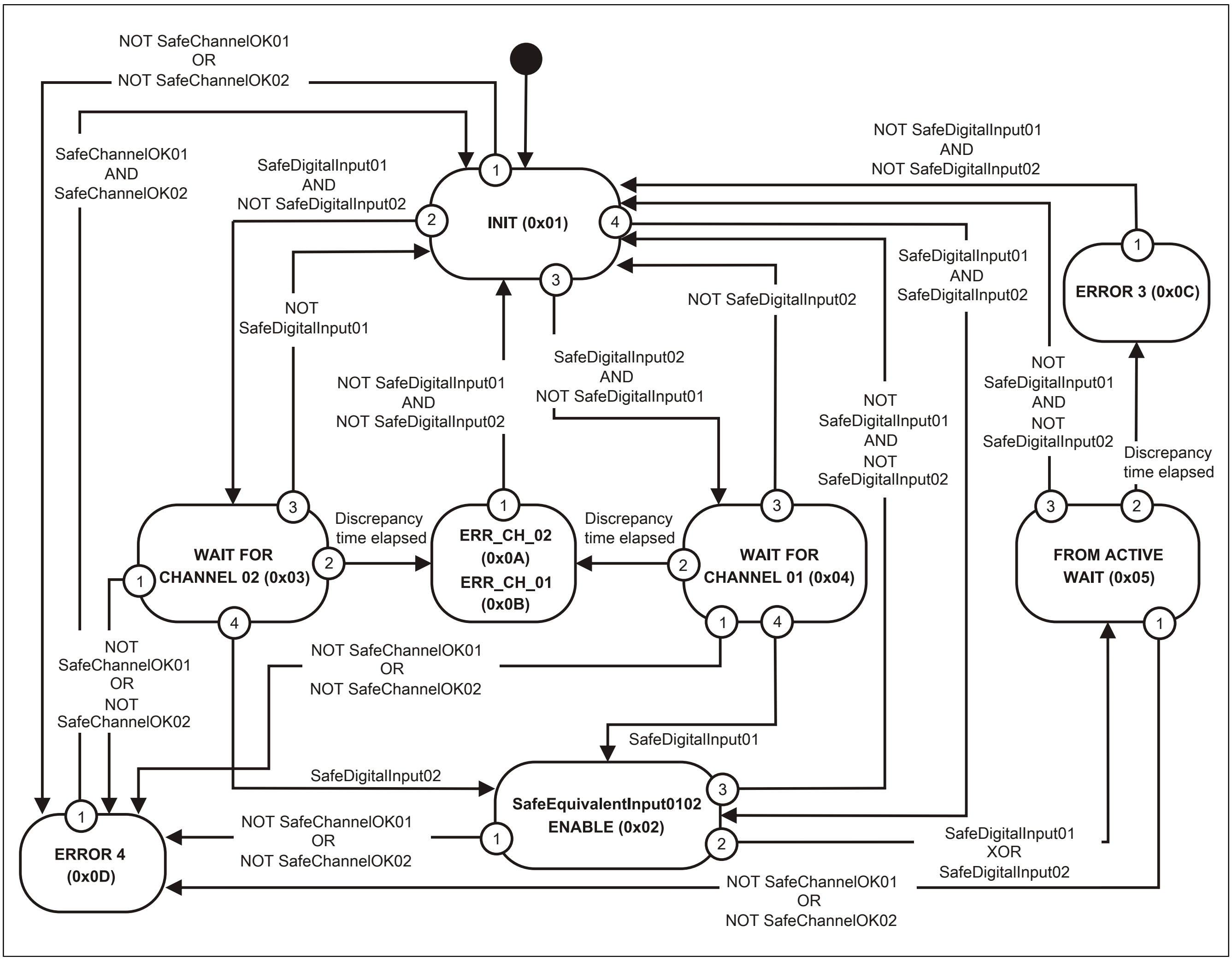


Figure 200: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

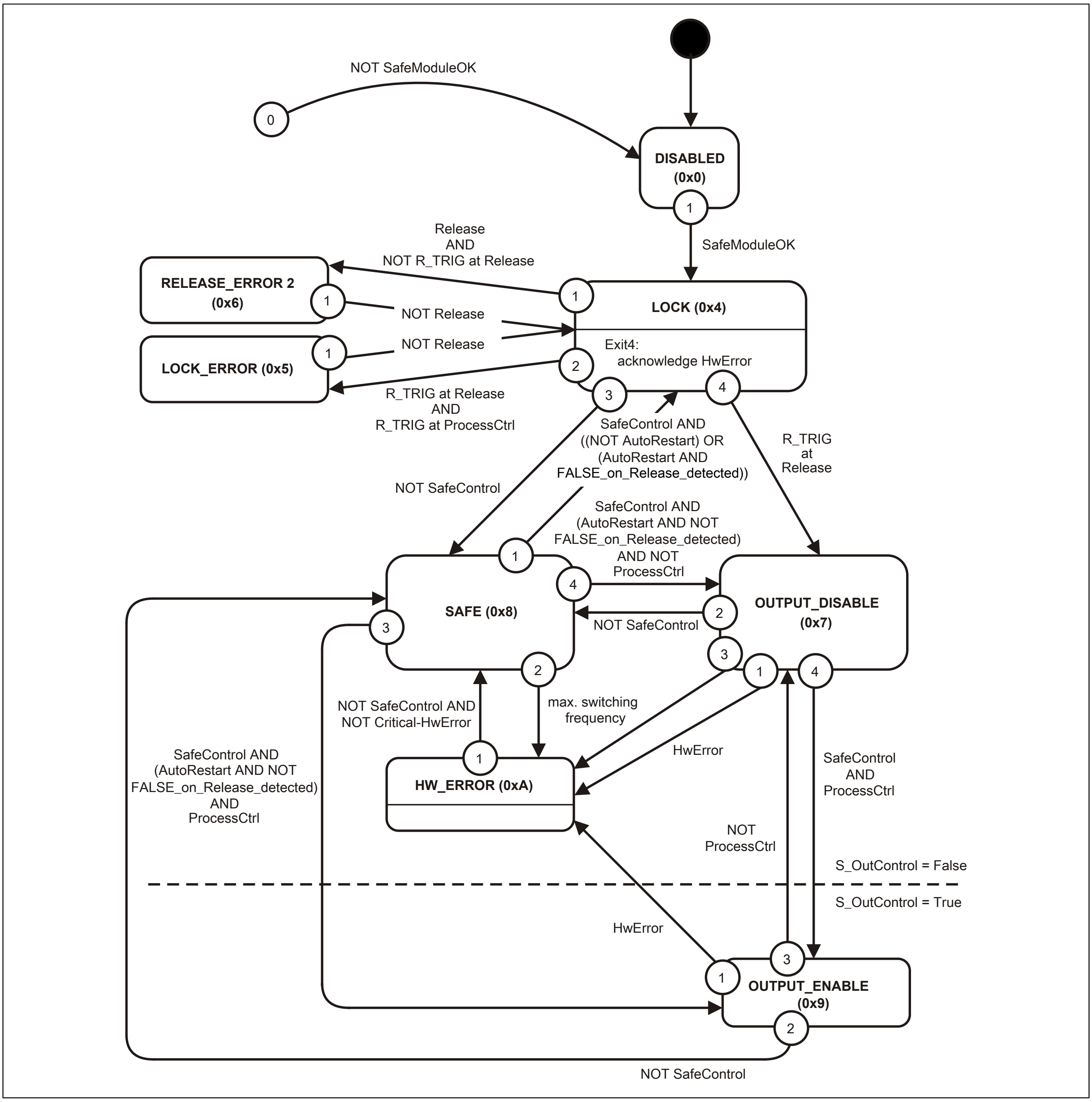


Figure 201: Restart interlock - State diagram

2.6.12.3 X20(c)SOx530

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 251: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 252: Organization of general notices

2.6.12.3.1 General information

The modules are equipped with 2 or 6 safe relay outputs.

The modules can be used for controlling floating actuators in safety-related applications up to PL e or SIL 3.

Safety relays are installed in the module. The positively driven feedback contacts are evaluated internally by the module. The safe digital output modules have a start interlock on error in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- 2 or 6 safe relay outputs
- Output type "Relay"
- Relay module for 230 VAC / 24 VDC
- Switching current 6 A
- Normally open contact
- Single-channel isolated outputs

Danger!

Risk of electric shock!

The terminal block is only permitted to conduct voltage when it is connected. It is not permitted to be disconnected or connected while voltage is applied or have voltage applied to it while it is removed under any circumstances.

2.6.12.3.1.1 Function

Safe relay outputs

The modules are equipped with safe relay outputs for floating control of actuators in safety-related applications up to PL e or SIL 3.

Safety relays are installed in the module. The positively driven feedback contacts are evaluated internally by the module. The B10d values are specified in the technical data for the safety-related perspective of the relay contacts. These values apply up to the specified maximum contact service life.

Safe digital output modules have a start interlock on error in the event of network errors. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

2.6.12.3.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.12.3.2 Overview

Module	**X20SO2530**	**X20SO6530**
Relay outputs		
Number of outputs	2	6
Switching voltage range	5 to 24 VDC, 5 to 230 VAC	
Switching current range	5 mA to 6 A	5 mA to 6 A (hardware revision < B5: 2 A)
Overload protection and short-circuit protection	External 6 A gL/gG fuse (blow-out fuse), LS automat C characteristic 1.6 A	

Table 253: Digital output modules

2.6.12.3.3 Order data


X20SO2530


X20SO6530

Model number	Short description
	Digital output modules
X20SO2530	X20 safe digital output module, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A
X20cSO2530	X20 safe digital output module, coated, 2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A
X20SO6530	X20 safe digital output module, 6 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A
	Required accessories
	Bus modules
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous
	Terminal blocks
X20TB72	X20 terminal block, 12-pin, safety-keyed, 240 VAC, red

Table 254: X20SO2530, X20cSO2530, X20SO6530 - Order data

2.6.12.3.4 Technical data

Model number	X20SO2530	X20cSO2530	X20SO6530
Short description			
I/O module	2 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A		6 relays with 1 normally open contact each, 230 VAC / 6 A, 24 VDC / 6 A
General information			
B&R ID code	0xD205	0xDD86	0xF22A
System requirements			
Automation Studio	3.0.81.15 or later	4.0.16 or later	4.2.5 or later
Automation Runtime	3.00 or later	V3.08 or later	4.2 or later
SafeDESIGNER	2.70 or later	3.1.0 or later	4.2.0 or later
Safety Release	1.2 or later	1.7 or later	1.10 or later
Status indicators	I/O function per channel, operating state, module status		
Diagnostics			
Module run/error	Yes, using status LED and software		
Outputs	Yes, using status LED and software		
Blackout mode			
Scope	Module		
Function	Module function		
Standalone mode	No		
Max. I/O cycle time	1 ms		
Power consumption			
Bus	0.26 W		
Internal I/O	1.15 W		1.65 W
Electrical isolation			
Channel - Bus	Yes		
Channel - Channel	Yes		
Certifications			
CE	Yes		
EAC	Yes		
UL	cULus E115267 Industrial control equipment		In preparation
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5		-
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X		
DNV GL	In preparation		
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013		
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3		
Functional safety	EN 50156-1:2004		-
Relays			
EN 50155	Yes		No
EN 50205	Yes		
Safety characteristics			
EN ISO 13849-1:2015			
MTTFD	2500 years		
Mission time	Max. 20 years		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
PFH / PFH_d			
Module	$<1*10^{-10}$		
openSAFETY wired	Negligible		
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour		
PFD	$<2*10^{-5}$		
Proof test interval (PT)	20 years		
Safe relay channels			
EN ISO 13849-1:2015			
Category	Cat. 1 if the relay channel is used individually, Cat. 4 if 2 relay channels are connected in series [1)]		
PL	PL c if the relay channel is used individually, PL e if 2 relay channels are connected in series [1)]		
B10d			
DC1, 24 VDC	6 A / 780,000		6 A / 1,000,000
AC1, 230 VAC	6 A / 780,000		6 A / 200,000

Table 255: X20SO2530, X20cSO2530, X20SO6530 - Technical data

Model number	X20SO2530	X20cSO2530	X20SO6530
AC15, 230 VAC	3 A / 1,960,000		5 A / 100,000
DC13, 24 VDC	5 A / 780,000		4 A / 300,000 [2]
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
SIL CL	SIL 1 if the relay channel is used individually, SIL 3 if 2 relay channels are connected in series [1]		
I/O power supply			
Nominal voltage	24 VDC		
Voltage range	24 VDC -15% / +20%		
Integrated protection	Reverse polarity protection		
Relay outputs			
Variant	2 relays, each with 1 normally open contact, internal high-side and low-side control		6 relays, each with 1 normally open contact, internal high-side and low-side control
Diagnostic status	Contact position determined by positively driven contacts		
Max. switching frequency	10 Hz		
Switching delay			
0 → 1	<50 ms		
1 → 0	<50 ms		
Isolation voltage between channel and bus	Safe disconnection of 300 VAC per EN 50178		
Isolation voltage between channel and channel	Tested at 1350 VAC		
Contact resistance (without terminal block)	20 mΩ		
Contact service life	See "Contact service life".		
Short-circuit protection, overload protection	External 6 A gL/gG fuse (blow-out fuse), LS automat C characteristic 1.6 A		
Switching voltage range	5 to 24 VDC, 5 to 230 VAC		
Switching current range	5 mA to 6 A		5 mA to 6 A (hardware revision < B5: 2 A)
Coil voltage	24 VDC -15% / +20%		
Short-circuit proof	Yes, 1000 A (with specified short-circuit / overload protection)		
Max. inrush current	30 A for 20 ms		AC: 50 A for 100 ms, DC: 10 A for 200 ms
Overvoltage category per EN 60664-1	II		
Max. switching capacity			
AC1	230 VAC / 6 A		
AC15	230 VAC / 3 A		230 VAC / 5 A
DC1	24 VDC / 6 A		
DC13	24 VDC / 5 A / 0.1 Hz		24 VDC / 4 A / 0.1 Hz
Operating conditions			
Mounting orientation			
Horizontal	Yes		
Vertical	Yes		
Installation elevation above sea level	0 to 2000 m, no limitation		
Degree of protection per EN 60529	IP20		
Ambient conditions			
Temperature			
Operation			
Horizontal mounting orientation [3]	0 to 60°C	-25 to 60°C	0 to 60°C
Vertical mounting orientation	0 to 50°C	-25 to 50°C	0 to 50°C
Derating	See section "Derating".		
Storage	-40 to 85°C		
Transport	-40 to 85°C		
Relative humidity			
Operation	5 to 95%, non-condensing	Up to 100%, condensing	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing		
Transport	5 to 95%, non-condensing		
Mechanical properties			
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.		
Spacing	$25^{+0.2}$ mm		

Table 255: X20SO2530, X20cSO2530, X20SO6530 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) In this case, a protective circuit (parallel diode via load) is necessary.
3) Compared to the specification in the X20 system user's manual, in which the angle of the horizontal mounting orientation is 70°, this applies only up to an angle of 85° on the X20(c)SO2530. Below this, the derating for face-up installation must be applied.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter **"Installation notes for X20 modules" on page 23.**

Derating

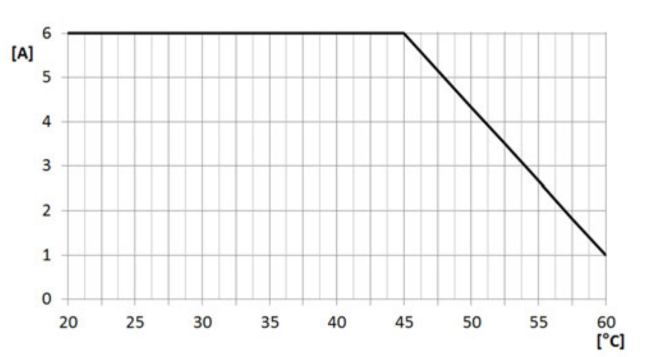
The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus or to the left by the derating penalty by the following measures in a horizontal mounting orientation.

Module	X20SO2530	X20SO6530
Derating bonus		
At 24 VDC	+0°C	
Dummy module on the left	+0°C	
Dummy module on the right	+2.5°C	
Dummy module on the left and right	+2.5°C	
With double PFH / PFH_d	+0°C	
Hardware revision < B5	+0°C	-5°C

Table 256: Derating bonus / Derating penalty

The maximum nominal current per channel depends on the operating temperature and mounting orientation. The resulting nominal current per channel is listed in the following diagrams.

Horizontal (0 to 60°C, coated: -25 to 60°C)	Vertical (0 to 50°C, coated: -25 to 50°C)
X20SO2530	



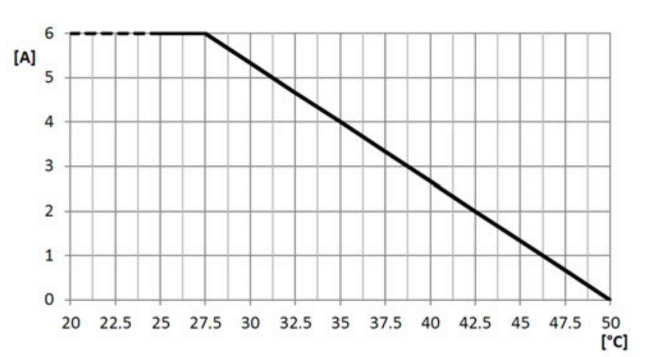
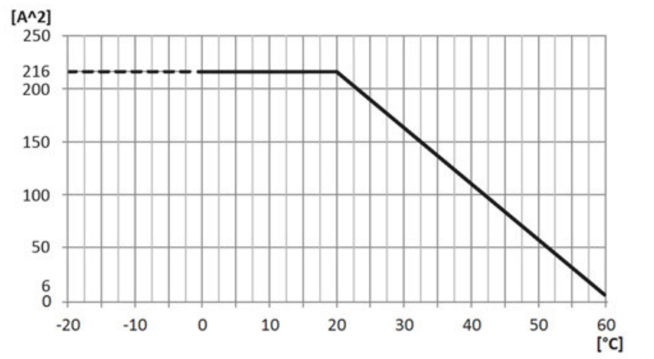


Table 257: Derating in relation to operating temperature and mounting orientation

The max. squared summation current (i.e. sum of the square of the nominal currents) depends on the operating temperature and mounting orientation. The resulting max. squared summation current is listed in the following diagrams.

Horizontal (0 to 60°C)	Vertical (0 to 50°C)
X20SO6530	



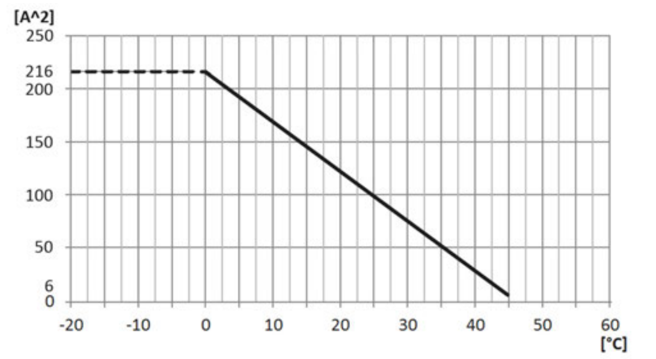


Table 258: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

Contact service life of relay outputs

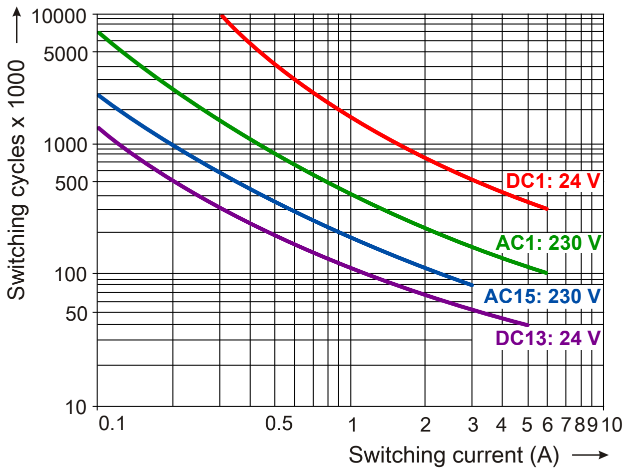
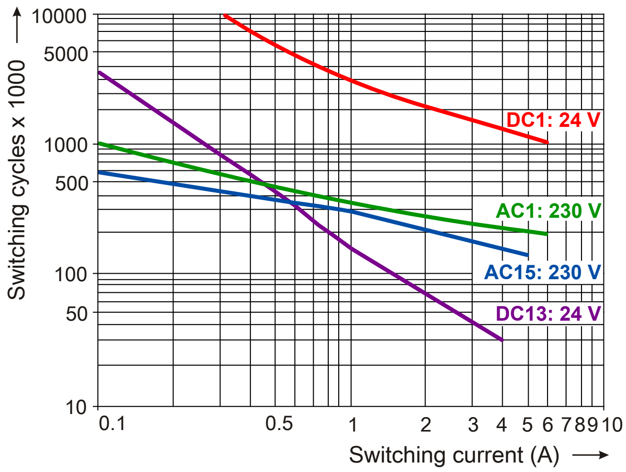
Contact service life of relay outputs	
X20SO2530	X20SO6530
	

Table 259: Contact service life of relay outputs

2.6.12.3.5 LED status indicators

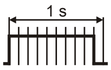
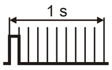
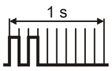
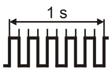
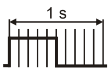
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Mode "Reset"
			Double flash	Updating firmware
			Blinking	Mode PREOPERATIONAL
			On	Mode RUN
	e	Red	Off	Module not supplied with power or everything OK
			Pulsating	Bootloader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Solid red / Single green flash		Invalid firmware
	1 to 6	Output state of the corresponding digital output		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or startup not yet completed
		Orange	On	Output set
X20SO2530	SE	Red	Off	Mode RUN or I/O component not provided with voltage
			 1 s	Boot phase, missing X2X Link or defective processor
			 1 s	Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in state PREOPERATIONAL.
			 1 s	Safe communication channel not OK
			 1 s	The firmware for this module is a non-certified pilot customer version.
X20SO6530			 1 s	Boot phase, faulty firmware
			On	Safety state active for the entire module (= state "FailSafe")
		The "SE" LEDs separately indicate the status of safety processor 1 (LED "S") and safety processor 2 (LED "E").		

Table 260: Status indicators

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.12.3.6 Pinouts

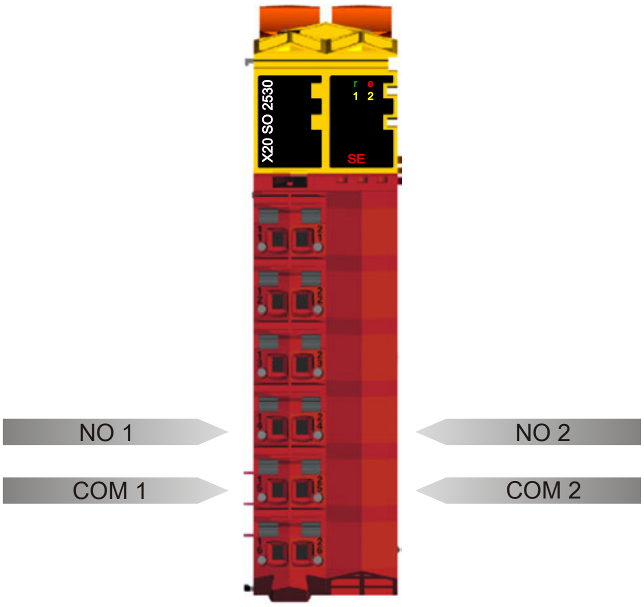


Figure 202: X20SO2530 - Pinout

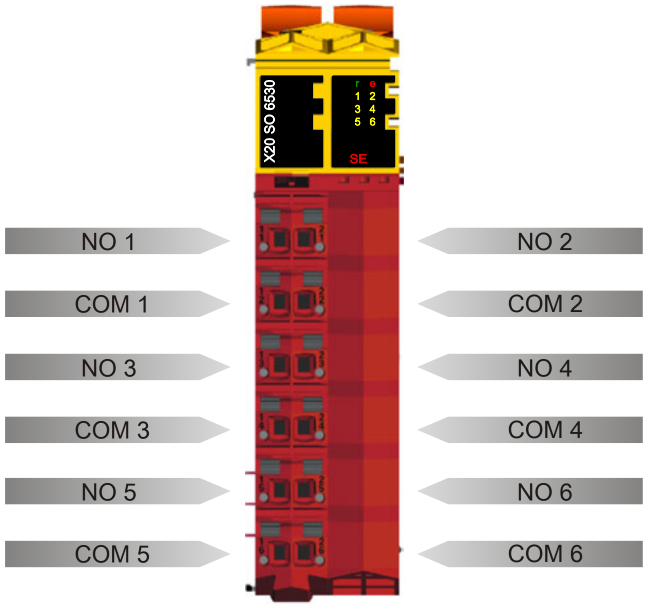


Figure 203: X20SO6530 - Pinout

2.6.12.3.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.12.3.7.1 Connecting safety-oriented actuators for relay outputs

The connection example shown here only represents a selection of the possible wiring methods. However, the following must always be taken into consideration:

- Two relay channels must be connected in series for applications that correspond to EN ISO 13849-1:2015 above category 1.
- Relay contacts must be protected with a fuse (see technical data for the module).

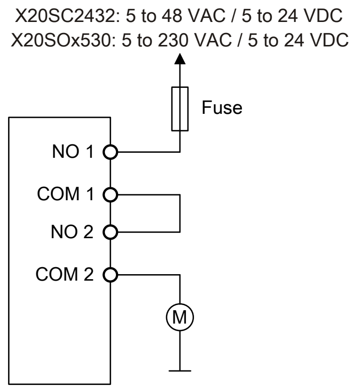


Figure 204: Connecting safety-oriented actuators for relay outputs

Danger!

Make sure that a proper protective circuit is used for the relay contacts (see technical data for the module). Also consider that operation outside of the specification is not permitted.

Operating outside of the specification or not using a protective circuit can cause the relay contacts to melt simultaneously, resulting in a loss of safety functionality.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Information:

Using signal "SafeDigitalOutputxxyy" and "SafeDigitalOutputxx" at the same time is not permitted and prevented by the system.

Using signal "SafeDigitalOutputxxyy" causes a switch-on sequence to be activated that switches on relay 2 with a 20 ms delay. This behavior is necessary to prevent simultaneous melting of the two relay contacts in certain operating states. Controlling two independent EN ISO 13849-1:2015 Category 1 actuators using signal "SafeDigitalOutputxxyy" must therefore be avoided since this causes delayed activation of the actuator on channel 2.

2.6.12.3.8 Error detection

2.6.12.3.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.12.3.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

> **Danger!**
>
> **Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.**

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

Connecting safety-oriented actuators for relay outputs

Danger!

A relay channel does not have error detection for wiring problems. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures or by the connected device.

Danger!

The user is responsible for ensuring that each relay channel is cut off at least 1x per week so that the appropriate internal tests can be completed.

2.6.12.3.9 Output circuit diagram

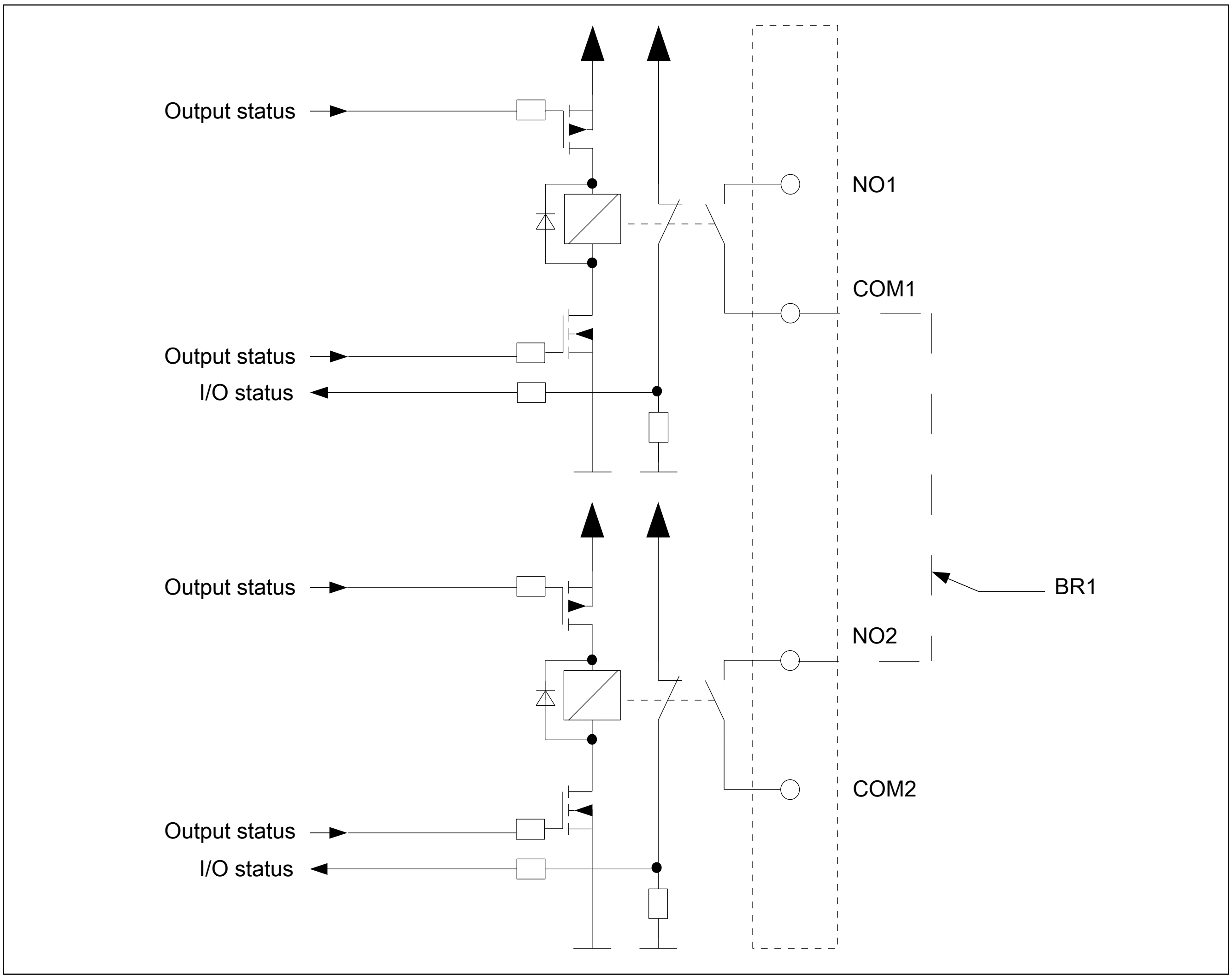


Figure 205: Output circuit diagram

2.6.12.3.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.12.3.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time	
X20SO2530	X20SO6530
500 µs	

Maximum I/O update time	
X20SO2530	X20SO6530
1000 µs + 50 ms	

2.6.12.3.12 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.12.3.13 Restart behavior

Each digital input channel is not equipped with an internal start interlock on error, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network. It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example. Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal "start interlock on error", which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the output as described in section ""Start interlock on error" state diagram".

This sequence is necessary in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

2.6.12.3.14 X20SO2530 - Register description

2.6.12.3.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 261: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
SerialNumber
ModuleID
HardwareVariant
FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 262: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx
DigitalOutputxxyy</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 263: I/O configuration parameters: Output signal path

2.6.12.3.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 264: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.

<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td><td>No</td><td>-</td></tr>
<tr><td>Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.

<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td><td>Yes</td><td>-</td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 265: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 266: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.12.3.14.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 267: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

Parameter	Description	Default value	Unit
Manual Configuration	This parameter makes it possible to manually and individually configure the safety response time for the module. The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module. **Parameter value** – **Description** Yes – Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals. No – The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.	No	-
Safe Data Duration	This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module. For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added. • Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)	20000	µs
Additional Tolerated Packet Loss	This parameter specifies the number of additional tolerated lost packets during data transfer. • Permissible values: 0 to 10	0	Packets
Packets per Node Guarding	This parameter specifies the maximum number of packets used for node guarding. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.	5	Packets

Table 268: SafeDESIGNER parameters: Safety Response Time

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior"). **Parameter value** – **Description** Yes-ATTENTION – "Automatic restart" function is activated. No – "Automatic restart" function is not activated.	No	-

Table 269: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.12.3.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) 1)</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) 1)</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>DigitalOutputxxyy</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal for combined channel SO xx/yy</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxxyy</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe combined channel SO xx/yy</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>ReleaseOutputxxyy</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of combined channel SO xx/yy</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table>
<tr><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr>
<tr><td>Reserved</td><td>Reserved</td><td>Channel 2</td><td>Channel 1</td></tr>
</table></td></tr>
</table>

Table 270: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Information:

Using signal "SafeDigitalOutputxxyy" and "SafeDigitalOutputxx" at the same time is not permitted and prevented by the system.

Using signal "SafeDigitalOutputxxyy" causes a switch-on sequence to be activated that switches on relay 2 with a 20 ms delay. This behavior is necessary to prevent simultaneous melting of the two relay contacts in certain operating states. Controlling two independent EN ISO 13849-1:2015 Category 1 actuators using signal "SafeDigitalOutputxxyy" must therefore be avoided since this causes delayed activation of the actuator on channel 2.

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

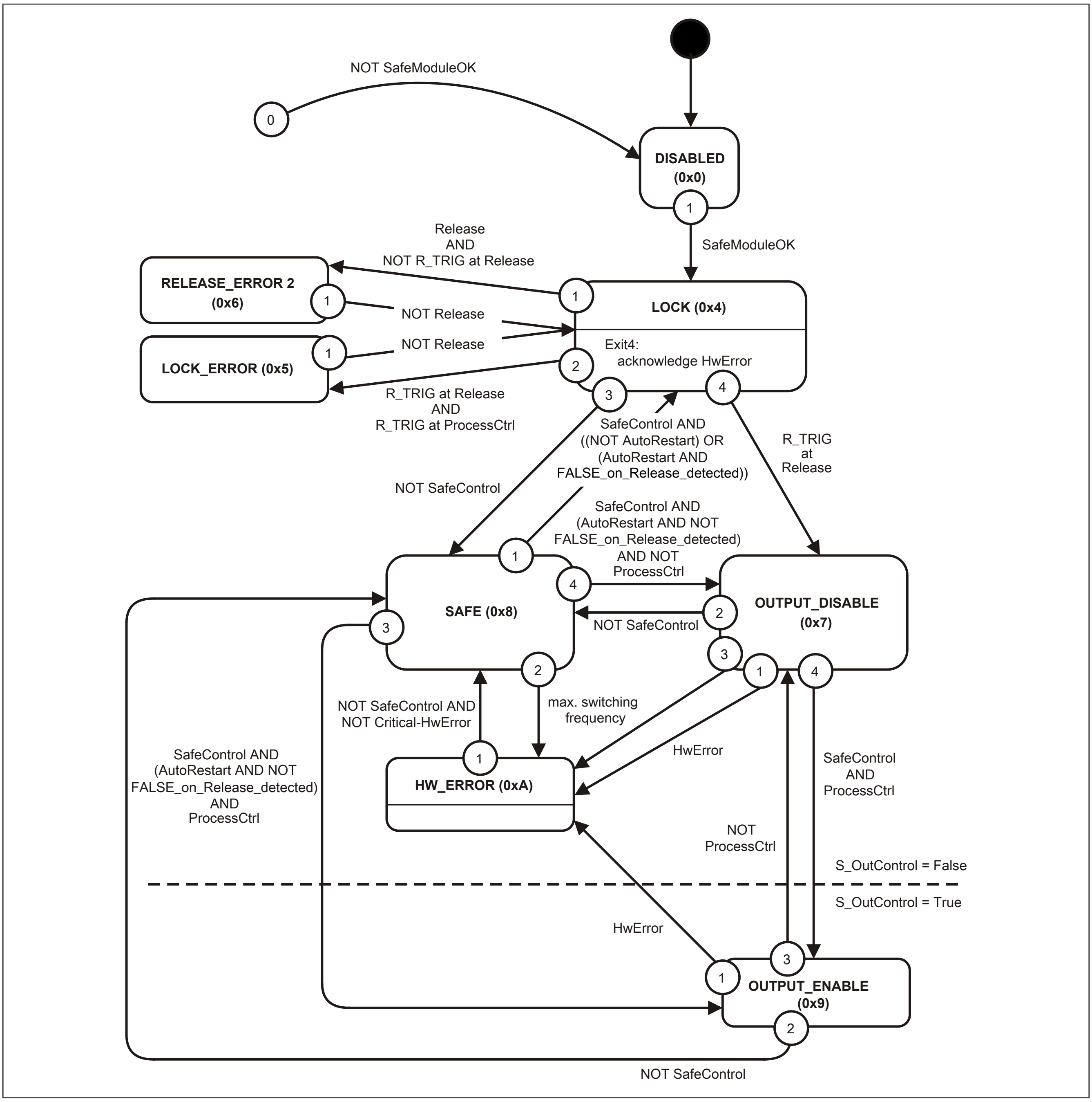


Figure 206: Restart interlock - State diagram

2.6.12.3.15 X20SO6530 - Register description

2.6.12.3.15.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 271: I/O configuration parameters: Function model

Group: General

Parameter	Description	Default value	Unit
Module supervised	System behavior when a module is missing **Parameter value** / **Description** On: A missing module triggers service mode. Off: A missing module is ignored.	On	-
Blackout mode	This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode). **Parameter value** / **Description** On: Blackout mode is enabled. Off: Blackout mode is disabled.	Off	-
Channel state information	This parameter enables/disables the channel-specific status information in the I/O mapping.	On	-
State number for start interlock on error	This parameter enables/disables the status information for the start interlock on error.	Off	-
SafeDOMAIN ID	In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller. • Permissible values: 1 to 1024	Assigned automatically	-
SafeNODE ID	Unique safety address of the module • Permissible values: 2 to 1023	Assigned automatically	-

Table 272: I/O configuration parameters: General

Group: Output signal path

Parameter	Description	Default value	Unit
Digital output xx Digital output xxyy	This parameter specifies the mode that can be used by the standard application to access the output channel. **Parameter value** / **Description** Direct: The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are available in the I/O mapping accordingly. Via SafeLOGIC: The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" and "DigitalOutputxxyy" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.	Direct	-

Table 273: I/O configuration parameters: Output signal path

2.6.12.3.15.2 Parameters in SafeDESIGNER

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 274: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 275: SafeDESIGNER parameters: Safety Response Time

2.6.12.3.15.3 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) 1)</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>MissionTimeEnd</td><td>(Read) 1)</td><td>-</td><td>UDINT</td><td>Expiration of the mission time.
Specified in format YYYYMMDD (e.g. 20250506 = 2025-05-06, 0 = Not stored).</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate</td><td>(Read) 1)</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) 1)</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>DigitalOutputxxyy</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal for combined channel SO xx/yy</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxxyy</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe combined channel SO xx/yy</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutput</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for start interlock on error</td></tr>
<tr><td>PhysicalStateOutputxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBOutputState</td><td>Read</td><td>-</td><td>UDINT</td><td>State number of start interlock on error of channel x.
See section ""Start interlock on error" state diagram"
<table>
<tr><th>Bit 23 to 20</th><th>Bit 19 to 16</th><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr>
<tr><td>Channel 6</td><td>Channel 5</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table></td></tr>
</table>

Table 276: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Information:

Using signal "SafeDigitalOutputxxyy" and "SafeDigitalOutputxx" at the same time is not permitted and prevented by the system.

Using signal "SafeDigitalOutputxxyy" causes a switch-on sequence to be activated that switches on relay 2 with a 20 ms delay. This behavior is necessary to prevent simultaneous melting of the two relay contacts in certain operating states. Controlling two independent EN ISO 13849-1:2015 Category 1 actuators using signal "SafeDigitalOutputxxyy" must therefore be avoided since this causes delayed activation of the actuator on channel 2.

"Start interlock on error" state diagram

The "start interlock on error" works independently of the "Enabling principle", i.e. the behavior described in section "Restart behavior" is not influenced by the configuration of the enabling principle or by the chronological position of the standard switching signal "DigitalOutputxx".

The following state diagram illustrates the effect of the "start interlock on error" integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBOutputState".

Information:

To set an output channel, at least one time interval from one network cycle is required between the rising edge on signal "SafeDigitalOutputxx" and the rising edge on signal "ReleaseOutput". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

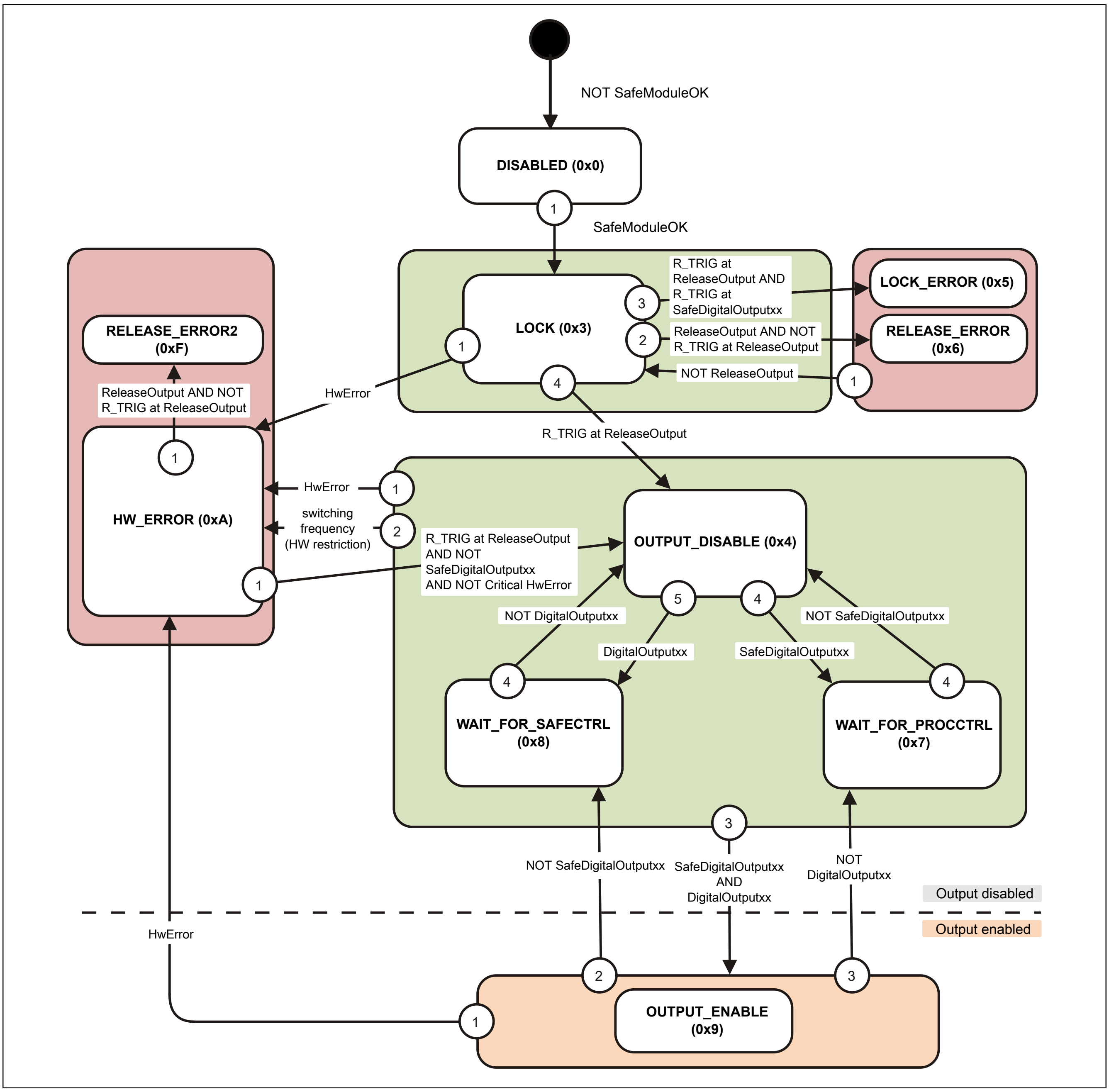


Figure 207: Start interlock on error - State diagram

2.6.13 Power supply modules

2.6.13.1 Overview

Model number	Short description	Page
X20SP1130	X20 power supply module, with integrated safe cutoff function, for internal I/O power supply, 24 VDC, 10 A, 1 safe type B1 digital output, 24 VDC, 10 A, without OSSD, note the list of permitted modules in the potential group	623

2.6.13.2 X20SP1130

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 277: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 278: Organization of general notices

2.6.13.2.1 General information

The modules are equipped with 1 safe digital output. The nominal output current is 10 A. Power supply modules are used for the internal I/O power supply.

A safe digital output channel is integrated in the module for cutting off the I/O power supply of connected X20 modules in safety-related applications up to PL e or SIL 3.

The output is designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case.
In addition, the instructions for cutting off X20 potential groups must be observed.
Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- 1 safe digital output with 10 A
- Source circuit
- Output type B
- 24 VDC power supply module for internal I/O power supply
- Safe cutoff of potential groups with standard outputs
- Safely switched potential can also be tapped externally
- Integrated output protection

2.6.13.2.1.1 Function

Power supply module

The power supply module is used for the internal I/O power supply. A safe digital output channel is integrated in the module for cutting off the I/O power supply of connected X20 modules in safety-related applications up to PL e or SIL 3.

The modules arranged in the X20 potential group must support the "safe cutoff of an X20 potential group" operating principle.

The potential switched with the safe digital output channel is conducted on pins 11 and 21 of the terminal block so that externally connected actuators can be cut off in this way as well. The output is designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Safe digital outputs

The module is equipped with a safe digital output channel. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The output is designed using semiconductor technology so that its safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without reference potential (e.g. relays, valves). For actuators with reference potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the wiring in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are readable.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

2.6.13.2.2 Overview

Module	X20SP1130
Number of outputs	1
Nominal voltage	24 VDC
Nominal output current	10 A
Output protection	Protection for switching inductive loads

Table 279: Power supply modules

2.6.13.2.3 Order data

Model number	Short description	Figure
	Power supply modules	
X20SP1130	X20 power supply module, with integrated safe cutoff function, for internal I/O power supply, 24 VDC, 10 A, 1 safe type B1 digital output, 24 VDC, 10 A, without OSSD, note the list of permitted modules in the potential group	
	Required accessories	
	Bus modules	
X20BM23	X20 power supply bus module, for X20 SafeIO power supply modules, internal I/O power supply interrupted to the left	
X20BM26	X20 power supply bus module, for X20 SafeIO power supply modules, with node number switch, internal I/O power supply interrupted to the left	
	Terminal blocks	
X20TB52	X20 terminal block, 12-pin, safety-keyed	

Table 280: X20SP1130 - Order data

2.6.13.2.4 Technical data

Model number	X20SP1130
Short description	
I/O module	1 safe type B1 digital output, 24 VDC, 10 A, without OSSD, note list of permitted modules in the potential group
General information	
B&R ID code	0x1DBF
System requirements	
Automation Studio	3.0.81.15 or later
Automation Runtime	3.00 or later
SafeDESIGNER	2.70 or later
Safety Release	1.2 or later
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using status LED and software
Outputs	Yes, using status LED and software
Blackout mode	
Scope	Module
Function	Module function
Standalone mode	No
Max. I/O cycle time	800 µs
Power consumption	
Bus	0.2 W
Internal I/O	1.5 W
Electrical isolation	
Channel - Bus	Yes
Certifications	
CE	Yes
KC	Yes
EAC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X
DNV GL	In preparation
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3
Functional safety	EN 50156-1:2004
Safety characteristics	
EN ISO 13849-1:2015	
Category	Cat. 4
PL	PL e
DC	>94%
MTTFD	2500 years
Mission time	Max. 20 years
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
SIL CL	SIL 3
SFF	>90%
PFH / PFH_d	
Module	$<1*10^{-10}$
openSAFETY wired	Negligible
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour
PFD	$<2*10^{-5}$
Proof test interval (PT)	20 years
Input I/O power supply 1)	
Input voltage	24 VDC -15% / +20%
Fuse	Required line fuse: Max. 10 A, slow-blow
Reverse polarity protection	No
Safe digital outputs	
Variant	FET, 2x positive switching, type B1, output level readable
Nominal voltage	24 VDC

Table 281: X20SP1130 - Technical data

Model number	X20SP1130
Nominal output current	10 A
Output protection	Protection for switching inductive loads [2)]
Braking voltage when switching off inductive loads	1 VDC
Diagnostic status	Output monitoring, current measurement (shutdown in the event of overcurrent)
Error detection time	2 s
Isolation voltage between channel and bus	500 V_{eff}
Leakage current when switched off	1 mA
Residual voltage	≤200 mVDC at nominal output current
Switching voltage	I/O power supply minus residual voltage
Max. switching frequency	5 times per minute with max. 2 Hz
Test pulse length	Without test pulse
Max. capacitive load	1 mF
Minimum load	15 mA
Current on loss of ground	
I_{OUT}	<1 mA
I_{GND}	<50 mA
Operating conditions	
Mounting orientation	
Horizontal	Yes
Vertical	Yes
Installation elevation above sea level	0 to 2000 m, no limitation
Degree of protection per EN 60529	IP20
Ambient conditions	
Temperature	
Operation	
Horizontal mounting orientation	0 to 60°C
Vertical mounting orientation	0 to 35°C
Derating	See section "Derating".
Storage	-40 to 85°C
Transport	-40 to 85°C
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.
Spacing	$25^{+0.2}$ mm

Table 281: X20SP1130 - Technical data

1) If a hardware revision < B9 or firmware version < 320 is used, then the power supply used must be able to charge a capacitance of 4 mF in a time period of 2 ms.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SP1130
Derating bonus	
At 24 VDC	+0°C
Dummy module on the left	+2.5°C
Dummy module on the right	+0°C
Dummy module on the left and right	+5°C
With double PFH / PFH_d	+0°C

Table 282: Derating bonus

The maximum nominal current per channel depends on the operating temperature and the mounting orientation. The resulting nominal current per channel can be found in the following table.

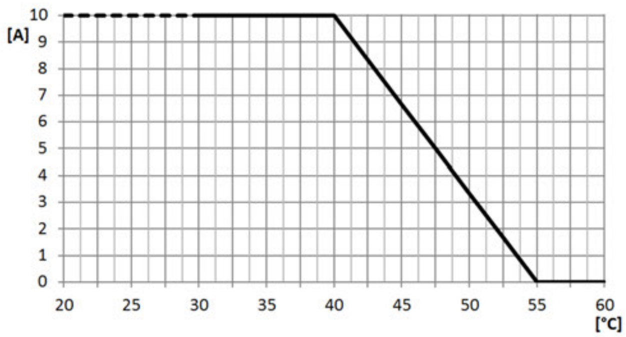
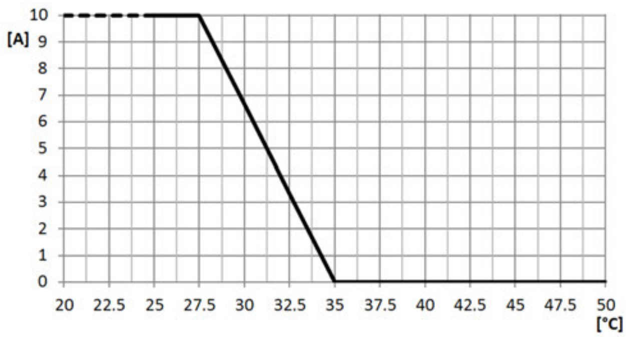
Horizontal (0 to 60°C)	Vertical (0 to 35°C)
	

Table 283: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.13.2.5 LED status indicators

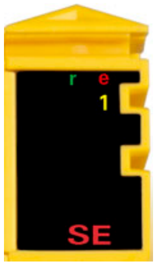
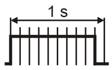
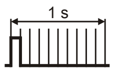
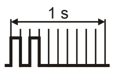
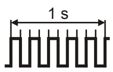
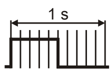
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1	Output status of the digital output		
		Red	On	Warning/Error on output channel, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 284: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.13.2.6 Pinout

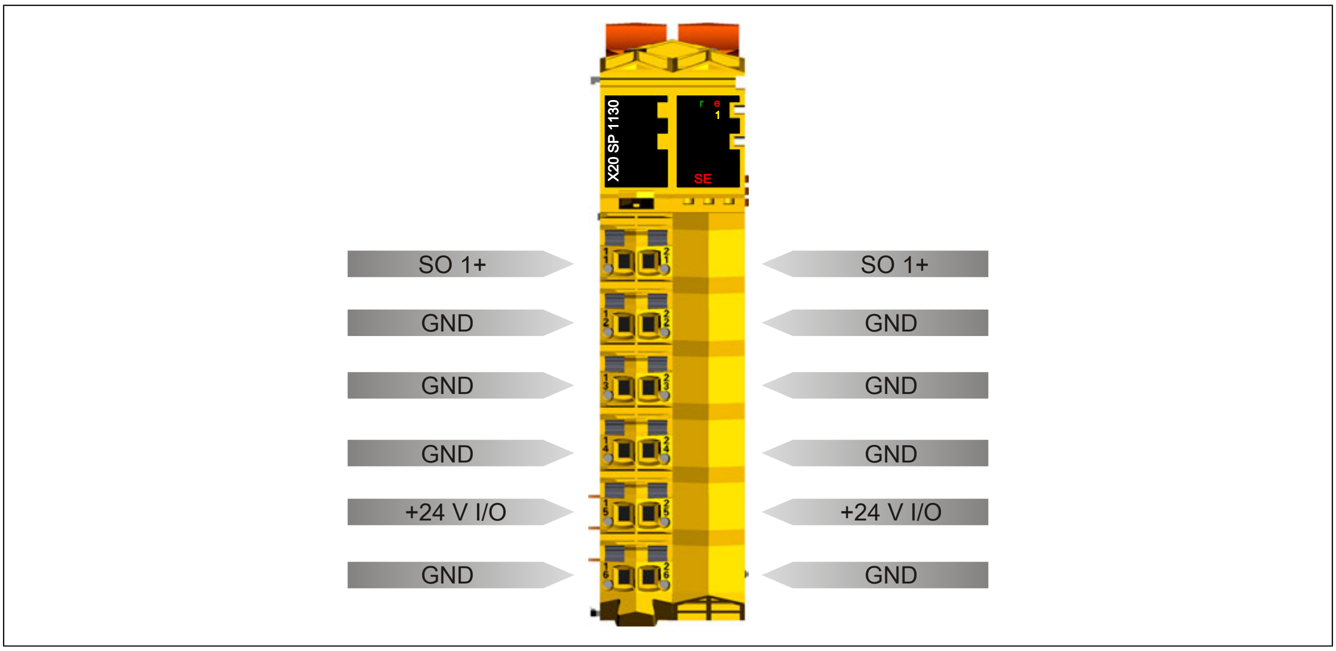


Figure 208: X20SP1130 - Pinout

2.6.13.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.13.2.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

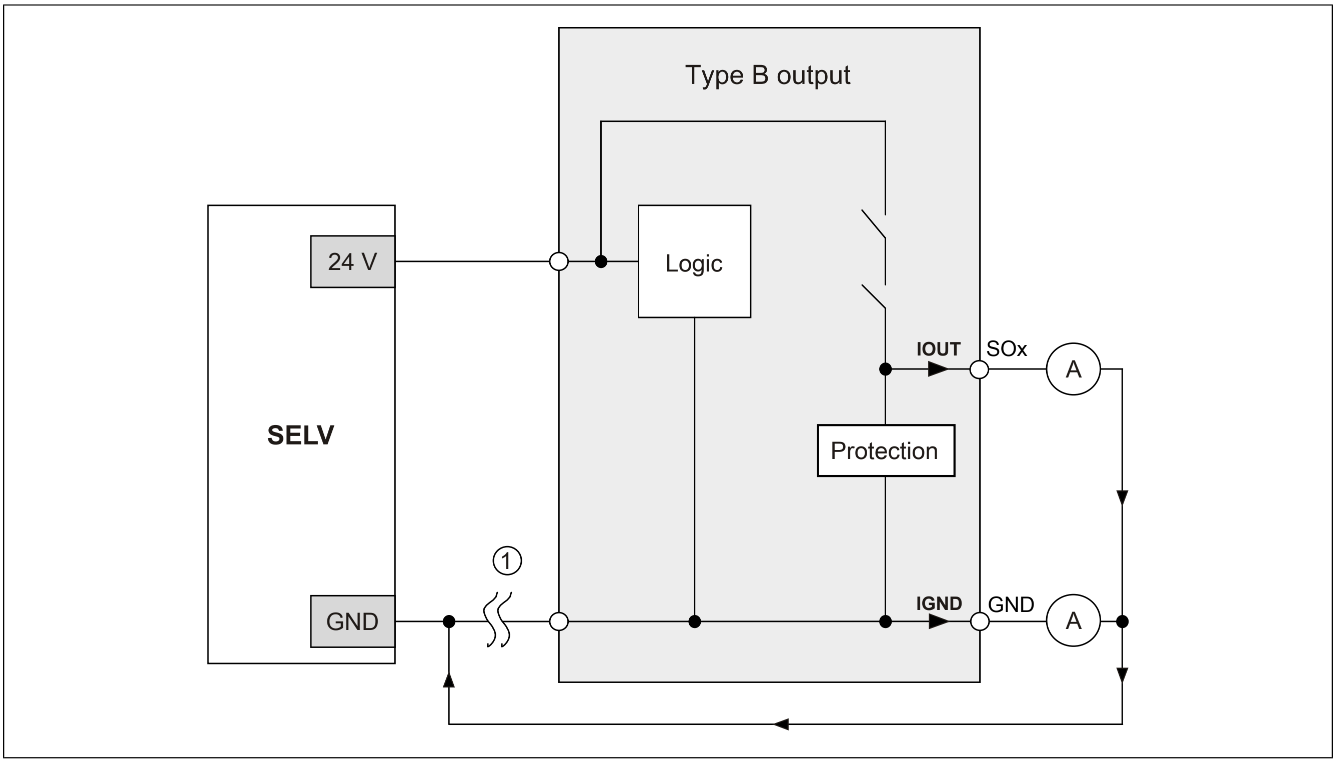


Figure 209: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

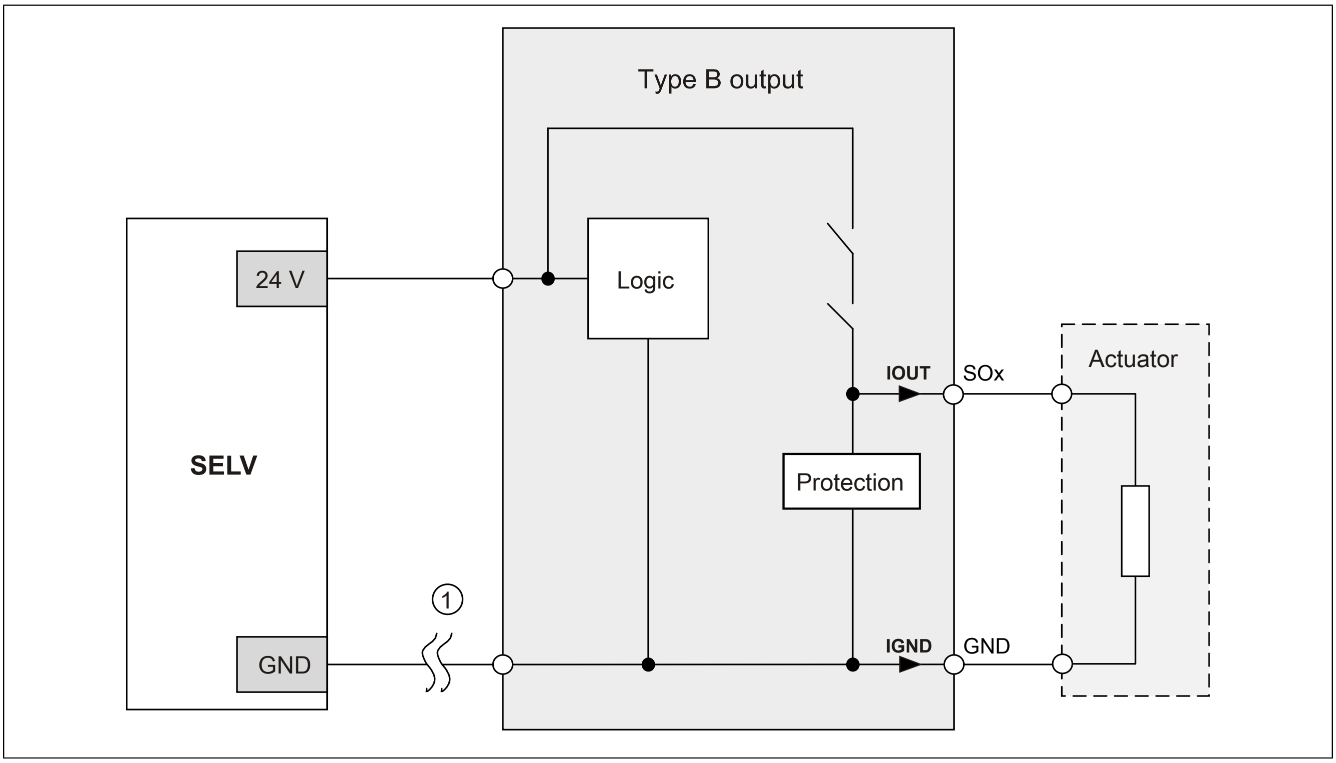


Figure 210: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

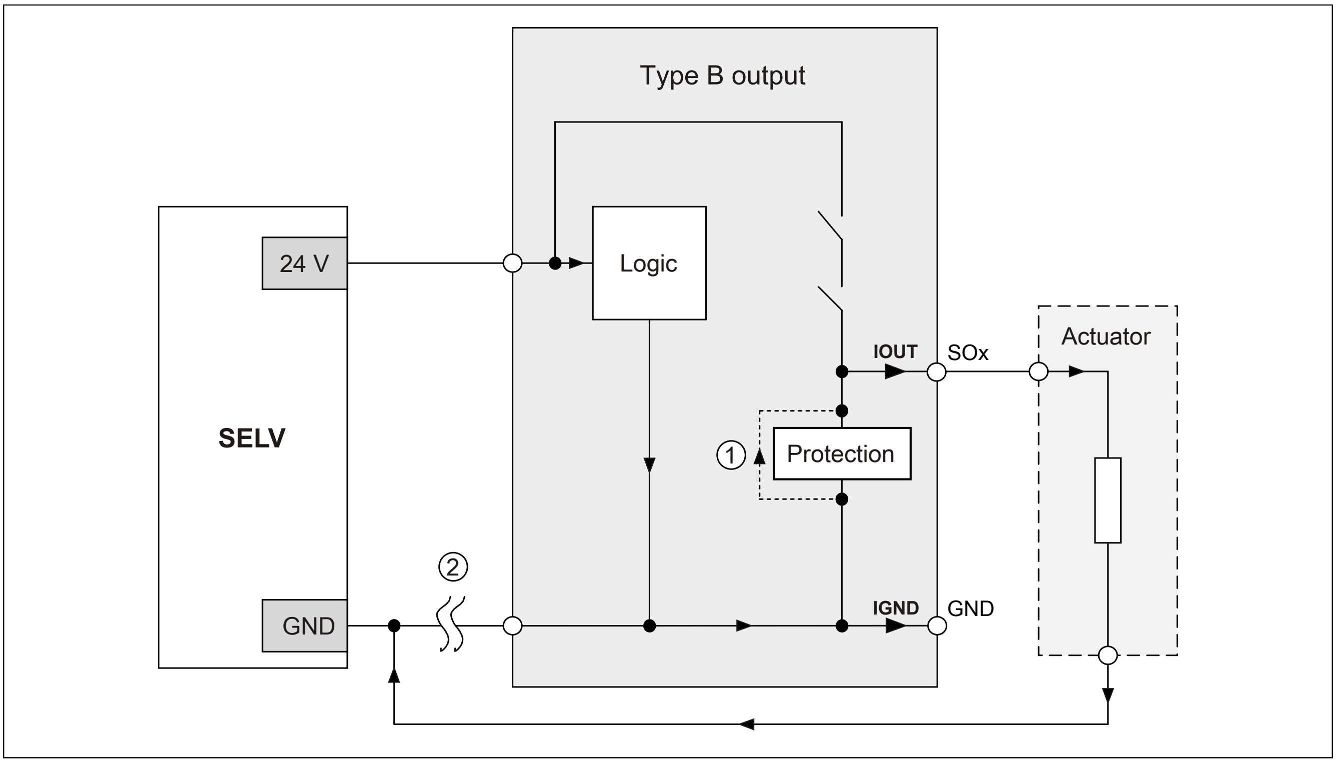


Figure 211: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

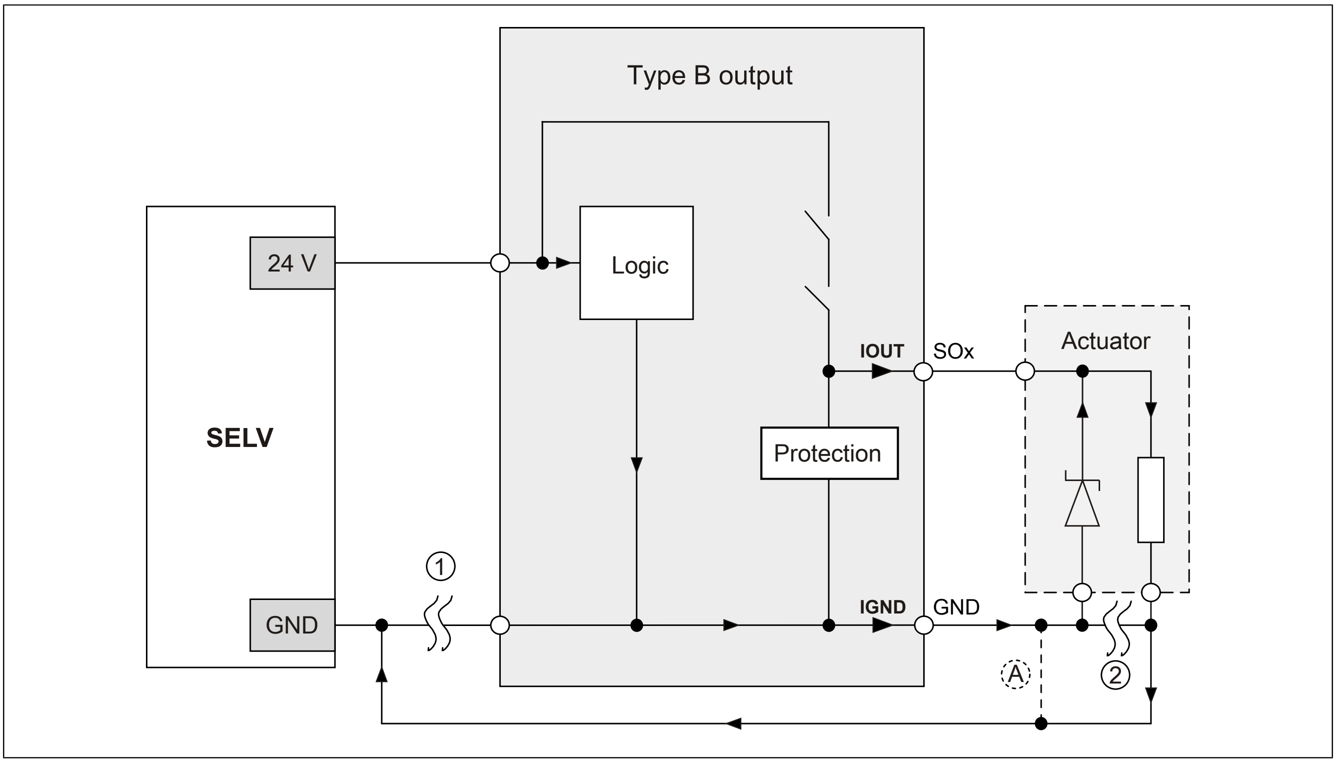


Figure 212: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

X20SP1130 with permissible modules and external GND

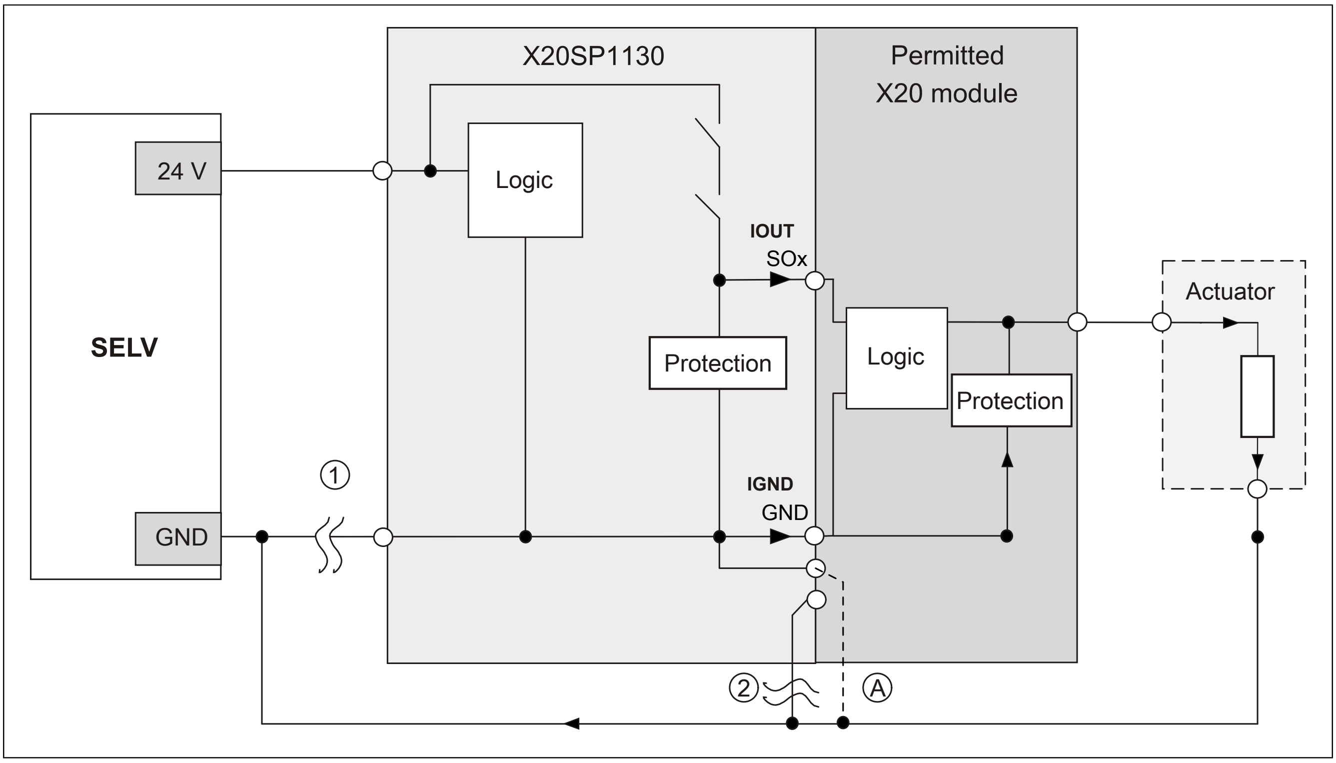


Figure 213: X20SP1130 - Possible connection error

If the permissible module is wired without external GND, then current will not flow on the X20SP1130 if the GND connection is lost. However, if an external GND connection is used, then the currents I_{OUT} and I_{GND} could flow in the following scenarios:

Fault sequence:

- Fault ①: Loss of GND on the X20SP1130
- Fault ②: Loss of GND on return line to the X20SP1130 terminal block

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open line fault in ② → see connection Ⓐ.

2.6.13.2.7.2 Connecting safety-oriented actuators for Type B outputs

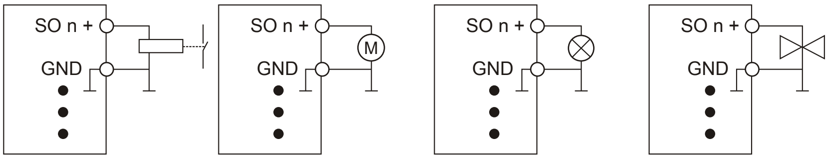


Figure 214: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.13.2.7.3 Operating principle "safe cutoff of an X20 potential group"

The operating principle "Safe cutoff of an X20 potential group" enables the user to execute safety-related functions within an X20 potential group that is supplied with module X20SP1130 per standard EN ISO 13849-1:2015 or EN ISO 13849-2:2012 and EN 62061:2013. The safety function is limited to cutting off or interrupting the power to connected actuators.

Correct functionality was proven in a separate certification test. The result of the certification test is documented in certificate "FS certificate for the operating principle safe cutoff of an X20 potential group". The certificate is available for download from the B&R website (www.br-automation.com).

The user and danger warnings in the documentation associated with the certificate must be observed in all cases (see chapter "Safe cutoff of a potential group" of the "Integrated safety technology" user's manual (MASAFETY-ENG))!

Functionality:
An X20SP1130 module provides the I/O supply for the potential group. When the functional safe state is requested or a FAILSAFE state occurs, this power supply module cuts off the I/O supply of the potential group. The power is then also cut off for all actuators connected to this potential group.

2.6.13.2.8 Error detection

2.6.13.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

2.6.13.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

<u>Type B output channels</u>

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

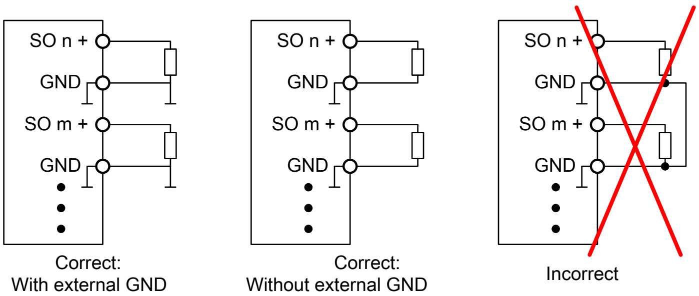


Figure 215: Invalid wiring

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between GND and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Open circuit (output type A and B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td><td rowspan="8">Not detected</td><td rowspan="3">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td><td>Not detected [2)]</td><td>Not detected [2)]</td></tr>
<tr><td>X20SO6300</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
</table>

Table 285: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx	**Not detected**	Detected	**Not detected**	Detected
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				

Table 285: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.13.2.9 Output circuit diagram

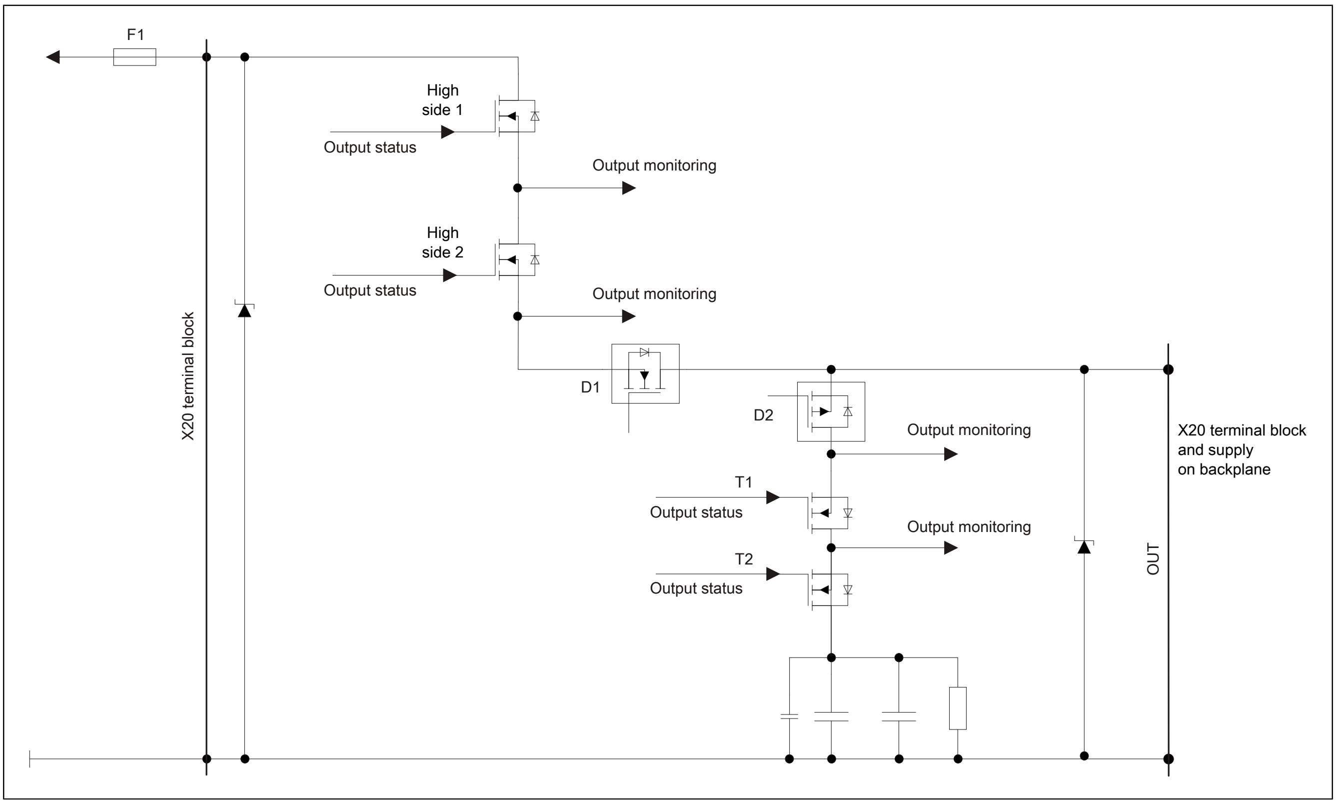


Figure 216: Output circuit diagram

2.6.13.2.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 μs

2.6.13.2.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
400 μs

Maximum I/O update time
1600 μs

2.6.13.2.12 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.13.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.13.2.14 Register description

2.6.13.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 286: I/O configuration parameters: Function model

Group: General

Parameter	Description	Default value	Unit
Module supervised	System behavior when a module is missing **Parameter value** / **Description** On: A missing module triggers service mode. Off: A missing module is ignored.	On	-
Module information (up to AS 3.0.90)	This parameter enables/disables the module-specific information in the I/O mapping: • SerialNumber • ModuleID • HardwareVariant • FirmwareVersion	Off	-
Blackout mode (hardware upgrade 1.10.0.6 or later)	This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode). **Parameter value** / **Description** On: Blackout mode is enabled. Off: Blackout mode is disabled.	Off	-
Output status information	This parameter enables/disables the channel-specific status information in the I/O mapping.	On	-
Restart inhibit state information	This parameter enables/disables restart interlock status information.	Off	-
SafeLOGIC ID	In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller. • Permissible values: 1 to 1024	Assigned automatically	-
SafeMODULE ID	Unique safety address of the module • Permissible values: 2 to 1023	Assigned automatically	-

Table 287: I/O configuration parameters: General

Group: Output signal path

Parameter	Description	Default value	Unit
DigitalOutputxx	This parameter specifies the mode that can be used by the standard application to access the output channel. **Parameter value** / **Description** Direct: The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly. Via SafeLOGIC: The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.	Direct	-

Table 288: I/O configuration parameters: Output signal path

2.6.13.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 289: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
• Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
• Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
• Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 290: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 291: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.13.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 292: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 293: SafeDESIGNER parameters: Safety Response Time

Group: SafeDigitalOutputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr>
<tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr>
</table></td></tr>
</table>

Table 294: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.13.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeChannelOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO x</td></tr>
<tr><td>CurrentOKxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Status of current measurement of channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".</td></tr>
</table>

Table 295: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

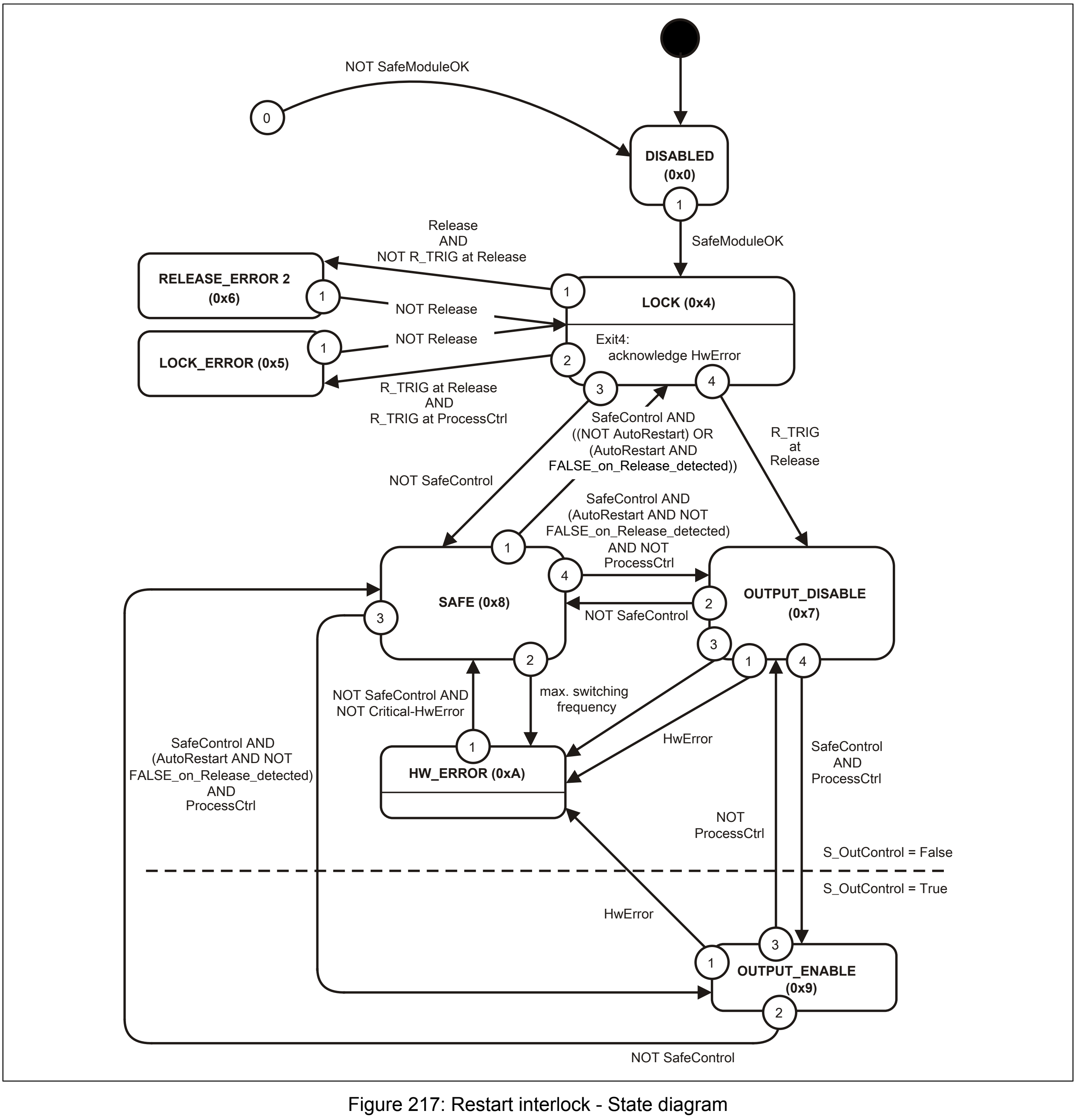


Figure 217: Restart interlock - State diagram

2.6.14 Analog input modules

2.6.14.1 Overview

Model number	Short description	Page
X20SA4430	X20 safe current input module, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	653
X20cSA4430	X20 safe current input module, coated, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	653

2.6.14.2 X20(c)SA4430

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 296: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 297: Organization of general notices

2.6.14.2.1 General information

The modules are equipped with 2 safe analog input pairs for current measurement. Each input pair has its own sensor power supply. The channels with their respective sensor supplies are galvanically isolated from each other. It is possible to acquire current signals in the range of 0.5 to 25 mA.

The safe analog input modules are suitable for safely acquiring current signals for safety-related applications up to PL e or SIL 3.

These modules are designed for X20 16-pin terminal blocks.

- 2 safe analog input pairs for current measurement 0.5 to 25 mA
- 24-bit digital converter resolution
- Channels individually galvanically isolated
- Sensor power supplies galvanically isolated
- Input filter configurable

2.6.14.2.1.1 Function

Safe analog inputs

This safe analog input module is suitable for safely acquiring current signals for safety-related applications up to PL e or SIL 3.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

2.6.14.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.14.2.2 Overview

Module	X20SA4430
Number of inputs	2x 2
Measurement range	Up to firmware version 321: 3.6 to 21 mA, firmware version 322 or later: 0.5 to 25 mA
Digital converter resolution	24-bit
Note	Electrical isolation between channels

Table 298: Safe analog input module

2.6.14.2.3 Order data

Model number	Short description	Figure
	Analog input modules	
X20SA4430	X20 safe current input module, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	
X20cSA4430	X20 safe current input module, coated, 2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated, configurable input filter and switching thresholds	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	
	Terminal blocks	
X20TB5F	X20 terminal block, 16-pin, safety-keyed	

Table 299: X20SA4430, X20cSA4430 - Order data

2.6.14.2.4 Technical data

Model number	X20SA4430	X20cSA4430
Short description		
I/O module	2x 2 safe analog inputs, 4 to 20 mA, channels individually galvanically isolated	
General information		
B&R ID code	0xB8B5	0xDD9F
System requirements		
Automation Studio	3.0.81.15 or later	4.0.16 or later
Automation Runtime	3.00 or later	V3.08 or later
SafeDESIGNER	2.81 or later	3.1.0 or later
Safety Release	1.4 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using status LED and software	
Inputs	Yes, using status LED and software	
Blackout mode		
Scope	Module	
Function	Module function	
Standalone mode	No	
Max. I/O cycle time	2 ms	
Power consumption		
Bus	0.25 W	
Internal I/O	1.7 W	
Electrical isolation		
Channel - Bus	Yes	
Channel - Channel	Yes	
Channel pair - Channel pair	Yes	
Certifications		
CE	Yes	
KC	Yes	-
EAC	Yes	-
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013	
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3	
Functional safety	EN 50156-1:2004	
Safety characteristics		
Note	The following characteristic values apply only to the use of input channel pairs. Assessing the channels from a safety point of view when they are used individually is not possible. 1)	
EN ISO 13849-1:2015		
Category	Cat. 4 (SHUNTTEST enabled), Cat. 3 (SHUNTTEST disabled)	
PL	PL e (SHUNTTEST enabled), PL d (SHUNTTEST disabled)	
DC	>94% (regardless of whether SHUNTTEST is enabled or disabled)	
MTTFD	2200 years (regardless of whether SHUNTTEST is enabled or disabled)	
Mission time	Max. 20 years	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 3 (regardless of whether SHUNTTEST is enabled or disabled)	
SFF	>90% (regardless of whether SHUNTTEST is enabled or disabled)	
PFH / PFH_d		
Module	$<1*10^{-9}$ (regardless of whether SHUNTTEST is enabled or disabled)	
openSAFETY wired	Negligible	
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour	
PFD	$<1*10^{-4}$ (regardless of whether SHUNTTEST is enabled or disabled)	
Proof test interval (PT)	20 years	
I/O power supply		
Nominal voltage	24 VDC	

Table 300: X20SA4430, X20cSA4430 - Technical data

Model number	X20SA4430	X20cSA4430
Voltage range	24 VDC -15% / +20%	
Analog inputs		
Input type	Differential input	
Digital converter resolution	24-bit	
Conversion time	See chapter "I/O update time".	
Output format	SAFEINT	
Load	Up to hardware revision D3: 230 to 420 Ω, hardware revision E0 or later: 185 to 245 Ω	
Input protection	Protection against external supply voltages and overcurrent	
Open-circuit detection	Yes, using software	
Permissible input signal		
Voltage	Max. 30.5 V	
Conversion procedure	Sigma-delta	
Max. error at 25°C		
Gain		
0.5 to <4 mA	<0.3% 2)	
4 to 25 mA	<0.08% 2)	
Offset		
0.5 to <4 mA	<2 µA	
4 to 25 mA	<6.3 µA	
Max. gain drift		
0.5 to <4 mA	<1.225 µA/°C	
4 to 25 mA	<1.225 µA/°C	
Max. offset drift		
0.5 to <4 mA	<0.735 µA/°C	
4 to 25 mA	<0.735 µA/°C	
Common-mode rejection		
DC	>70 dB	
50 Hz	>70 dB	
Common-mode range	Between the inputs ±50 V	
Nonlinearity	<0.003%	
Measurement range	Up to firmware version 321: 3.6 to 21 mA, firmware version 322 or later: 0.5 to 25 mA	
Input filter		
Hardware	1st-order low pass / cutoff frequency 500 Hz	
Software	Sinc^3 filter	
Resolution	1 µA/LSB	
Overload detection	Yes, using software	
Test voltage between		
Channel and bus	500 VDC	
To ground	500 VDC	
Channel pair and channel pair	500 VDC	
Safety-related accuracy per channel		
Cat. 3	0.184 mA	
Cat. 4	0.49 mA	
Filter time	Configurable between 1 and 66.7 ms	
Sensor power supply		
Nominal voltage	29 VDC ±5%	
Nominal output current	Max. 60 mA	
Short-circuit proof	Yes, continuous	
Electrical isolation		
Sensor power supply - Channel	No	
Sensor power supply - Sensor power supply	Yes	
Behavior on short circuit	Voltage cutoff	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level	0 to 2000 m, no limitation	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	0 to 60°C	-40 to 60°C 3)
Vertical mounting orientation	0 to 40°C	-40 to 40°C 4)
Derating	See section "Derating".	
Storage	-40 to 85°C	
Transport	-40 to 85°C	

Table 300: X20SA4430, X20cSA4430 - Technical data

Model number	X20SA4430	X20cSA4430
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm	

Table 300: X20SA4430, X20cSA4430 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) Based on the current measured value
3) Up to hardware upgrade <1.10.9.0: -25 to 60°C
4) Up to hardware upgrade <1.10.9.0: -25 to 40°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

Modules next to the X20SA4430 can only have a maximum power consumption of 1 W. Beginning at 50°C (horizontal mounting orientation) and 35°C (vertical mounting orientation), a dummy module must be inserted next to the X20SA4430.

	Number of usable signal pairs
Horizontal mounting orientation up to 50°C	2
Horizontal mounting orientation up to 55°C	1
Vertical mounting orientation up to 35°C	2
Vertical mounting orientation from 35 to 40°C	1

Table 301: Derating in relation to operating temperature and mounting orientation

2.6.14.2.4.1 Safety-oriented measurement precision

The following aspects need to be taken into consideration with regard to the safety-oriented measurement precision of a safe analog input module or temperature module:

- The safety-related precision per channel is specified in the technical data.
- The measurement precision of a signal is the result of: Safety-related precision of the channel + Measurement precision of the sensor + Quality of the signal link of the sensor at the measurement point (depends on the installation)
- From a safety standpoint, a channel pair (i.e. signal pair) must always be observed. The measurement precision acquired for the signal pair must be taken into consideration when specifying the "Limit Threshold Equivalent" parameter. The "Limit Threshold Equivalent" parameter must be set as small as possible, but its value should not fall below the functional measurement precision.
- From a safety point of view, a guaranteed measurement precision per signal pair is the result of:
  ± ("Limit Threshold Equivalent" + Measurement precision of signal)

2.6.14.2.5 LED status indicators

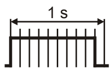
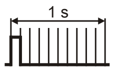
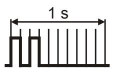
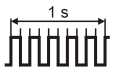
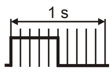
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 4	Input state of the corresponding analog input		
		Red	On	Warning/Error on an input channel
			Blinking	Open circuit on corresponding channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Channel being used and signal OK
			Blinking	Channel outside of the limits configured in SafeDESIGNER
			Off	Channel not used
	12, 34	Input state of the corresponding analog input channel pair		
		Red	On	Warning/Error on this channel pair
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Signal on channel pair OK
			Off	Signal on channel pair not OK
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 302: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.14.2.6 Pinout

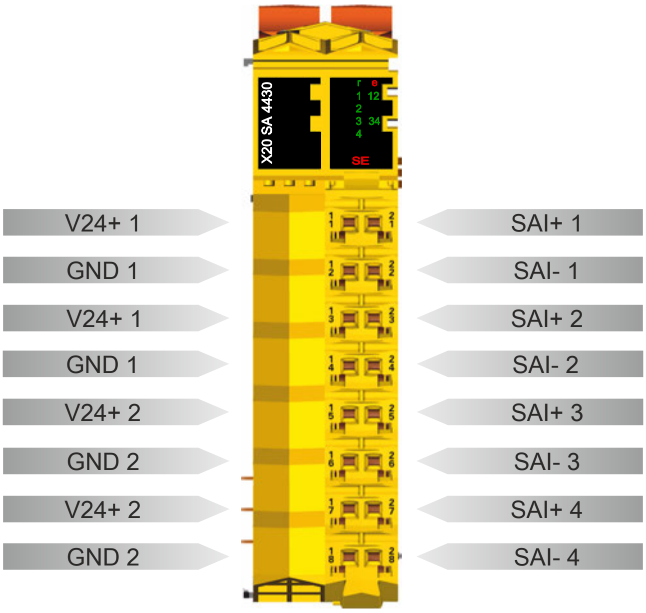


Figure 218: X20SA4430 - Pinout

2.6.14.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods.

The following must be taken into consideration during installation:

- The line resistance must be added to the module's load.
- Make sure that long cables are laid neatly and properly.
- All wiring must be shielded.
- All installed wiring must provide short-circuit protection and voltage disturbance protection (fault exclusion per EN ISO 13849-2:2012, appendix D.2.4, table D.4).

Information:

The analog inputs must be wired; otherwise, the module changes to state "FailSafe".

2.6.14.2.7.1 Channel pair applications with 2 sensors

The following channel pair applications are sufficient to achieve max. PL e (EN ISO 13849-1:2015), max. SIL 3 (EN 62061:2013), max. SIL 3 (IEC 61508:2010) or max. SIL 3 (IEC 61511:2004).

X20SA4430 - 2-wire connection, 2x SIL 2

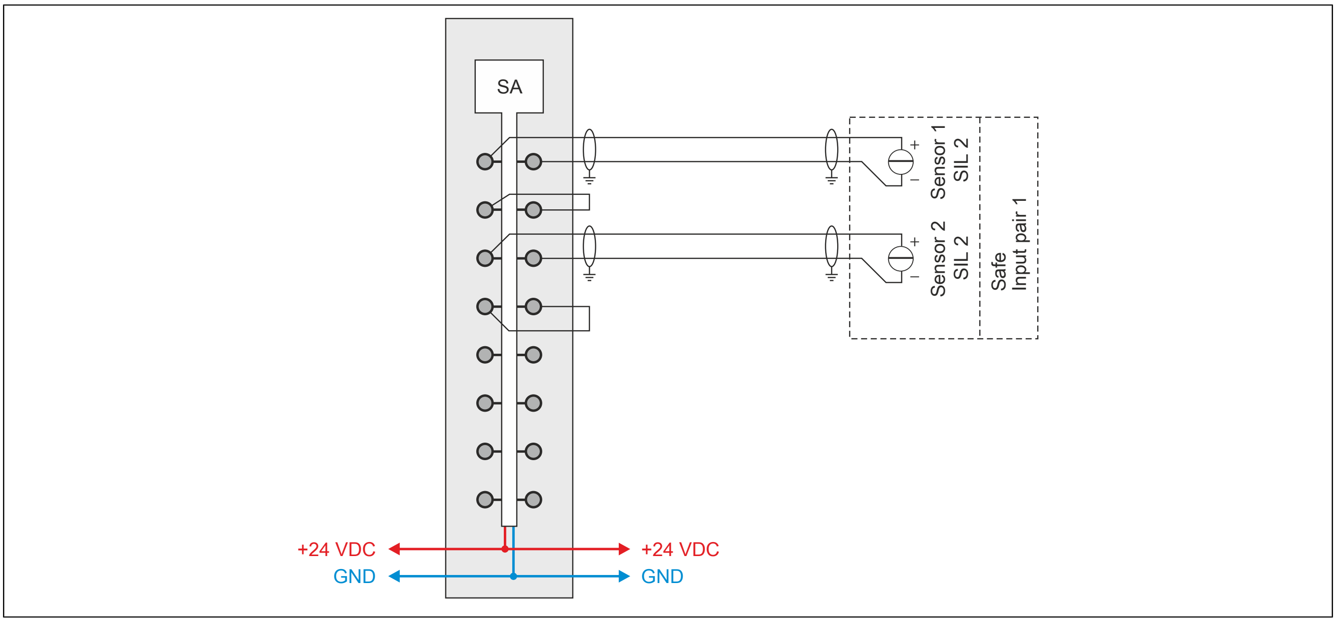


Figure 219: X20SA4430 - 2-wire connection, 2x SIL 2

X20SA4430 - 3-wire connection, 2x SIL 2

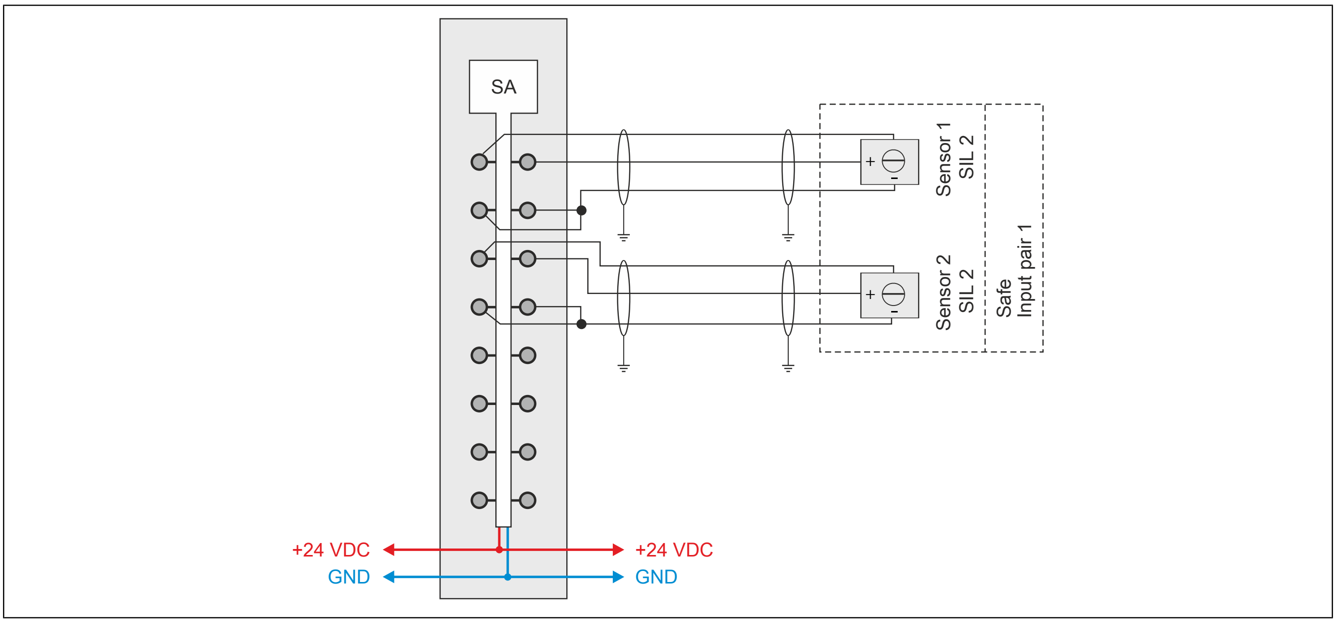


Figure 220: X20SA4430 - 3-wire connection, 2x SIL 2

X20SA4430 - 4-wire connection, 2x SIL 2

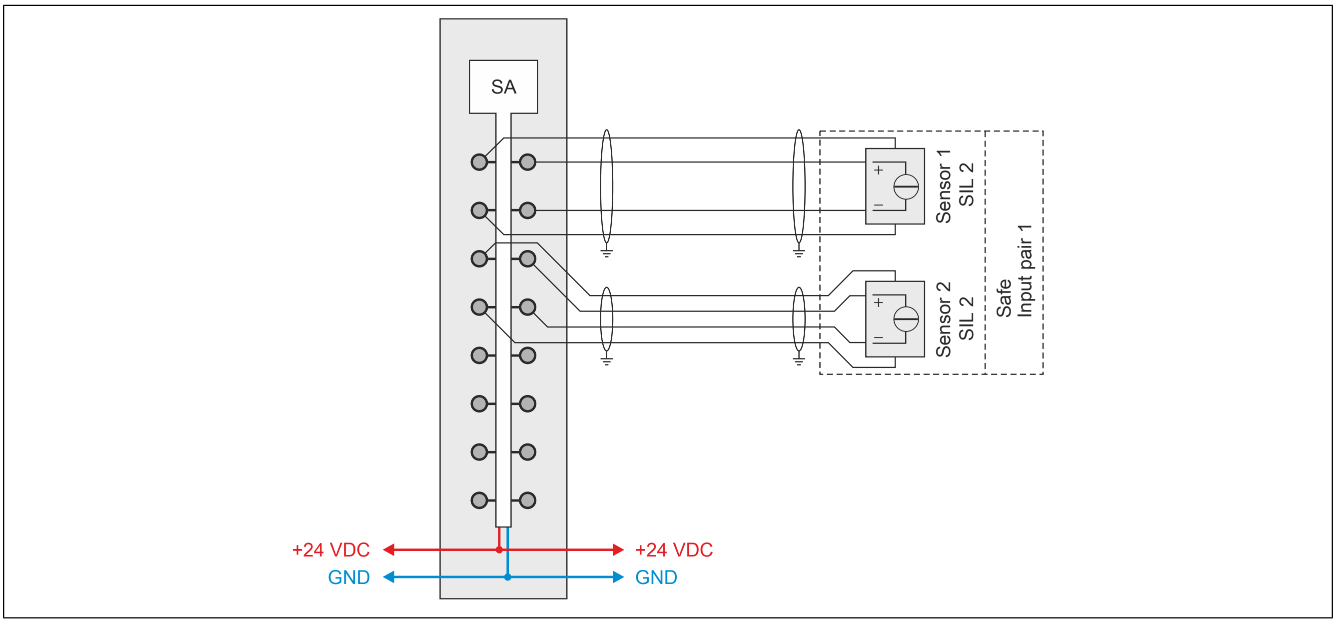


Figure 221: X20SA4430 - 4-wire connection, 2x SIL 2

2.6.14.2.7.2 Channel pair applications with only one sensor

The following channel pair applications are sufficient to achieve max. PL e (EN ISO 13849-1:2015), max. SIL 3 (EN 62061:2013), max. SIL 3 (IEC 61508:2010) or max. SIL 3 (IEC 61511:2004).

X20SA4430 - 2-wire connection, 1x SIL 3

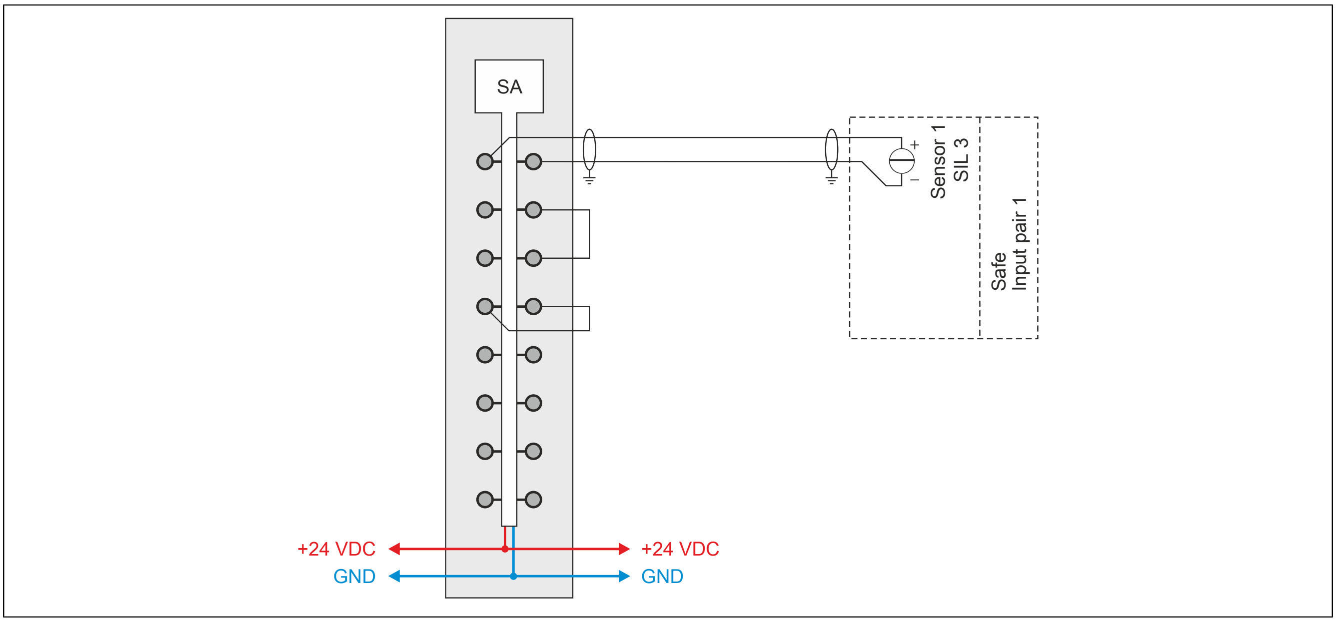


Figure 222: X20SA4430 - 2-wire connection, 1x SIL 3

X20SA4430 - 3-wire connection, 1x SIL 3

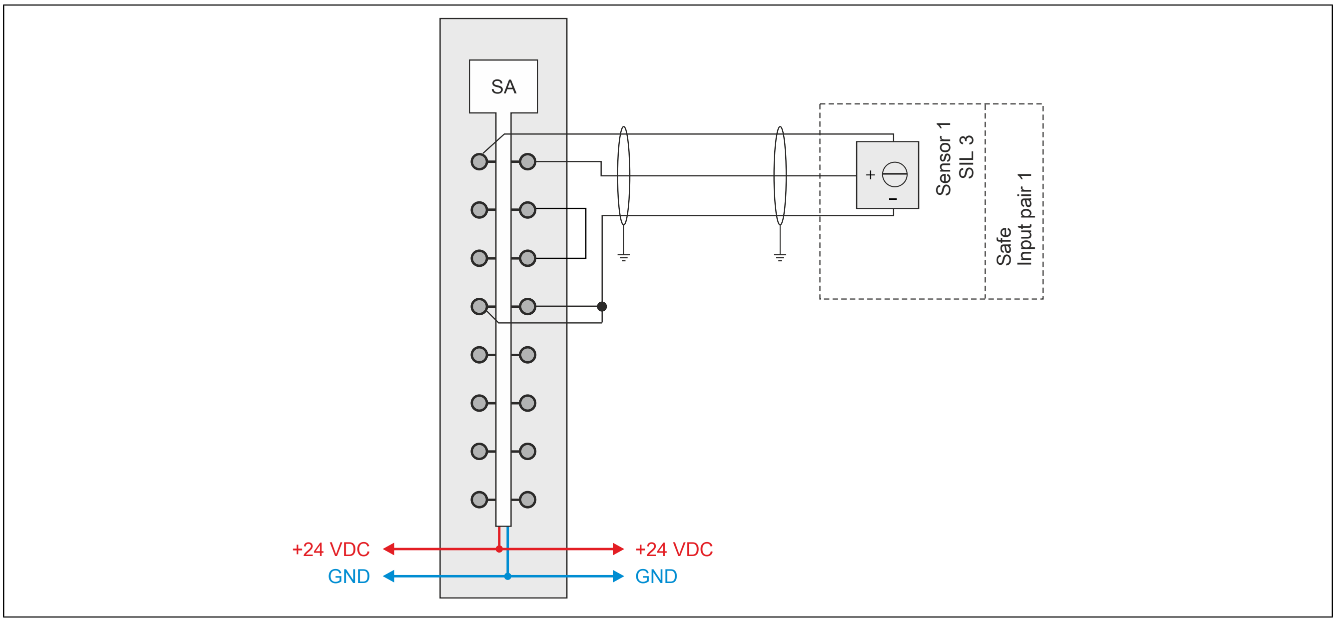


Figure 223: X20SA4430 - 3-wire connection, 1x SIL 3

X20SA4430 - 4-wire connection, 1x SIL 3

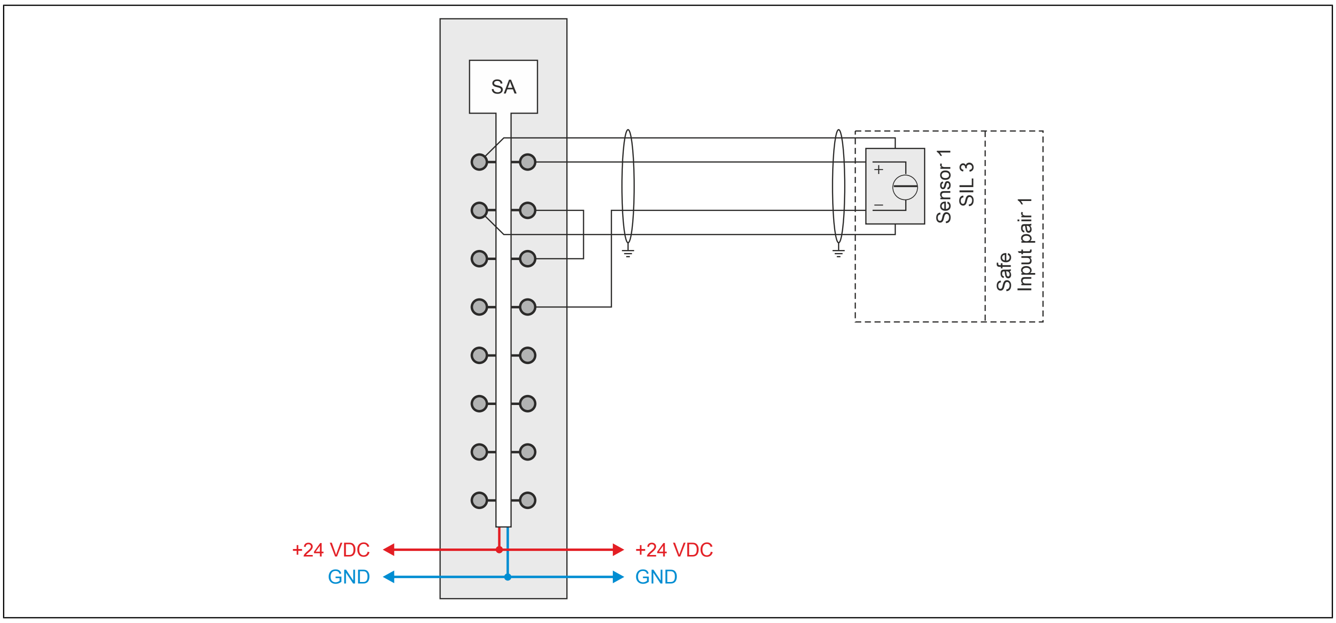


Figure 224: X20SA4430 - 4-wire connection, 1x SIL 3

2.6.14.2.8 Error detection

2.6.14.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.14.2.8.2 Wiring errors

The wiring errors described in the following section are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Errors	Detection	Comment
Open line	Detected	Module switches to the FAILSAFE state
Short circuit between T+ or T- and external 24 V or GND	Not detected	Signal distortion usually does not result due to the electrical isolation of the channels; nevertheless, it is mandatory to use shielded signal lines. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between T+ and T-	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Reverse polarity of T+ and T-	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 303: Error detection for safe inputs of type "Thermocouple"

Errors	Detection	Comment
Open circuit on Sense+ or Sense-	Detected	Channel errors
Short circuit between Sense+, Sense- and external 24 V or GND	Not detected	Signal distortion usually does not result due to the electrical isolation of the channels; nevertheless, it is mandatory to use shielded signal lines. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between Sense+ and Sense-	Detected	Channel errors
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 304: Error detection for safe inputs of type "PT100 / PT1000"

Errors	Detection	Comment
Open line	Detected	Channel errors
Short circuit between signal lines	May not be detected	The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between signal and supply line	May not be detected	The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Reverse polarity of signal lines	Detected	Module switches to the FAILSAFE state
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 305: Error detection for safe inputs of type "Current"

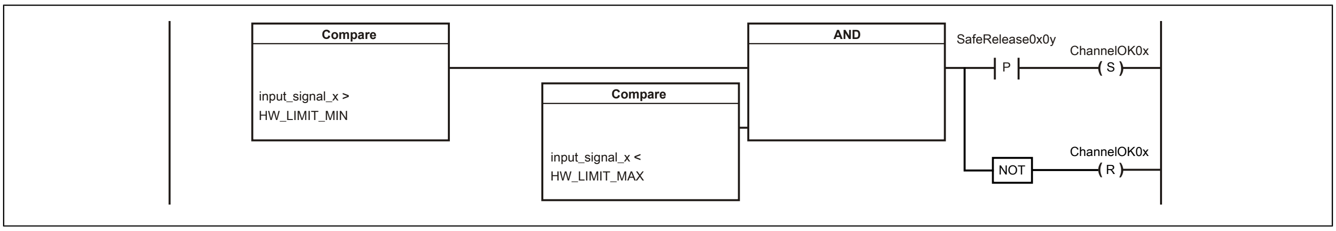
2.6.14.2.8.3 Signal errors

"HW_LIMIT_MIN" designates the lower limit of the measurement range specified in the technical data.
"HW_LIMIT_MAX" designates the upper limit of the measurement range specified in the technical data.

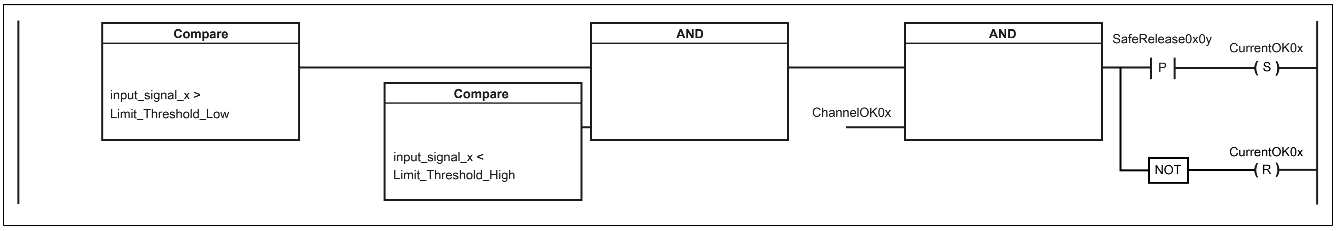
A reset must be performed in order to leave an error state.
For this to be possible, a valid signal must be received at the analog input for the duration of the I/O update time. Then the error can be acknowledged by a rising edge on signal "SafeRelease0x0y".

Signal evaluation takes place in 3 stages:

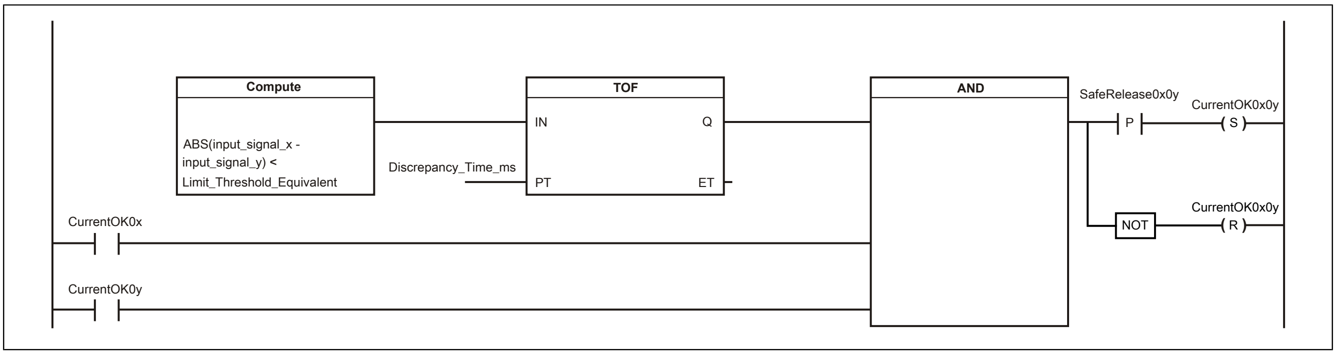
Stage 1: Evaluation of signals against absolute limits



Stage 2: Evaluation of signals against configurable limits



Stage 3: Evaluation of signals against configurable signal pair limits



2.6.14.2.8.4 Channel diagnostics

Channel electronics are automatically tested internally by the module. Here, a test signal is generated in the module and applied to each channel once per hour for a maximum time of 1 s. To avoid signal distortion, the signal value of the channel being tested is frozen during this time.

Only one channel is tested at a time. Per IEC 61508:2010, the module is considered a 1oo2D system for the duration of the channel test. The resulting probability of a dangerous state was taken into account in the safety characteristics in chapter 5.

Up to firmware version 321, the behavior for the duration of channel diagnostics is structured as follows:

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. Since the signal value of the channel being tested is frozen for the duration of channel diagnostics, the arithmetic mean value during this period of channel diagnostics for the safe signal is taken from the frozen value of the diagnosed channel and the signal value of the non-diagnosed channel.

In firmware version 322 and later, the behavior for the duration of channel diagnostics is structured as follows:

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. For the duration of channel diagnostics, however, it is not the arithmetic mean value that is used, but the signal value of the channel that is not currently being diagnosed.
If the behavior of firmware version 321 is desired for compatibility reasons, this can be implemented using parameter "Measurement Result while Testing = Averaged".
An active channel test is indicated by channel "TestActive".

The sequence for channel diagnostics is independent of the firmware version and structured as follows:

		X20SA4430	X20ST4492
Diagnostic Window 1	Hourly	SAI1	TC1, Sense 1
Diagnostic Window 2	Hourly, 15 min after Diagnostic Window 1	SAI3	TC4, Sense 2
Diagnostic Window 3	Hourly, 30 min after Diagnostic Window 1	SAI4	TC3
Diagnostic Window 4	Hourly, 45 min after Diagnostic Window 1	SAI2	TC2

Table 306: Channel diagnostics sequence

In order to meet the stringent requirements of Cat. 4 per EN ISO 13849-1:2015, the shunts of the channel electronics must be tested (shunt test) despite the multi-channel structure. For a proper shunt test, the slew rate of the input signals must be limited to 220 µA/ms.

For steeper signal edges and parameter configuration "Disable Shunttest = No", the module switches to state FAILSAFE if necessary, which affects the entire module. Note that very noisy signal sources or signals with high frequencies likewise result in excessively steep signal edges and can trigger a shunt test error.

Information:

If problems with the slew rate of input signals or shunt test occur, the shunt test can be disabled with parameter "Disable Shunttest = Yes-ATTENTION". In this context, note that the module meets only the requirements of Cat. 3 per EN ISO 13849-1:2015.

2.6.14.2.9 Module function

The safe analog input module is suitable for safely acquiring current signals for safety-related applications up to PL e or SIL 3.

> ## Danger!
>
> **Possible failure of safety function**
> **Dangerous system behavior due to incorrect use of analog signal values**
>
> **When using analog signal values, note the information listed in the data sheet regarding the functionality, precision and scope of the data.**

The current drawn via the input terminals is converted into measurement voltages via shunts 1 and 2, smoothed by the hardware filters (1st-order low pass / cutoff frequency 500 Hz) and digitized in the subsequent A/D converters.

The filter values configured in the software are applied during digitalization in the A/D converter.

The signals then pass through the 3 stages of digital signal processing.

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. At this point, also note the information about channel diagnostics.

The validity of analog signals is represented by the associated status signals. These binary status signals (data type SAFEBOOL) must also be evaluated each time the analog signals are used. A binary status signal with the status FALSE indicates an invalid value in the analog signal. In these situations, the analog signal is no longer permitted to be used for safety-related assessments.

To exit an error state, a reset must be carried out. For this to be possible, a valid signal must be received at the analog input for the duration of the I/O update time. The error can then be acknowledged by a rising edge on signal "SafeRelease0x0y".

An optional sensor power supply is available to provide power to the sensors. If the sensor is supplied externally, the 2-wire connection examples must be used. Current measurement protects the module's internal sensor power supply against overload.

2.6.14.2.10 Input circuit diagram

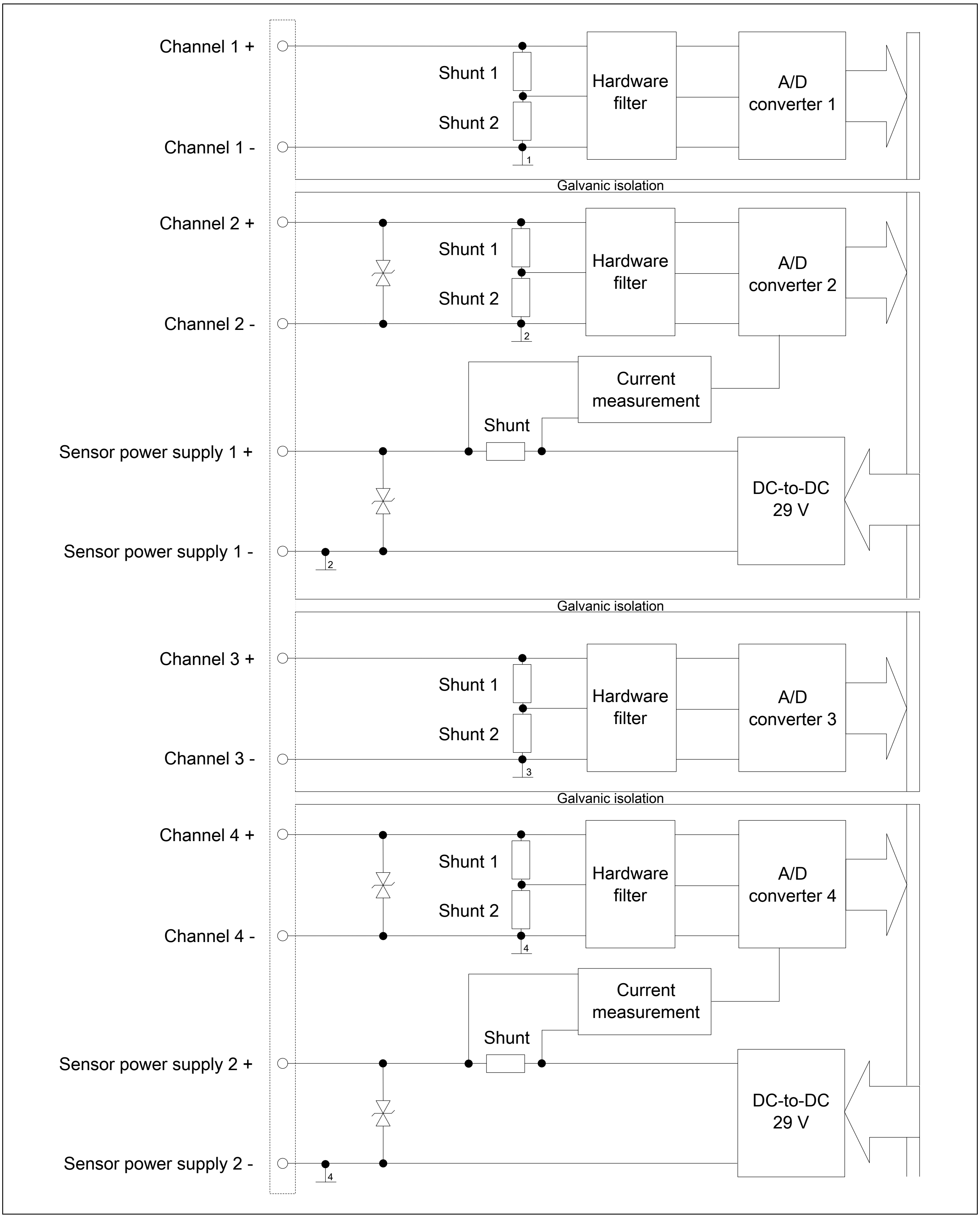


Figure 225: Input circuit diagram

2.6.14.2.11 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 μs

2.6.14.2.12 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

> **Danger!**
>
> **With regard to the I/O update time, an I/O update time of 200 ms should generally be considered for analog input modules with firmware version 301 or lower. The maximum I/O update time is 400 ms.**
>
> **The I/O update time has been optimized in firmware version 302 and later. The optimized times are listed in the table for the maximum I/O update time.**

Configured filter	Maximum I/O update time
1 ms	17 ms
2 ms	19 ms
10 ms	35 ms
16.7 ms	50 ms
20 ms	55 ms
33.3 ms	82 ms
40 ms	95 ms
66.7 ms	122 ms

2.6.14.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.14.2.14 Register description

2.6.14.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 307: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.1.1 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 308: I/O configuration parameters: General

2.6.14.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
<tr><td>Input_Filter_ms</td><td>This parameter sets the filter time of A/D converters.</td><td>1</td><td>ms</td></tr>
<tr><td rowspan="2">Disable_Shunttest</td><td>This parameter can be used to disable automatic testing of the measurement shunts for all of the module's channels. This increases the tolerance of the module in relation to the interference on the input signal.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The automatic testing of the measurement shunts is disabled.</td></tr>
<tr><td>No</td><td>The automatic testing of the measurement shunts is not disabled.</td></tr>
</table>
</td></tr>
</table>

Table 309: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_Shunttest = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements of Cat. 4 per EN ISO 13849-1:2015.
As a result, the module meets the requirements up to max. Cat. 3 per EN ISO 13849-1:2015.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 310: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeCurrentxxyy

Parameter	Description	Default value	Unit
Limit_Threshold_High_1, Limit_Threshold_High_2, Limit_Threshold_High_3, Limit_Threshold_High_4	This parameter specifies the current maximum permissible analog input value. • Permissible values: 3600 to 21,000 µA (corresponds to 3.6 to 21 mA)	20000	µA
Limit_Threshold_Low_1, Limit_Threshold_Low_2, Limit_Threshold_Low_3, Limit_Threshold_Low_4	This parameter specifies the current minimum permissible analog input value. • Permissible values: 3600 to 21,000 µA (corresponds to 3.6 to 21 mA)	4000	µA
Limit_Threshold_Equivalent_1, Limit_Threshold_Equivalent_2, Limit_Threshold_Equivalent_3, Limit_Threshold_Equivalent_4	This parameter specifies the maximum permissible deviation between the analog input values. • Permissible values: 0 to 21,000 µA (corresponds to 0 to 21 mA)	20000	µA
Discrepancy_Time_1_ms, Discrepancy_Time_2_ms, Discrepancy_Time_3_ms, Discrepancy_Time_4_ms	This parameter specifies the maximum time for the "Dual-channel evaluation" function in which the difference between both analog input values is permitted to exceed the limit value. • Permissible values: 0 to 10,000 ms (corresponds to 0 to 10 s)	0	ms

Table 311: SafeDESIGNER parameters: SafeCurrentxxyy

Parameters "Limit_Threshold_High_x", "Limit_Threshold_Low_x", "Limit_Threshold_Equivalent_x" and "Discrepancy_Time_x_ms" together make up a parameter set. The channels "SafeThrSelector_xxyy_Bit1" and "SafeThrSelector_xxyy_Bit2" are available in the SafeDESIGNER application to determine which parameter set in the module is enabled, i.e. it is possible to change the parameter set at runtime.

2.6.14.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 312: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 313: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Input Filter</td><td>This parameter sets the filter time of A/D converters.
Permissible values: 1 ms, 2 ms, 10 ms, 16.7 ms, 20 ms, 33.3 ms, 40 ms, 66.7 ms</td><td>1</td><td>ms</td></tr>
<tr><td rowspan="2">Disable Shunttest</td><td>This parameter can be used to disable automatic testing of the measurement shunts for all of the module's channels. This increases the tolerance of the module in relation to the interference on the input signal.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the measurement shunts is disabled ("Yes-ATTENTION" = SHUNTTEST disabled).</td></tr>
<tr><td>No</td><td>Automatic testing of the measurement shunts is not disabled ("No" = SHUNTTEST enabled).</td></tr>
</table></td></tr>
<tr><td rowspan="2">Measurement Result while Testing</td><td>This parameter enables the signal behavior specified prior to firmware version 321 for the duration of signal diagnostics (see chapter "Channel diagnostics").</td><td>Single channel</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Averaged</td><td>During testing, the safe analog signal results from the mean value of the individual signals.</td></tr>
<tr><td>Single channel</td><td>During testing, the safe analog signal corresponds to the individual signal of the channel that is not currently being diagnosed.</td></tr>
</table></td></tr>
</table>

Table 314: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable Shunttest = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements of Cat. 4 per EN ISO 13849-1:2015.
As a result, the module meets the requirements up to max. Cat. 3 per EN ISO 13849-1:2015.

Group: SafeCurrentxxyy

Parameter	Description	Default value	Unit
Limit Threshold High 1, Limit Threshold High 2, Limit Threshold High 3, Limit Threshold High 4	This parameter specifies the current maximum permissible analog input value. • Permissible values: 500 to 25,000 µA (corresponds to 0.5 to 25 mA) (up to hardware upgrade 1.10.1.0: 3600 to 21,000 µA - corresponds to 3.6 to 21 mA)	20000	µA
Limit Threshold Low 1, Limit Threshold Low 2, Limit Threshold Low 3, Limit Threshold Low 4	This parameter specifies the current minimum permissible analog input value. • Permissible values: 500 to 25,000 µA (corresponds to 0.5 to 25 mA) (up to hardware upgrade 1.10.1.0: 3600 to 21,000 µA - corresponds to 3.6 to 21 mA)	4000	µA
Limit Threshold Equivalent 1, Limit Threshold Equivalent 2, Limit Threshold Equivalent 3, Limit Threshold Equivalent 4	This parameter specifies the maximum permissible deviation between the analog input values. • Permissible values: 0 to 25,000 µA (corresponds to 0 to 25 mA) (up to hardware upgrade 1.10.1.0: 0 to 21,000 µA - corresponds to 0 to 21 mA)	100	µA
Discrepancy Time 1, Discrepancy Time 2, Discrepancy Time 3, Discrepancy Time 4	This parameter specifies the maximum time for the "Dual-channel evaluation" function in which the difference between both analog input values is permitted to exceed the limit value. • Permissible values: 0 to 10,000 ms (corresponds to 0 to 10 s)	0	ms

Table 315: SafeDESIGNER parameters: SafeCurrentxxyy

The parameters "Limit Threshold High x", "Limit Threshold Low x", "Limit Threshold Equivalent x" and "Discrepancy Time x" together make up a parameter set. The channels "SafeThrSelector_xxyy_Bit1" and "SafeThrSelector_xxyy_Bit2" are available in the SafeDESIGNER application to determine which parameter set in the module is enabled, i.e. it is possible to change the parameter set at runtime.

2.6.14.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read)[1)]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read)[1)]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>SafeChannelOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel xx</td></tr>
<tr><td>SafeCurrentOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of current range evaluation of channel xx</td></tr>
<tr><td>SafeCurrentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel current evaluation of channel xxyy</td></tr>
<tr><td>TestActive</td><td>Read</td><td>Read</td><td>BOOL</td><td>Indication of an active channel test</td></tr>
<tr><td>EquivalentThresholdxxyy</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Limit value "Limit Threshold Equivalent" currently in use (see "SafeDESIGNER parameters: SafeCurrentxxyy")</td></tr>
<tr><td>DiscrepanceTimeThresholdxxyy</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Limit value "Discrepancy Time" currently in use (see "SafeDESIGNER parameters: SafeCurrentxxyy")</td></tr>
<tr><td>SafeCurrentxxyy</td><td>Read</td><td>Read</td><td>SAFEINT</td><td>(Current channel xx + Current channel yy)/2
<table>
<tr><th>Values</th><th>Input signal</th></tr>
<tr><td>0 to 20000</td><td>Current signal 0 to 20 mA</td></tr>
</table></td></tr>
</table>

Table 316: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>Currentxx</td><td>Read</td><td>Read</td><td>INT</td><td>Current channel xx
<table><tr><th>Values</th><th>Input signal</th></tr><tr><td>0 to 20000</td><td>Current signal 0 to 20 mA</td></tr></table></td></tr>
<tr><td>SafeThrSelector_xxyy_Bit1</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td rowspan="2"><table><tr><th>**_Bit1</th><th>**_Bit2</th><th>Parameters currently being used</th></tr><tr><td>0</td><td>0</td><td>Parameter set 1</td></tr><tr><td>1</td><td>0</td><td>Parameter set 2</td></tr><tr><td>0</td><td>1</td><td>Parameter set 3</td></tr><tr><td>1</td><td>1</td><td>Parameter set 4</td></tr></table></td></tr>
<tr><td>SafeThrSelector_xxyy_Bit2</td><td>-</td><td>Write</td><td>SAFEBOOL</td></tr>
<tr><td>SafeReleasexxyy</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Release signal - Channel xxyy</td></tr>
</table>

Table 316: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

The validity of analog signals is represented by the associated status signals. These binary status signals (data type SAFEBOOL) must also be evaluated each time the analog signals are used. A binary status signal with the status FALSE indicates an invalid value in the analog signal. When this happens, the analog signal is no longer permitted to be used for safety-related assessments.

2.6.15 Temperature measurement modules

2.6.15.1 Overview

Model number	Short description	Page
X20ST4492	X20 safe temperature input module, 2x 2 safe analog inputs for thermocouples, Type: J, K, N, S, R, C, T, resolution 0.1°C, 1x 2 safe analog inputs for PT100/PT1000 sensors, channel pairs galvanically isolated, integrated compensation of terminal temperature, integrated temperature sensor in terminal block X20TB5E, configurable input filter and switching thresholds	684

2.6.15.2 X20ST4492

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 317: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 318: Organization of general notices

2.6.15.2.1 General information

The module is equipped with 2 safe analog input pairs for J, K, N, S, R, C and T thermocouple sensors and 1 safe analog input pair for PT100/PT1000 resistance temperature measurement.

The safe temperature module is suitable for safely acquiring temperatures for safety-related applications up to PL e or SIL 3.

This module is designed for X20 16-pin terminal blocks.

- 2 safe analog input pairs for thermocouples
- For sensor types J, K, N, S, R, C, T, raw value measurement
- 1 safe analog input pair for resistance temperature measurement
- For PT100 and PT1000
- Configurable sensor type per channel
- 24-bit digital converter resolution
- Galvanically isolated analog channel pairs
- Input filter configurable
- Integrated terminal temperature compensation
- 2x PT1000 sensor integrated in the terminal (X20TB5E)
- 2x external PT1000 sensor can be connected (X20TB5F)

2.6.15.2.1.1 Function

Safe temperature measurement

This safe temperature module is suitable for safely connecting PT100, PT1000 or thermocouples for safety-related applications up to PL e or SIL 3.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

2.6.15.2.2 Overview

Module	X20ST4492
Thermocouple inputs	
Number of inputs	2x 2 thermocouple inputs
Measurement range	-270.0 to 1768.0 °C
Sensor	Fe-CuNi: Type J NiCr-Ni: Type K NiCrSi-NiSi: Type N PtRh10-Pt: Type S PtRh13-Pt: Type R WRe5-WRe26: Type C Cu-CuNi: Type T Sensor specification in accordance with EN IEC 60584-1:2010
Voltage measurement	Yes: ±65 mV
Terminal temperature compensation	Yes: 1x 2 PT100/PT1000 inputs available on module
Digital converter resolution	24-bit
PT100/PT1000 inputs	
Number of inputs	1x 2 PT100/PT1000 inputs, e.g. for terminal temperature compensation
Measurement range	Firmware version 295: -40.0 to 130.0°C, firmware version 301 or later: -200.0 to 850.0°C
Sensor	PT100 PT1000
Measurement type	2-wire measurement
Digital converter resolution	24-bit

Table 319: Safe temperature module

2.6.15.2.3 Order data

Model number	Short description	Figure
	Analog input modules	
X20ST4492	X20 safe temperature input module, 2x 2 safe analog inputs for thermocouples, Type: J, K, N, S, R, C, T, resolution 0.1°C, 1x 2 safe analog inputs for PT100/PT1000 sensors, channel pairs galvanically isolated, integrated compensation of terminal temperature, integrated temperature sensor in terminal block X20TB5E, configurable input filter and switching thresholds	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
	Terminal blocks	
X20TB5E	X20 terminal block, 16-pin, safety-keyed, 2x PT1000 integrated for terminal temperature compensation	
X20TB5F	X20 terminal block, 16-pin, safety-keyed	

Table 320: X20ST4492 - Order data

2.6.15.2.4 Technical data

Model number	**X20ST4492**
Short description	
I/O module	2x 2 safe analog inputs for thermocouples, 1x 2 safe analog inputs for PT100/ PT1000 sensors, channel pairs galvanically isolated, integrated terminal temperature compensation, integrated temperature sensor in terminal block X20TB5E
General information	
B&R ID code	0xB419
System requirements	
Automation Studio	3.0.81.15 or later
Automation Runtime	3.00 or later
SafeDESIGNER	2.81 or later
Safety Release	1.4 or later
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using status LED and software
Inputs	Yes, using status LED and software
Blackout mode	
Scope	Module
Function	Module function
Standalone mode	No
Max. I/O cycle time	2 ms
Power consumption	
Bus	0.25 W
Internal I/O	1.2 W
Electrical isolation	
Channel - Bus	Yes
Channel - Channel	No
Channel pair - Channel pair	Yes
Certifications	
CE	Yes
KC	Yes
EAC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3
Functional safety	EN 50156-1:2004
Safety characteristics	
Note	The following characteristic values apply only to the use of input channel pairs. Assessing the channels from a safety point of view when they are used individually is not possible. 1)
EN ISO 13849-1:2015	
Category	Cat. 4
PL	PL e
DC	>94%
MTTFD	2200 years
Mission time	Max. 20 years
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
SIL CL	SIL 3
SFF	>90%
PFH / PFH_d	
Module	$<1*10^{-9}$
openSAFETY wired	Negligible
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour
PFD	$<1*10^{-4}$
Proof test interval (PT)	20 years

Table 321: X20ST4492 - Technical data

Model number	X20ST4492
I/O power supply	
Nominal voltage	24 VDC
Voltage range	24 VDC -15% / +20%
Thermocouple temperature inputs	
Input	Thermocouple
Digital converter resolution	24-bit
Filter time	Configurable between 1 and 66.7 ms
Output format	SAFEINT
Measurement range	
Sensor temperature	
Type J: Fe-CuNi	-210.0 to 1200.0°C
Type K: NiCr-Ni	-270.0 to 1372.0°C
Type N: NiCrSi-NiSi	-270.0 to 1300.0°C
Type S: PtRh10-Pt	-50.0 to 1768.0°C
Type R: PtRh13-Pt	-50.0 to 1768.0°C
Type C: WRe5-WRe26	0 to 2320.0°C
Type T: Cu-CuNi	-270.0 to 400.0°C
Voltage	±65 mV
Max. internal resistance of source during voltage measurement	20 Ω
Terminal temperature compensation	Internal / External
Sensor standard	EN 60584
Resolution	
Sensor temperature	1 LSB = 0.1°C
Voltage	1 LSB = 2 µV
Conversion procedure	Sigma-delta
Linearization method	Internal
Permissible input signal	Max. ±1 V
Input filter	1st-order low pass / cutoff frequency 500 Hz
Basic accuracy [2)]	
Type J	0.10%
Type K	0.11%
Type N	0.11%
Type S	0.17%
Type R	0.17%
Type C	0.15%
Type T	0.11%
Voltage	0.06%
Max. gain drift [3)]	0.013 %/°C
Max. offset drift [4)]	
Type J	0.0021 %/°C
Type K	0.0026 %/°C
Type N	0.0030 %/°C
Type S	0.0090 %/°C
Type R	0.0080 %/°C
Type C	0.0046 %/°C
Type T	0.0050 %/°C
Voltage	0.0013 %/°C
Terminal temperature compensation	
Accuracy of internal terminal temperature	15°C at static temperatures and during safe operation
Common-mode rejection	
DC	>70 dB
50 Hz	>70 dB
Common-mode range	±4 V within channel pair, ±50 V between 2 channel pairs
Crosstalk between channels	≤70 dB
Isolation voltage between channel and bus	500 VDC
Safety-related accuracy per channel [4)]	
Type J	2.5%
Type K	2.9%
Type N	3.3%
Type S	8.3%
Type R	7.4%
Type C	4.8%
Type T	4.6%
Voltage	1.6%
Resistance measurement temperature inputs	
Measurement range	
PT100	Firmware version 295: -40.0 to 130.0°C, firmware version 301 or later: -200.0 to 850.0°C
PT1000	Firmware version 295: -40.0 to 130.0°C, firmware version 301 or later: -200.0 to 850.0°C
Basic accuracy [2)]	
PT100	1.1%
PT1000	0.3%
Measurement current	262 µA ±5%

Table 321: X20ST4492 - Technical data

Model number	X20ST4492
Max. gain drift 3)	0.004 %/°C
Max. offset drift	
PT100	0.03 %/°C
PT1000	0.003 %/°C
Temperature sensor resolution	
PT100	1 LSB = 0.1°C
PT1000	1 LSB = 0.1°C
Input filter	
Cutoff frequency	500 Hz 1st-order
Max. line length	50 m
Max. line resistance	5 Ω
Safety-related accuracy per channel 4)	
PT100	4%
PT1000	2%
Operating conditions	
Mounting orientation	
Horizontal	Yes
Vertical	Yes
Installation elevation above sea level	0 to 2000 m, no limitation
Degree of protection per EN 60529	IP20
Ambient conditions	
Temperature	
Operation	
Horizontal mounting orientation	0 to 60°C
Vertical mounting orientation	0 to 50°C
Derating	See section "Derating".
Storage	-40 to 85°C
Transport	-40 to 85°C
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.
Spacing	$25^{+0.2}$ mm

Table 321: X20ST4492 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) Based on the entire measurement range at 25°C
3) Based on the measured value
4) Based on the entire measurement range

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

Starting at a temperature of 55°C (horizontal mounting orientation), dummy modules must be connected next to the X20ST4492.

Module	X20ST4492
Derating bonus	
Dummy module on the left	+0°C
Dummy module on the right	+2.5°C
Dummy module on the left and right	+5°C

Table 322: Derating bonus

2.6.15.2.4.1 Safety-oriented measurement precision

The following aspects need to be taken into consideration with regard to the safety-oriented measurement precision of a safe analog input module or temperature module:

- The safety-related precision per channel is specified in the technical data.
- The measurement precision of a signal is the result of: Safety-related precision of the channel + Measurement precision of the sensor + Quality of the signal link of the sensor at the measurement point (depends on the installation)
- From a safety standpoint, a channel pair (i.e. signal pair) must always be observed. The measurement precision acquired for the signal pair must be taken into consideration when specifying the "Limit Threshold Equivalent" parameter. The "Limit Threshold Equivalent" parameter must be set as small as possible, but its value should not fall below the functional measurement precision.
- From a safety point of view, a guaranteed measurement precision per signal pair is the result of:
  ± ("Limit Threshold Equivalent" + Measurement precision of signal)
- On input channels for PT100/PT1000 sensors, the line resistance must be taken into consideration from a safety point of view.
- On input channels for thermocouples, the measurement precision of the signal for the terminal temperature must also be added.
- When using the X20TB5E terminal, the measurement precision of the signal for the terminal temperature is specified in the technical data.

2.6.15.2.5 LED status indicators

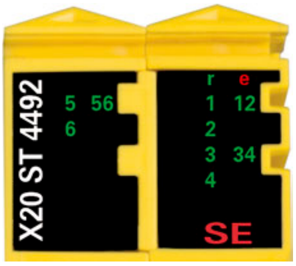
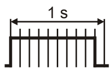
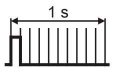
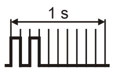
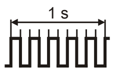
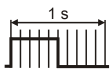
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 to 6	Input state of the corresponding analog input		
		Red	On	Warning/Error on the input channel
			Blinking	Open circuit on corresponding channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Channel being used and signal OK
			Blinking	Channel outside of the limits configured in SafeDESIGNER
			Off	Channel not used
	12, 34, 56	Input state of the corresponding analog input channel pair		
		Red	On	Warning/Error on this channel pair
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Signal on channel pair OK
			Off	Signal on channel pair not OK
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 323: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.15.2.6 Pinout

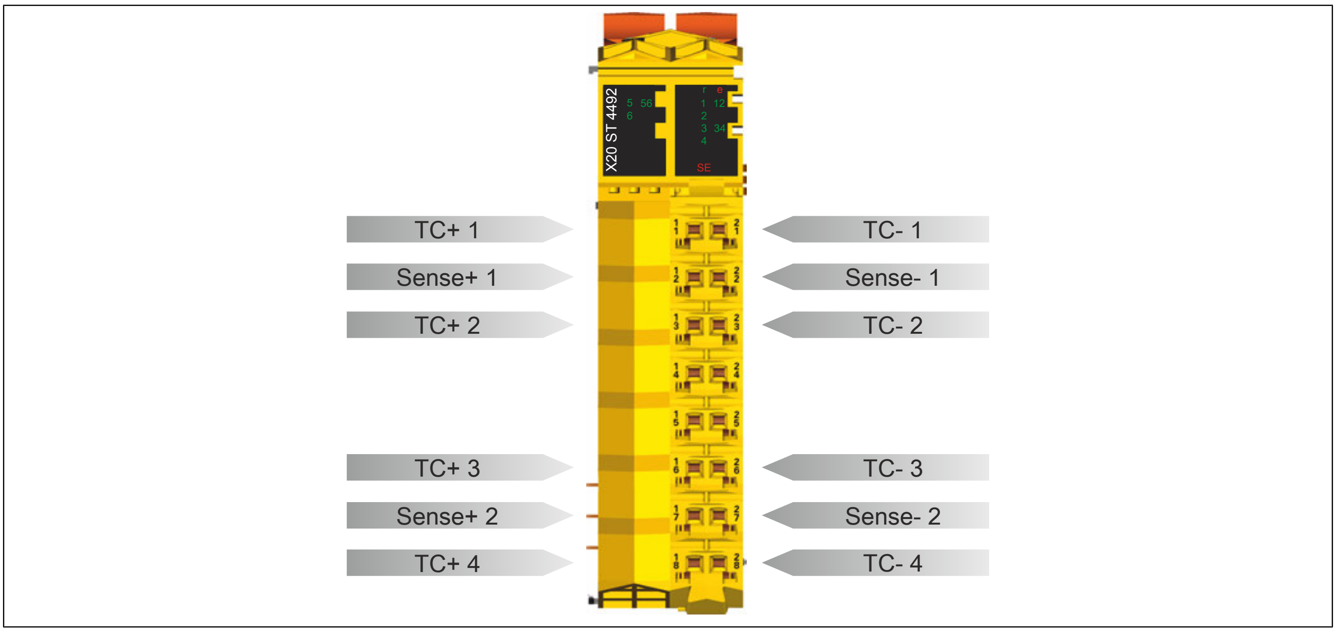


Figure 226: X20ST4492 - Pinout

2.6.15.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods.

The following must be taken into consideration during installation:

- The maximum permissible cable length is 50 m.
- The maximum resistance per stranded wire is 5 ohms.
- All wiring must be shielded.
- All installed wiring must provide short-circuit protection and voltage disturbance protection (fault exclusion per EN ISO 13849-2:2012, appendix D.2.4, table D.4).
- The wiring for the PT100/PT1000 channels must be installed in such a way that contact resistances do not change. This is because they must be included in the safety-related measurement precision (see section "Safety-oriented measurement precision").

Information:

The thermocouple inputs must be wired, otherwise, the module switches to the "FailSafe" state.

2.6.15.2.7.1 Channel pair applications

The following channel pair applications are sufficient to achieve max. PL e (EN ISO 13849-1:2015), max. SIL 3 (EN 62061:2013), max. SIL 3 (IEC 61508:2010) or max. SIL 3 (IEC 61511:2004).

X20ST4492 - Safe thermocouple input pair with X20TB5E for acquiring terminal temperature

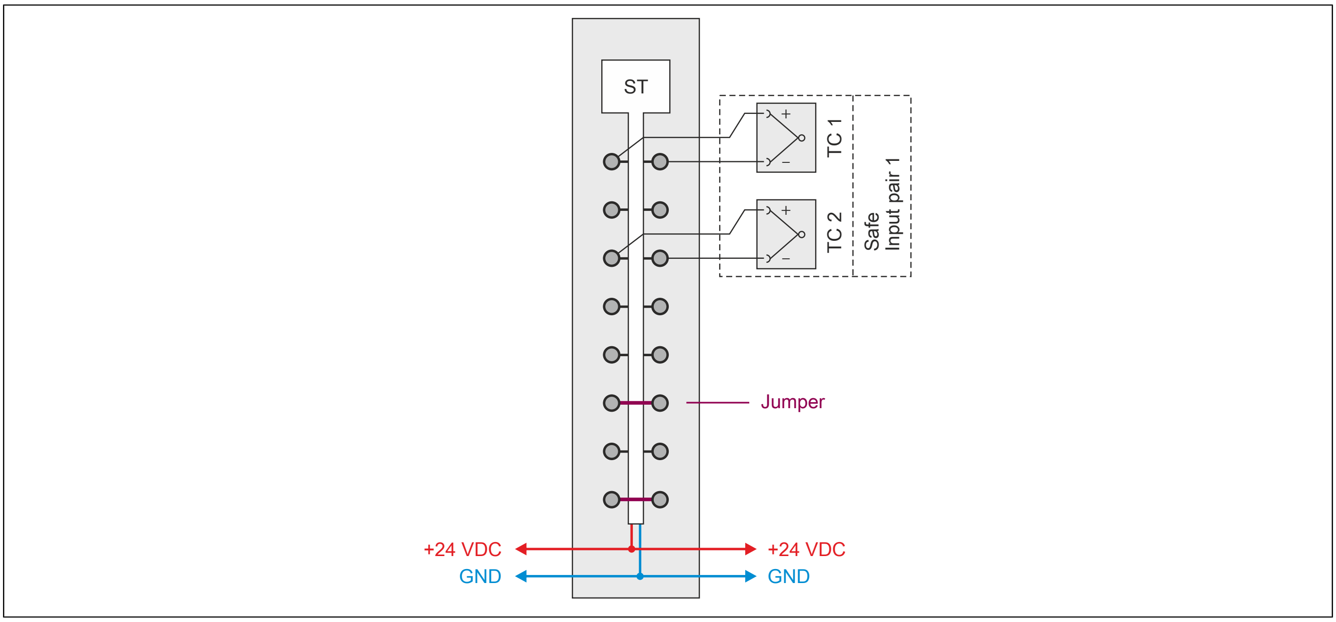


Figure 227: X20ST4492 - Safe thermocouple input pair with X20TB5E for acquiring terminal temperature

X20ST4492 - Safe thermocouple input pair, remote terminal temperature compensation, PT100/PT1000 2-wire connection

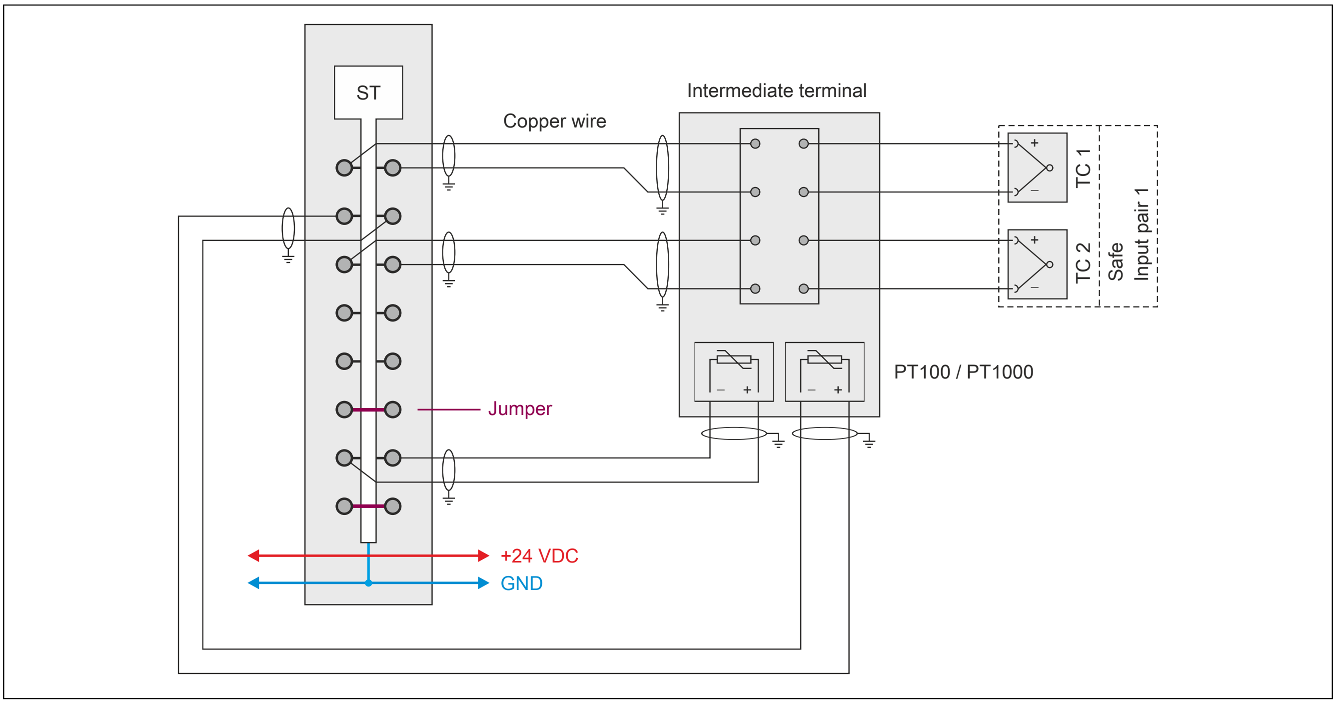


Figure 228: X20ST4492 - Safe thermocouple input pair, remote terminal temperature compensation, PT100/PT1000 2-wire connection

X20ST4492 - Safe PT100/PT1000 input pair, 2-wire connection

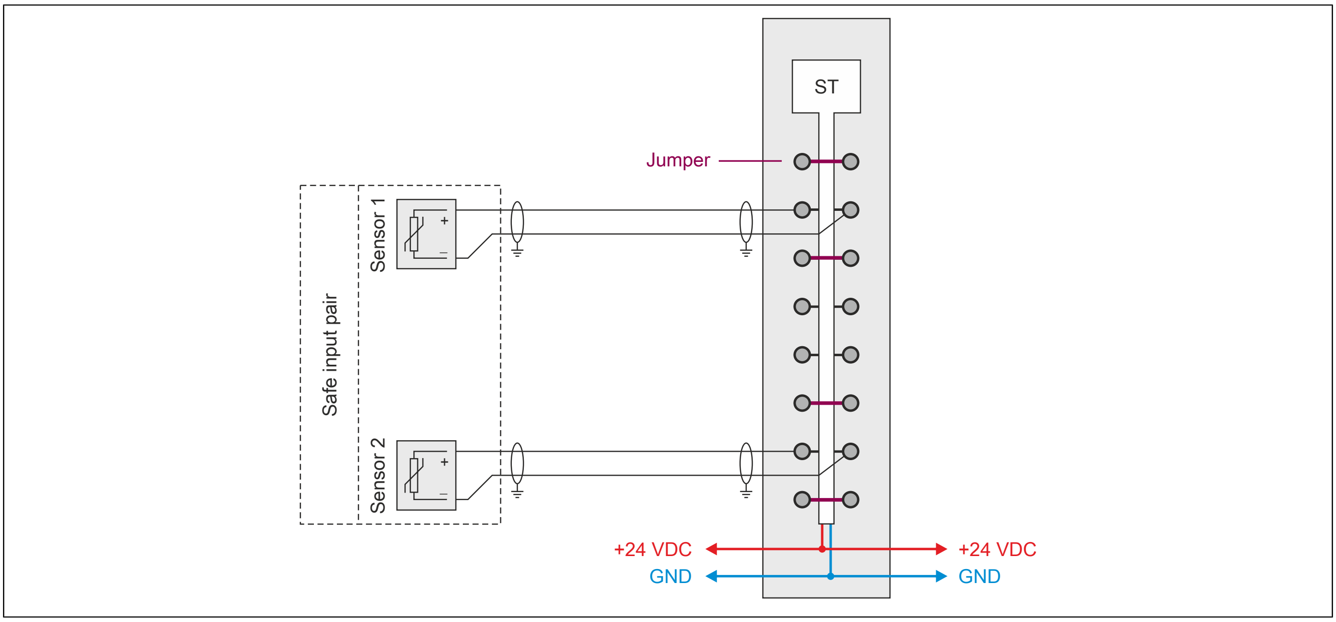


Figure 229: X20ST4492 - Safe PT100/PT1000 input pair, 2-wire connection

2.6.15.2.8 Error detection

2.6.15.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.15.2.8.2 Wiring errors

The wiring errors described in the following section are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Errors	Detection	Comment
Open line	Detected	Module switches to the FAILSAFE state
Short circuit between T+ or T- and external 24 V or GND	Not detected	Signal distortion usually does not result due to the electrical isolation of the channels; nevertheless, it is mandatory to use shielded signal lines. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between T+ and T-	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Reverse polarity of T+ and T-	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 324: Error detection for safe inputs of type "Thermocouple"

Errors	Detection	Comment
Open circuit on Sense+ or Sense-	Detected	Channel errors
Short circuit between Sense+, Sense- and external 24 V or GND	Not detected	Signal distortion usually does not result due to the electrical isolation of the channels; nevertheless, it is mandatory to use shielded signal lines. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between Sense+ and Sense-	Detected	Channel errors
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 325: Error detection for safe inputs of type "PT100 / PT1000"

Errors	Detection	Comment
Open line	Detected	Channel errors
Short circuit between signal lines	May not be detected	The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Short circuit between signal and supply line	May not be detected	The user must take appropriate measures to ensure that this error does not lead to a safety-critical state. Signal and supply lines must be installed in such a way that fault exclusion is possible per EN ISO 13849-2:2012, table D.5.
Reverse polarity of signal lines	Detected	Module switches to the FAILSAFE state
Disturbance voltage	Not detected	This error results in signal distortion that may be detected by dual-channel evaluation in some circumstances. Shielded cables are mandatory for all signal lines. Different installation paths must be used for the cabling of both signals of the signal pair. The user must take appropriate measures to ensure that this error does not lead to a safety-critical state.

Table 326: Error detection for safe inputs of type "Current"

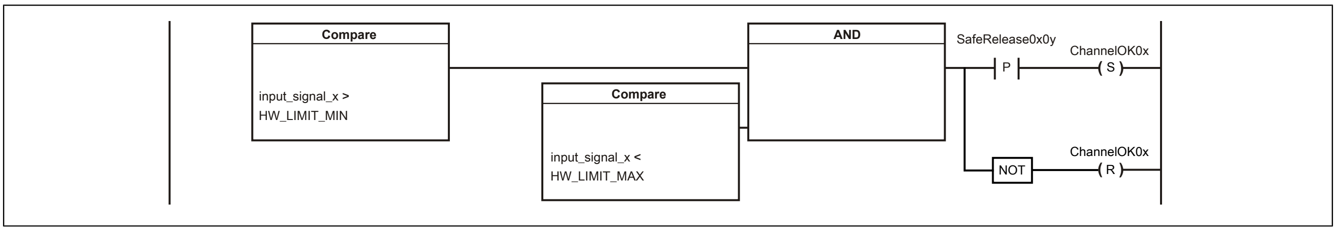
2.6.15.2.8.3 Signal errors

"HW_LIMIT_MIN" designates the lower limit of the measurement range specified in the technical data.
"HW_LIMIT_MAX" designates the upper limit of the measurement range specified in the technical data.

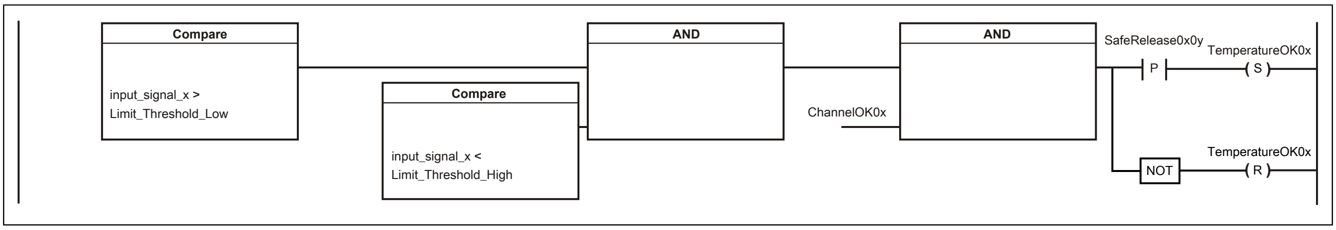
A reset must be performed in order to leave an error state.
For this to be possible, a valid signal must be received at the analog input for the duration of the I/O update time. Then the error can be acknowledged by a rising edge on signal "SafeRelease0x0y".

Signal evaluation takes place in 3 stages:

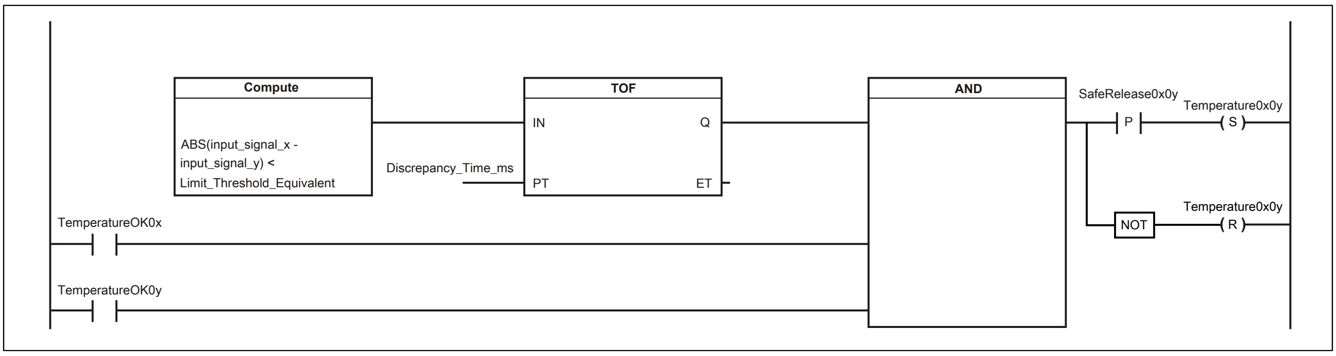
Stage 1: Evaluation of signals against absolute limits



Stage 2: Evaluation of signals against configurable limits



Stage 3: Evaluation of signals against configurable signal pair limits



2.6.15.2.8.4 Channel diagnostics

Channel electronics are automatically tested internally by the module. Here, a test signal is generated in the module and applied to each channel once per hour for a maximum time of 1 s. To avoid signal distortion, the signal value of the channel being tested is frozen during this time.

Only one channel is tested at a time. Per IEC 61508:2010, the module is considered a 1oo2D system for the duration of the channel test. The resulting probability of a dangerous state was taken into account in the safety characteristics in chapter 5.

Up to firmware version 321, the behavior for the duration of channel diagnostics is structured as follows:

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. Since the signal value of the channel being tested is frozen for the duration of channel diagnostics, the arithmetic mean value during this period of channel diagnostics for the safe signal is taken from the frozen value of the diagnosed channel and the signal value of the non-diagnosed channel.

In firmware version 322 and later, the behavior for the duration of channel diagnostics is structured as follows:

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. For the duration of channel diagnostics, however, it is not the arithmetic mean value that is used, but the signal value of the channel that is not currently being diagnosed.
If the behavior of firmware version 321 is desired for compatibility reasons, this can be implemented using parameter "Measurement Result while Testing = Averaged".
An active channel test is indicated by channel "TestActive".

The sequence for channel diagnostics is independent of the firmware version and structured as follows:

		X20SA4430	X20ST4492
Diagnostic Window 1	Hourly	SAI1	TC1, Sense 1
Diagnostic Window 2	Hourly, 15 min after Diagnostic Window 1	SAI3	TC4, Sense 2
Diagnostic Window 3	Hourly, 30 min after Diagnostic Window 1	SAI4	TC3
Diagnostic Window 4	Hourly, 45 min after Diagnostic Window 1	SAI2	TC2

Table 327: Channel diagnostics sequence

2.6.15.2.9 Module function

The safe temperature module is suitable for safely connecting PT100, PT1000 or thermocouples for safety-related applications up to PL e or SIL 3.

> ## Danger!
>
> **Possible failure of safety function**
> **Dangerous system behavior due to incorrect use of analog signal values**
>
> **When using analog signal values, note the information listed in the data sheet regarding the functionality, precision and scope of the data.**

The signal taken via the input terminals is smoothed by the hardware filter (1st-order low pass / cutoff frequency 500 Hz) and digitalized in the subsequent A/D converters.

The filter values configured in the software are applied during digitalization in the A/D converter.

The signals then pass through the 3 stages of digital signal processing.

The safe analog input channels (data type SAFEINT) are formed by the arithmetic mean value of the two individual signals. At this point, also note the information about channel diagnostics.

The validity of analog signals is represented by the associated status signals. These binary status signals (data type SAFEBOOL) must also be evaluated each time the analog signals are used. A binary status signal with the status FALSE indicates an invalid value in the analog signal. In these situations, the analog signal is no longer permitted to be used for safety-related assessments.

To exit an error state, a reset must be carried out. For this to be possible, a valid signal must be received at the analog input for the duration of the I/O update time. The error can then be acknowledged by a rising edge on signal "SafeRelease0x0y".

2.6.15.2.10 Input circuit diagram

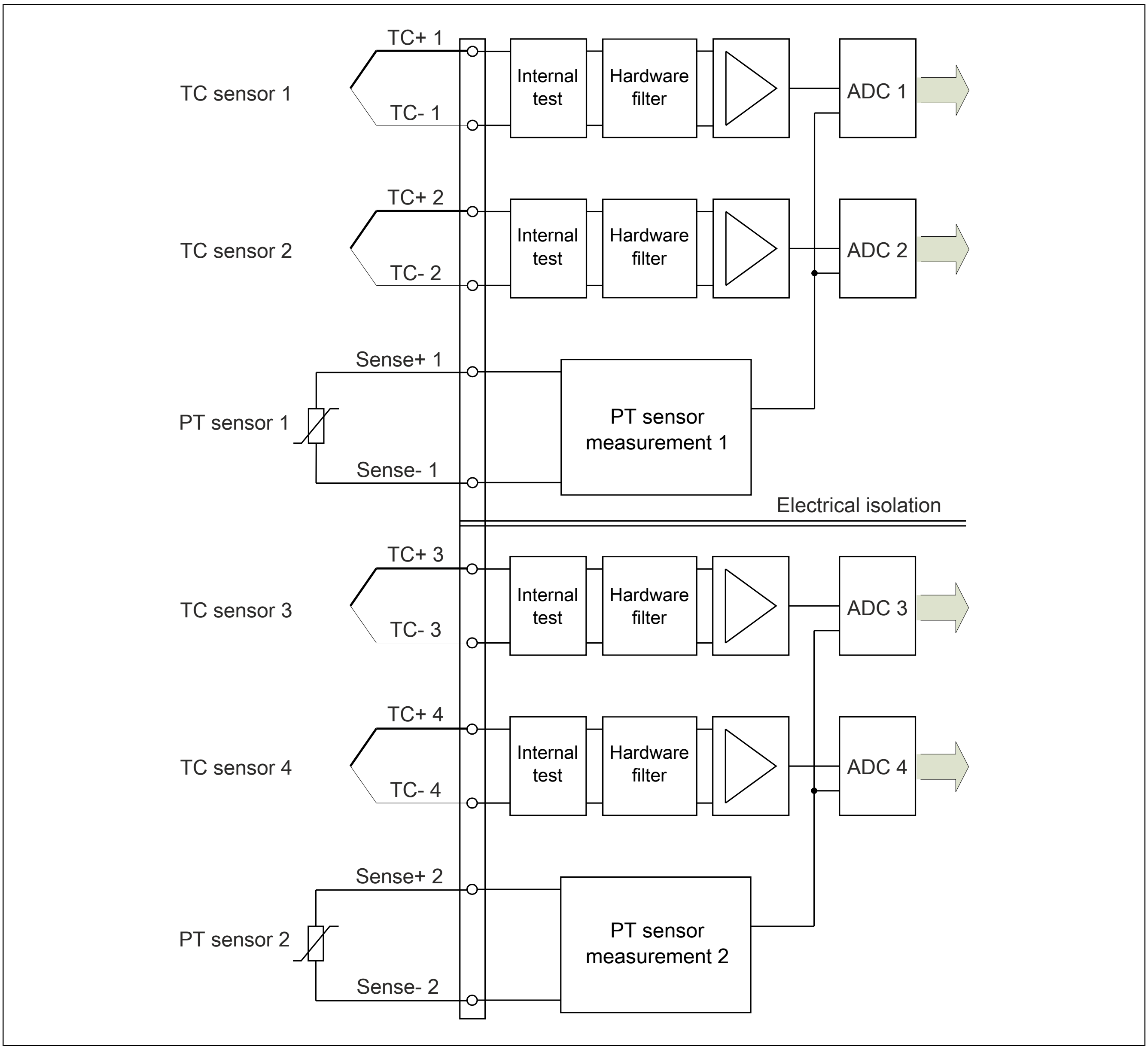


Figure 230: Input circuit diagram

2.6.15.2.11 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.15.2.12 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Danger!

With regard to the I/O update time, an I/O update time of 200 ms should generally be considered for temperature modules up to firmware version 301. The maximum I/O update time is 400 ms.

The I/O update time has been optimized in firmware version 302 and later. The optimized times are listed in the table for the maximum I/O update time.

Configured filter	Maximum I/O update time
1 ms	32 ms
2 ms	40 ms
10 ms	86 ms
16.7 ms	132 ms
20 ms	152 ms
33.3 ms	240 ms
40 ms	284 ms
66.7 ms	372 ms

2.6.15.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example. Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.15.2.14 Register description

2.6.15.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 328: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.1.1 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 329: I/O configuration parameters: General

2.6.15.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td rowspan="2">TwoChannelMode</td><td>This value sets the channels being used for dual-channel evaluation.</td><td>Channel12</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Channel12</td><td>Channels 1 and 2 as well as Channels 3 and 4 are used for dual-channel evaluation.</td></tr>
<tr><td>Channel13</td><td>Channels 1 and 3 as well as Channels 2 and 4 are used for dual-channel evaluation.</td></tr>
</table></td></tr>
<tr><td>Input_Filter_ms</td><td>This parameter sets the filter time of A/D converters.</td><td>1</td><td>ms</td></tr>
</table>

Table 330: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 331: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeTemperatureInputxx

Parameter	Description	Default value	Unit
Sensor_Type (for SafeTemperatureInput01-04)	This parameter can be used to specify the type of sensor connected. • Permissible values: Type J, Type K, Type N, Type S, Type R, Type C, Type T, Voltage [µV]	Type J	-
Sensor_Type (for SafeTemperatureInput05-06)	This parameter can be used to specify the type of sensor connected. • Permissible values: PT100, PT1000	PT1000	-

Table 332: SafeDESIGNER parameters: SafeTemperatureInputxx

Danger!

The use or configuration of an incorrect TC sensor type CANNOT be detected by the module. The temperature value acquired by the module is incorrect.

Make sure during validation that the correct TC sensor type is installed and configured.

Information:

The use or configuration of an incorrect PT sensor type is detected by the module. The module will switch to the FAILSAFE state.

Group: SafeTemperatureInputxxyy

Parameter	Description	Default value	Unit
Limit_Threshold_High_1, Limit_Threshold_High_2, Limit_Threshold_High_3, Limit_Threshold_High_4	This parameter specifies the current maximum permissible analog input value. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -32,768 to +32,767	1000	0.1°C 2 µV
Limit_Threshold_Low_1, Limit_Threshold_Low_2, Limit_Threshold_Low_3, Limit_Threshold_Low_4	This parameter specifies the current minimum permissible analog input value. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -32768 to +32767	0	0.1°C 2 µV
Limit_Threshold_Equivalent_1, Limit_Threshold_Equivalent_2, Limit_Threshold_Equivalent_3, Limit_Threshold_Equivalent_4	This parameter specifies the limit value for the "Dual-channel evaluation" function for the maximum temperature difference between the two channels. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -32768 to +32767	1000	0.1°C 2 µV
Discrepancy_Time_1_ms, Discrepancy_Time_2_ms, Discrepancy_Time_3_ms, Discrepancy_Time_4_ms	This parameter specifies the maximum time for the "Dual-channel evaluation" function in which the difference between both analog input values is permitted to exceed the limit value. • Permissible values: 0 to 10,000 ms (corresponds to 0 to 10 s)	0	ms

Table 333: SafeDESIGNER parameters: SafeTemperatureInputxxyy

Parameters "Limit_Threshold_High_x", "Limit_Threshold_Low_x", "Limit_Threshold_Equivalent_x" and "Discrepancy_Time_x_ms" together make up a parameter set. The channels "SafeThrSelector_xxyy_Bit1" and "SafeThrSelector_xxyy_Bit2" are available in the SafeDESIGNER application to determine which parameter set in the module is enabled, i.e. it is possible to change the parameter set at runtime.

2.6.15.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

Parameter	Description	Default value	Unit
Min required FW Rev	This parameter is reserved for future functional expansions.	Basic Release	-
Optional	This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.	No	-
	Parameter value: No — This module is absolutely necessary for the application. The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No". After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.		
	Parameter value: Yes — This module is not necessary for the application. The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller. After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.		
	Parameter value: Startup — This module is optional. The system determines how the module will proceed during startup. If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set. If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.		
	Parameter value: NotPresent — This module is not necessary for the application. The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present. Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior. After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.		
External UDID	This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.	No	-
	Parameter value: Yes-ATTENTION — The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.		
	Parameter value: No — The UDID is specified by a teach-in procedure during startup.		

Table 334: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 335: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Two-Channel Mode</td><td>This value sets the channels being used for dual-channel evaluation.</td><td>Channel 1-2</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Channel 1-2</td><td>Channels 1 and 2 as well as Channels 3 and 4 are used for dual-channel evaluation.</td></tr>
<tr><td>Channel 1-3</td><td>Channels 1 and 3 as well as Channels 2 and 4 are used for dual-channel evaluation.</td></tr>
</table></td></tr>
<tr><td>Input Filter</td><td>This parameter sets the filter time of A/D converters.
• Permissible values: 1 ms, 2 ms, 10 ms, 16.7 ms, 20 ms, 33.3 ms, 40 ms, 66.7 ms</td><td>1</td><td>ms</td></tr>
<tr><td rowspan="2">Measurement Result while Testing</td><td>This parameter enables the signal behavior specified prior to firmware version 321 for the duration of signal diagnostics (see chapter "Channel diagnostics").</td><td>Single channel</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Averaged</td><td>During testing, the safe analog signal results from the mean value of the individual signals.</td></tr>
<tr><td>Single channel</td><td>During testing, the safe analog signal corresponds to the individual signal of the channel that is not currently being diagnosed.</td></tr>
</table></td></tr>
</table>

Table 336: SafeDESIGNER parameters: Module Configuration

Group: SafeTemperaturexx

Parameter	Description	Default value	Unit
Sensor Type (for SafeTemperatureInput01-04)	This parameter can be used to specify the type of sensor connected. • Permissible values: Type J, Type K, Type N, Type S, Type R, Type C, Type T, Voltage (µV)	Type J	-
Sensor Type (for SafeTemperatureInput05-06)	This parameter can be used to specify the type of sensor connected. • Permissible values: PT100, PT1000	PT1000	-

Table 337: SafeDESIGNER parameters: SafeTemperaturexx

Danger!

The use or configuration of an incorrect TC sensor type CANNOT be detected by the module. The temperature value acquired by the module is false.

Make sure during validation that the correct TC sensor type is installed and configured.

Information:

The use or configuration of an incorrect PT sensor type is detected by the module. The module will switch to the FAILSAFE state.

Group: SafeTemperaturexxyy

Parameter	Description	Default value	Unit
Limit Threshold High 1, Limit Threshold High 2, Limit Threshold High 3, Limit Threshold High 4	This parameter specifies the current maximum permissible analog input value. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -2,147,483,648 to +2,147,483,647	1000	0.1°C 2 µV
Limit Threshold Low 1, Limit Threshold Low 2, Limit Threshold Low 3, Limit Threshold Low 4	This parameter specifies the current minimum permissible analog input value. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -2,147,483,648 to +2,147,483,647	0	0.1°C 2 µV
Limit Threshold Equivalent 1, Limit Threshold Equivalent 2, Limit Threshold Equivalent 3, Limit Threshold Equivalent 4	This parameter specifies the limit value for the "Dual-channel evaluation" function for the maximum temperature difference between the two channels. • Permissible values during temperature measurement: Corresponds to the thermocouple type • Permissible values during voltage measurement: -2,147,483,648 to +2,147,483,647	1000	0.1°C 2 µV
Discrepancy Time 1, Discrepancy Time 2, Discrepancy Time 3, Discrepancy Time 4	This parameter specifies the maximum time for the "Dual-channel evaluation" function in which the difference between both analog input values is permitted to exceed the limit value. • Permissible values: 0 to 10,000 ms (corresponds to 0 to 10 s)	0	ms

Table 338: SafeDESIGNER parameters: SafeTemperaturexxyy

The parameters "Limit Threshold High x", "Limit Threshold Low x", "Limit Threshold Equivalent x" and "Discrepancy Time x" together make up a parameter set. The channels "SafeThrSelector_xxyy_Bit1" and "SafeThrSelector_xxyy_Bit2" are available in the SafeDESIGNER application to determine which parameter set in the module is enabled, i.e. it is possible to change the parameter set at runtime.

2.6.15.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read)[1)]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) [1)]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read)[1)]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>ChannelOKxx</td><td>Read</td><td>-</td><td>BOOL</td><td>Status of physical channel xx</td></tr>
<tr><td>SafeChannelOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel xx</td></tr>
<tr><td>SafeTemperatureOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel temperature evaluation xx/yy</td></tr>
<tr><td>TestActive</td><td>Read</td><td>Read</td><td>BOOL</td><td>Indication of an active channel test</td></tr>
<tr><td>EquivalentThresholdxxyy</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Limit value "Limit Threshold Equivalent" currently in use (see "SafeDESIGNER parameters: SafeTemperaturexxyy")</td></tr>
<tr><td>DiscrepanceTimeThresholdxxyy</td><td>(Read)[1)]</td><td>-</td><td>UINT</td><td>Limit value "Discrepancy Time" currently in use (see "SafeDESIGNER parameters: SafeTemperaturexxyy")</td></tr>
<tr><td>SafeTemperaturexxyy</td><td>Read</td><td>Read</td><td>SAFEINT</td><td>(Temperature channel xx + Temperature channel yy)/2</td></tr>
<tr><td>TemperatureOKxx</td><td>Read</td><td>-</td><td>BOOL</td><td>Status of temperature evaluation xx</td></tr>
<tr><td>Temperature_A</td><td>Read</td><td>-</td><td>INT</td><td>Temperature of the temperature channel selected with "TempChnl_Select_A"</td></tr>
</table>

Table 339: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>TempChnl_Select_A</td><td>Write</td><td>-</td><td>USINT</td><td>Selection of the temperature to be transferred on the "Temperature_A" channel</td></tr>
<tr><td>SafeThrSelector_xxyy_Bit1</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td rowspan="2"><table><tr><th>**_Bit1</th><th>**_Bit2</th><th>Parameters currently being used</th></tr><tr><td>0</td><td>0</td><td>Parameter set 1</td></tr><tr><td>1</td><td>0</td><td>Parameter set 2</td></tr><tr><td>0</td><td>1</td><td>Parameter set 3</td></tr><tr><td>1</td><td>1</td><td>Parameter set 4</td></tr></table></td></tr>
<tr><td>SafeThrSelector_xxyy_Bit2</td><td>-</td><td>Write</td><td>SAFEBOOL</td></tr>
<tr><td>SafeReleasexxyy</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Release signal - Channel xxyy</td></tr>
</table>

Table 339: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

The validity of analog signals is represented by the associated status signals. These binary status signals (data type SAFEBOOL) must also be evaluated each time the analog signals are used. A binary status signal with the status FALSE indicates an invalid value in the analog signal. When this happens, the analog signal is no longer permitted to be used for safety-related assessments.

2.6.16 Counter modules

2.6.16.1 Overview

Model number	Short description	Page
X20SD1207	X20 safe digital counter module, 1 safe digital counter channel, 7 kHz, 24 VDC	713
X20cSD1207	X20 safe digital counter module, coated, 1 safe digital counter channel, 7 kHz, 24 VDC	713

2.6.16.2 X20(c)SD1207

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 340: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 341: Organization of general notices

2.6.16.2.1 General information

The modules are equipped with 1 safe digital input for measuring the velocity. Velocity information can be acquired from AB signals up to a maximum frequency of 7 kHz.

The safe counter modules are suitable for safely acquiring velocities for safety applications up to PL e or SIL 3.

These modules are designed for X20 12-pin terminal blocks.

- 1 safe digital counter input with up to 7 kHz counter frequency
- For encoder inputs A-A, A-B, A-A/-B-B/
- Sink circuit
- Input filter configurable

2.6.16.2.1.1 Function

Safe counter function

This safe counter module is suitable for safely acquiring speed information from AB signals up to a maximum frequency of 7 kHz for safety-related applications up to PL e or SIL 3.

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.16.2.1.2 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

> **Information:**
>
> **For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.**

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.16.2.2 Overview

Module	X20SD1207
Counter function	
Number of counter channels	1
Nominal voltage	24 VDC
Input circuit	Sink
Function modes	A-A, A-B, A-A/-B-B/
Input frequency	Max. 7 kHz
Encoder supply	
Nominal voltage	24 VDC
Nominal output current	80 mA

Table 342: Counter and positioning modules

2.6.16.2.3 Order data

Model number	Short description	Figure
	Counter and positioning modules	
X20SD1207	X20 safe digital counter module, 1 safe digital counter channel, 7 kHz, 24 VDC	
X20cSD1207	X20 safe digital counter module, coated, 1 safe digital counter channel, 7 kHz, 24 VDC	
	Required accessories	
	Bus modules	
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous	
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous	
X20cBM33	X20 bus module, coated, for X20 SafeIO modules, internal I/O power supply continuous	
	Terminal blocks	
X20TB52	X20 terminal block, 12-pin, safety-keyed	

Table 343: X20SD1207, X20cSD1207 - Order data

2.6.16.2.4 Technical data

Model number	X20SD1207	X20cSD1207
Short description		
I/O module	1 safe digital counter channel, 7 kHz, 24 VDC	
General information		
B&R ID code	0xCAC1	0xE1CB
System requirements		
Automation Studio	3.0.90 or later	4.0.16 or later
Automation Runtime	3.00 or later	V3.08 or later
SafeDESIGNER	2.91 or later	3.1.0 or later
Safety Release	1.5 or later	1.7 or later
Status indicators	I/O function per channel, operating state, module status	
Diagnostics		
Module run/error	Yes, using status LED and software	
Inputs	Yes, using status LED and software	
Blackout mode		
Scope	Module	
Function	Module function	
Standalone mode	No	
Max. I/O cycle time	2 ms	
Power consumption		
Bus	0.25 W	
Internal I/O	0.75 W	
Electrical isolation		
Channel - Bus	Yes	
Channel - Channel	No	
Certifications		
CE	Yes	
EAC	Yes	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
DNV GL	Temperature: **A** (0 - 45°C) Humidity: **B** (up to 100%) Vibration: **A** (0.7 g) EMC: **B** (bridge and open deck)	
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013	
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3	
Functional safety	EN 50156-1:2004	
Safety characteristics		
EN ISO 13849-1:2015		
Category	Cat. 4 The special instructions in the "Connection examples" section must be followed. [1]	
PL	PL e	
DC	>94%	
MTTFD	2500 years	
Mission time	Max. 20 years	
IEC 61508:2010, IEC 61511:2004, EN 62061:2013		
SIL CL	SIL 3	
SFF	>90%	
PFH / PFH_d		
Module	$<1*10^{-10}$	
openSAFETY wired	Negligible	
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour	
PFD	$<2*10^{-5}$	
Proof test interval (PT)	20 years	
Encoder power supply		
Output voltage	I/O power supply minus residual voltage	
Nominal output current	80 mA	
Residual voltage	<0.4 VDC	

Table 344: X20SD1207, X20cSD1207 - Technical data

Model number	X20SD1207	X20cSD1207
Protective measures		
Short-circuit proof	Thermal limit determined by PTC	
I/O power supply		
Nominal voltage	24 VDC	
Voltage range	24 VDC -15% / +20%	
Integrated protection	Reverse polarity protection	
Safe digital counter inputs		
Nominal voltage	24 VDC	
Input characteristics per EN 61131-2	Type 1	
Input filter		
Hardware	<10 µs	
Software	Configurable between 0 and 100 s	
Input frequency	Max. 7 kHz	
Input circuit	Sink	
Input voltage	24 VDC -15% / +20%	
Input current at 24 VDC	2.48 mA	
Input resistance	9.68 kΩ	
Isolation voltage between channel and bus	500 V_{eff}	
Switching threshold		
Low	<5 VDC	
High	>15 VDC	
Line length	Max. 30 m shielded	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level	0 to 2000 m, no limitation	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	0 to 60°C	-40 to 60°C [2]
Vertical mounting orientation	0 to 50°C	-40 to 50°C [3]
Derating	See section "Derating".	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.	
Spacing	$25^{+0.2}$ mm	

Table 344: X20SD1207, X20cSD1207 - Technical data

1) The related danger warnings in the technical data sheet must also be observed.
2) Up to hardware upgrade <1.10.1.0: -25 to 60°C
3) Up to hardware upgrade <1.10.1.0: -25 to 50°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SD1207
Derating bonus	
At 24 VDC	+2.5°C
Dummy module to the left	+0°C
Dummy module to the right	+2.5°C
Dummy module to the left and right	+5°C
With double PFH / PFH_d	+0°C

Table 345: Derating bonus

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

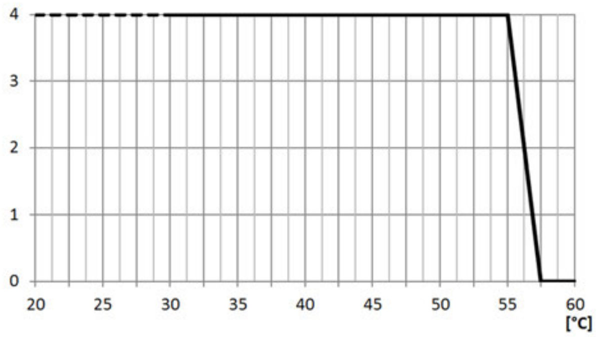
Horizontal (0 to 60°C, coated: -40 to 60°C)	Vertical (0 to 50°C, coated: -40 to 50°C)
	No derating

Table 346: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.16.2.5 LED status indicators

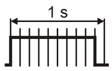
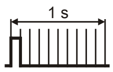
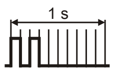
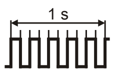
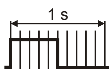
Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	A, B, $\overline{A}$, $\overline{B}$	Input state of the corresponding digital input		
		Red	On	Warning/Error on the input channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
			Off	No warning / No error
		Green	On	Input set
			Off	Input not set
	p	This LED is reserved for future functional expansions.		
	v	Status of speed evaluation		
		Red	On	Warning/Error on evaluation channel, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Evaluation channel set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 347: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.16.2.6 Pinout

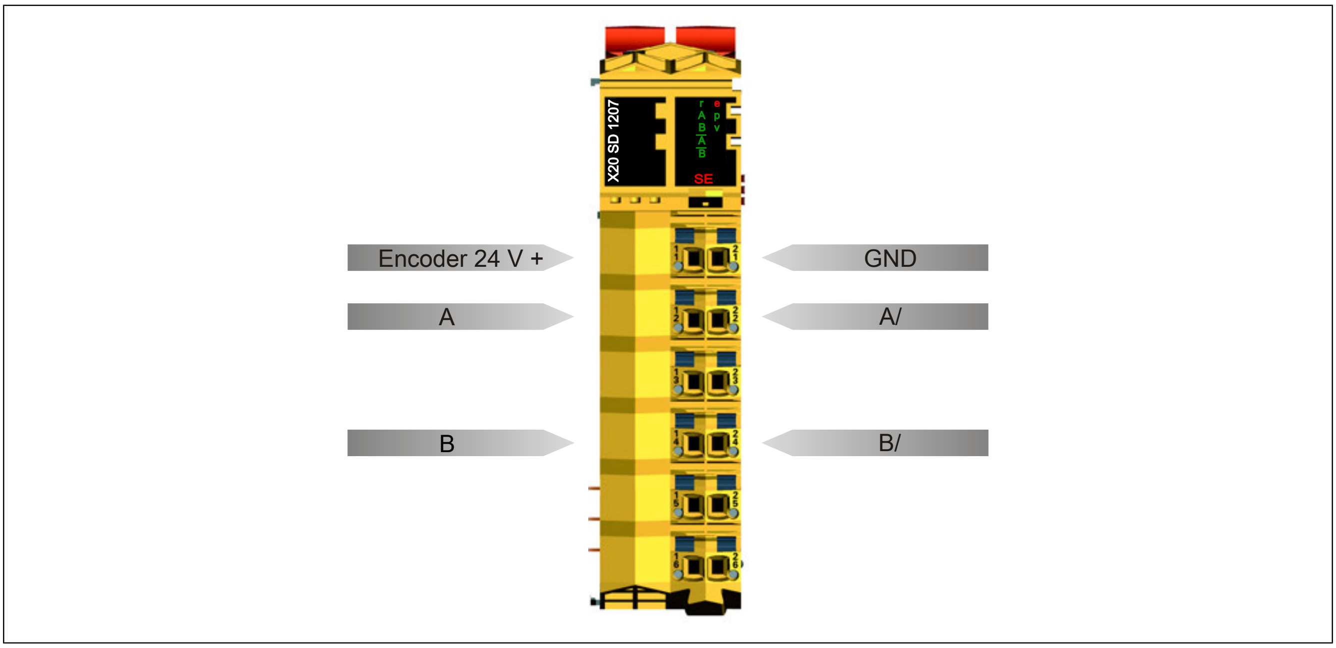


Figure 231: X20SD1207 - Pinout

2.6.16.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into consideration in each case.

2.6.16.2.7.1 Function mode A-A - Single-channel encoder

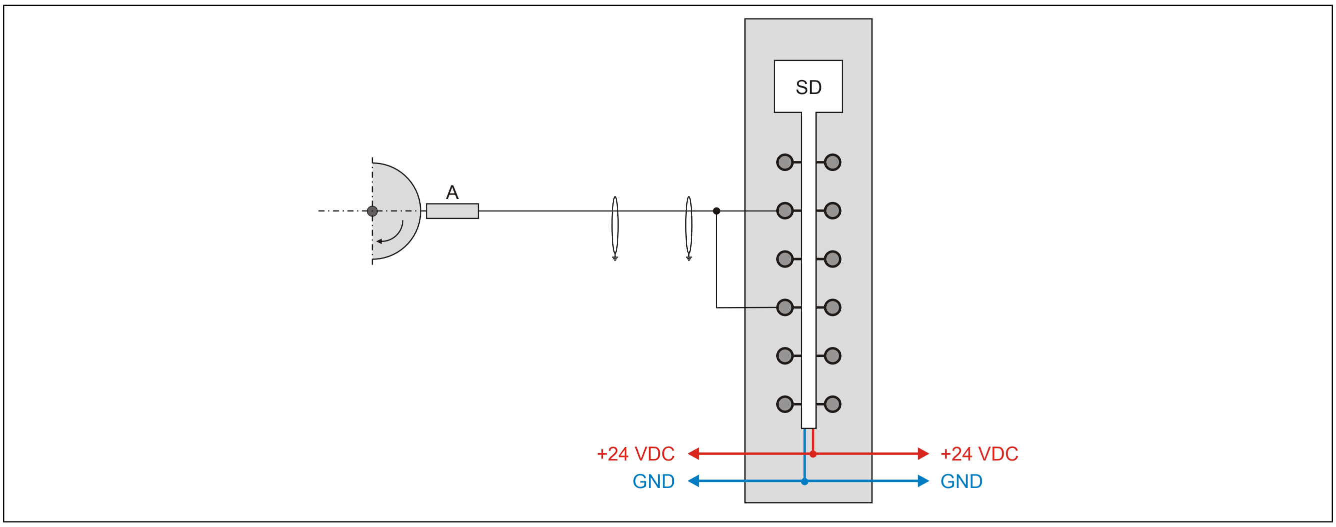


Figure 232: X20SD1207 - Function mode A-A - Single-channel encoder

Signal form A-A

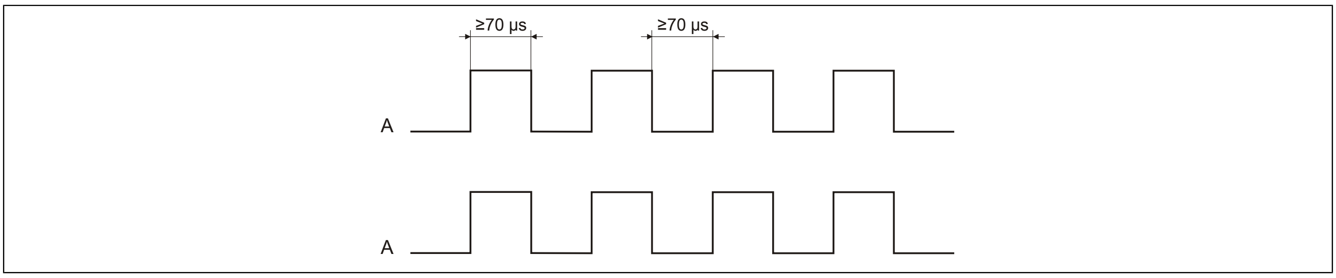


Figure 233: Signal form A-A

Function mode	A-A - Single-channel encoder
Category in accordance with EN ISO 13849-1:2015 (module and encoder)	CAT 2
Safe recording of the rotary speed	Yes, if rotary speed >0
Safe recording of the direction of rotation	No
Safe stall detection	No
Encoder wiring instructions	
• Shielded cables should be used for encoder wiring. • Cable length - Max. 30 m	
Information regarding the encoder	
• The encoder must be taken into consideration when assessing and validating the safety chain. • Encoders with output signal test pulses (OSSD) are not permitted to be used because the test pulses would result in incorrect measurement results on the counter channel. • The encoder signal levels must be compatible with the input channels. Here, the characteristic values listed in the technical data must be taken into account.	
Information regarding the encoder supply	
• The design of the encoder supply must ensure proper operation and the correct signal level (<5 VDC low, >15 VDC high).	

2.6.16.2.7.2 Function mode A-A - Two-channel encoder

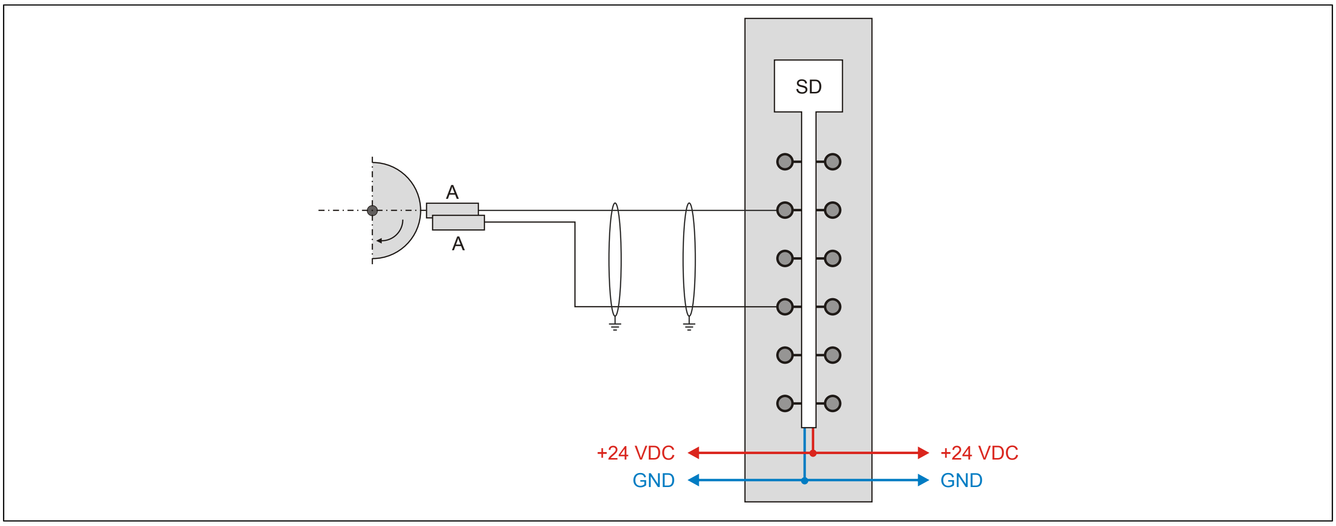


Figure 234: X20SD1207 - Function mode A-A - Two-channel encoder

Signal form A-A

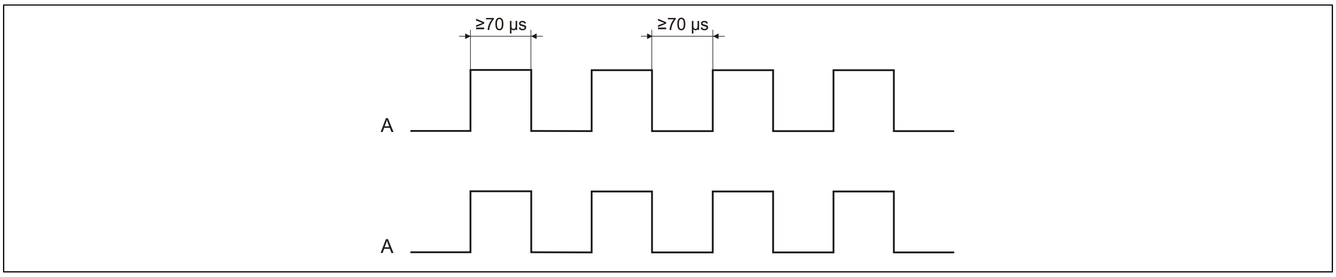


Figure 235: Signal form A-A

Function mode	**A-A - Two-channel encoder**
Category in accordance with EN ISO 13849-1:2015 (module and encoder)	CAT 4
Safe recording of the rotary speed	Yes, if rotary speed >0
Safe recording of the direction of rotation	No
Safe stall detection	No
Encoder wiring instructions	
• Two separate and shielded lines must be used to wire both encoders.	
Information regarding the encoder	
• The encoder must be taken into consideration when assessing and validating the safety chain. • Encoders with output signal test pulses (OSSD) are not permitted to be used because the test pulses would result in incorrect measurement results on the counter channel. • The encoder signal levels must be compatible with the input channels. Here, the characteristic values listed in the technical data must be taken into account. • The two "A" signals must be generated by independent encoders.	
Information regarding the encoder supply	
• The design of the encoder supply must ensure proper operation and the correct signal level (<5 VDC low, >15 VDC high).	

2.6.16.2.7.3 Function mode A-B

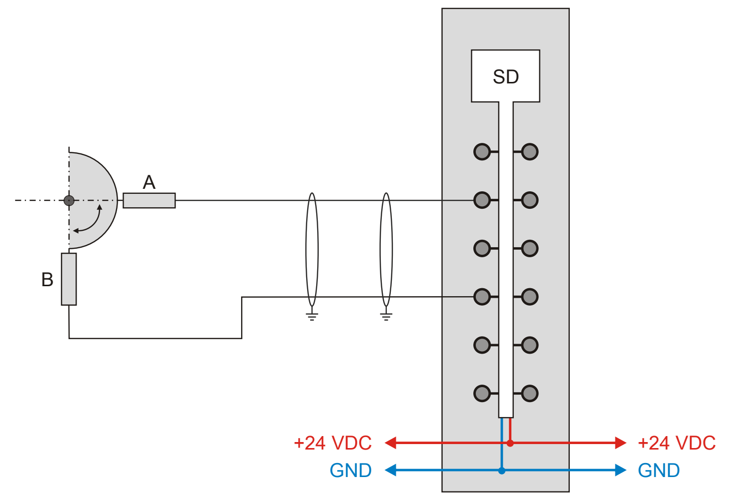


Figure 236: X20SD1207 - Function mode A-B

Signal form A-B

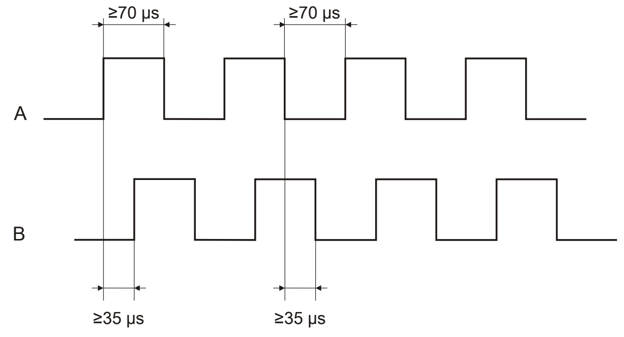


Figure 237: Signal form A-B

Function mode	A-B
Category in accordance with EN ISO 13849-1:2015 (module and encoder)	CAT 4
Safe recording of the rotary speed	Yes, if rotary speed >0
Safe recording of the direction of rotation	No
Safe stall detection	No
Encoder wiring instructions	
• Shielded cables should be used for encoder wiring. • Cable length - Max. 30 m	
Information regarding the encoder	
• The encoder must be taken into consideration when assessing and validating the safety chain. • Encoders with output signal test pulses (OSSD) are not permitted to be used because the test pulses would result in incorrect measurement results on the counter channel. • The encoder signal levels must be compatible with the input channels. Here, the characteristic values listed in the technical data must be taken into account. • The "A" and "B" signals must be generated by independent encoders. If "AB" encoders are used, it is necessary to ensure that the "A" signal is generated in the encoder independent of the "B" signal.	
Information regarding the encoder supply	
• The design of the encoder supply must ensure proper operation and the correct signal level (<5 VDC low, >15 VDC high).	

2.6.16.2.7.4 Function mode A-A/-B-B/

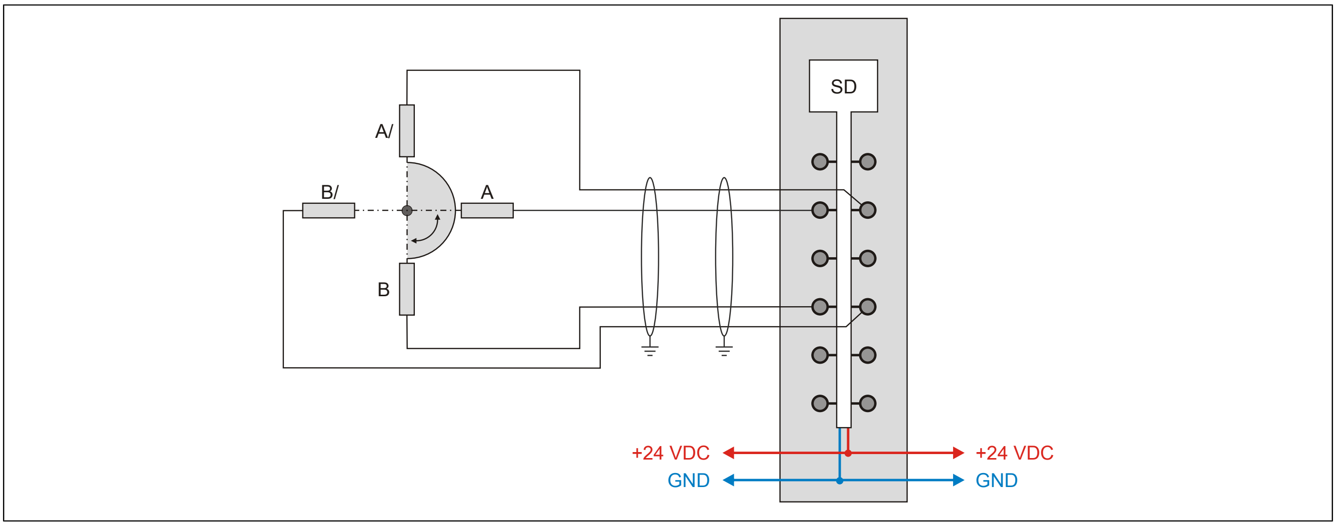


Figure 238: X20SD1207 - Function mode A-A/-B-B/

Signal form A-A/-B-B/

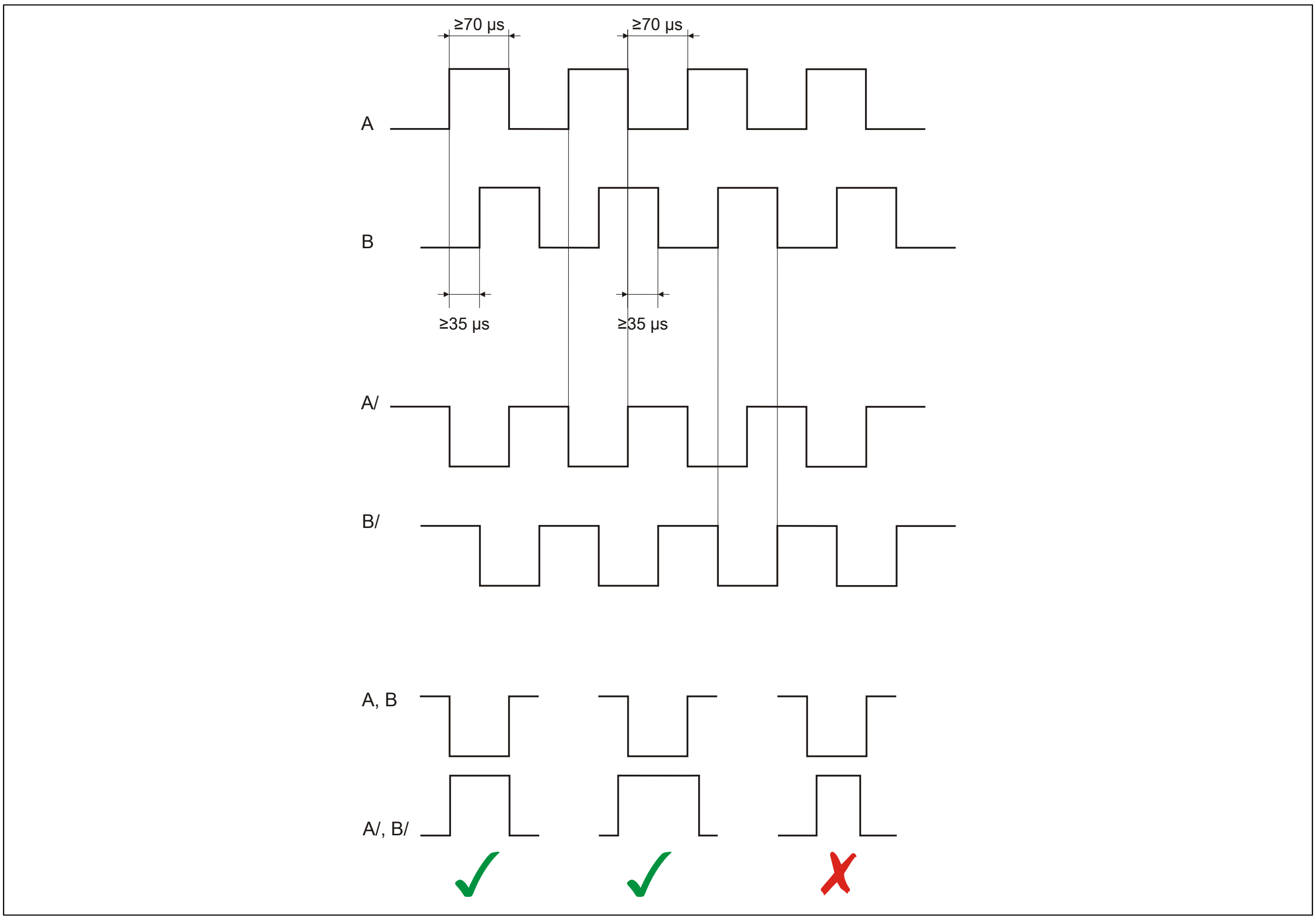


Figure 239: Signal form A-A/-B-B/

Function mode	A-A/-B-B/
Category in accordance with EN ISO 13849-1:2015 (module and encoder)	CAT 4
Safe recording of the rotary speed	Yes, if rotary speed >0
Safe recording of the direction of rotation	Yes
Safe stall detection	Yes
Encoder wiring instructions	
• Shielded cables should be used for encoder wiring. • Cable length - Max. 30 m	
Information regarding the encoder	
• The encoder must be taken into consideration when assessing and validating the safety chain. • Encoders with output signal test pulses (OSSD) are not permitted to be used because the test pulses would result in incorrect measurement results on the counter channel. • The encoder signal levels must be compatible with the input channels. Here, the characteristic values listed in the technical data must be taken into account. • The "A", "A/", "B" and "B/" signals must be generated by independent encoders. If "AA/BB/" encoders are used, it is necessary to ensure that all signals are generated in the encoder independent of the others.	
Information regarding the encoder supply	
• The design of the encoder supply must ensure proper operation and the correct signal level (<5 VDC low, >15 VDC high).	

2.6.16.2.8 Error detection

2.6.16.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.16.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

> **Danger!**
>
> **Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.**

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

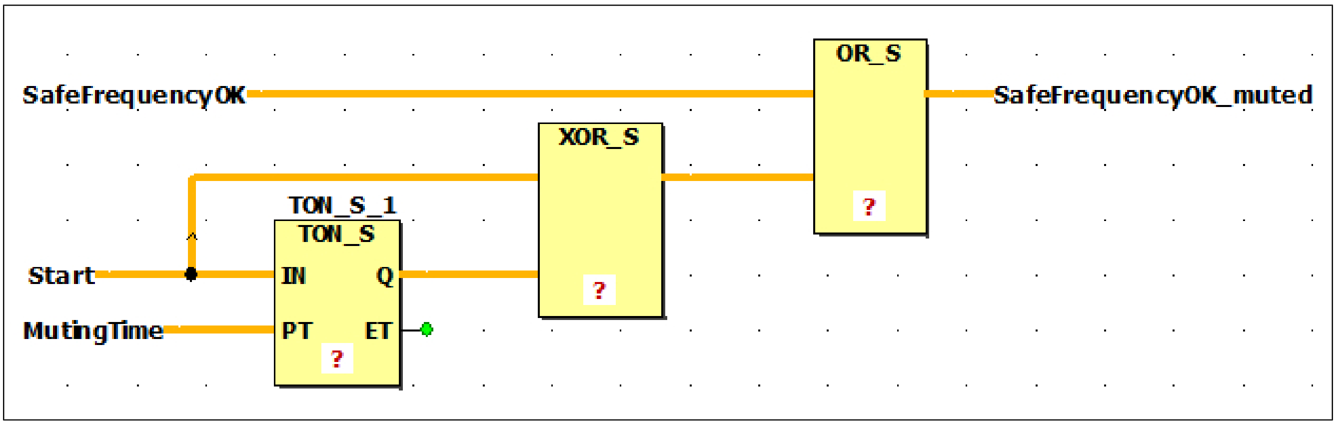
Function mode A-A and A-B

In these modes, the module identifies a safe frequency signal ("SafeFrequency").
Detecting wiring errors only functions properly with dynamic signals and not when in a stationary state. Thus, signal "SafeFrequency" is not permitted to be evaluated when in a stationary state.
This situation is indicated by the "SafeFrequencyOK" status signal.

Status signal "SafeFrequencyOK" is determined as follows:

- SAFETRUE, if pulses are detected on the counter channel within the time specified for "Timebase"
- SAFEFALSE, if no pulses are detected on the counter channel within the time specified for "Timebase", or a different problem is found on the module

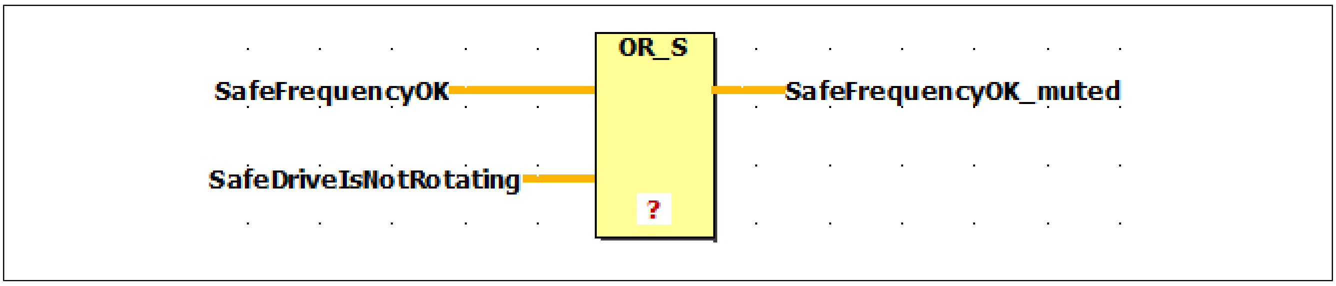
Because "SafeFrequency" is not permitted to be evaluated when in a stationary state, a deadlock situation can occur, for example in an application that monitors the max. speed when starting up the drive (drive cannot start because signal "SafeFrequencyOK" is not SAFETRUE, and signal "SafeFrequencyOK" cannot become SAFETRUE because the drive does not start).
The following SafeDESIGNER code snippet could be used, for example, to solve this problem:



Variable	Type	Source	Description
SafeFrequencyOK	SAFEBOOL	X20SD1207	This status signal indicates the validity of signal "SafeFrequency".
Start	SAFEBOOL	Applications	A rising edge on this signal indicates that a start request has been sent for the rotary movement.
MutingTime	SAFETIME	Applications	This signal defines the max. time the drive needs to detect pulses on the counter channel. The "Timebase" parameter must also be taken into consideration during this time. **Important:** **Monitoring functions are not active during this time. Therefore, this time must be as short as possible. Alternate methods must be used to ensure that no dangerous states can occur within this time.**
SafeFrequencyOK_muted	SAFEBOOL	-	This signal can now be used to further evaluate the rotary movement.

Table 348: Code snippet: Timed muting of signal "SafeFrequencyOK"

As soon as a safe signal for determining the rotary movement is available, the following SafeDESIGNER code snippet can be used:



Variable	Type	Source	Description
SafeFrequencyOK	SAFEBOOL	X20SD1207	This status signal indicates the validity of signal "SafeFrequency".
SafeDriveIsNotRotating	SAFEBOOL	Applications	This signal indicates if a rotary movement is taking place or not.
SafeFrequencyOK_muted	SAFEBOOL	-	This signal can now be used to further evaluate the rotary movement.

Table 349: Code snippet: Muting signal "SafeFrequencyOK" using an additional signal

Function mode A-A/-B-B/

In mode "A-A/-B-B/", wiring error detection is always available regardless if in a stationary state or not. In this mode, it is also permitted to evaluate signal "SafeFrequency" when in a stationary state and to implement safe stall detection.

2.6.16.2.9 Input circuit diagram

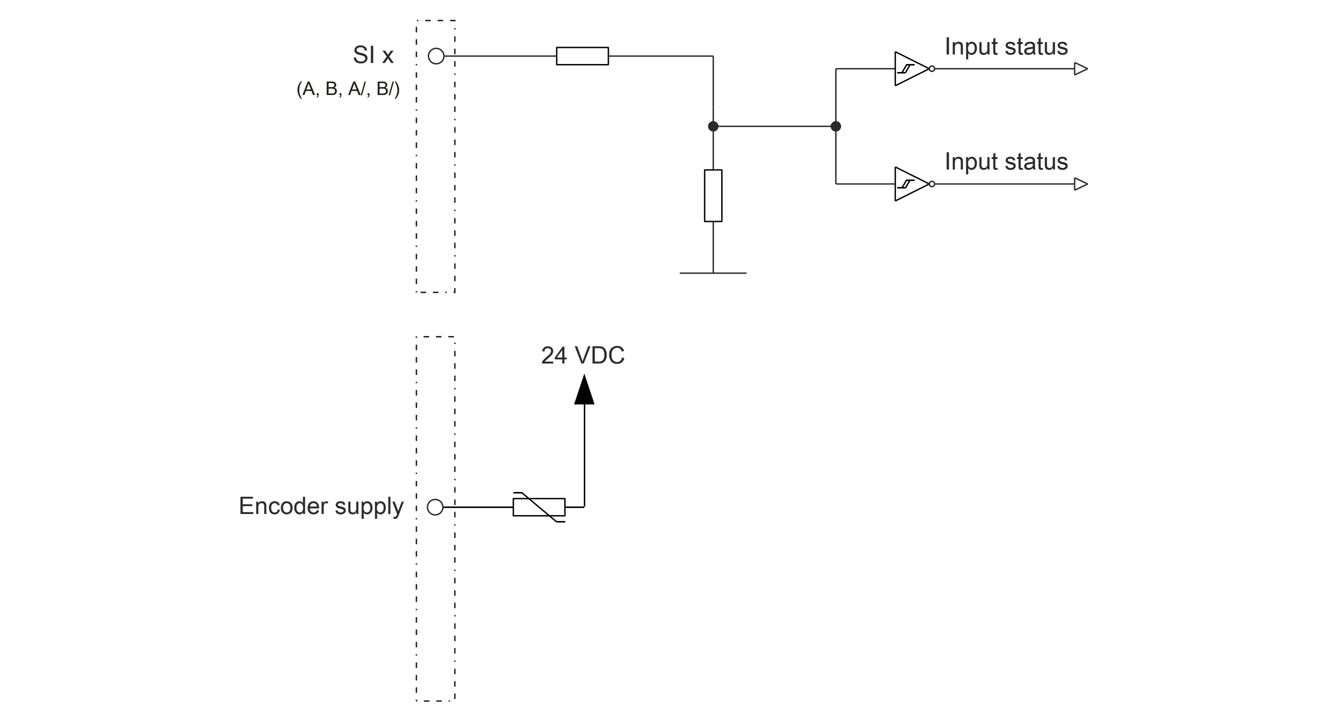


Figure 240: Input circuit diagram

2.6.16.2.10 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.16.2.11 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time. This depends on the "Timebase" configured in SafeDESIGNER.

Timebase	I/O update time	Maximum I/O update time
10 ms	2 ms	12 ms
50 ms	2 ms	52 ms
100 ms	2 ms	102 ms
500 ms	5 ms	505 ms
1000 ms	10 ms	1010 ms
5000 ms	50 ms	5050 ms
10 s	0.1 s	10.1 s
50 s	0.5 s	50.5 s
100 s	1 s	101 s

Danger!

Configuring parameter "Timebase" lengthens the safety response time!

2.6.16.2.12 Precision

The precision of the frequency value measured by the module is determined by the module's resolution and basic accuracy. In firmware version 300 and later, measurement precision has been significantly improved.

2.6.16.2.12.1 Precision in firmware version 297

Timebase	Resolution in mode "A-A"			Resolution in mode "A-B" and "A-A/-B-B/"			Basic accuracy
	Inc/s	Inc/min	Inc/h	Inc/s	Inc/min	Inc/h	
10 ms	±60 Inc/s	±60 Inc/s	±60 Inc/s	±30 Inc/s	±30 Inc/s	±30 Inc/s	±5% of measured value
50 ms	±12 Inc/s	±12 Inc/s	±12 Inc/s	±6 Inc/s	±6 Inc/s	±6 Inc/s	±5% of measured value
100 ms	±6 Inc/s	±6 Inc/s	±6 Inc/s	±3 Inc/s	±3 Inc/s	±3 Inc/s	±5% of measured value
500 ms	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value
1 s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value
5 s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value
10 s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value
50 s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value
100 s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±1 Inc/s	±5% of measured value

Table 350: Precision in firmware version 297

Danger!

The safe precision of the safe counter module is the result of adding the resolution and the basic accuracy (see table above).

2.6.16.2.12.2 Precision in firmware version 300 and later

Configuring the "Unit" parameter			Basic accuracy
Inc/s	Inc/min	Inc/h	
±1 Inc/s	±1 Inc/min	±1 Inc/h	±3% of measured value

Table 351: Precision in firmware version 300 and later

Danger!

The safe precision of the safe counter module is the result of adding the resolution and the basic accuracy (see table above).

2.6.16.2.13 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example. Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.16.2.14 Register description

2.6.16.2.14.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 352: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 353: I/O configuration parameters: General

2.6.16.2.14.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table></td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table></td></tr>
<tr><td>Function mode</td><td>This parameter can be used to select the mode for input signal evaluation.</td><td>A-A</td><td>-</td></tr>
</table>

Table 354: SafeDESIGNER parameters: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td></td><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Mode A-A</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The frequency setting can only accept positive values in this mode.</td></tr>
<tr><td>Mode A-B</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The frequency setting can only accept positive values in this mode.</td></tr>
<tr><td>Mode A-Ai-B-Bi</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The combination of the inputs can be used to differentiate between a positive and negative direction.
The frequency setting can accept positive and negative values in this mode.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">Unit</td><td>This parameter can be used to set the unit that should be used when the module transfers the frequency.</td><td>Increment / s</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Increment / s</td><td>The frequency that has been determined will be shown in increments per second.</td></tr>
<tr><td>Increment / min</td><td>The frequency that has been determined will be shown in increments per minute.</td></tr>
<tr><td>Increment / h</td><td>The frequency that has been determined will be shown in increments per hour.</td></tr>
</table>
</td></tr>
<tr><td>Timebase</td><td>This parameter specifies the time for calculating the average value of the frequency.
• Permissible values: 10 ms, 50 ms, 100 ms, 500 ms, 1 s, 5 s, 10 s, 50 s, 100 s</td><td>10</td><td>ms</td></tr>
</table>

Table 354: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

Configuring parameter "Timebase" lengthens the safety response time!

Group: Safety_Response_Time

Parameter	Description	Default value	Unit
Manual_Configuration	This parameter makes it possible to manually and individually configure the safety response time for the module. The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module. **Parameter value** – **Description** Yes – Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals. No – The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.	No	-
Synchronous_Network_Only	This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime. **Parameter value** – **Description** Yes – In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times. No – No requirement for synchronization of the networks	Yes	-
Max_X2X_CycleTime_us	This parameter specifies the maximum X2X cycle time used to calculate the safety response time. • Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)	5000	µs
Max_Powerlink_CycleTime_us	This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time. • Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)	5000	µs
Max_CPU_CrossLinkTask_CycleTime_us	This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time. • Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)	5000	µs
Min_X2X_CycleTime_us	This parameter specifies the minimum X2X cycle time used to calculate the safety response time. • Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)	200	µs
Min_Powerlink_CycleTime_us	This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time. • Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)	200	µs
Min_CPU_CrossLinkTask_CycleTime_us	This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time. • Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)	0	µs
Worst_Case_Response_Time_us	This parameter specifies the limit value for monitoring the safety response time. • Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)	50000	µs
Node_Guarding_Lifetime	This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".	5	-

Table 355: SafeDESIGNER parameters: Safety_Response_Time

2.6.16.2.14.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 356: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 357: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Function Mode</td><td>This parameter can be used to select the mode for input signal evaluation.</td><td>Mode A-B</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Mode A-A</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The frequency setting can only accept positive values in this mode.</td></tr>
<tr><td>Mode A-B</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The frequency setting can only accept positive values in this mode.</td></tr>
<tr><td>Mode A-Ai-B-Bi</td><td>In this mode, the frequency of the pulses on the inputs is determined. The frequencies of the relevant inputs are checked to see if they are the same, and a channel error is triggered if there are any deviations.
The combination of the inputs can be used to differentiate between a positive and negative direction.
The frequency setting can accept positive and negative values in this mode.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Unit</td><td>This parameter can be used to set the unit that should be used when the module transfers the frequency.</td><td>Increment / s</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Increment / s</td><td>The frequency that has been determined will be shown in increments per second.</td></tr>
<tr><td>Increment / min</td><td>The frequency that has been determined will be shown in increments per minute.</td></tr>
<tr><td>Increment / h</td><td>The frequency that has been determined will be shown in increments per hour.</td></tr>
</table></td></tr>
<tr><td>Timebase</td><td>This parameter specifies the time for calculating the average value of the frequency.
• Permissible values: 10 ms, 20 ms, 50 ms, 100 ms, 200 ms, 500 ms, 1,000 ms, 2,000 ms, 5,000 ms, 10,000 ms, 20,000 ms, 50,000 ms, 100,000 ms</td><td>10</td><td>ms</td></tr>
</table>

Table 358: SafeDESIGNER parameters: Module Configuration

Danger!

Configuring the "Timebase" parameter lengthens the safety response time!

2.6.16.2.14.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table>
</td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>SafeChannelOK</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>No errors in frequency evaluation</td></tr>
<tr><td>SafeFrequency</td><td>Read</td><td>Read</td><td>SAFEINT</td><td>Current frequency</td></tr>
<tr><td>SafeFrequencyOK</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the frequency being output is OK</td></tr>
<tr><td>Reset</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal</td></tr>
</table>

Table 359: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

Danger!

The validity of analog signals is represented by the associated status signals. These binary status signals (data type SAFEBOOL) must also be evaluated each time the analog signals are used. A binary status signal with the status FALSE indicates an invalid value in the analog signal. When this happens, the analog signal is no longer permitted to be used for safety-related assessments.

2.6.17 reACTION modules

2.6.17.1 Overview

Model number	Short description	Page
X20SRT402	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	741
X20SRT806	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs	741
X20SRT842	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs	741

2.6.17.2 X20SRTxxx

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 2.6.5.2.7 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 360: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 361: Organization of general notices

2.6.17.2.1 General information

The reACTION Technology modules are equipped with 4 to 8 high-speed safe digital inputs and 2 to 6 high-speed safe digital outputs. They are designed for a nominal voltage of 24 VDC.

The modules can be used to read in digital signals and control actuators in safety-related applications up to PL e or SIL 3.

Ultrafast reACTION Technology makes it possible to control internal I/O channels with cycle times down to 100 µs. All commands that can be used for reACTION programs are available as function blocks in special libraries (e.g. AsIORTI). Programming in compliance with IEC 61131-3 requirements takes place in the Function Block Diagram editor in Automation Studio.

The modules are equipped with filters that are individually configurable for switch-on and switch-off behavior. The modules also provide pulse signals for diagnosing the sensor line.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side low-side" variant (output type A) is limited to actuators without reference potential (e.g. relays, valves). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

These modules are designed for X20 12-pin terminal blocks.

- reACTION Technology module
- 4 to 8 high-speed safe digital inputs, sink circuit
- 4 pulse outputs
- Software input filter configurable for each channel
- 4 high-speed safe digital outputs, output type A with 3 A, source circuit
- 2 or 6 high-speed safe digital outputs, output type B with 50 mA or 0.2 A, source circuit
- Cycle time for the safe reACTION task starting at 125 µs
- Integrated output protection

2.6.17.2.1.1 reACTION Technology

This module is equipped with ultrafast reACTION Technology. This allows the I/O channels integrated in the reACTION module to be controlled with cycle times down to 100 µs. In particular, this new technology allows time-critical subprocesses to be managed using standard hardware, which lowers hardware costs by reducing the load on the controller and allowing it to be scaled down accordingly.

All commands that can be used for reACTION programs are available as function blocks in special libraries (e.g. AsIORTI). Programming in compliance with IEC 61131-3 requirements takes place in the Function Block Diagram editor in Automation Studio.

reACTION
TECHNOLOGY

2.6.17.2.1.2 Blackout mode

In blackout mode, module functionality persists even if the network fails. Without this function, the safe state would always be initiated on the affected module if the network fails. In addition, blackout mode can allow partial operation to resume or coordinated shutdown scenarios to be initiated. Blackout mode also makes it possible to boot a module without a network based on a configuration saved on the module beforehand.

2.6.17.2.1.3 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open SAFETY

2.6.17.2.2 Overview

Module	X20SRT402	X20SRT806	X20SRT842
Safe digital inputs			
Number of inputs	4	8	8
Nominal voltage	24 VDC		
Input filter Hardware Software	 ≤130 µs Configurable between 0 and 500 ms		
Input circuit	Sink		
Pulse outputs			
Design	Push-Pull		
Switching voltage	I/O power supply minus residual voltage		
Safe digital HS-LS outputs			
Number of outputs	-		4
Nominal voltage	-		24 VDC
Nominal output current	-		3 A
Total nominal current	-		10 A 1)
Output protection	-		Thermal short circuit shut-down, integrated protection for switching inductive loads
Safe digital HS-HS outputs			
Number of outputs	2	6	2
Nominal voltage	24 VDC		
Nominal output current	0.2 A		50 mA
Total nominal current	0.4 A	1.2 A	100 mA
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads		

Table 362: Digital mixed modules

1) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.

2.6.17.2.3 Order data

X20SRT402 X20SRT806 X20SRT842

Model number	Short description
	reACTION Technology modules
X20SRT402	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 4 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SRT806	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs
X20SRT842	X20 safe digital mixed module, reACTION Technology for safety, 100 µs safety cycle time, 8 safe digital inputs, configurable input filter, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs
	Required accessories
	Bus modules
X20BM33	X20 bus module, for X20 SafeIO modules, internal I/O power supply continuous
X20BM36	X20 bus module, for X20 SafeIO modules, with node number switch, internal I/O power supply continuous
	Terminal blocks
X20TB52	X20 terminal block, 12-pin, safety-keyed

Table 363: X20SRT402, X20SRT806, X20SRT842 - Order data

2.6.17.2.4 Technical data

Model number	X20SRT402	X20SRT806	X20SRT842
Short description			
I/O module	4 safe digital inputs, 4 pulse outputs, 24 VDC, 2 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs, reACTION Technology	8 safe digital inputs, 4 pulse outputs, 24 VDC, 6 safe type B2 digital outputs, 24 VDC, 0.2 A, OSSD <10 µs, reACTION Technology	8 safe digital inputs, 4 pulse outputs, 24 VDC, 4 safe type A digital outputs, 24 VDC, 3 A, OSSD <500 µs, 2 safe type B2 digital outputs, 24 VDC, 50 mA, OSSD <500 µs, reACTION Technology
General information			
B&R ID code	0xE7EC	0xE759	0xE7F7
System requirements			
Automation Studio	4.2.5 or later		
Automation Runtime	4.2 or later		
SafeDESIGNER	4.2.2 or later		
Safety Release	1.10 or later		
Status indicators	I/O function per channel, operating state, module status		
Diagnostics			
Module run/error	Yes, using status LED and software		
Outputs	Yes, using status LED and software		
Inputs	Yes, using status LED and software		
reACTION-capable I/O channels	Yes		
Blackout mode			
Scope	Module		
Function	Programmable		
Standalone mode	Yes		
Max. I/O cycle time	800 µs		
Power consumption			
Bus	0.4 W		
Internal I/O	2.5 W		
Electrical isolation			
Channel - Bus	Yes		
Channel - Channel	No		
Type of signal lines	Shielded cables must be used for all signal lines. [1]		
Certifications			
CE	Yes		
UL	cULus E115267 Industrial control equipment		
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X		
DNV GL	In preparation		
Functional safety	cULus FSPC in preparation ANSI UL 1998 in preparation		
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3		
Functional safety	EN 50156-1:2004		
Safety characteristics			
EN ISO 13849-1:2015			
MTTFD	2500 years		
Mission time	Max. 20 years		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
PFH / PFH_d			
Module	$<1*10^{-10}$		
openSAFETY wired	Negligible		
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour		
PFD	$<2*10^{-5}$		
Proof test interval (PT)	20 years		
Safe digital inputs			
EN ISO 13849-1:2015			
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [2]		
PL	PL e		
DC	>94%		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
SIL CL	SIL 3		
SFF	>90%		

Table 364: X20SRT402, X20SRT806, X20SRT842 - Technical data

Model number	X20SRT402	X20SRT806	X20SRT842
Safe digital outputs			
EN ISO 13849-1:2015			
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" 2)		
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" 2)		
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" 2)		
IEC 61508:2010, IEC 61511:2004, EN 62061:2013			
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" 2)		
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" 2)		
I/O power supply			
Nominal voltage	24 VDC		
Voltage range	24 VDC -15% / +20%		
Integrated protection	Reverse polarity protection		
Safe digital inputs			
Nominal voltage	24 VDC		
Input characteristics per EN 61131-2	Type 1		
Input filter			
Hardware	≤130 µs		
Software	Configurable between 0 and 500 ms		
Input circuit	Sink		
Input voltage	24 VDC -15% / +20%		
Input current at 24 VDC	Max. 3.28 mA		
Input resistance	Min. 7.33 kΩ		
Error detection time	100 ms		
Isolation voltage between channel and bus	500 V_{eff}		
Switching threshold			
Low	<5 VDC		
High	>15 VDC		
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line		
Safe digital HS-LS outputs			
Variant	-		FET, 1x positive switching, 1x negative switching, type A, output level readable
Nominal voltage	-		24 VDC
Nominal output current	-		3 A
Total nominal current	-		10 A 3)
Output protection	-		Thermal short-circuit shutdown, integrated protection for switching inductive loads 4)
Braking voltage when switching off inductive loads	-		Max. 90 VDC 5)
Error detection	-		1 s
Isolation voltage between channel and bus	-		500 V_{eff}
Peak short-circuit current	-		Max. 100 A
Leakage current when switched off	-		<1 mA
Residual voltage	-		≤1 VDC at nominal current
Switching voltage	-		I/O power supply minus residual voltage
Max. switching frequency	-		1000 Hz
Test pulse length	-		Max. 500 µs
Max. capacitive load	-		100 nF
Safe digital HS-HS outputs			
Variant	FET, 2x positive switching, type B2, output level readable		
Nominal voltage	24 VDC		
Nominal output current	0.2 A		50 mA
Total nominal current	0.4 A	1.2 A	100 mA
Output protection	Active shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads 4)		
Braking voltage when switching off inductive loads	Max. 45 VDC		
Error detection time	1 s		
Isolation voltage between channel and bus	500 V_{eff}		
Peak short-circuit current	Max. 10 A		500 mA
Leakage current when switched off	<100 µA		<1 mA
Residual voltage	≤1.2 VDC at nominal current		≤3 VDC at nominal current
Switching voltage	I/O power supply minus residual voltage		
Max. switching frequency	100 Hz		
Test pulse length	Max. 10 µs		Max. 500 µs
Max. capacitive load	100 nF		

Table 364: X20SRT402, X20SRT806, X20SRT842 - Technical data

Model number	X20SRT402	X20SRT806	X20SRT842
Current on loss of ground			
I_{OUT}	<100 µA		
I_{GND}	<200 mA		<50 mA [6]
Pulse outputs			
Variant	Push-Pull		
Nominal output current	50 mA		
Output protection	Shutdown of individual channels in the event of overload or short circuit [4]		
Peak short-circuit current	0.5 A for 120 µs		
Short-circuit current	15 mA_{eff}		
Leakage current when switched off	0.1 mA		
Residual voltage	≤4 VDC		
Switching voltage	I/O power supply minus residual voltage		
Total nominal current	200 mA		
Operating conditions			
Mounting orientation			
Horizontal	Yes		
Vertical	Yes		
Installation elevation above sea level	0 to 2000 m, no limitation		
Degree of protection per EN 60529	IP20		
Ambient conditions			
Temperature			
Operation			
Horizontal mounting orientation	0 to 60°C		
Vertical mounting orientation	0 to 50°C		
Derating	See section "Derating".		
Storage	-40 to 85°C		
Transport	-40 to 85°C		
Relative humidity			
Operation	5 to 95%, non-condensing		
Storage	5 to 95%, non-condensing		
Transport	5 to 95%, non-condensing		
Mechanical properties			
Note	Order 2x safety-keyed terminal block separately. Order 1x safety-keyed bus module separately.		
Spacing	$25^{+0.2}$ mm		

Table 364: X20SRT402, X20SRT806, X20SRT842 - Technical data

1) For more information, see the Installation/EMC guide.
2) The related danger warnings in the technical data sheet must also be observed.
3) The module's total nominal current is limited to 10 A. The output currents of group "Safe digital HS-HS outputs" must be included.
4) The protective function is provided for max. 30 minutes for a continuous short circuit.
5) Due to the internal protective circuit, this braking voltage only takes effect starting at a load of typ. 250 mA.
6) The value for this module is limited to 50 mA by the nominal output current of the HS-HS outputs.

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For detailed information about installation, see chapter "Installation notes for X20 modules" on page 23.

Derating

The derating curve refers to standard operation and can be shifted to the right by the specified derating bonus if in a horizontal mounting orientation.

Module	X20SRT402	X20SRT806	X20SRT842
Derating bonus			
At 24 VDC	+2.5°C		+5°C
At 20.4 VDC	+7.5°C		+10°C
Dummy module on the left	+2.5°C		
Dummy module on the right	+0°C		
Dummy module on the left and right	+2.5°C		+5°C
Pulse output	+7.5°C 1)		+5°C 1)
4 safe inputs (SI)	+0°C	+2.5°C 2)	+0°C
With double PFH / PFH_d	+15°C 3)		

Table 365: Derating bonus

1) Pulse output loaded with maximum 2 safe inputs (SI)
2) Only 4 safe inputs (SI) in use
3) Hardware revision C0 or later and hardware upgrade 1.10.2.0 or later

Inputs

The number of inputs that should be used at the same time depends on the operating temperature and the mounting orientation. The resulting amount can be looked up in the following table.

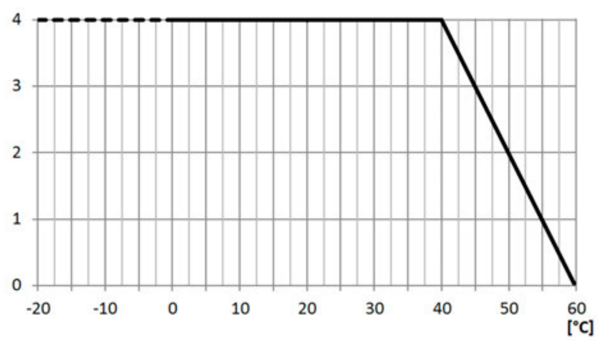
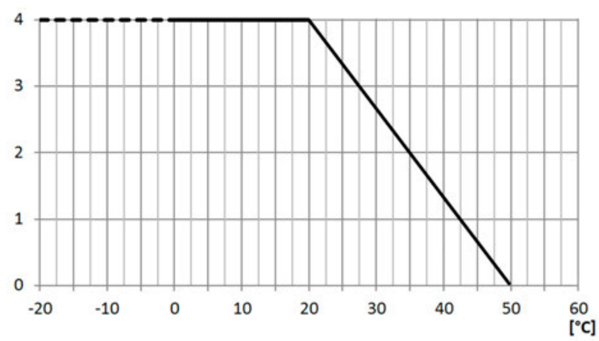
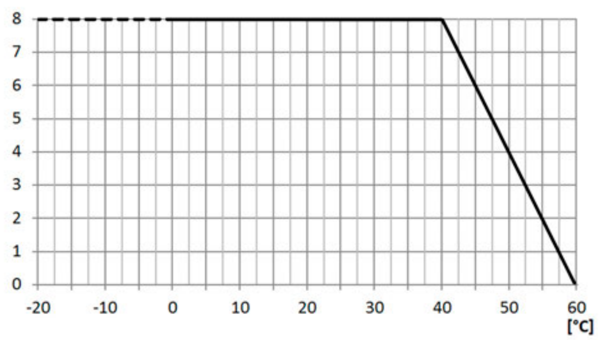
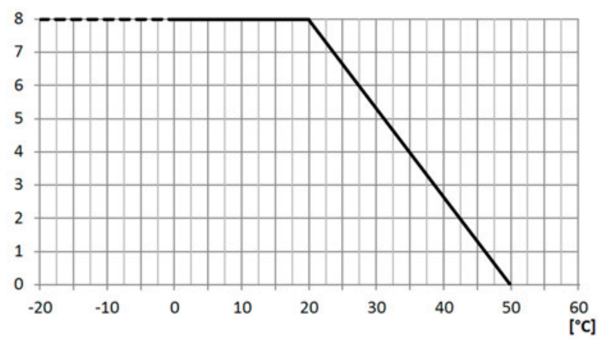
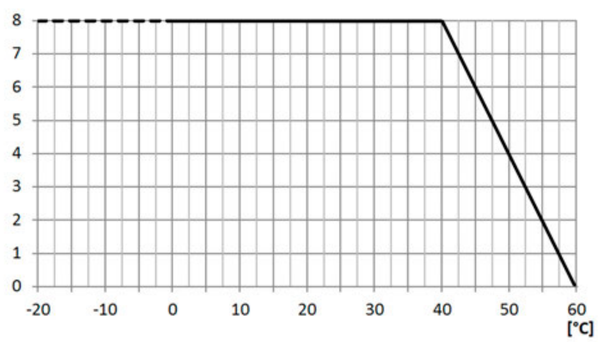
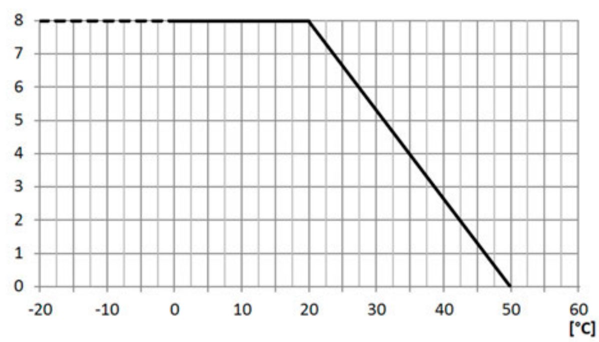
Horizontal (0 to 60°C)	Vertical (0 to 50°C)
X20SRT402	
	
X20SRT806	
	
X20SRT842	
	

Table 366: Derating in relation to operating temperature and mounting orientation

Outputs

The maximum total nominal current depends on the operating temperature and the mounting orientation. The resulting total nominal current can be found in the following table.

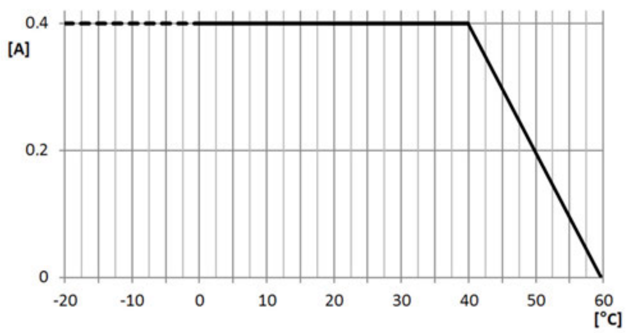
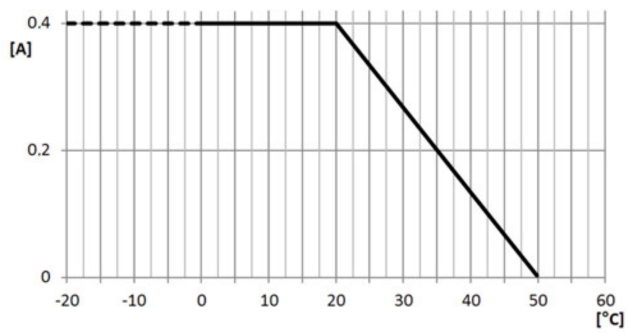
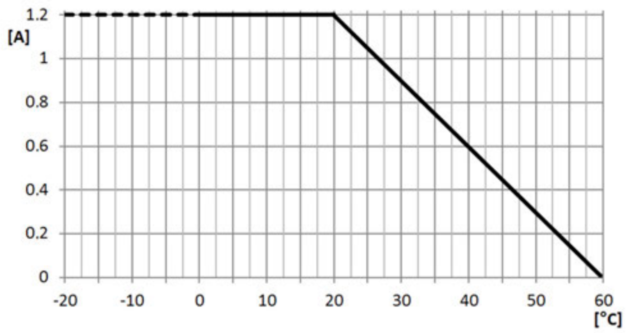
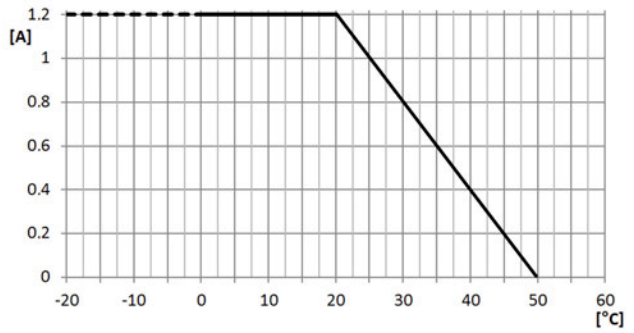
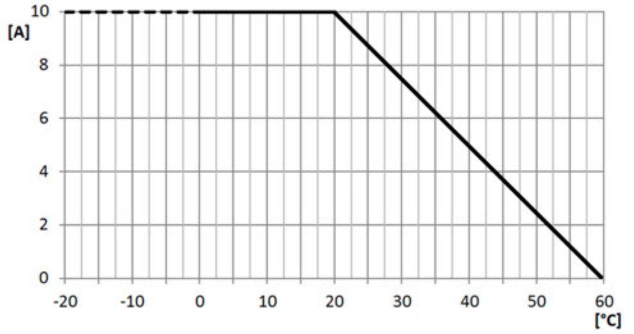
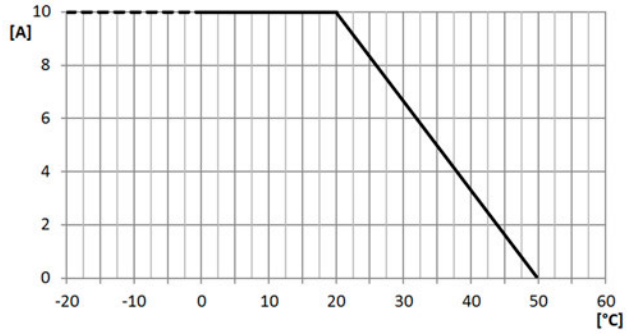
Horizontal (0 to 60°C)	Vertical (0 to 50°C)
X20SRT402	
	
X20SRT806	
	
X20SRT842	
	

Table 367: Derating in relation to operating temperature and mounting orientation

Information:

Regardless of the values specified in the derating curve, the module cannot be operated above the values specified in the technical data.

2.6.17.2.5 LED status indicators

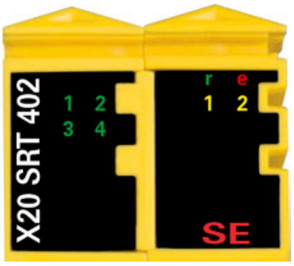
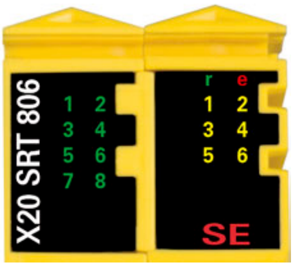
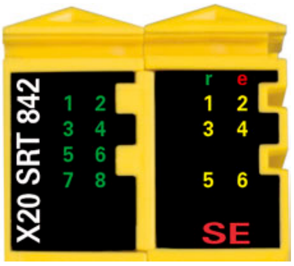
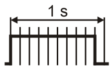
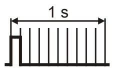
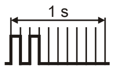
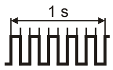
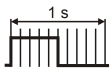
Figure	LED	Color	Status	Description
  	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Solid red / Single green flash		Invalid firmware
	1 to 8	Input state of the corresponding digital input The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an input channel
			Blinking	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or startup not yet completed
		Green	On	Input set
	1 to 6	Output status of the corresponding digital output The number of channel LEDs varies depending on the number of channels on the module type.		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or startup not yet completed
		Orange	On	Output set
	SE	Red	Off	RUN mode or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version. No reACTION application exists on the module
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= state "FailSafe")
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 368: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

2.6.17.2.6 Pinouts

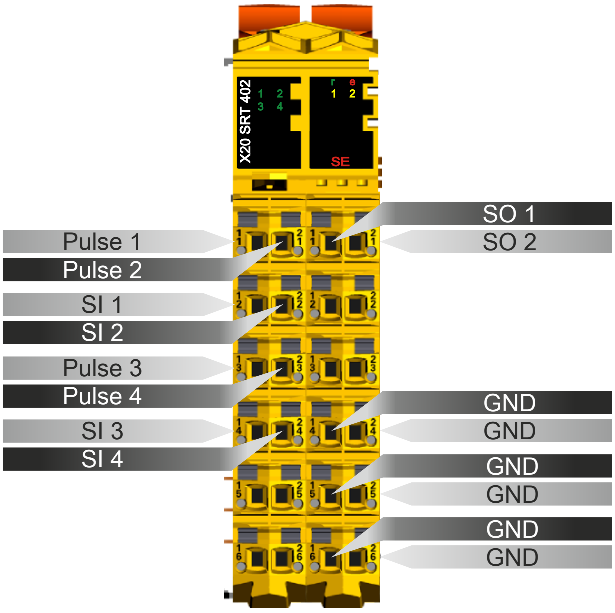


Figure 241: X20SRT402 - Pinout

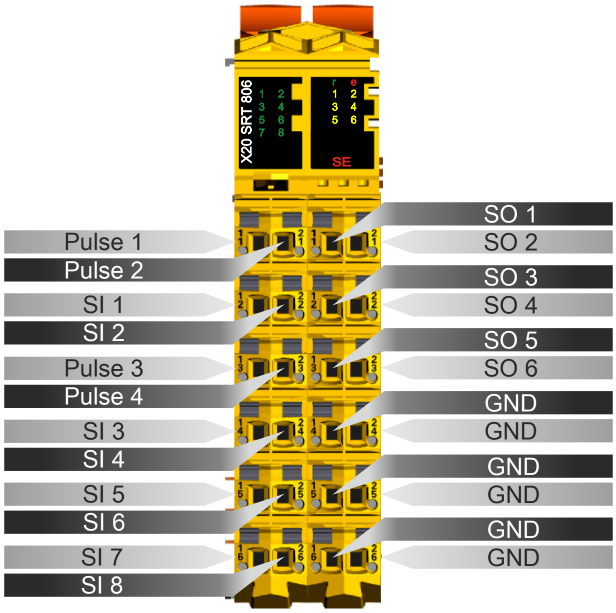


Figure 242: X20SRT806 - Pinout

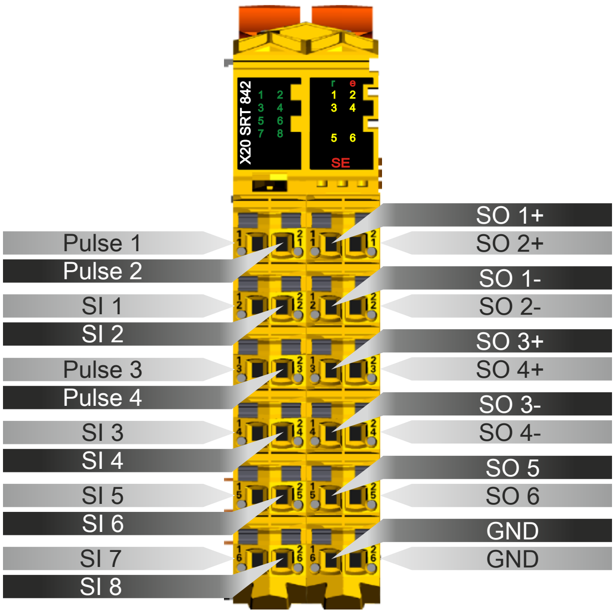


Figure 243: X20SRT842 - Pinout

2.6.17.2.7 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

2.6.17.2.7.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

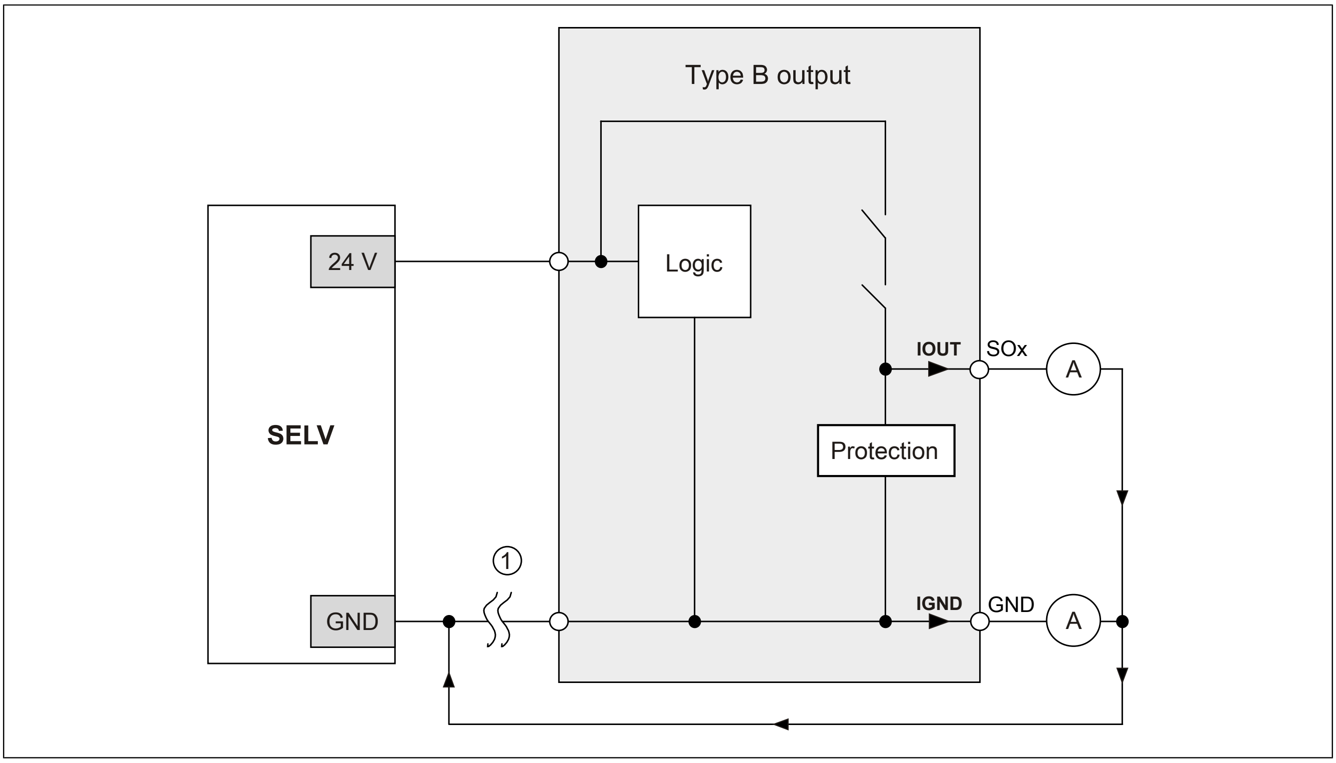


Figure 244: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

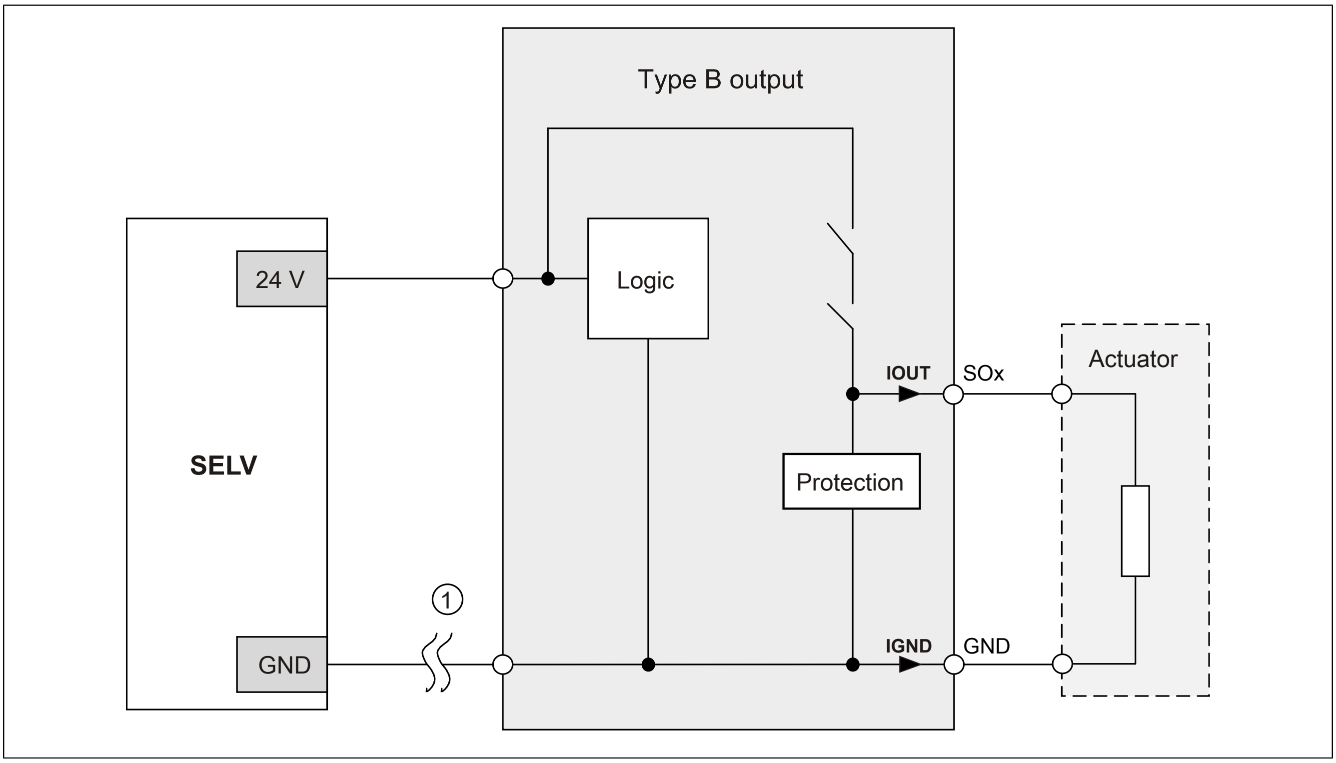


Figure 245: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

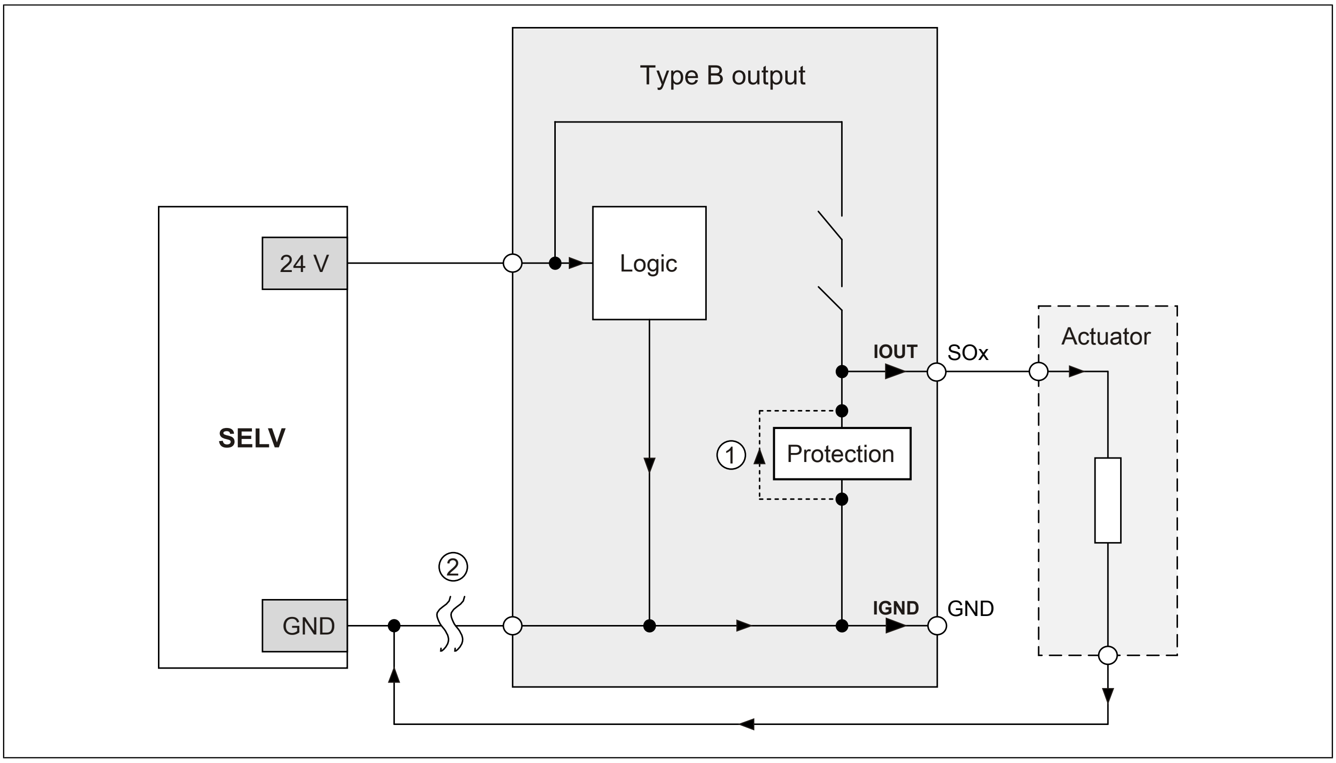


Figure 246: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

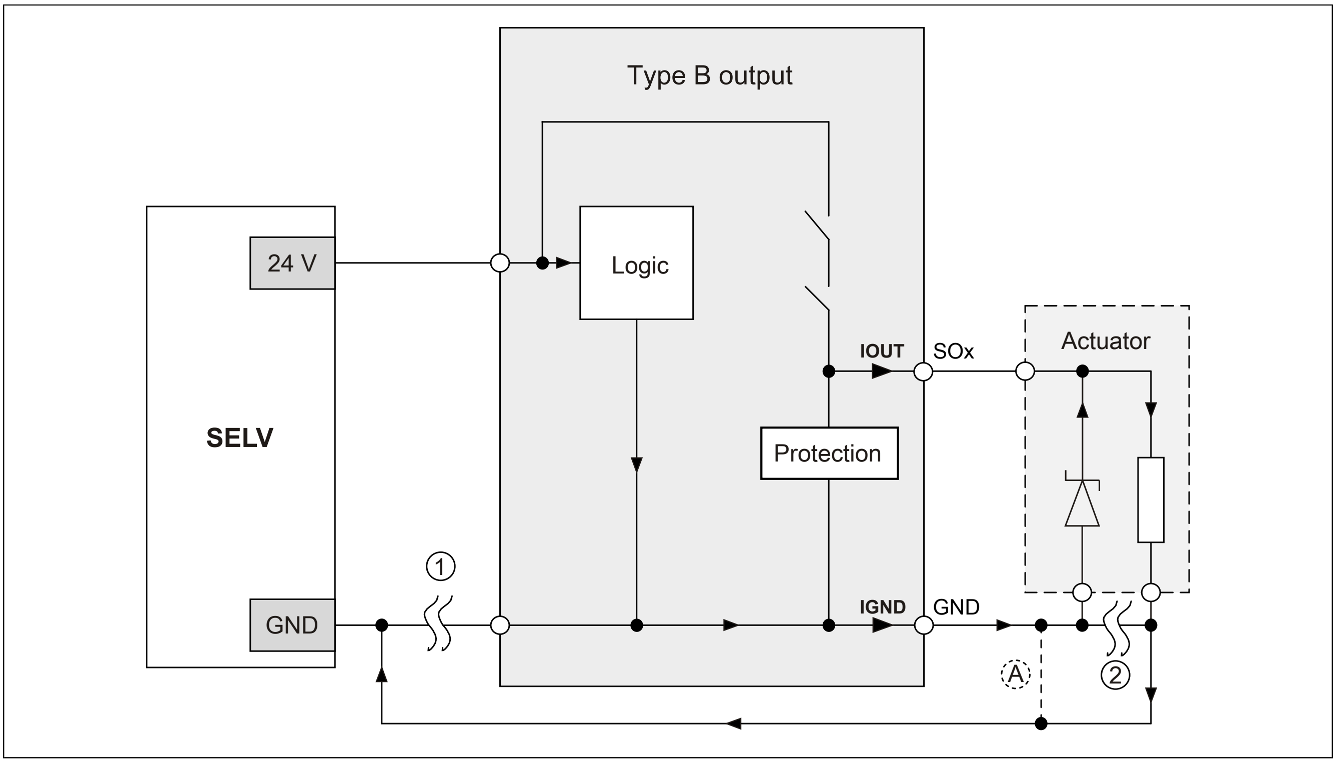


Figure 247: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

2.6.17.2.7.2 Connecting single-channel sensors with contacts

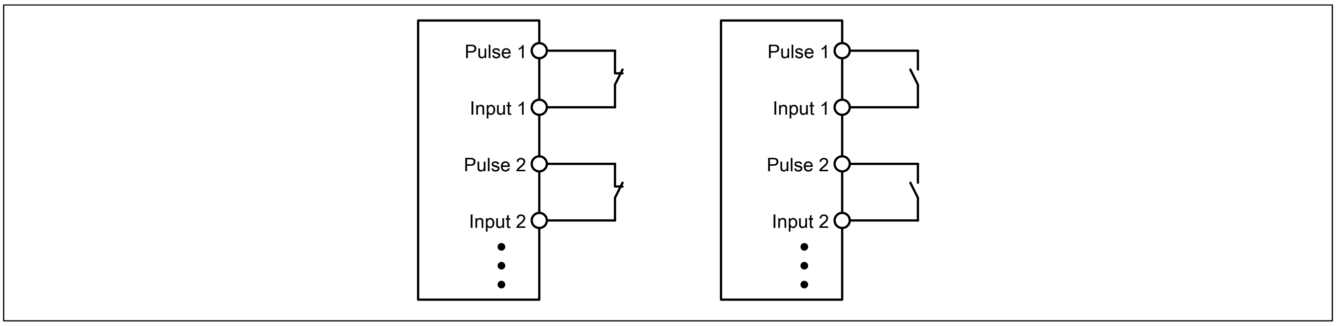


Figure 248: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.17.2.7.3 Connecting two-channel sensors with contacts

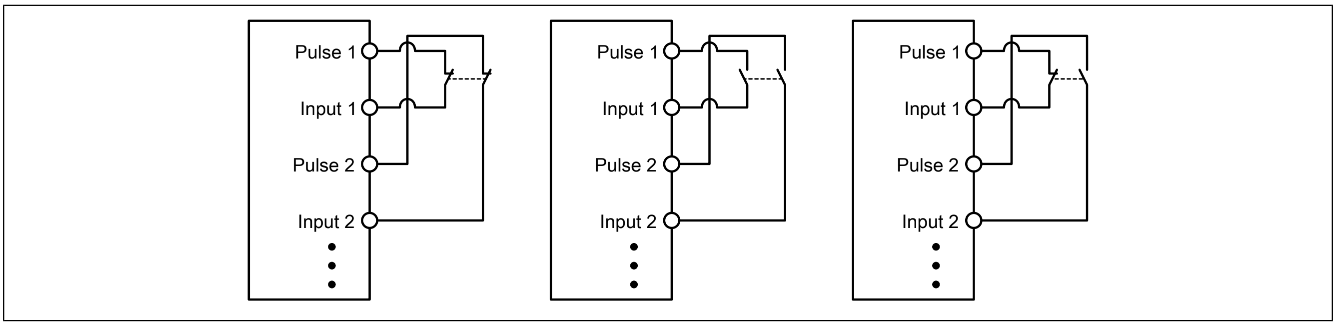


Figure 249: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

2.6.17.2.7.4 Connecting multi-channel sensors with contacts

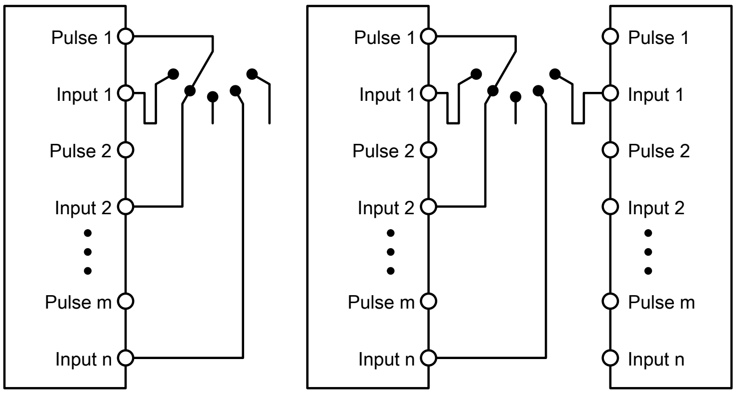


Figure 250: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

2.6.17.2.7.5 Connecting electronic sensors

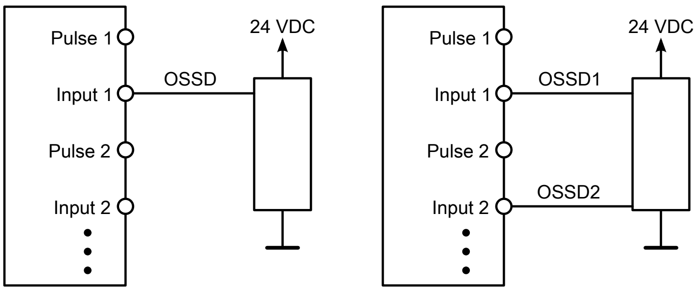


Figure 251: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

2.6.17.2.7.6 Using the same pulse signals

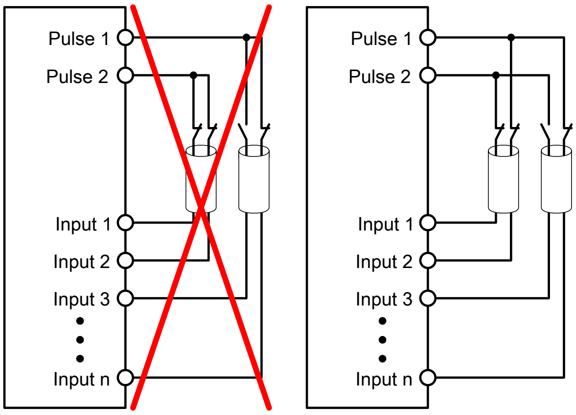


Figure 252: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

2.6.17.2.7.7 Connecting safety-oriented actuators for Type A outputs

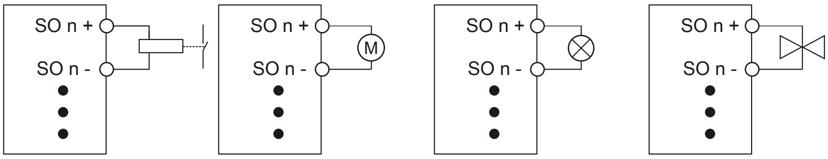


Figure 253: Connecting safety-oriented actuators for Type A outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

2.6.17.2.7.8 Connecting safety-oriented actuators for Type B outputs

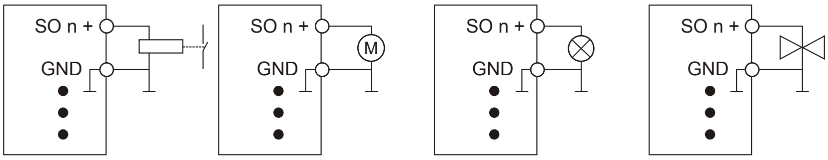


Figure 254: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

2.6.17.2.8 Error detection

2.6.17.2.8.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

> **Danger!**
>
> **Operating the safety module in BOOT mode is not permitted.**

> **Danger!**
>
> **A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.**

2.6.17.2.8.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

> **Danger!**
>
> **Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.**

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

Type A output channels

> **Danger!**
>
> **Type A output channels also cut off the load on the GND side. Check whether the actuator you have connected permits a cutoff on the GND side. X20 and X67 systems do not support this type of cutoff, for example.**

> **Danger!**
>
> **Note that wiring SOx+ directly to GND via an actuator is not permitted; wiring 24 VDC directly to SOx- via an actuator is also not permitted.**
>
> **These types of errors will not be detected by the module. The user must prevent these types of errors through careful wiring.**

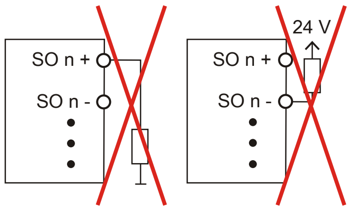


Figure 255: Invalid wiring

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

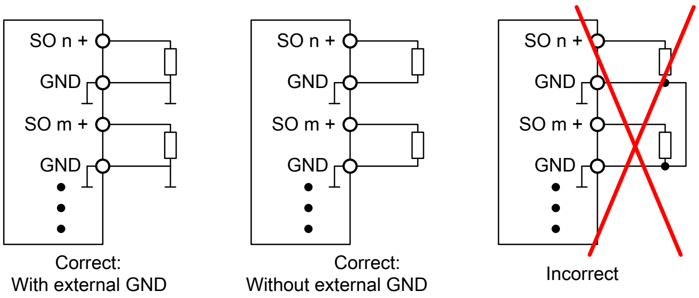


Figure 256: Invalid wiring

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 369: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1]	**Not detected**
Open circuit	**Not detected**	Detected[1]

Table 370: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1]
Ground fault on signal input (active signal)	Detected[1]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1]
Open circuit (inactive signal)	**Not detected**

Table 371: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Safety actuator connection

<table>
<tr><th rowspan="3">Error / module</th><th colspan="2">Disable OSSD = No</th><th colspan="2">Disable OSSD = Yes-ATTENTION</th></tr>
<tr><th colspan="4">Error on output</th></tr>
<tr><th>Switched off</th><th>Switched on</th><th>Switched off</th><th>Switched on</th></tr>
<tr><td colspan="5">Ground fault on SOx+ (output type A) or SOx (output type B)</td></tr>
<tr><td>All SO types</td><td>Not detected</td><td>Detected</td><td>Not detected</td><td>Detected</td></tr>
<tr><td colspan="5">Ground fault on SOx- (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Not detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx+ shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">SOx shorted to 24 VDC (output type B)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">SOx- shorted to 24 VDC (output type A)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">GND shorted to 24 VDC</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx+ (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Detected</td><td rowspan="4">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SOx1x0</td></tr>
<tr><td colspan="5">Cross fault between SOx (output type B) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td rowspan="7">Detected [1)]</td><td rowspan="5">Not detected</td><td rowspan="7">Detected [1)]</td><td rowspan="7">Not detected</td></tr>
<tr><td>X20SLXxxx</td></tr>
<tr><td>X20SRTxxx</td></tr>
<tr><td>X20SO6300</td></tr>
<tr><td>X20SP1130</td></tr>
<tr><td>X20SC2212</td><td rowspan="2">Detected [1)]</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td colspan="5">Cross fault between SOx- (output type A) and the other signal (high)</td></tr>
<tr><td>X20SC0xxx</td><td>Detected</td><td>Detected</td><td>Detected</td><td>Not detected</td></tr>
</table>

Table 372: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
Cross fault between GND and the other signal (high)				
X20SC0xxx	**Not detected**	**Not detected**	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SO6300				
X20SP1130				
X20SC2212				
X67SC4122.L12				
Open circuit (output type A and B)				
X20SC0xxx	**Not detected**	**Not detected**	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0		**Not detected** 2)		**Not detected** 2)
X20SO6300		**Not detected**		**Not detected**
X20SP1130				
X20SC2212				
X67SC4122.L12				
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx	**Not detected**	Detected	**Not detected**	Detected
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				

Table 372: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

2.6.17.2.9 Input circuit diagram

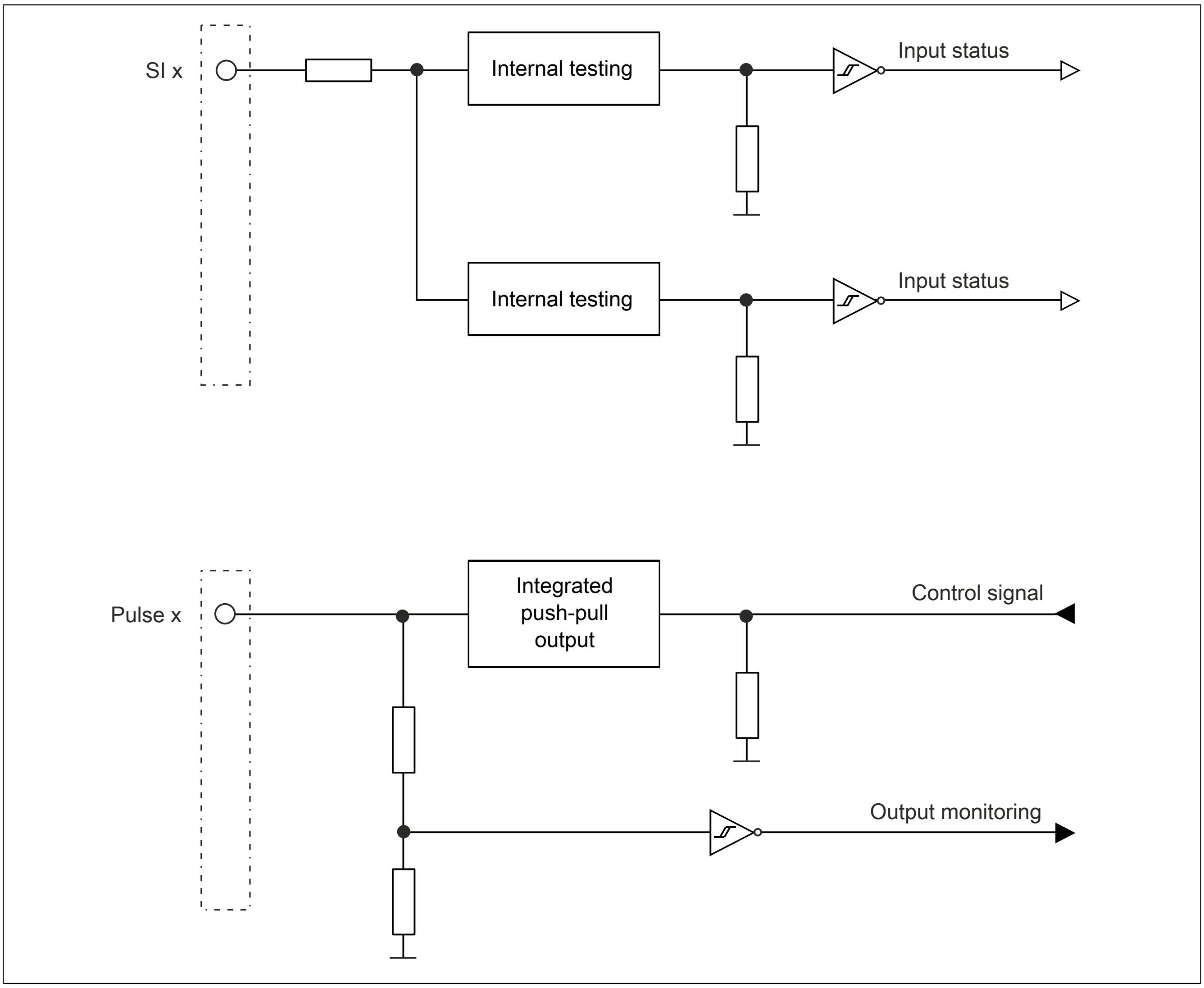


Figure 257: Input circuit diagram

2.6.17.2.10 Type A output circuit diagram

Type A digital output channels are designed for positive and GND switching inside the module.

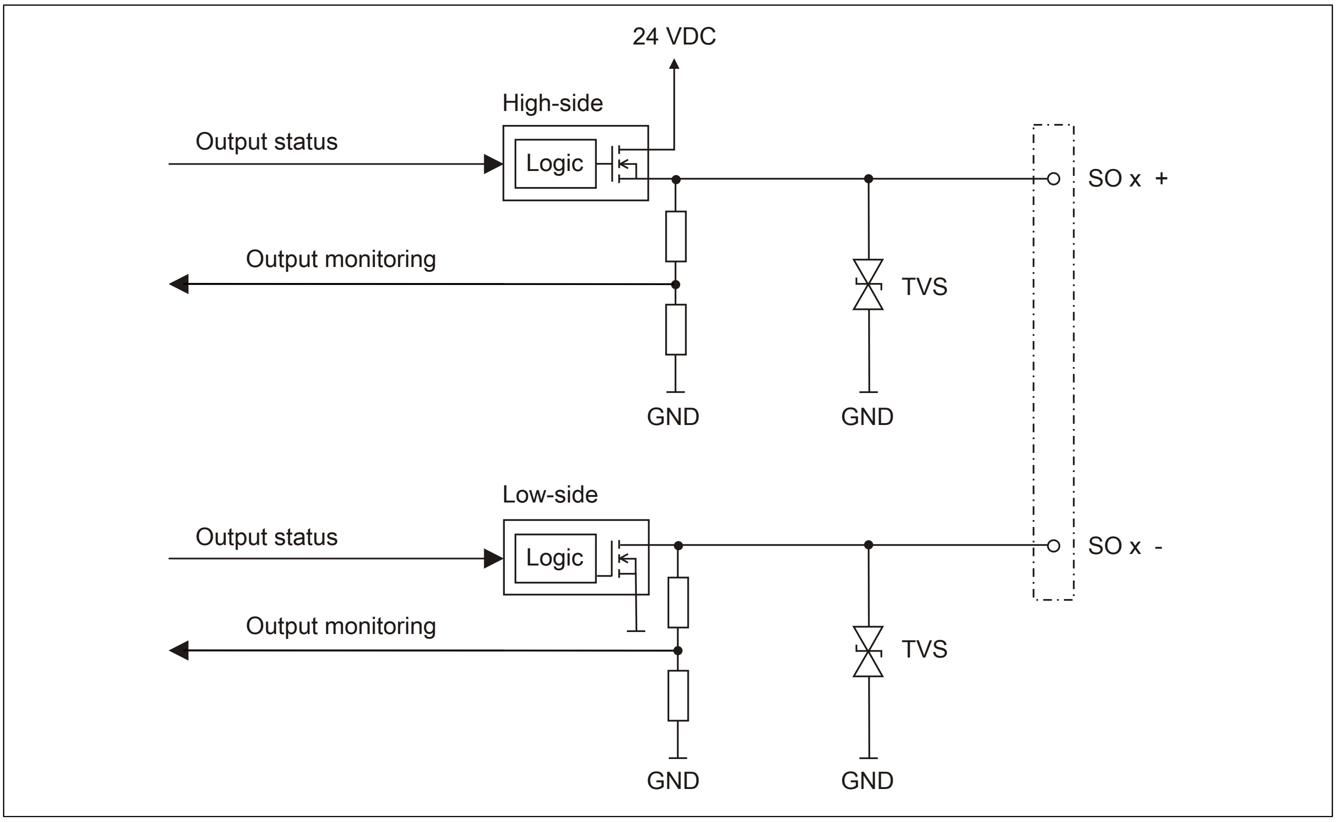


Figure 258: Type A output circuit diagram

2.6.17.2.11 Type B output circuit diagram

Type B digital output channels are designed for positive and positive switching inside the module.

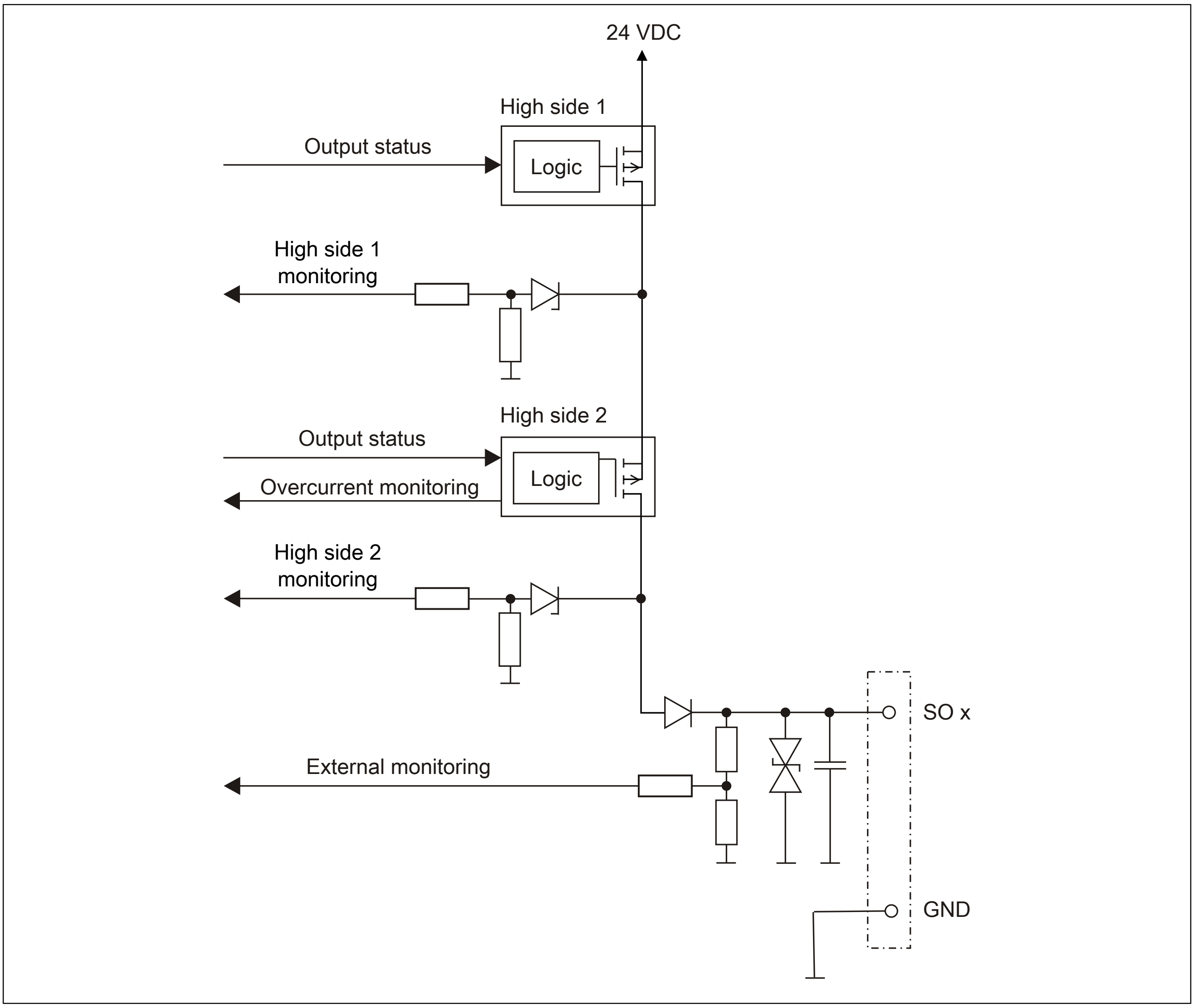


Figure 259: Type B output circuit diagram

2.6.17.2.12 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

2.6.17.2.13 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
100 µs

Maximum I/O update time for input channels when operated as a safe reACTION module
130 µs + Filter time (see chapter "Filter")

Danger!

Up to firmware versions ≤320, a minimum switch-off filter corresponding to 3 times the cycle time of the safe reACTION program must be added with configuration "Pulse Mode = Internal" (default value).

Maximum I/O update time for input channels when operated as a safe mixed module
1150 µs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels when operated as a safe reACTION module
20 µs

Maximum I/O update time for output channels when operated as a safe mixed module
1300 µs

2.6.17.2.14 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

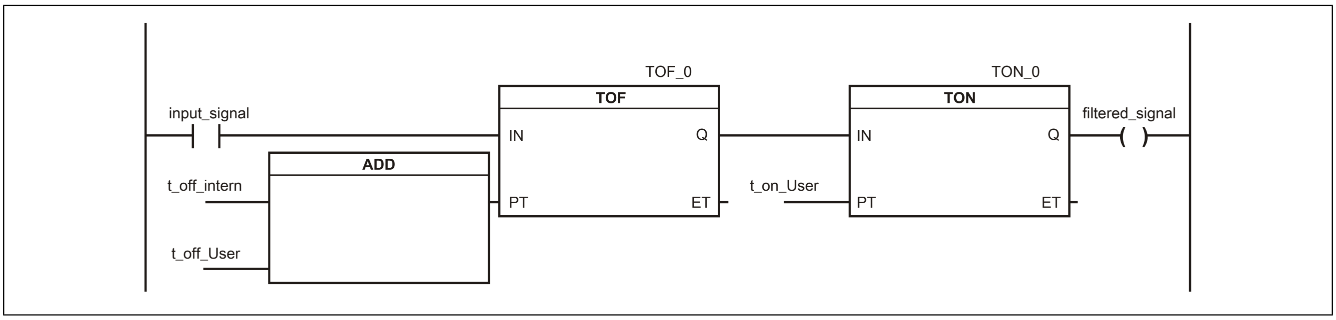


Figure 260: SI input filter - Diagram 1

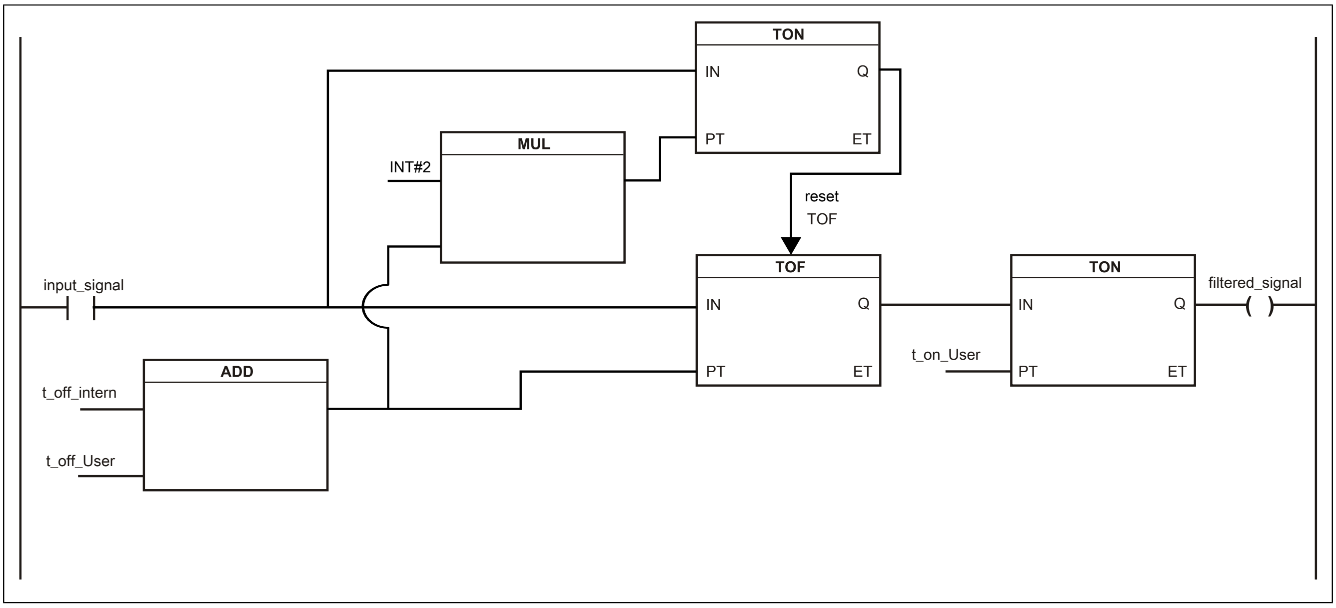


Figure 261: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

2.6.17.2.15 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

2.6.17.2.16 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.17.2.17 reACTION Technology

For more information about reACTION Technology for safety, see Safety technology → SafeDESIGNER → User manual → reACTION Technology for safety in Automation Help.

Danger!

For firmware versions ≤325, module X20SRTxxx must be restarted manually after an application download by SafeDESIGNER. If a restart is not performed, then new parameters or the application itself may not be applied by module X20SRTxxx.

2.6.17.2.18 Blackout mode

Blackout mode allows users to continue execution of the application in lower-level subsystems if components of the B&R system fail. In this way, the B&R system – independently of redundancy technology – makes it possible to respond to system-critical situations based on the specific application.

The use of blackout-capable modules is recommended for the following requirements:

- Exit routines on system failure, e.g. to enable the opening of a press if the system fails.
- Stopping or controlled setting of an output on system failure, e.g. to automatically close inflow valves.
- Deceleration sequences on system failure, e.g. to reduce motor speeds before transmitting a stop command.

If blackout-capable modules are configured accordingly, blackout mode will be carried out if the network connection to the higher-level controller or CPU is interrupted.

As soon as the network disturbance has been corrected, blackout mode is stopped by the modules and bumpless synchronization with the network takes place.

Requirements for operation

The following requirements must be met in order to use blackout mode:

- The module being used must support blackout mode.
- Parameter "Blackout mode" must be enabled in Automation Studio.

2.6.17.2.18.1 Areas of use

Through the use of blackout-capable modules, a part of the control system can also remain functional if a disturbance in the network or X2X Link connection between the modules occurs.

Loss of POWERLINK connection

Initial situation

Several stations in an application are connected to the CPU via network cables. A fault occurs that interrupts data transfer between the CPU and stations.

Effect

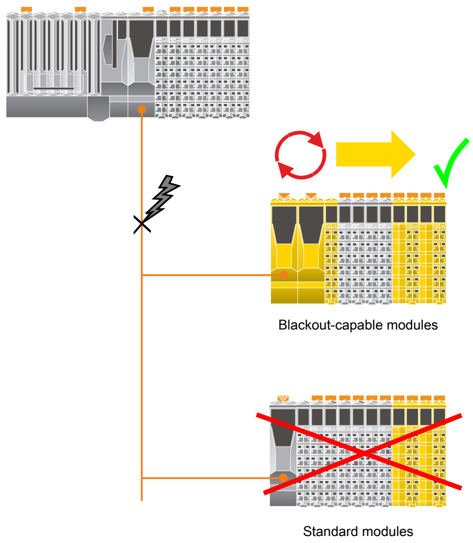
Non-blackout modules are reset and operated according to their default characteristics.

Blackout-capable modules show the following behavior:

- The programmed function continues to be executed.
- Subordinate networks continue to work.
- Data from the CPU is initialized with "0".
- After the disturbance has been corrected, the module bumplessly returns to the higher-level network.

Warning!

Blackout mode causes data from the CPU to be initialized with "0". If blackout mode is used in combination with "output inversion", this can lead to the unwanted setting of outputs.



Loss of X2X Link connection

Initial situation

Modules in an application are connected to the network via X2X Link cables. A defect in the X2X Link cable causes the data transfer between the CPU and modules to be interrupted.

Effect

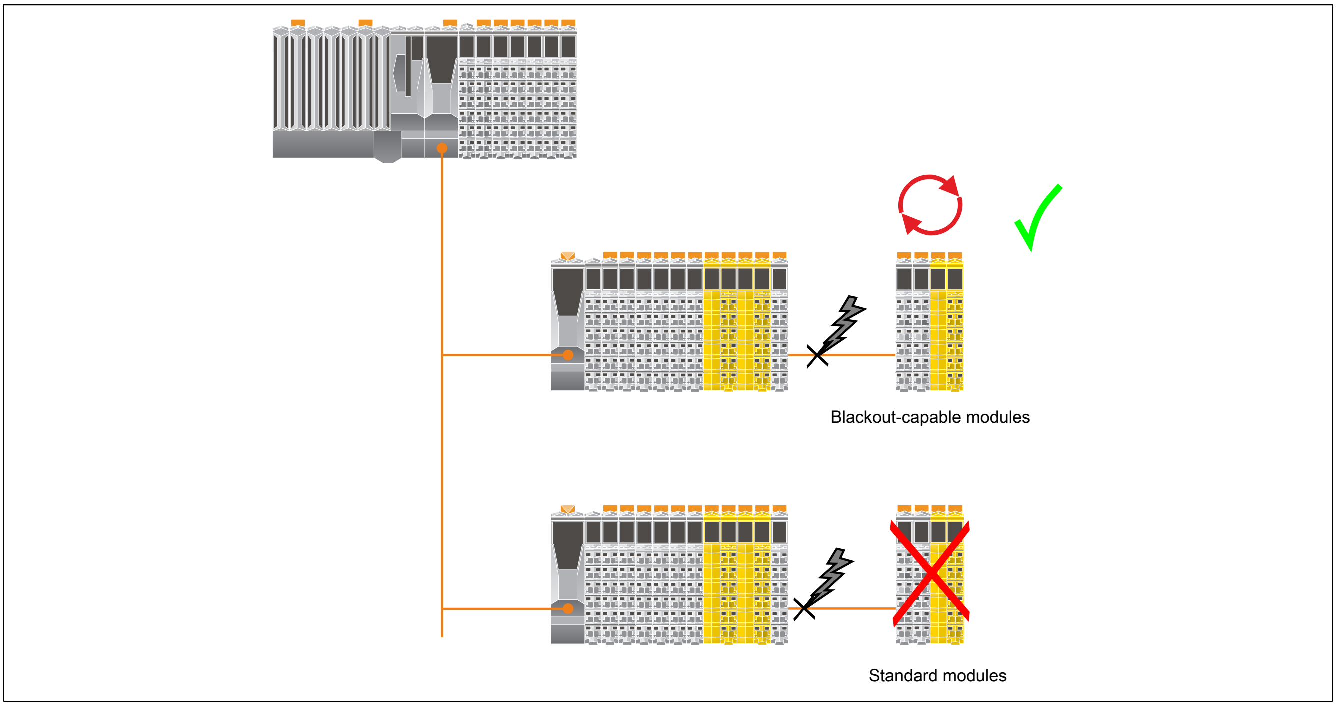
Non-blackout modules are reset and operated according to their default characteristics.

Blackout-capable modules show the following behavior:

- The programmed function continues to be executed.
- Subordinate networks continue to work.
- Data from the CPU is initialized with "0".
- After the disturbance has been corrected, the module bumplessly returns to the higher-level network.

Warning!

Blackout mode causes data from the CPU to be initialized with "0". If blackout mode is used in combination with "output inversion", this can lead to the unwanted setting of outputs.



2.6.17.2.18.2 Programming blackout mode

Blackout mode cannot be detected by the blackout-capable modules themselves. If it is necessary to program specific blackout behavior in an application, an indirect method must therefore be chosen.

One possibility is to implement a counter in the blackout-capable module's higher-level CPU and query it cyclically. Blackout mode would make itself noticeable in this case by a counter value that no longer changes or a counter value of zero.

Blackout-capable modules can be divided into 2 categories:

- **Programmable modules**
  The blackout function is programmed using existing function blocks. In other words, the existing technologies for application programming or reACTION Technology are used.
  The blackout function is executed largely independently of other system components.
- **Standard function modules**
  These modules are not programmable and maintain their default behavior in blackout mode.

2.6.17.2.18.3 Standalone function

The standalone function is an extension of blackout mode. After switching on the power supply, blackout mode is enabled immediately regardless of whether a network connection exists. This means that after switching on the power supply, the module begins executing the most recently saved configuration or application without waiting for activity or synchronization with a higher-level CPU or SafeLOGIC controller.

As soon as the network is active, bumpless synchronization between the module and existing network takes place.

> ## Warning!
>
> **Standalone modules act identically to blackout mode on system startup and until the network connection is established. Their use therefore requires extreme caution!**

Requirements for operation

The following requirements must be met in order to use the standalone function:

- The module being used must support the standalone function.
- Parameter "Standalone mode" must be enabled in Automation Studio.
- For the standalone function on the bus controller (e.g. X20SL8101), blackout mode is enabled for at least 1 module on the local X2X Link network.
- The module must have been operated with a CPU at least once in order to have a valid configuration.

> ## Information:
>
> **The use of the standalone function in connection with DNA is not permitted. Static addresses must be used.**

> ## Warning!
>
> **The following aspects need to be taken into account in particular:**
>
> - **The module must be clearly (and permanently) identified to highlight its distinctive behavior from the standard.**
> - **Service technicians must be well-versed with the special characteristics of these modules.**
> - **Before connecting the terminal block to a module with an enabled standalone function, at least one of the following conditions must be met:**
>   - **It must be ensured that the module is really meant to be operated with the standalone function and the configuration on the module has been checked for correctness.**
>   - **The flashing sequence of the module indicates the "normal, network-connected operational state" of the module.**

Area of application

Initial situation

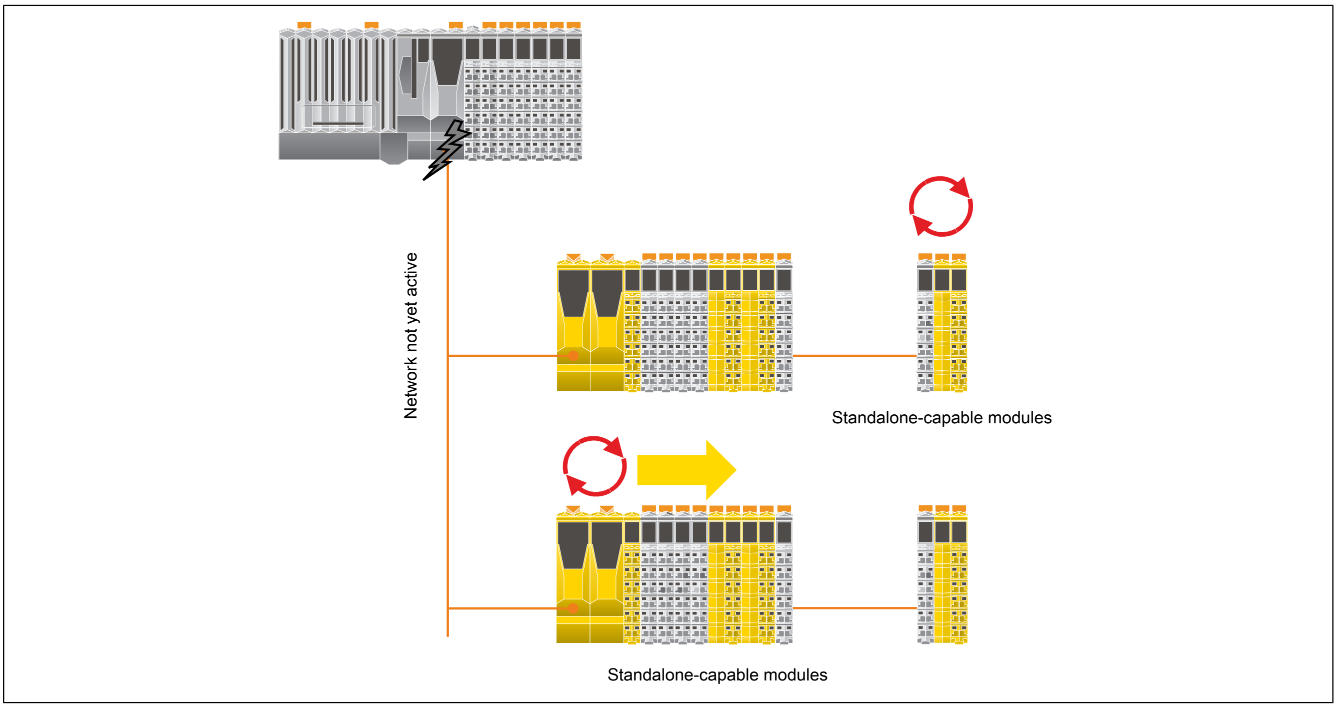
Several stations in an application are connected to the CPU via network cables. After the entire system has been switched off and on, a fault results in the network connection not being established.

Effect

Non-standalone modules are put into the active state only after the application starts up.

Standalone-capable modules show the following behavior:

- The boot procedure is started without waiting on a higher-level network.
- The module behaves identically to blackout mode.
- As soon as the network becomes active, it is bumplessly added to the higher-level network.



2.6.17.2.19 Register description

2.6.17.2.19.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 373: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode</td><td>This parameter enables blackout or standalone mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Off</td><td>Both blackout mode and standalone mode are disabled.</td></tr><tr><td>Blackout mode</td><td>Blackout mode is enabled.</td></tr><tr><td>Standalone mode</td><td>Standalone mode is enabled. This makes it possible to start up the reACTION module without an active communication connection.</td></tr></table></td></tr>
<tr><th colspan="4">reACTION - Properties</th></tr>
<tr><td>reACTION object</td><td>This parameter defines reACTION task to be executed.
Note:
Value "Managed by library" is predefined by default, i.e. the module is operated as a safe mixed module without reACTION Technology.</td><td>Managed by library</td><td>-</td></tr>
<tr><td>Cycle time</td><td>The desired cycle time for the reACTION program is specified with this parameter.</td><td>100</td><td>μs</td></tr>
</table>

Table 374: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 375: I/O configuration parameters: Output signal path

2.6.17.2.19.2 Parameters in SafeDESIGNER

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 376: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

Parameter	Description	Default value	Unit
Manual Configuration	This parameter makes it possible to manually and individually configure the safety response time for the module. The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module. **Parameter value** – **Description** Yes – Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals. No – The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.	No	-
Safe Data Duration	This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module. For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added. • Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)	20000	µs
Additional Tolerated Packet Loss	This parameter specifies the number of additional tolerated lost packets during data transfer. • Permissible values: 0 to 10	0	Packets
Packets per Node Guarding	This parameter specifies the maximum number of packets used for node guarding. • Permissible values: 1 to 255 **Note** • The larger the configured value, the greater the amount of asynchronous data traffic. • This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.	5	Packets

Table 377: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

Parameter	Description	Default value	Unit
Disable OSSD	This parameter can be used to switch off automatic testing of the output driver for all of the module's channels. **Parameter value** – **Description** Yes-ATTENTION – Automatic testing of the output driver is switched off. No – Automatic testing of the output driver is enabled.	No	-

Table 378: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">All available pulse outputs can be specified as "Pulse Source". The default values can be determined using the following table:
<table>
<tr><th>Channel</th><th>Default "Pulse Source"</th></tr>
<tr><td>1, 5</td><td>Channel 1</td></tr>
<tr><td>2, 6</td><td>Channel 2</td></tr>
<tr><td>3, 7</td><td>Channel 3</td></tr>
<tr><td>4, 8</td><td>Channel 4</td></tr>
</table></td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table></td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal "low phases".
Note:
If the selected value is too low, it may result in toggling of the input signal.
• Permissible values: 0 to 500,000 μs (corresponds to 0 to 0.5 s)</td><td>0.
Firmware version 321 or later: 1000</td><td>μs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 μs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>μs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error.
• Permissible values: 0 to 10,000,000 μs (corresponds to 0 to 10 s)</td><td>50000</td><td>μs</td></tr>
<tr><td>Two-Channel Processing Mode</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td>-</td></tr>
</table>

Table 379: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 380: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

2.6.17.2.19.3 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGN-ER</th><th>Access via reACTION program</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.2.0 or later)</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) 1)</td><td>-</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxxyy_state</td><td>Read</td><td>-</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 11 = Channel 1 to 4</td><td>Bit no. 0 to 3 = Channel 1 to 4</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
</table>

Table 381: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGN-ER</th><th>Access via reACTION program</th><th>Data type</th><th>Description</th></tr>
<tr><td>SafetyCycleTime</td><td>(Read) 1)</td><td>-</td><td>-</td><td>UDINT</td><td>Currently used reACTION "Cycle time" (see "I/O configuration parameters: General")</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>-</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>RTCycleTime
(hardware upgrade 1.10.2.0 or later)</td><td>Read</td><td>-</td><td>-</td><td>USINT</td><td>Time needed by the reACTION module to run through the loaded program once</td></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelInputxxyy</td><td>Read</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>SafeBoolSrtInputxx</td><td>-</td><td>Write</td><td>Read</td><td>SAFEBOOL</td><td>SafeLOGIC to reACTION program communication channel</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>-</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelOkxxyy</td><td>Read</td><td>Read</td><td>-</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>Read</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>-</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeDigitalSrtOutputxx</td><td>-</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe reACTION channel xx</td></tr>
<tr><td>SafeOutputModeSelectxx</td><td>-</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Control of output xx.
0: Output xx is controlled by the SafeLOGIC controller and reACTION program.
1: Output xx is controlled only by the reACTION program.</td></tr>
<tr><td>SafeBoolSrtOutputxx</td><td>-</td><td>Read</td><td>Write</td><td>SAFEBOOL</td><td>reACTION program to SafeLOGIC communication channel</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>-</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>-</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateChannelxx</td><td>Read</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>-</td><td>UDINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table><tr><th>Bit 23 to 20</th><th>Bit 19 to 16</th><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr><tr><td>Channel 6</td><td>Channel 5</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table></td></tr>
</table>

Table 381: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

reACTION Technology for safety - Channel list

The following overview shows the assignment of I/O channels to reACTION function blocks. Please note that the number of channels differs depending on the module type.

For example, a module with 4 inputs offers only 4 channels of type "SafeDigitalInput". 8 channels of type "SafeBoolSrtInput" and "SafeBoolSrtOutput" are still available, however.

Channel	Function block	
	rtiDin	rtiDout
SafeDigitalInput01 1)	Channel 1	-
SafeDigitalInput02 1)	Channel 2	-
SafeDigitalInput03 1)	Channel 3	-
SafeDigitalInput04 1)	Channel 4	-
SafeDigitalInput05 1)	Channel 5	-
SafeDigitalInput06 1)	Channel 6	-
SafeDigitalInput07 1)	Channel 7	-
SafeDigitalInput08 1)	Channel 8	-
SafeTwoChannelInput0102 1)	Channel 21	-
SafeTwoChannelInput0304 1)	Channel 22	-
SafeTwoChannelInput0506 1)	Channel 23	-
SafeTwoChannelInput0708 1)	Channel 24	-
PhysicalStateChannel01 1)	Channel 101	-
PhysicalStateChannel02 1)	Channel 102	-
PhysicalStateChannel03 1)	Channel 103	-
PhysicalStateChannel04 1)	Channel 104	-
PhysicalStateChannel05 1)	Channel 105	-
PhysicalStateChannel06 1)	Channel 106	-
DigitalOutput01 1)	Channel 111	-
DigitalOutput02 1)	Channel 112	-
DigitalOutput03 1)	Channel 113	-
DigitalOutput04 1)	Channel 114	-
DigitalOutput05 1)	Channel 115	-
DigitalOutput06 1)	Channel 116	-
SafeBoolSrtInput01	Channel 801	-
SafeBoolSrtInput02	Channel 802	-
SafeBoolSrtInput03	Channel 803	-
SafeBoolSrtInput04	Channel 804	-
SafeBoolSrtInput05	Channel 805	-
SafeBoolSrtInput06	Channel 806	-
SafeBoolSrtInput07	Channel 807	-
SafeBoolSrtInput08	Channel 808	-
SafeDigitalSrtOutput01 1)	-	Channel 901
SafeDigitalSrtOutput02 1)	-	Channel 902
SafeDigitalSrtOutput03 1)	-	Channel 903
SafeDigitalSrtOutput04 1)	-	Channel 904
SafeDigitalSrtOutput05 1)	-	Channel 905
SafeDigitalSrtOutput06 1)	-	Channel 906
SafeOutputModeSelect01 1)	-	Channel 911
SafeOutputModeSelect02 1)	-	Channel 912
SafeOutputModeSelect03 1)	-	Channel 913
SafeOutputModeSelect04 1)	-	Channel 914
SafeOutputModeSelect05 1)	-	Channel 915
SafeOutputModeSelect06 1)	-	Channel 916
SafeBoolSrtOutput01	-	Channel 921
SafeBoolSrtOutput02	-	Channel 922
SafeBoolSrtOutput03	-	Channel 923
SafeBoolSrtOutput04	-	Channel 924
SafeBoolSrtOutput05	-	Channel 925
SafeBoolSrtOutput06	-	Channel 926
SafeBoolSrtOutput07	-	Channel 927
SafeBoolSrtOutput08	-	Channel 928

1) The number of channels actually available depends on the module type.

The following list can be copied directly to the reACTION variable declaration. The channels are defined as constants and can be used with the channel names when developing a reACTION program.

```
VAR CONSTANT
    SafeDigitalInput01 : INT := 1;
    SafeDigitalInput02 : INT := 2;
    SafeDigitalInput03 : INT := 3;
    SafeDigitalInput04 : INT := 4;
    SafeDigitalInput05 : INT := 5;
    SafeDigitalInput06 : INT := 6;
    SafeDigitalInput07 : INT := 7;
    SafeDigitalInput08 : INT := 8;
    SafeTwoChannelInput0102 : INT := 21;
    SafeTwoChannelInput0304 : INT := 22;
    SafeTwoChannelInput0506 : INT := 23;
    SafeTwoChannelInput0708 : INT := 24;
    PhysicalStateChannel01 : INT := 101;
    PhysicalStateChannel02 : INT := 102;
    PhysicalStateChannel03 : INT := 103;
    PhysicalStateChannel04 : INT := 104;
    PhysicalStateChannel05 : INT := 105;
    PhysicalStateChannel06 : INT := 106;
    DigitalOutput01 : INT := 111;
    DigitalOutput02 : INT := 112;
    DigitalOutput03 : INT := 113;
    DigitalOutput04 : INT := 114;
    DigitalOutput05 : INT := 115;
    DigitalOutput06 : INT := 116;
    SafeBoolSrtInput01 : INT := 801;
    SafeBoolSrtInput02 : INT := 802;
    SafeBoolSrtInput03 : INT := 803;
    SafeBoolSrtInput04 : INT := 804;
    SafeBoolSrtInput05 : INT := 805;
    SafeBoolSrtInput06 : INT := 806;
    SafeBoolSrtInput07 : INT := 807;
    SafeBoolSrtInput08 : INT := 808;
    SafeDigitalSrtOutput01 : INT := 901;
    SafeDigitalSrtOutput02 : INT := 902;
    SafeDigitalSrtOutput03 : INT := 903;
    SafeDigitalSrtOutput04 : INT := 904;
    SafeDigitalSrtOutput05 : INT := 905;
    SafeDigitalSrtOutput06 : INT := 906;
    SafeOutputModeSelect01 : INT := 911;
    SafeOutputModeSelect02 : INT := 912;
    SafeOutputModeSelect03 : INT := 913;
    SafeOutputModeSelect04 : INT := 914;
    SafeOutputModeSelect05 : INT := 915;
    SafeOutputModeSelect06 : INT := 916;
    SafeBoolSrtOutput01 : INT := 921;
    SafeBoolSrtOutput02 : INT := 922;
    SafeBoolSrtOutput03 : INT := 923;
    SafeBoolSrtOutput04 : INT := 924;
    SafeBoolSrtOutput05 : INT := 925;
    SafeBoolSrtOutput06 : INT := 926;
    SafeBoolSrtOutput07 : INT := 927;
    SafeBoolSrtOutput08 : INT := 928;

END_VAR
```

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

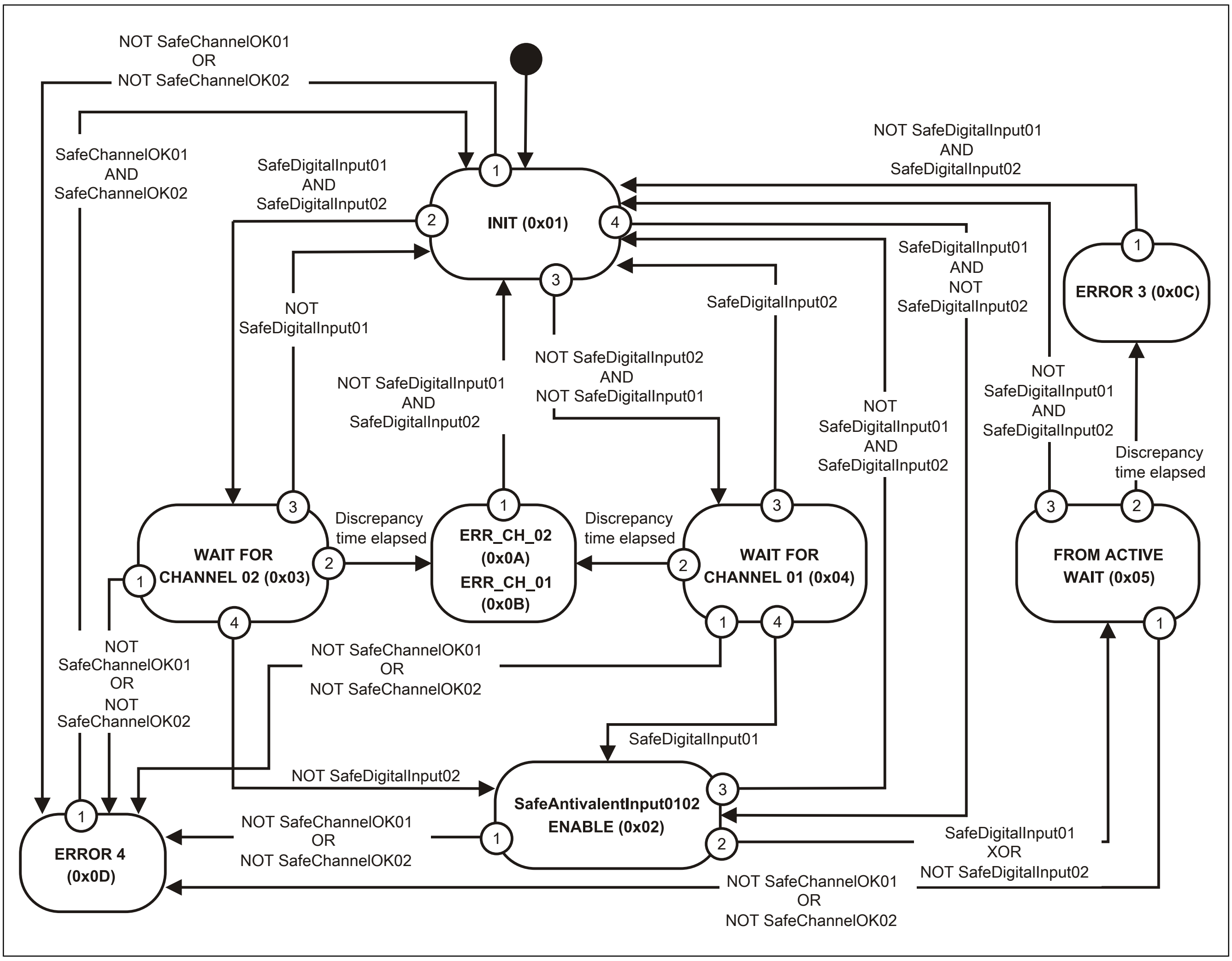


Figure 262: "Antivalent" function block - State diagram

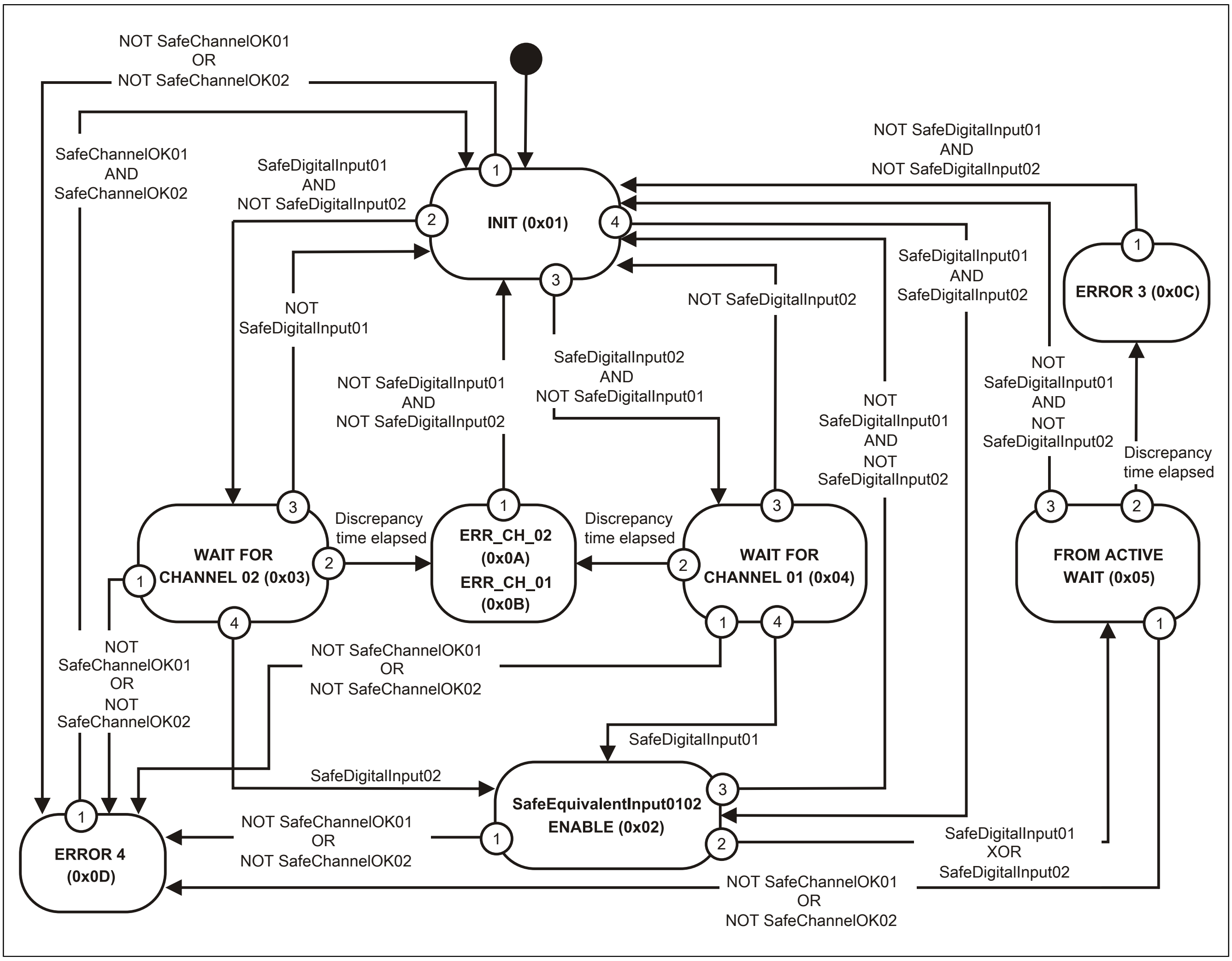


Figure 263: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

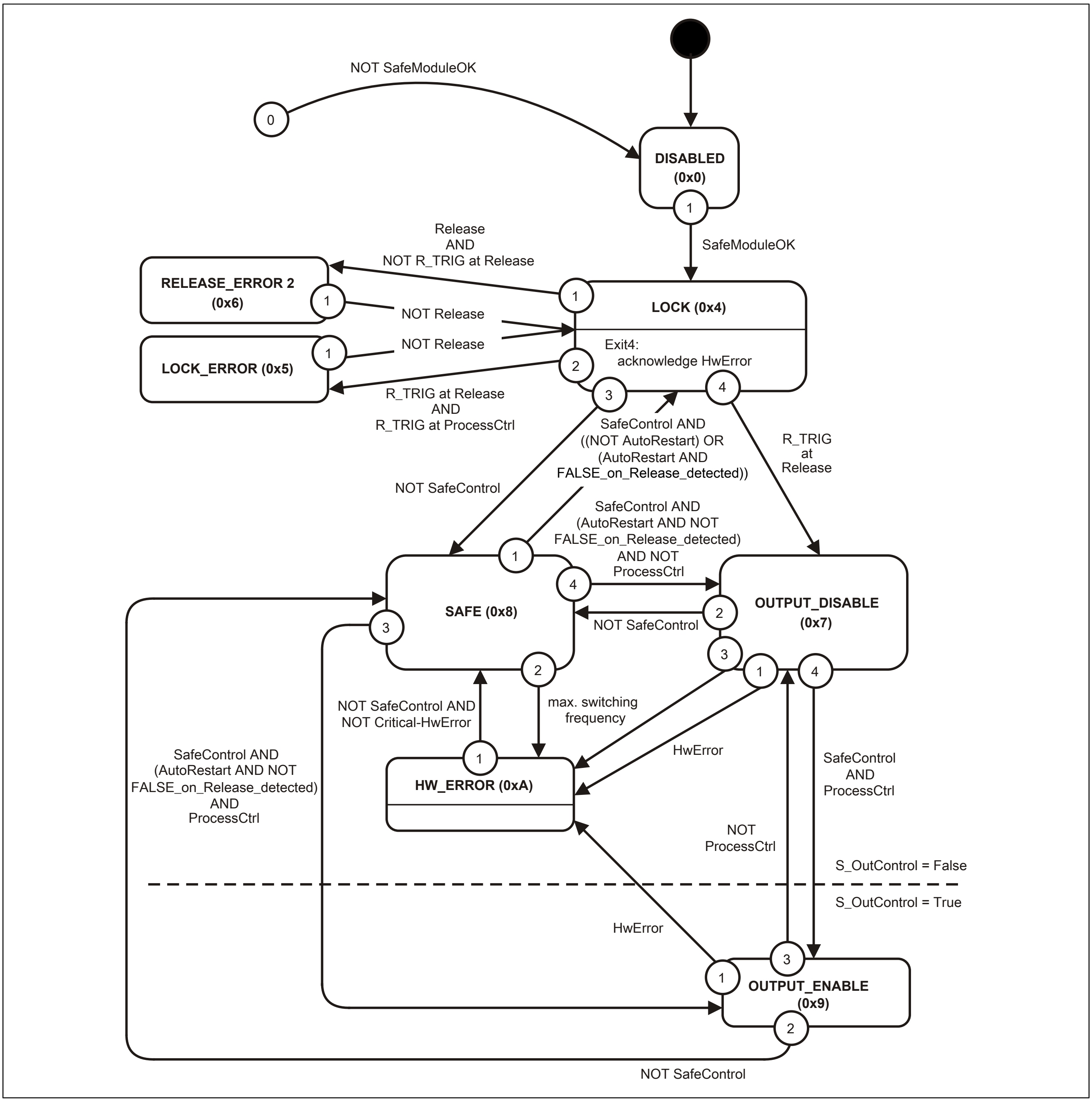
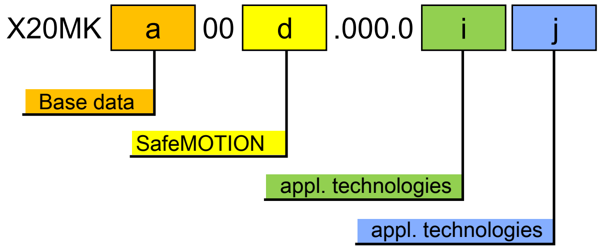


Figure 264: Restart interlock - State diagram

2.6.18 Safety Technology Guarding

"Safety Technology Guarding" defines the range of functions available for applications using X20SL81xx- or X20cSL81xx-series SafeLOGIC controllers. Licenses are stored on a SafeKEY dongle. The functions required for the application must be put together in the X20MK configurator by selecting a SafeKEY with a sufficient amount of memory, a coated/non-coated variant and the necessary technology functions.
Each solution is delivered exclusively as a set (X20MKXXXX.XXX.XXX) consisting of the SafeKEY and the activated licenses for the selected technology functions.

Order key



The following explanation of the order key shows which components and technology functions are included in a configured set.

> ## Information:
>
> **Please note that a set can only be ordered if the combination was generated beforehand using the X20MK configurator.**

Base data

a	X20MK0211: X20 SafeKEY 2 MB	X20MK0213: X20 SafeKEY 8 MB	X20cMK0213: X20 SafeKEY coated 8 MB	X20SF0001: SafeLOGIC 20 SN base	X20SF0002: SafeLOGIC 100 SN base	X20SF0003: SafeLOGIC 300 SN base
1	•			•		
2		•		•		
3	•				•	
4		•			•	
5		•				•
X			•			•
Y			•		•	
Z			•	•		

SafeMOTION

d	X20SF1101: SafeMOTION base functions	X20SF1102: SafeROBOTICS base functions
0		
1	•	
2		•

Application technologies 2

i	X20SF2106: Speed to SafeSPEED converter	Reserved	Reserved	Reserved	Reserved
0					
1	•				

Application technologies 1

j	X20SF2101: Press Control Utilities	X20SF2102: Safe Remanent Data	X20SF2103: PROFIsafe device interface	X20SF2104: C programming extension	Reserved
0					
1	•				
2		•			
3	•	•			
4			•		
5	•		•		
6		•	•		
7	•	•	•		
8				•	
9	•			•	
A		•		•	
B	•	•		•	
C			•	•	
D	•		•	•	
E		•	•	•	
F	•	•	•	•	

2.6.18.1 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation.
The modules' electronics are fully compatible with the corresponding X20 modules.

Information:

For simplification purposes, only images and module IDs of uncoated modules are used in this data sheet.

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, Method 4, exposure 21 days

Contrary to the specifications for X20 system modules without safety certification and despite the tests performed, X20 safety modules are **NOT suited for applications with corrosive gases (EN 60068-2-60)!**

2.6.18.2 Storage medium

Model number	Short description	Figure
	Accessories	
X20MK0211	X20 SafeKEY, 2 MB, for X20SL81xx series	
X20MK0213	X20 SafeKEY, 8 MB, for X20SL81xx series	
X20cMK0213	X20 SafeKEY, coated, 8 MB, for X20SL81xx series	

Table 382: X20MK0211, X20MK0213, X20cMK0213 - Model numbers

2.6.18.2.1 Technical data

Model number	X20MK0211	X20MK0213	X20cMK0213
General information			
Memory size	2 MB	8 MB	
Application memory			
Guaranteed erase/write cycles	100,000		
Certifications			
CE	Yes		
UL	cULus E115267 Industrial control equipment		
DNV GL	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)		
LR	ENV1		
Operating conditions			
Mounting orientation			
Horizontal	Yes		
Vertical	Yes		
Installation elevation above sea level	0 to 2000 m, no limitation		
Degree of protection per EN 60529	IP20		
Ambient conditions			
Temperature			
Operation			
Horizontal mounting orientation	0 to 60°C		-40 to 60°C
Vertical mounting orientation	0 to 60°C		-40 to 60°C
Derating	-		
Storage	-40 to 85°C		
Transport	-40 to 85°C		

Table 383: X20MK0211, X20MK0213, X20cMK0213 - Technical data

2.6.18.3 Technology functions

Model number	Short description
X20SF0001	SafeLOGIC 20 SN base, for projects with up to 20 openSAFETY nodes; actual number determined in the SafeDESIGNER project. Each module and each instance of SafeLOGIC-to-SafeLOGIC communication with a link to global variables counts as a node.
X20SF0002	SafeLOGIC 100 SN base, for projects with up to 100 openSAFETY nodes; actual number determined in the SafeDESIGNER project. Each module and each instance of SafeLOGIC-to-SafeLOGIC communication with a link to global variables counts as a node.
X20SF0003	SafeLOGIC 300 SN base, for projects with up to 300 openSAFETY nodes; actual number determined in the SafeDESIGNER project. Each module and each instance of SafeLOGIC-to-SafeLOGIC communication with a link to global variables counts as a node.
X20SF1101	SafeMOTION base functions, for projects with SafeMOTION functions, access to the following SafeDESIGNER libraries: PLCopen_Motion_SF_2, openSAFETY_Motion_SF
X20SF1102	SafeROBOTICS base functions, for projects with SafeROBOTICS functions, implicitly contains a license for X20SF1101 SafeMOTION base functions, access to the following SafeDESIGNER libraries: PLCopen_Motion_SF_2, openSAFETY_Motion_SF, RoboticCtrl_SF_3
X20SF2101	Press Control Utilities, for press applications, access to the following SafeDESIGNER libraries: PLCopen_Press_SF
X20SF2102	Safe Remanent Data, for storing remanent data on the SafeKEY, access to the following SafeDESIGNER library functions: Utilities_SF/SF_RemanentData_SAFEDWORD, Utilities_SF/SF_RemanentData_SAFEDINT
X20SF2104	C programming extension, support for function blocks created using SafeDESIGNER's C programming extension

2.6.18.4 Version history

Version	Date	Comment
1.141	April 2019	• Chapter 2.6.18.2.1 "Technical data": – Added application memory. – Limited installation elevation to 2000 m. – Updated temperature range. – Updated standards. • Editorial changes.
1.102	September 2018	Chapter 2.6.18 "Safety Technology Guarding": Corrected "Base data": X20cMK0213 = 8 MB
1.101	March 2016	Editorial changes.
1.100	February 2016	First edition as a product-specific manual

Table 384: Version history

2.7 Accessories

2.7.1 Additional equipment for X20 modules and CPUs

The following additional equipment is available for X20 modules and CPUs:

X20 modules	Plain text tag, accessory locking clip and tag holder that also serves as a terminal locking clip
X20 CPU	Plain text tag

Installation of these accessories is described in the section "Installing accessories" on page 133.

Figure 265: Additional equipment for X20 modules

Figure 266: Additional equipment for X20 CPUs

2.7.1.1 Tag holders, terminal locking clips

Model number	Short description	Figure
	Tag holders, terminal locking clips	
X20AC0SC1	X20 terminal locking clip and tag holder for plain text tag	
X20AC0SC1.0100	X20 terminal locking clip and tag holder for plain text tag, 100 pcs. per package	

2.7.1.2 Plain text tag for X20 modules

Model number	Short description	Figure
	Plain text cover	
X20AC0SH1	X20 plain text tag	
X20AC0SH1.0100	X20 plain text tag, 100 pcs. per package	
	X20 legend strips	
X20AC0LB2.0100	X20 slide-in labels for X20 plain text tag, paper, white, perforated, 88 labels on A4 sheets, 100 sheets per package	

Table 385: X20AC0SH1, X20AC0SH1.0100, X20AC0LB2.0100 - Order data

2.7.1.3 Plain text tag for X20 CPUs

Model number	Short description	Figure
	X20 CPU Label	
X20AC0SH2.0010	X20 CPU labels, 10 pcs.	

Table 386: X20AC0SH2.0010 - Order data

2.7.1.4 Accessory locking clips

Model number	Short description	Figure
	Accessory locking clips	
X20AC0AX1	X20 accessory locking clip	
X20AC0AX1.0100	X20 accessory locking clip, 100 pcs. per package	

2.7.2 Locking plates

The locking plate protects the modules on the outside against dirt and damage.

Model number	Short description	Figure
	Locking plates	
X20AC0SL1	X20 locking plate, left	
X20AC0SR1	X20 locking plate, right	
X20AC0SL1.0010	X20 locking plate, left, 10 pcs. per package	
X20AC0SR1.0010	X20 locking plate, right, 10 pcs. per package	

2.7.3 Cable shield clamp

For information on use, see section "X20 cable shield clamp" on page 73.

Model number	Short description	Figure
	Cable shield clamp	
X20AC0SG1.0010	X20 cable shield grounding clamp, 10 pcs. per package	
X20AC0SG1.0100	X20 cable shield grounding clamp, 100 pcs. per package	

2.7.4 Shielding bracket

The X20 shielding bracket provides an easy and space-saving way to ground the cable shielding (see "X20 shielding bracket" on page 73).

Model number	Short description	Figure
	Shielding bracket	
X20AC0SF7.0010	X20 shielding bracket 66 mm 10 pcs.	
X20AC0SF9.0010	X20 shielding bracket 88 mm 10 pcs.	

Table 387: X20AC0SF7.0010, X20AC0SF9.0010 - Order data

2.7.5 End clamp set

Model number	Short description	Figure
	End clamp set	
X20AC0RF1	X20 end clamp set for high vibrations	

Table 388: X20AC0RF1 - Order data

2.7.6 Shield connection clamp

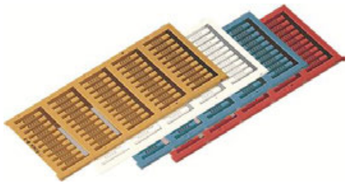
Model number	Short description
	Shield connection clamp
X20AC0SA08.0010	X20 shield connection clamp 3 to 8 mm, 10 pieces
X20AC0SA14.0010	X20 shield connection clamp 3 to 14 mm, 10 pieces
X20AC0SA20.0010	X20 shield connection clamp 5 to 20 mm, 10 pieces
X20AC0SA35.0010	X20 shield connection clamp 20 to 35 mm, 10 pieces

Table 389: X20AC0SA08.0010, X20AC0SA14.0010, X20AC0SA20.0010, X20AC0SA35.0010 - Order data

2.7.7 Terminal labeling

Each terminal connection is identified clearly directly on the terminal. It is also possible to attach tags to label the terminals individually.

The labeling tool is needed for attachment (see "Label tags" on page 138).



X20AC0M0x / X20AC0M1x

X20AC0M21

	Blank X20 tag labels (10.4 x 2.4 mm)
X20AC0M01	Blank X20 tag labels, white, set for 16 modules
X20AC0M02	Blank X20 tag labels, red, set for 16 modules
X20AC0M03	Blank X20 tag labels, blue, set for 16 modules
X20AC0M04	Blank X20 tag labels, orange, set for 16 modules
	Printed X20 tag labels (10.4 x 2.4 mm)
X20AC0M11	Printed X20 tag labels, white, set for 16 modules, label text: Module (modules 1 to 16), terminal (1 to 192)
X20AC0M12	Printed X20 tag labels, red, set for 16 modules, label text: +24 V
X20AC0M13	Printed X20 tag labels, blue, set for 16 modules, label text: GND
X20AC0M14	Printed X20 tag labels, orange, set for 16 modules, label text: Module (modules 1 to 16), terminal (1 to 192)
	Blank X20 tag labels, 10 pcs. per package (10.4 x 2.4 mm)
X20AC0M01.0010	Blank X20 tag labels, white, set for 16 modules, 10 pcs. per package
X20AC0M02.0010	Blank X20 tag labels, red, set for 16 modules, 10 pcs. per package
X20AC0M03.0010	Blank X20 tag labels, blue, set for 16 modules, 10 pcs. per package
X20AC0M04.0010	Blank X20 tag labels, orange, set for 16 modules, 10 pcs. per package
	Printed X20 tag labels, 10 pcs. per package (10.4 x 2.4 mm)
X20AC0M11.0010	Printed X20 tag labels, white, set for 16 modules, 10 pcs. per package, label text: Module (modules 1 to 16), terminal (1 to 192)
X20AC0M12.0010	Printed X20 tag labels, red, set for 16 modules, 10 pcs. per package, label text: +24 V
X20AC0M13.0010	Printed X20 tag labels, blue, set for 16 modules, 10 pcs. per package, label text: GND
X20AC0M14.0010	Printed X20 tag labels, orange, set for 16 modules, 10 pcs. per package, label text: Module (modules 1 to 16), terminal (1 to 192)
	Blank X20 tag labels, large (10.4 x 7.0 mm)
X20AC0M21	Blank X20 tag labels, large white, set for 48 modules
X20AC0M21.0010	Blank X20 tag labels, large white, set for 48 modules, 10 pcs. per package

2.7.8 Labeling tool

The labeling tool is needed to attach the tag labels.

Model number	Short description	Figure
	Labeling tool	
X20AC0MT1	X20 labeling tool for X20 tag labels	

2.7.9 Screwdriver

The screwdriver was developed specially for use with terminal blocks X20TB1E and X20TB1F to prevent damage to the terminals.

Model number	Short description	Figure
	Screwdriver	
X20AC0SD1	B&R screwdriver	

Table 390: X20AC0SD1 - Order data

2.7.10 Pre-assembled cables

2.7.10.1 POWERLINK/Ethernet cables

	Short description, model number		
Length	Connection cable - RJ45 to RJ45		Attachment cables - RJ45 to M12
0.2 m		X20CA0E61.00020	
0.25 m		X20CA0E61.00025	
0.3 m		X20CA0E61.00030	
0.35 m		X20CA0E61.00035	
0.4 m		X20CA0E61.00040	
0.5 m		X20CA0E61.00050	
1 m		X20CA0E61.00100	X67CA0E41.0010
1.5 m		X20CA0E61.00150	
2 m		X20CA0E61.00200	X67CA0E41.0020
3 m		X20CA0E61.00300	X67CA0E41.0030
4 m		X20CA0E61.00400	
5 m		X20CA0E61.00500	X67CA0E41.0050
6 m		X20CA0E61.00600	
8 m		X20CA0E61.00800	
9 m		X20CA0E61.00900	
10 m	X20CA3E61.0100	X20CA0E61.01000	
11 m		X20CA0E61.01100	
12 m		X20CA0E61.01200	
13 m		X20CA0E61.01300	
14 m		X20CA0E61.01400	
15 m	X20CA3E61.0150	X20CA0E61.01500	X67CA3E41.0150 X67CA0E41.0150
16 m		X20CA0E61.01600	
17 m		X20CA0E61.01700	
19 m		X20CA0E61.01900	
20 m	X20CA0E61.0200 X20CA3E61.0200	X20CA0E61.02000	
25 m	X20CA0E61.0250		
30 m	X20CA0E61.0300		
35 m	X20CA0E61.0350		
40 m	X20CA0E61.0400		
50 m	X20CA0E61.0500		X67CA0E41.0500
60 m	X20CA0E61.0600		
100 m	X20CA0E61.1000		
			

Length	Tolerances for cable lengths
X20CA0E61.xxxx	
10 to 100 m	+2% of the length
X20CA0E61.xxxxx	
0.2 to 0.5 m	+0.01 m
1 to 5 m	+0.04 m
6 to 20 m	+1% of the length
X67CAxE41.xxxx	
0 to <10 m	+10 cm
10 m to <50 m	+2% of the length

2.7.10.1.1 Technical data

Product ID	X20CA0E61.xxxxx	X20CA0E61.xxxx	X67CA0E41	X20CA3E61	X67CA3E41
General information					
Note	Lead-free	Halogen-free			
Durability	Flame-retardant in accordance with IEC 60332-3-24	Flame-retardant in accordance with IEC 60332-1-2		Oil resistant in accordance with EN 60811-2-1 Flame-retardant in accordance with IEC 60332-1-2 UV resistant in accordance with UL 2556	
Connection	RJ45 to RJ45		RJ45 to M12; 4-pin	RJ45 to RJ45	RJ45 to M12; 4-pin
Type	Connection cables		Attachment cables	Connection cables	Attachment cables
Cable cross section					
AWG	4x 2x 26 AWG	4x 22 AWG			
mm²	4x 2x 0.14 mm²	4x 0.34 mm²			
RoHS-compliant[1]	Yes				
Cable construction					
Complete shielding	Aluminum-clad foil (overlapping), tinned copper braiding, 85% covering				
Outer sheathing					
Material	Polyvinyl chloride(PVC)	Polyurethane (PUR)			
Color	Black	Green			
Labeling	"B&R" + Model number + Revision number				
Lines					
Wire insulation	Polyethylene (PE)				
Wire colors	Blue-white, blue, orange-white, orange, green-white, green, brown-white, brown	White, yellow, blue, orange			
Type	Tinned copper stranded wire Stranded wire (4x 2x 26 AWG)	Tinned copper stranded wire Fine stranded wire (7x 0.25 mm / 7x 22 AWG)			
Stranding	Twisted pair wires	4-wire twisted pair			
Electrical characteristics					
Operating voltage	Max. 125 V	-			
Conductor resistance	≤145 Ω/km at 20°C	≤120 Ω/km at 20°C			
Transfer properties	Category 5 according to EN 50288-2-2 (2004) / IEC 61156-6 (2002)	Category 5 / Class D up to 100 MHz in accordance with ISO/IEC 11801 (EN50173-1), ISO/IEC 24702 (EN 50173-3)			
Transfer rate	10/100 Mbit/s				
Insulation resistance	≥5 GΩ/km at 20°C	≥500 MΩ/km at 20°C			
Operating conditions					
EN 60529 protection					
Cables	IP67				
Male M12 connector	-		IP67, only when screwed in	-	IP67, only when screwed in
RJ45 connector	IP20, only when connected properly				
Environmental conditions					
Temperature					
Transport	-40 to 80°C	-40 to 70°C			
Fixed installation	-40 to 80°C	-40 to 70°C			
Flexible installation	-10 to 60°C	-20 to 60°C			
Mechanical characteristics					
Dimensions					
Length	Various				15 m
Diameter	6.7 mm ±0.2 mm	6.5 mm ±0.2 mm			
Bend radius	≥8x outer diameter	≥7x outer diameter			
Drag chain data					
Acceleration	-			4 m/s²	
Flex cycles	-			Min. 3 million	
Speed	-			4 m/s	
Weight	0.058 kg/m	0.064 kg/m	0.061 kg/m		

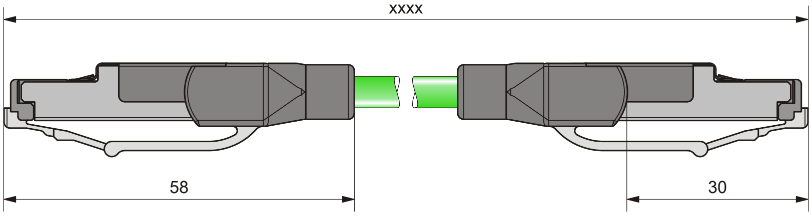
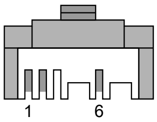
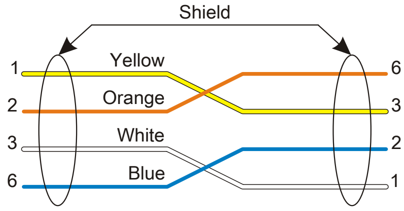
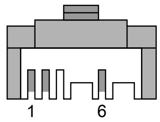
Table 391: X20CAxE61 / X67CAxE41 - Technical data

1) RoHS (Restriction of the use of certain Hazardous Substances) limits the use of the following substances in electrical and electronic devices: lead, mercury, cadmium, chrome VI as well as flame-retardant polybrominated biphenyl (PBB) and polybrominated diphenyl ether (PBDE).

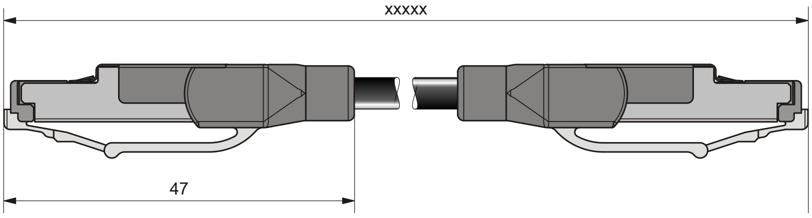
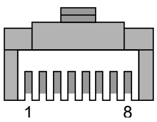
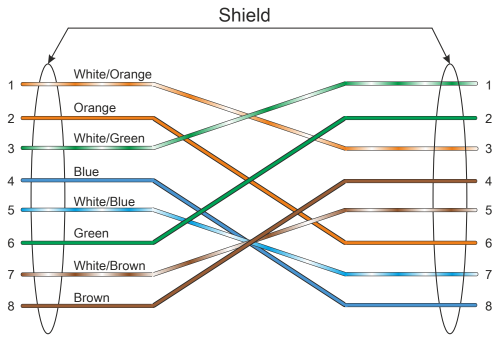
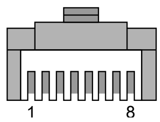
2.7.10.1.2 X20CA0E61.xxxx and X20CA3E61.xxxx

This cable is offered in 2 variants:

- X20CA0E61: Standard design
- X20CA3E61: Can be used in cable drag chains

<table>
<tr><th colspan="3">Dimensions</th></tr>
<tr><td colspan="3">
</td></tr>
<tr><th colspan="3">Pinout</th></tr>
<tr><th>Male RJ45 connector</th><th></th><th>Male RJ45 connector</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

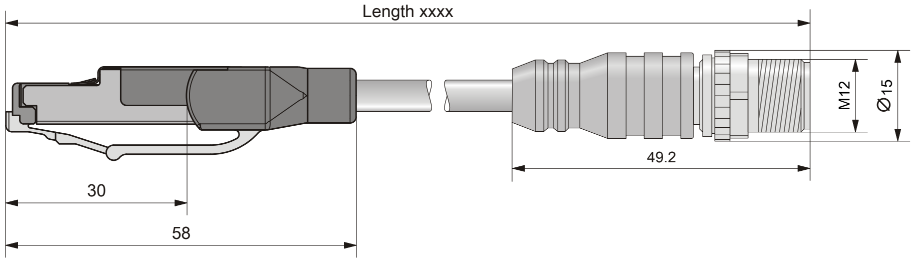
2.7.10.1.3 X20CA0E61.xxxxx

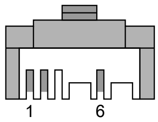
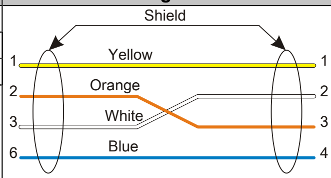
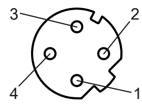
<table>
<tr><th colspan="3">Dimensions</th></tr>
<tr><td colspan="3">
</td></tr>
<tr><th colspan="3">Pinout</th></tr>
<tr><th>Male RJ45 connector</th><th></th><th>Male RJ45 connector</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

2.7.10.1.4 X67CA0E41.xxxx and X67CA3E41.xxxx

This cable is offered in 2 variants:

- X67CA0Exx: Standard variant
- X67CA3Exx: Can be used in cable drag chains

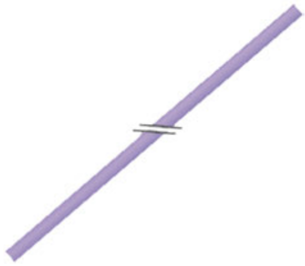
Dimensions


Pinout				
Male RJ45 connector	**Pin**	**Description**	**Diagram**	**M12 connector**
	1 - 1	TXD		 D-coded
	2 - 3	RXD		
	3 - 2	TXD\		
	6 - 4	RXD\		

2.7.10.2 X2X Link cables

	Short description, model number	
Length	Attachment cables	Connection cables
0.3 m		X20CA0X68.0003
1 m	X20CA0X48.0010	X20CA0X68.0010
2 m	X20CA0X48.0020	X20CA0X68.0020
5 m	X20CA0X48.0050	X20CA0X68.0050
10 m	X20CA0X48.0100	X20CA0X68.0100
20 m	X20CA0X48.0200	
		

Length	Tolerances for cable lengths
0 to <10 m	+10 cm
10 m to <50 m	+2% of the length

	Short description, model number
Length	X2X Link cables for custom assembly
100 m	X67CA0X99.1000
500 m	X67CA0X99.5000
	

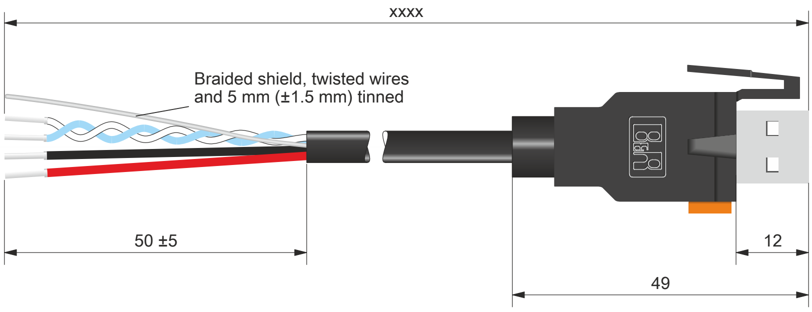
2.7.10.2.1 Technical data

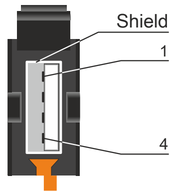
Product ID	X20CA0X48	X20CA0X68	X67CA0X99
General information			
Note			Halogen-free
Durability	Flame-retardant according to VW-1 and FT1		Flame-retardant
Connection	USB A, male		-
Type	Attachment cables	Connection cables	-
Cable cross section			
Data cables			
AWG	2x 24 AWG		
mm²	2x 0.25 mm²		
Supply lines			
AWG	2x 22 AWG		
mm²	2x 0.34 mm²		
RoHS-compliant[1)]	Yes		
Cable construction			
Signal lines			
Shield	Paired shield with aluminum foil		
Stranding	Twisted pair wires		
Cable stranding	7/36 (28 AWG) with filler		0.35 mm² (22 AWG) with filler
Complete shielding	Tinned copper braiding, coverage >85%		
Outer sheathing			
Material	Polyvinyl chloride(PVC)		Thermoplastic polyurethane (TPU)
Color	Black		Violet
Labeling	"B&R" + Model number + Revision number		B&R X67CA0X99.xxxx
Lines			
Type	Tinned copper stranded wire		Tinned copper ETB1 Data line: Fine stranded wire (19x 0.13 mm) Supply line: Fine stranded wire (19x 0.15 mm)
Wire colors			
Data cables	Blue, white		
Supply lines	Red, black		
Wire insulation			
Data cables	Polyethylene foam		Cell polyethylene (PE)
Supply lines	SR PVC		Polyethylene (PE)
Electrical characteristics			
Operating voltage	30 V		Max. 2500 VAC
Degree of insulation	-		Category II in accordance with IEC 61076-2
Conductor resistance	Data line: <93.2 Ω/km at 20°C Supply line: <55 Ω/km at 20°C		Data line: ≤78 Ω/km Supply line: ≤55 Ω/km
Operating conditions			
EN 60529 protection			
Connector/Coupling	IP20, only when connected properly		-
Environmental conditions			
Temperature	-25 to 80°C		-40 to 80°C
Fixed installation	-25 to 80°C		-40 to 80°C
Flexible installation	-20 to 80°C		-25 to 60°C
Mechanical characteristics			
Dimensions			
Length	Various		
Diameter	7 mm ± 0.19 mm		6.9 mm ±0.2 mm
Bend radius	≥8x outer diameter		≥15x outer diameter
Drag chain data			
Acceleration	-		Max. 4 m/s²
Flex cycles	-		Min. 2 million
Speed	-		Max. 3 m/s
Weight	-		0.063 kg/m

Table 392: X20CA0Xx8 / x67CA0X99 - Technical data

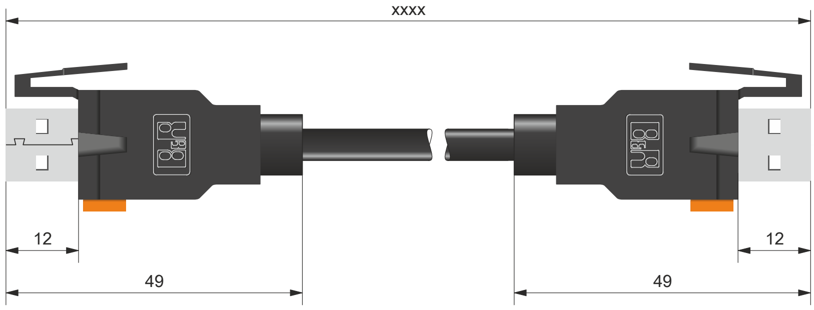
1) RoHS (Restriction of the use of certain Hazardous Substances) limits the use of the following substances in electrical and electronic devices: lead, mercury, cadmium, chrome VI as well as flame-retardant polybrominated biphenyl (PBB) and polybrominated diphenyl ether (PBDE).

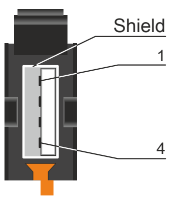
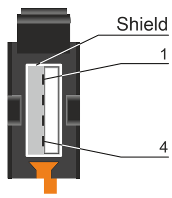
2.7.10.2.2 X20CA0X48.xxxx



Pinout				
Open	**Pin**	**Name**	**Wire color**	**Male connector**
For custom Wiring	1	X2X+	Red	 Keying for X3A
	2	X2X\	Blue	
	3	X2X	White	
	4	X2X⊥	Black	
	Male connector	SHLD	Shield	

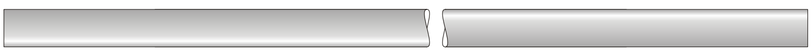
2.7.10.2.3 X20CA0X68.xxxx



Pinout				
Male connector	**Pin**	**Name**	**Wire color**	**Male connector**
	1	X2X+	Red	
	2	X2X\	Blue	
	3	X2X	White	
	4	X2X⊥	Black	
	Male connector	SHLD	Shield	

2.7.10.2.4 X67CA0X99.xxxx

Dimensions



Pinout			
	Description	**Wire colors**	
For custom Wiring	X2X+	Red	For custom Wiring
	X2X	White	
	X2X⊥	Black	
	X2X\	Blue	
	SHLD	-	

2.8 International and national certifications

Products and services from B&R comply with applicable standards. This includes international standards from organizations such as ISO, IEC and CENELEC, as well as national standards from organizations such as UL, CSA, DNV GL, etc. We are committed to ensuring the reliability of our products in an industrial environment.

Information:

Certifications that apply to a particular module are available at the following places:

- **The data sheet's technical data under "Certifications".**
- **At www.br-automation.com under "Products" in the "Certifications" area of the technical data**
- **On the side of the module housing**

2.8.1 Overview of certifications

Markings	Function	certificate authority	Region
CE	CE marking	Notified bodies	Europe (EU)
open SAFETY certified product	Functional safety (CE)	Notified bodies	Europe (EU)
Ex	Explosion protection (CE)	Notified bodies	Europe (EU)
c UL US LISTED	Underwriters Laboratories Inc. (UL) (certification for US and Canada)	UL	Canada USA
SP C US	Canadian Standards Association (CSA) (certification for US and Canada)	CSA	Canada USA
TYPE APPROVED PRODUCT DNV·GL MARITIME	Det Norske Veritas - Germanischer Lloyd (DNV GL)	DNV GL	Germany Norway
LR	Lloyd's Register (LR)	LR	Great Britain
РСТ	GOST-R	Federal agency on technical regulating and metrology	Former Russian Federation
EAC	Eurasian Conformity (EAC)	Federal agency on technical regulating and metrology	Eurasian Trade Union
KC	Korean Conformity (KC)	Radio Research Agency (RRA)	Korea
	Regulatory Compliance Mark (RCM)	ACMA	Australia Oceania

2.8.2 EU directives and standards (CE)

CE markings

Europe (EU)

The respective product complies with all applicable EU directives and relevant harmonized standards.

Certification of these products is performed in cooperation with accredited testing laboratories.

EMC directive 2014/30/EU

All devices satisfy the protection requirements of the "EMC directive" and are designed for industrial use:

Applicable standards from this directive:

EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing
EN 61000-6-2	Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity standard for industrial environments
EN 61000-6-4	Electromagnetic compatibility (EMC) - Part 6-4: Generic standards ; Generic standards - Emission standard for industrial environments

Low voltage directive 2014/35/EU

The low voltage directive applies to electrical equipment with a nominal voltage from 50 to 1000 VAC and from 75 to 1500 VDC.
All devices within the area of application of this directive satisfy the its protection requirements.

Applicable standard from this directive:

EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing

The corresponding declaration of conformity is available for download from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity PLC

Machinery directive 2006/42/EC

Functional safety

open SAFETY certified product

Europe (EU)

In accordance with the machinery directive, safety technology products are designed, developed, tested and labeled for special applications providing protection to machinery and personnel.

Certification of these products is performed exclusively in cooperation with EU-authorized bodies (Notified Bodies).

Applicable standards from this directive:

SIL 3:	
IEC 61508-1	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 1: General requirements
IEC 61508-2	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
IEC 61508-3	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements
IEC 61508-4	Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations
EN 62061	Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems
IEC 61511-1	Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements
PL e, Cat. 4:	
EN ISO 13849-1	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design
EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing

Declarations of conformity, certificates and any other safety-related documentation can be downloaded from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity FS PLC

Certificate

Homepage > Downloads > Certificates > Safety technology > X20, X67

User's manual

Homepage > Downloads > Safety technology > Integrated Safety Technology User's Manual

ATEX directive 2014/34/EU

ATEX / Zone 2

II 3G Ex nA nC IIA T5 Gc

Europe (EU)

Products with this marking are suitable for use in potentially explosive environments. The X20 system is certified for use in environments with explosive gases with a normal level of safety (Zone 2).

Certification of these products is performed exclusively in cooperation with EU-authorized bodies (Notified Bodies).

Each module is also accompanied by an information sheet providing detailed installation and safety guidelines.

Applicable standards from this directive:

EN 60079-0	Explosive atmospheres Part 0: Equipment - General requirements
EN 60079-15	Explosive atmospheres - Part 15: Equipment protection by type of protection "n"

The declaration of conformity and certificate can be downloaded from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity ATEX X20

Certificate

Homepage > Downloads > Certificates > ATEX > X20 > FTZÙ 09 ATEX 0083X

2.8.2.1 Overview of standards

Standard	Description
EN ISO 13849-1	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design
EN 50581	Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances
EN 55011 (CISPR 11)	Industrial, scientific and medical equipment - Radio frequency disturbance characteristics - Limits and methods of measurement
EN 55016-2-1 (CISPR 16-2-1)	Specification for radio disturbance and immunity measuring apparatus and methods - Part 2-1: Methods of measurement of disturbances and immunity - Conducted disturbance measurements
EN 55016-2-3 (CISPR 16-2-3)	Specification for radio disturbance and immunity measuring apparatus and methods - Part 2-3: Methods of measurement of disturbances and immunity - Radiated disturbance measurements
EN 55022 (CISPR 22)	Information technology equipment - Radio disturbance characteristics - Limits and methods of measurement
EN 60068-2-6	Environmental testing - Part 2-6: Procedures - Test Fc: Vibration (sinusoidal)
EN 60068-2-27	Environmental testing - Part 2-27: Test procedure - Test Ea and guidance: Shock
EN 60068-2-31[1)]	Environmental testing - Part 2-31: Test procedure - Test Ec: Rough handling shocks, mainly for devices
EN 60079-0	Explosive atmospheres Part 0: Equipment - General requirements
EN 60079-15	Explosive atmospheres - Part 15: Equipment protection by type of protection "n"
EN 60529	Degrees of protection provided by enclosures (IP code)
EN 60664-1	Insulation coordination for equipment within low-voltage systems - Part 1: Principles, requirements and tests
EN 60721-3-2	Classification of environmental conditions - Part 3: Classification of groups of environmental parameters and their severities - Section 2: Transport and handling
EN 60721-3-3	Classification of environmental conditions - Part 3: Classification of groups of environmental parameters and their severities - Section 3: Stationary use at weather-protected locations
EN 61000-4-2	Electromagnetic compatibility (EMC) - Part 4-2: Testing and measurement techniques - Electrostatic discharge immunity test
EN 61000-4-3	Electromagnetic compatibility (EMC) - Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test
EN 61000-4-4	Electromagnetic compatibility (EMC) - Part 4-4: Testing and measurement techniques - Electrical fast transient/burst immunity test
EN 61000-4-5	Electromagnetic compatibility (EMC) - Part 4-5: Testing and measuring techniques - Surge immunity test
EN 61000-4-6	Electromagnetic compatibility (EMC) - Part 4-6: Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields
EN 61000-4-8	Electromagnetic compatibility (EMC) - Part 4-8: Testing and measuring techniques - Power frequency magnetic field immunity test
EN 61000-4-11	Electromagnetic compatibility (EMC) - Part 4-11: Testing and measuring techniques - Voltage dips, short interruptions and voltage variations
EN 61000-4-29	Electromagnetic compatibility (EMC) - Part 4-29: Testing and measurement techniques - Voltage dips, short interruptions and voltage variations on d.c. input power port immunity tests
EN 61000-6-2	Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity standard for industrial environments
EN 61000-6-4	Electromagnetic compatibility (EMC) - Part 6-4: Generic standards - Emission standard for industrial environments
EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing
IEC 61508-1	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 1: General requirements
IEC 61508-2	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
IEC 61508-3	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements
IEC 61508-4	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 4: Definitions and abbreviations
IEC 61511-1	Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements
EN 62061	Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems

1) Replacement for EN 60068-2-32

2.8.2.2 Requirements for immunity to disturbances

Immunity	Test carried out in accordance with	Requirements in accordance with
Electrostatic discharge (ESD)	EN 61000-4-2	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
High-frequency electromagnetic fields (HF field)	EN 61000-4-3	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
High-speed transient electrical disturbances (Burst)	EN 61000-4-4	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Surge voltages (Surge)	EN 61000-4-5	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Conducted disturbances	EN 61000-4-6	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Magnetic fields with electrical frequencies	EN 61000-4-8	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Voltage dips (AC) Short-term interruptions (AC) Voltage fluctuations (AC)	EN 61000-4-11	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Short-term interruptions (DC) Voltage fluctuations (DC)	EN 61000-4-29	EN 61131-2: Product standard - Programmable logic controllers

Evaluation criteria for performance

Criteria	During testing	After testing
A	The PLC system shall continue to operate as intended. No loss of function or performance.	The PLC system shall continue to operate as intended.
B	Degradation of performance accepted. No change of operating mode. No irreversible loss of stored data.	The PLC system shall continue to operate as intended. Temporary degradation of performance must be self-recoverable.
C	Loss of functions accepted, but no destruction of hardware or software (program or data).	The PLC system shall continue to operate as intended automatically, after manual restart or power off/power on.
D	Degradation or failure of functionality that can no longer be restored.	PLC system permanently damaged or destroyed.

Electrostatic discharge (ESD)

Test carried out in accordance with EN 61000-4-2	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Contact discharge (CD) to conductive external parts	±4 kV Criteria B	
Air discharge (AD) to insulating external parts	±8 kV Criteria B	

High-frequency electromagnetic fields (HF field)

Test carried out in accordance with EN 61000-4-3	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Housing, completely wired	80 MHz to 1 GHz, 10 V/m 1.4 GHz to 2 GHz, 3 V/m 2 GHz to 2.7 GHz, 1 V/m Criteria A	

High-speed transient electrical disturbances (Burst)

Test carried out in accordance with EN 61000-4-4	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	±2 kV / 5 kHz Criteria B	
AC power outputs	±2 kV / 5 kHz 1) Criteria B	±2 kV / 5 kHz Criteria B
AC other I/Os	±2 kV / 5 kHz 1) Criteria B	-
DC mains inputs/outputs	±2 kV / 5 kHz 1) Criteria B	
Other I/Os and interfaces	±1 kV / 5 kHz 1) Criteria B	

1) Only for connections with a permitted line length >3 m.

Surge voltages (Surge)

Test carried out in accordance with EN 61000-4-5	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC mains inputs/outputs Line / line	±1 kV Criteria B	
AC mains inputs/outputs Line / ground	±2 kV Criteria B	
DC mains inputs/outputs Line / line	±0.5 kV 1) Criteria B	±0.5 kV Criteria B
DC power inputs Line / ground	±0.5 kV 1) Criteria B	±0.5 kV Criteria B
DC power outputs Line / ground	±0.5 kV 1) Criteria B	±0.5 kV Criteria B
Signal connections, unshielded Line / ground	±1 kV 1) Criteria B	
All shielded lines Line / ground	±1 kV 1) Criteria B	-

1) Only for connections with a permitted line length >30 m.

Conducted disturbances

Test carried out in accordance with EN 61000-4-6	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC mains inputs/outputs	10 V 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	
DC mains inputs/outputs	10 V 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	
Other I/Os and interfaces	10 V 1) 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	

1) Only for connections with a permitted line length >3 m.

Magnetic fields with electrical frequencies

Test carried out in accordance with EN 61000-4-8	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Housing, completely wired	30 A/m 3 axes (x, y, z) 50/60 Hz 1) Criteria A	

1) Mains frequency per manufacturer data

Voltage dips

<table>
<tr><th>Test carried out in accordance with
EN 61000-4-11</th><th>Requirements in accordance with
EN 61131-2 / Zone B</th><th>Requirements in accordance with
EN 61000-6-2</th></tr>
<tr><td rowspan="3">AC power inputs</td><td colspan="2">0% residual voltage
250/300 periods (50/60 Hz)[1]
20 attempts
Criteria C</td></tr>
<tr><td colspan="2">40% residual voltage
10/12 periods (50/60 Hz)[1]
20 attempts
Criteria C</td></tr>
<tr><td colspan="2">70% residual voltage
25/30 periods (50/60 Hz)[1]
20 attempts
Criteria C</td></tr>
</table>

1) Mains frequency per manufacturer data

Short-term interruptions

Test carried out in accordance with EN 61000-4-11 / EN 61000-4-29	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	0% residual voltage 0.5 periods (50/60 Hz)[1] 20 interruptions Criteria A	0% residual voltage 1 period (50/60 Hz)[1] 3 interruptions Criteria B
DC power inputs	0% residual voltage ≥10 ms (PS2) 20 interruptions Criteria A	-

1) Mains frequency per manufacturer data

Voltage fluctuations

Test carried out in accordance with EN 61000-4-11 / EN 61000-4-29	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	-15% / +10% Test duration per 30 minutes Criteria A	-
DC power inputs	-15% / +20% Test duration per 30 minutes Criteria A	-

2.8.2.3 Emission requirements

Phenomenon	Test carried out in accordance with	Limits in accordance with
Emissions related to lines	EN 55011 / EN 55022 EN 55016-2-1	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-4: Generic standard - Emissions in industrial sectors
Radiated emissions	EN 55011 / EN 55022 EN 55016-2-3	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-4: Generic standard - Emissions in industrial sectors

Emissions related to lines

Test carried out in accordance with EN 55011 / EN 55022 / EN 55016-2-1	Limits in accordance with EN 61131-2 / Zone B	Limits in accordance with EN 61000-6-4
AC mains connection 150 kHz to 30 MHz	150 kHz to 500 kHz 79 dB (µV) quasi-peak value 66 dB (µV) average value	
	500 kHz to 30 MHz 73 dB (µV) quasi-peak value 60 dB (µV) average value	
Telecommunications / network connection 150 kHz to 30 MHz	-	150 kHz to 500 kHz 97 to 87 dB (µV) quasi-peak value 53 to 40 dB (µA) quasi-peak value 84 to 74 dB (µV) average value 40 to 30 dB (µA) average value
	-	500 kHz to 30 MHz 87 dB (µV) quasi-peak value 43 dB (µA) quasi-peak value 74 dB (µV) average value 30 dB (µA) average value

Radiated emissions

Test carried out in accordance with EN 55011 / EN 55022 / EN 55016-2-3	Limits in accordance with EN 61131-2 / Zone B	Limits in accordance with EN 61000-6-4
Electric field / Measured from 10 m 30 MHz to 1 GHz	30 MHz to 230 MHz 40 dB (µV/m) quasi-peak value	
	230 MHz to 1 GHz 47 dB (µV/m) quasi-peak value	
Electric field / Measured from 3 m 1 GHz to 6 GHz[1)]	-	1 GHz to 3 GHz[1)] 76 dB (µV/m) peak value 56 dB (µV/m) average value
	-	3 GHz to 6 GHz[1)] 80 dB (µV/m) peak value 60 dB (µV/m) average value

1) Depending on highest internal frequency

2.8.2.4 Mechanical conditions

Testing	Test carried out in accordance with	Requirements in accordance with
Vibration (sinusoidal) / Operation	EN 60068-2-6	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-3 / Class 3M4
Shock / Operation	EN 60068-2-27	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-3 / Class 3M4
Vibration (sinusoidal) / Transport (packaged)	EN 60068-2-6	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
		EN 60721-3-2 / Class 2M3
Shock / Transport (packaged)	EN 60068-2-27	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
Free fall / Transport (packaged)	EN 60068-2-31[1)]	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-2 / Class 2M1
Toppling / Transport (packaged)	EN 60068-2-31	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
		EN 60721-3-2 / Class 2M3

1) Replacement for EN 60068-2-32

Vibration (sinusoidal) / Operation

Test carried out in accordance with EN 60068-2-6	Requirements in accordance with EN 61131-2		Requirements in accordance with EN 60721-3-3 / Class 3M4	
Vibration (sinusoidal) / Operation[1)]	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**
	5 to 8.4 Hz	Deflection 3.5 mm	2 to 9 Hz	Deflection 3 mm
	8.4 to 150 Hz	Acceleration 1 g[2)]	9 to 200 Hz	Acceleration 1 g[2)]
	20 sweeps for each axis[3)]			

1) Uninterrupted duty with movable frequency in all 3 axes (x, y, z); 1 octave per minute
2) 1 g = 10 m/s²
3) 2 sweeps = 1 frequency cycle (fmin → fmax → fmin)

Shock / Operation

Test carried out in accordance with EN 60068-2-27	Requirements in accordance with EN 61131-2	Requirements in accordance with EN 60721-3-3 / Class 3M4
Shock / Operation[1)]	Acceleration 15 g Duration 11 ms 18 shocks	Acceleration 10 g Duration 11 ms 18 shocks

1) Pulse (half-sine) stress in all 3 axes (x, y, z)

Vibration (sinusoidal) / Transport (packaged)

Test carried out in accordance with EN 60068-2-6	Requirements in accordance with EN 60721-3-2 / Class 2M1		Requirements in accordance with EN 60721-3-2 / Class 2M2		Requirements in accordance with EN 60721-3-2 / Class 2M3	
Vibration (sinusoidal) / Transport (packaged)[1)]	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**
	2 to 9 Hz	Deflection 3.5 mm	2 to 9 Hz	Deflection 3.5 mm	2 to 8 Hz	Deflection 7.5 mm
	9 to 200 Hz	Acceleration 1 g[2)]	9 to 200 Hz	Acceleration 1 g[2)]	8 to 200 Hz	Acceleration 2 g[2)]
	200 to 500 Hz	Acceleration 1.5 g[2)]	200 to 500 Hz	Acceleration 1.5 g[2)]	200 to 500 Hz	Acceleration 4 g[2)]
	20 sweeps for each axis[3)]					

1) Uninterrupted duty with movable frequency in all 3 axes (x, y, z); 1 octave per minute
2) 1 g = 10 m/s²
3) 2 sweeps = 1 frequency cycle (fmin → fmax → fmin)

Shock / Transport (packaged)

Test carried out in accordance with EN 60068-2-27	Requirements in accordance with EN 60721-3-2 / Class 2M1	Requirements in accordance with EN 60721-3-2 / Class 2M2
Shock / Transport (packaged)[1)]	Type I Acceleration 10 g Duration 11 ms 18 shocks	
	Type II -	Type II Acceleration 30 g Duration 6 ms 18 shocks

1) Pulse (half-sine) stress in all 3 axes (x, y, z)

Free fall / Transport (packaged)

Tests in accordance with EN 60068-2-31	Requirements in accordance with EN 61131-2 with shipping packaging		Requirements in accordance with EN 61131-2 with shipping packaging		Requirements in accordance with EN 60721-3-2 / Class 2M1	
Free fall / Transport (packaged)	**Weight**	**Height**	**Weight**	**Height**	**Weight**	**Height**
	<10 kg	1.0 m	<10 kg	0.3 m	<20 kg	0.25 m
	10 to 40 kg	0.5 m	10 to 40 kg	0.3 m	20 to 100 kg	0.25 m
	>40 kg	0.25 m	>40 kg	0.25 m	>100 kg	0.1 m
	5 attempts					

1) Replacement for EN 60068-2-32

Toppling / Transport (packaged)

Test carried out in accordance with EN 60068-2-31	Requirements in accordance with EN 60721-3-2 / Class 2M1		Requirements in accordance with EN 60721-3-2 / Class 2M2		Requirements in accordance with EN 60721-3-2 / Class 2M3	
Toppling / Transport (packaged)	**Weight**	**Required**	**Weight**	**Required**	**Weight**	**Required**
	<20 kg	Yes	<20 kg	Yes	<20 kg	Yes
	20 to 100 kg	-	20 to 100 kg	Yes	20 to 100 kg	Yes
	>100 kg	-	>100 kg	-	>100 kg	Yes
	Topple on all edges					

2.8.2.5 Electrical safety

Overvoltage category

Requirement per EN 61131-2	Definition per EN 60664-1
Overvoltage category II	Equipment of "Overvoltage category II" is energy-consuming equipment to be supplied by the fixed installation.

Pollution degree

Requirement per EN 61131-2	Definition per EN 60664-1
Pollution degree 2	Only non-conductive pollution occurs. Occasionally, however, temporary conductivity caused by condensation is to be expected.

Protection rating provided by enclosure (IP code)

Requirement per EN 61131-2	Meaning of codes per EN 60529	Meaning for the protection of equipment	Meaning for the protection of personnel
≥IP20	First number IP**2**x	Protected against solid foreign bodies with a diameter ≥12.5 mm.	Protected against touching dangerous parts with fingers.
	Second number IPx**0**	Not protected.	-

2.8.3 UL / CSA

UL

Ind. Cont. Eq.
E115267

Canada / USA

Underwriters Laboratories (UL)

Products with this marking have been tested by Underwriters Laboratories and are listed as "Industrial Control Equipment" in category NRAQ (programmable controllers) with file number E115267.

This marking is valid for the USA and Canada and simplifies the certification of your machines and systems in these regions.

Standards applied:

UL 508	Standard for industrial control equipment
UL 61010-1	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements
UL 61010-2-201	Standard for Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use - Part 2-201: Particular Requirements for Control Equipment
CSA C22.2 No. 142-M1987	Process control equipment
CSA C22.2 No. 61010-1	Safety requirements for electrical equipment for measurement, control and laboratory use - Part 1: General requirements
CSA C22.2 No. 61010-2-201	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 2-201: Particular requirements for control equipment

Certificate

Homepage > Downloads > Certificates > UL > X20 > E115267 UL Certificate of Compliance X20

CSA HazLoc

HazLoc Cl. I, Div. 2
Group ABCD
0-60°C T5

Canada / USA

Canadian Standards Association (CSA)

Products with this marking have been tested by the Canadian Standards Association and are suitable for use in potentially explosive environments.
The products are listed in CLASS 2258 (Process control equipment - For hazardous locations) with file number 244665.
The X20 system is certified for Hazardous Locations Class I, Division 2.
Each certified module is accompanied by an information sheet providing detailed installation and safety guidelines.

This marking is valid for the USA and Canada and simplifies the certification of your machines and systems in these regions.

Standards applied:

CSA C22.2 No. 0-M1991	General Requirements - Canadian Electrical Code Part II
CSA C22.2 No. 142-M1987	Process control equipment
CSA C22.2 No. 213-M1987	Non-incendive electrical equipment for use in Class I, Division 2 hazardous locations
UL Std No. 916:2007	Energy Management Equipment
ANSI/ISA 12.12.01:2007	Nonincendive Electrical Equipment for Use in Class I and II, Division 2 and Class III, Division 1 and 2 Hazardous (Classified) Locations

Certificate

Homepage > Downloads > Certificates > HazLoc > CSA > X20, X67 > 244665 CSA HazLoc Certificate of Compliance X20, X67

2.8.4 Offshore/Maritime

With regard to maritime certification, B&R will limit itself to DNV GL for the time being. DNV GL testing is performed in accordance with the applicable standards: DNV GL, IACS E10 and IEC 60945 Section 1c. As a result, the tests comply with the requirements of other maritime classification societies.

DNV GL

Germany
Norway

Det Norske Veritas - Germanischer Lloyd

Many B&R products are certified by DNV GL and suitable for use in maritime environments.

DNV GL Maritime certificates (type approval) are generally accepted by other classification societies during ship acceptance procedures.

For corresponding environmental categories, see the technical data for the respective product.

Standards applied:

DNVGL-CG-0339	Environmental test specification for electrical, electronic and programmable equipment and systems

Certificate

Website > Downloads > Certificates > Maritime > DNV GL > X20 / Power Panel T/C > DNV GL Type Approval Certificate

LR

15/20082

Great Britain

Lloyd's Register

Products are suitable for use in maritime environments in accordance with the guidelines set forth by the Lloyd's Register classification society.

Approval has been granted for marine, offshore and industrial applications for environmental categories ENV1, ENV2, and ENV3.
Safety technology components (Safety) are certified for ENV1 and ENV2

These environmental categories are defined in Lloyd's Register's Type Approval System,
Test Specification Number 1-2015.

Covered standard:

Test Specification Number 1-2015	Lloyd's Register's Type Approval System

Certificate

Homepage > Downloads > Certificates > Maritime > LR > X20 > Lloyds Register

2.8.5 Other certifications

GOST-R

GOST-R

Products with this marking have been tested by an accredited testing laboratory and approved for import to the Russian Federation (based on EU compliance).

EAC

Eurasian Conformity (EAC)

Products with this marking have been tested by an accredited testing laboratory and approved for import (based on EU compliance) to the newly founded Eurasian Economic Union (Russia, Belarus, Kazakhstan, etc.).

KC

Korean Conformity (KC)

Products with this marking have been tested by an accredited testing laboratory and approved for import to the Korean market (based on EU compliance).

RCM

Regulatory Compliance Mark (RCM)

Products with this marking have been tested by an accredited testing laboratory and certified by the ACMA. This marking is valid in Australia/Oceania and simplifies the certification of your machines and systems in these areas (based on EU compliance).

3 X67 system

3.1 General information

The following points listed in this chapter are excerpts from the X67 system user's manual (V3.00):

- "General information"
- "System characteristics"
- "Mechanical and electrical configuration"
- "Accessories"
- "International and national certifications"

3.1.1 Transport and storage

During transport and storage, devices must be protected against undue stress (mechanical loads, temperature, moisture, corrosive atmospheres, etc.).

Devices contain components sensitive to electrostatic charges that can be damaged by improper handling. It is therefore necessary to provide the required protective measures against electrostatic discharge when installing or removing these devices (see "Protection against electrostatic discharges" on page 819).

3.1.2 Mounting orientation

- Installation must be performed according to this documentation using suitable equipment and tools.
- Devices are only permitted to be installed by qualified personnel without voltage applied.
- General safety guidelines and national accident prevention regulations must be observed.
- Electrical installation must be carried out in accordance with applicable guidelines (e.g. line cross sections, fuses, protective ground connections).
- Take the necessary steps to protect against electrostatic discharges (see "Protection against electrostatic discharges" on page 819).

3.1.3 Operation

3.1.3.1 Protection against touching electrical parts

Danger!

In order to operate programmable logic controllers, operating and monitoring devices and the uninterruptible power supply, it is necessary for certain components to carry dangerous voltages. Touching one of these components can result in a life-threatening electric shock. There is a risk of death, serious injury or damage to property.

Before switching on the programmable logic controllers, operating and monitoring devices and uninterruptible power supply, it must be ensured that the housing is properly connected to ground potential (PE rail). The ground connection must also be made if the operating and monitoring device and uninterruptible power supply are only connected for testing purposes or only operated for a short time!

Before switching on the device, all voltage-carrying components must be securely covered. During operation, all covers must remain closed.

3.1.4 Environmentally friendly disposal

All B&R control components are designed to inflict as little harm on the environment as possible.

3.1.4.1 Separation of materials

It is necessary to separate different materials so the device can undergo an environmentally friendly recycling process.

Component	Disposal
X67 modules, Cables	Electronics recycling
Cardboard box / Paper packaging	Cardboard box / Paper recycling

Disposal must comply with applicable legal regulations.

3.2 Safety guidelines

3.2.1 Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 393: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 394: Organization of general notices

3.2.2 Protection against electrostatic discharges

Electrical components that can be damaged by ESD (**E**lectro**S**tatic **D**ischarges) must be handled properly.

3.2.2.1 Packaging

- Electrical components with a housing
  ...do not require special ESD packaging but must be handled properly
  (see "Electrical components with a housing" on page 820).
- Electrical components without a housing
  ...are protected by ESD-suitable packaging.

3.2.2.2 Guidelines for proper ESD handling

Electrical components with a housing

- Do not touch the male connector contacts on the device (bus data contacts).
- Do not touch the male connector contacts on connected cables
- Do not touch the contact tips on circuit boards

Electrical components without a housing

The following points apply in addition to the points listed under "Electrical components with a housing":

- Any persons handling electrical components or devices with installed electrical components must be grounded.
- Components are only permitted to be touched on their narrow sides or front plate.
- Components must always be placed on or stored in a suitable medium (ESD packaging, conductive foam, etc.).

> **Information:**
>
> **Metallic surfaces are not suitable storage surfaces.**

- Components must not be subjected to electrostatic discharge (e.g. caused by charged plastics).
- Observe a minimum distance of 10 cm from monitors and television sets.
- Measuring instruments and equipment must be grounded.
- Probe tips of galvanically isolated measuring instruments must be temporarily discharged on suitably grounded surfaces before taking measurements.

Individual components

- ESD protective measures for individual components are thoroughly implemented at B&R (conductive floors, footwear, arm bands, etc.).
- Increased ESD protective measures for individual components are not required for handling B&R products at customer locations.

3.3 System characteristics

Decentralized machine concepts increasingly call for distributed I/O concepts. Ideally, these components are installed directly on-site on nearly any area of the machine. To accomplish this, however, the I/O modules must be rated IP67.

The remote X67 system meets these demands perfectly. This system also makes it possible to reduce costs in many areas, including cabling, the control cabinet, commissioning and service. With a completely distributed structure, the X67 system provides the highest level of flexibility.

To get the most out of a remote I/O system, performance is key. The X67 meets this prerequisite with update times under one millisecond for 1000 digital and 50 analog inputs and outputs.

A maximum of 253 modules can be operated on a single line, with a distance up to 100 m between 2 modules.

Traditional I/O systems are located centrally in the control cabinet, with extensive wiring required for sensors and actuators. In addition, modular machine designs often require intermediate connections with multi-pin connectors. Remote I/O modules can only reach their full potential, however, if additional distribution boxes can be eliminated completely. This is why the optimal solution has to include I/O modules with robust IP67 protection that can be placed directly in harsh industrial environments.

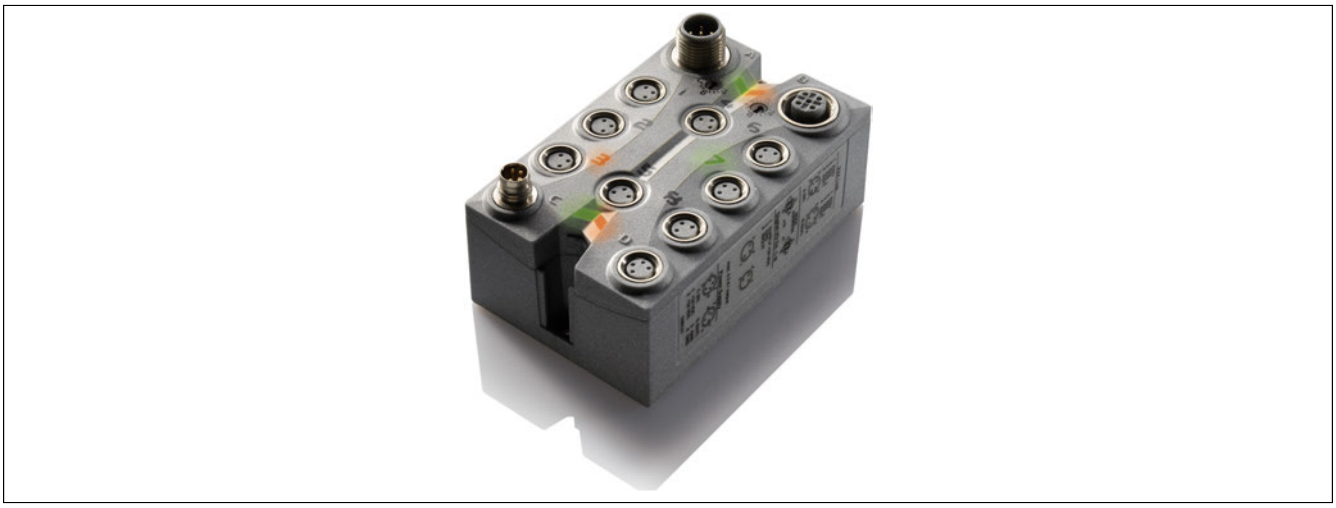
3.3.1 The X67 system

The X67 system consists of bus controller, I/O, function and system supply modules that can be wired using standard M8 and M12 connectors.

Bus controllers	
	Bus controllers are the components used to connect to fieldbus systems. Equipped with digital interfaces that can be configured as inputs or outputs, they are already full-fledged I/O modules. The ability to connect additional modules makes bus controllers extremely flexible and efficient. Like a modular system, the fieldbus device can be expanded. From the point of view of the fieldbus, it still remains a single device. The integrated X2X Link connection makes it possible to effortlessly connect various X67 modules over long distances. The X67 system is extremely effective in addition to inexpensive. If the fieldbus needs to be changed, only the bus controller changes. The rest stays the same – on the machine and throughout the documentation.
Digital modules	
	There are many different types of X67 digital modules available: • 8-/16-channel input modules • 8-channel output modules. Each channel can handle 2 amps. The maximum total load is 8 amps. • 8-/16-channel mixed modules with individually configurable channels • Valve control modules • Motor modules This flexibility reduces the number of modules while simplifying logistics and stock management. The number of inputs and outputs can always be tailored exactly to requirements.
Analog modules	
	The X67 system offers input and output modules, as well as mixed modules with 4 channels each for measuring current or voltage signals. Modules for measuring temperature using a resistance temperature detector or thermocouple round out the product range. A special male M12 connector for temperature compensation of the measurement point is also available as an accessory for these modules. One feature common to all analog modules is the complete shielding. The cable shield has seamless 360° contact with the shielding on the module.
Function modules	
	The X67 system offers special function modules: • Multifunctional counter module for absolute and incremental encoder and more • Communication module: Combining RS-232 or RS-485/RS-422 and digital I/O is a compact solution for many types of applications. This makes it possible to connect barcode readers and the corresponding trigger sensor with just one module.
System power supply	
	The ability to perform diagnostic functions must remain in every operating mode. This is an extremely important aspect of operational safety for the entire machine. For this reason, the power supply for I/O is completely isolated from the power supply for communication on X67 systems. Even if the I/O power supply is interrupted, communication and diagnostic capabilities remain. Flexible system supply modules are used for this purpose. A system supply module can supply 2 lines. Any number of system supply modules can be used in an X67 installation, allowing maximum availability by implementing a redundant supply design.

3.3.2 General product features

X67 modules have a plastic housing for use in harsh industrial environments. The devices are fully sealed and are therefore extremely resistant to mechanical stress. Integrated LEDs provide clear status indications on the machine, logically distributed on the individual channels, for the X2X Link status and the entire I/O section. Solutions such as central fastening using two screws allow easy installation, even when using wedge nuts in standard aluminum frames.



All connections are made using standard M8 or M12 connectors. The X2X Link connectors are keyed to prevent mix-ups with the M12 analog connectors.

X2X Link is based on shielded copper cables. Each module has one integrated male and female connector, one X2X Link input and one X2X Link output. An additional T-connector is not needed. Each module is operated synchronously. That means reading inputs or writing outputs takes place synchronous to the X2X Link cycle. In addition to cyclic communication, X2X Link also offers acyclic communication, e.g. to load parameters onto the module.

By default, it is not necessary to configure the node number switches on X67 modules. The modules are automatically identified by the system when booting using their X2X Link position.

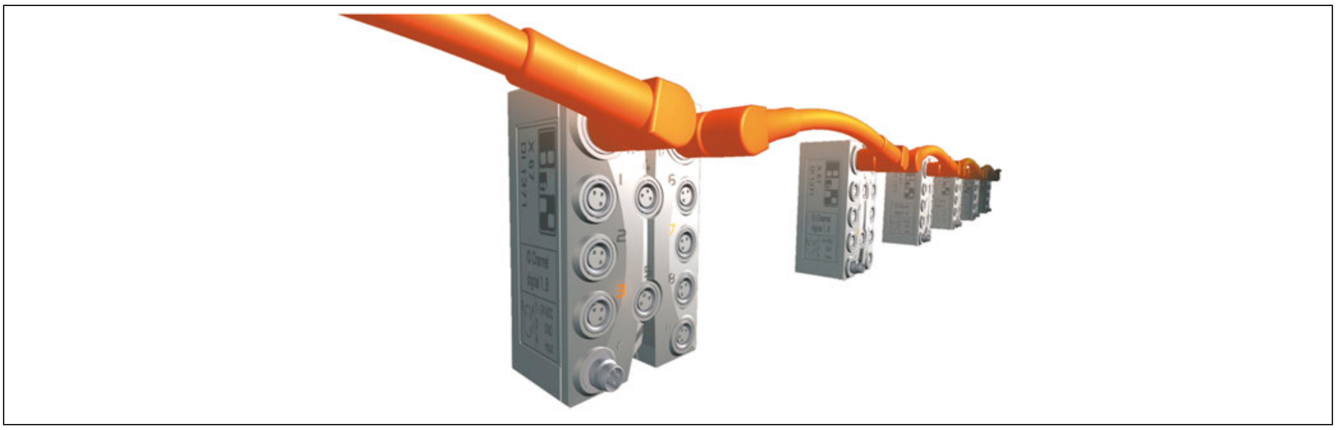


Figure 267: X67 system - X2X Link communication

3.3.3 Reduced costs

Reduced wiring

Instead of having to extensively wire each individual sensor or actuator to the control cabinet over long distances, the X67 system reduces the amount of work down to a single bus cable and a 24 VDC power supply. This applies to the entire machine. Considerable potential for savings exists even when compared to passive distributors since connecting a sensor to the X67 system replaces all of the input wiring to the control cabinet.

The shortest commissioning times

Connection via the mechanical, pre-assembled standard cables make this possible. Wiring errors are a thing of the past. Preparation for operation starts with the construction of the machine. Lengthy inspection of the wiring is no longer necessary.

Minimized service costs

Correcting errors is a snap with individual sensors and actuators that can be quickly replaced simply by unplugging as well as extensive diagnostic functions that allow errors to be detected immediately.

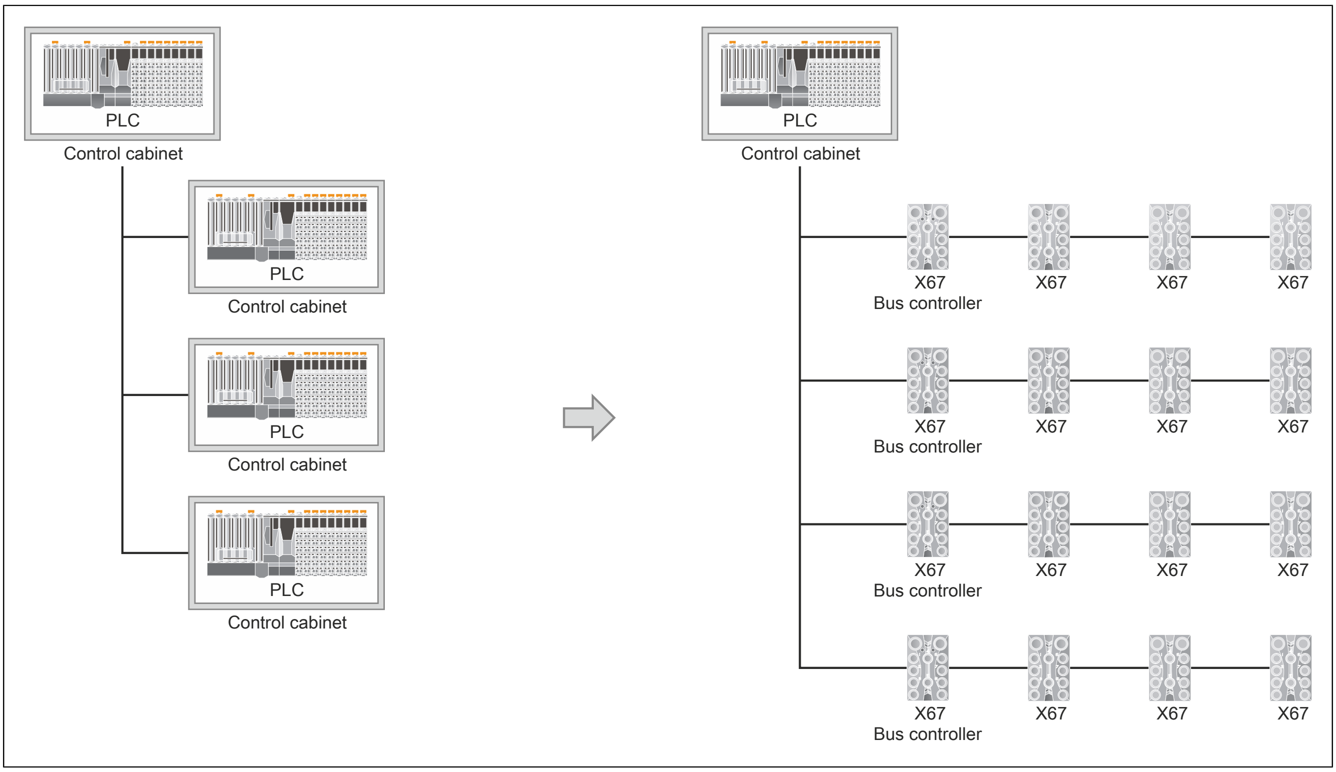
3.3.4 Flexibility

One system for all machine designs
Whether a compact machine or a large plant, this I/O system can be adapted to the machine's architecture to meet every demand for every level of performance. The X67 system offers ultimate freedom.

Unlimited expansion possibilities
The X67 system is extremely flexible, handling removable machine modules, optional expansions and even future upgrades to the machine architecture with ease.

Open communication
POWERLINK, CAN bus, CANopen, DeviceNet, PROFIBUS DP, etc. – regardless of the selected fieldbus, the I/O system remains X67.

Minimum control cabinet space
This system opens up the space normally needed for laying cables or placing terminals, I/O modules or additional distribution boxes.



Open
X67 is an I/O solution for all standard fieldbus systems and for direct connections to B&R controllers. The fieldbus may change, but the I/O system always remains the same.

Compact
Optimal ergonomics and an extremely compact design allow the X67 system to fit anywhere on the machine.

Flexible
100 m module spacing without limitations offers sufficient reserves, regardless of whether modules sit close together or distances must be covered.

Fast
Cycle times well below a millisecond also guarantee the necessary reserves for your application. Synchronous I/O processing goes without saying.

Safe
Communication and I/O are completely isolated electrically. Disturbances or voltage dips on the I/O side do not affect the bus. Performing diagnostics is always possible.

Powerful
I/O power via 2 pairs of leads provides up to 8 amps for outputs or supplies additional modules.

Shielded
Seamless 360° shield grounding from the cable over the connector directly on the threading of the M12 connector, through to the metal backplane of the module and over the mounting screws straight to the machine provides a complete ground connection for all bus and analog signals.

Centered
The central position of the two mounting screws prevents misalignment of the housing in standard aluminum frames with wedge nut installations.

Adaptable
Digital channels that can be configured as inputs or outputs allow the solution to be tailored to the requirements and reduce the total number and variety of modules needed.

Unmistakable
Visual status indicators on the modules and advanced status messages via the bus enable clear-cut diagnostics. Warning and error thresholds for I/O power supply, single-channel diagnostics and open circuit detection are just a few examples.

Robust
These completely sealed modules are the epitome of robustness, with features for maximizing electromagnetic immunity hidden inside.

Plug-and-run
Pre-assembled standard cables and automatic module identification reduce installation and commissioning work to an absolute minimum.

Protected
These systems are equipped with integrated reverse polarity protection, short circuit protection, protection when switching inductances and the highest level of protection for the electronics as well.

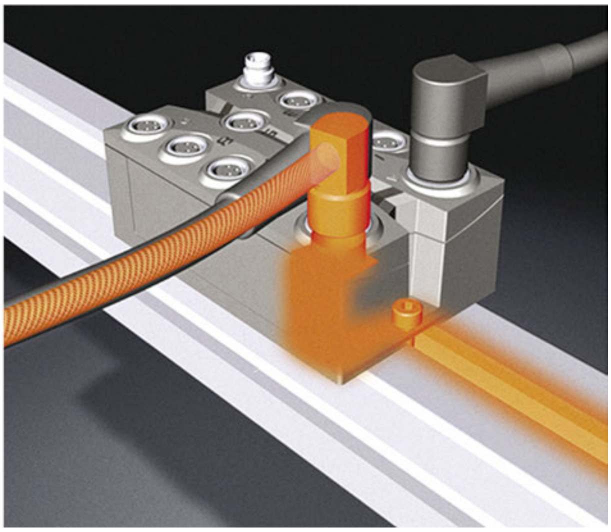
Well-supplied
Many sensors and actuators require a 24 VDC power supply. With X67 modules, this is integrated in all digital connections and also provides protection against short circuits.

Expandable
X67 systems can be expanded by 250 modules with up to 100 m between them.

Multi-talented
Synchronous I/O processing, adjustable software filters, integrated counter functions, flexible standard functions, and more – these are intelligent products perfect for the most versatile applications.

3.3.5 EMC concept

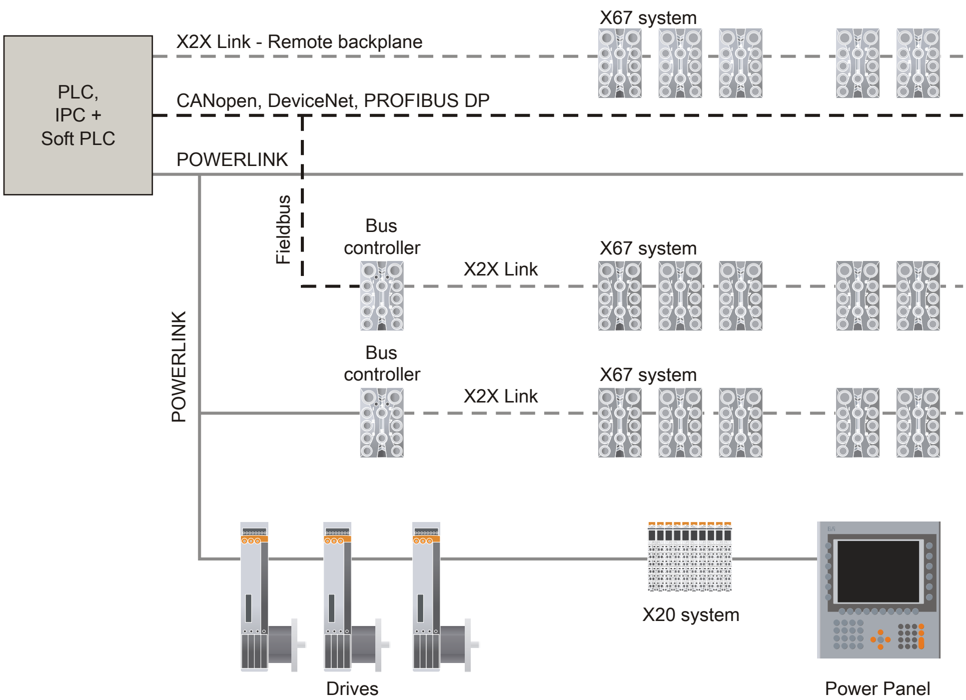
An important feature of the X67 construction is the sophisticated EMC concept. The cable shield is brought into the X67 module via the connector (complete 360° shielding). Inside the X67 housing, all components including the base plate make contact with the same ground. The final link in the chain is the connection between the base plate mounting screw and the machine which completes the ground contact from the cable to the machine. This is done for bus connections and analog connections using M12 connectors.



3.3.6 Communication

The goal of development was to free individual modules from the backplane to achieve a real remote system. The X67 system uses a cable to replace the conventional backplane and connect the modules together. The name of this "decentralized backplane" is **"X2X Link"** .

The X67 system offers many connection possibilities: X2X Link for direct connection of CPUs or IPCs with aPCI modules or PCI cards. Indirect connections or connections to non-B&R CPUs utilize the various fieldbus systems POWERLINK, CAN, CANopen, DeviceNet, and PROFIBUS DP.

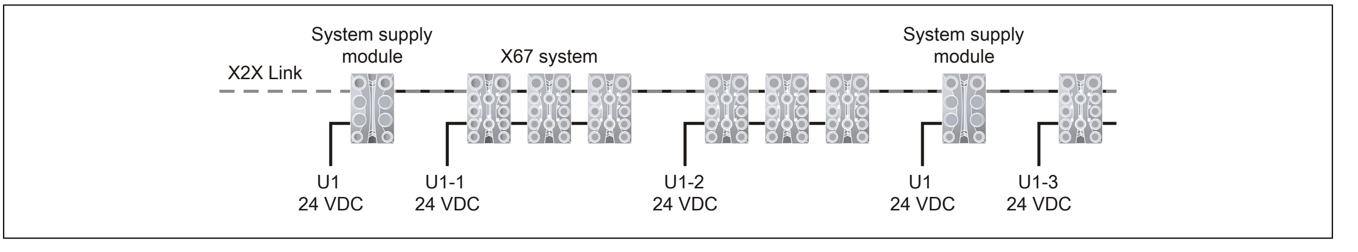


3.3.7 System power supply

The X67's decentralized structure allows modules to be placed in different power supply groups as needed. For example, this allows various modules to be connected to different voltage protection circuits or different emergency stop groups to be implemented.

The entire X2X Link network is operated totally independently of the I/O power supply. In addition to the communication lines, the connection cable contains 2 wires used to supply the X2X Link electronics for each module. Electrically, this is totally isolated from the I/O section. For this reason, power failures on the I/O side (e.g. due to short circuit, open circuit or emergency stop) only stop operation of the I/O section. The bus section continues to function with the corresponding status messages being sent to the CPU. This feature is essential in allowing errors to be analyzed quickly and corrected.

The X2X Link power supply is guaranteed by system supply modules.

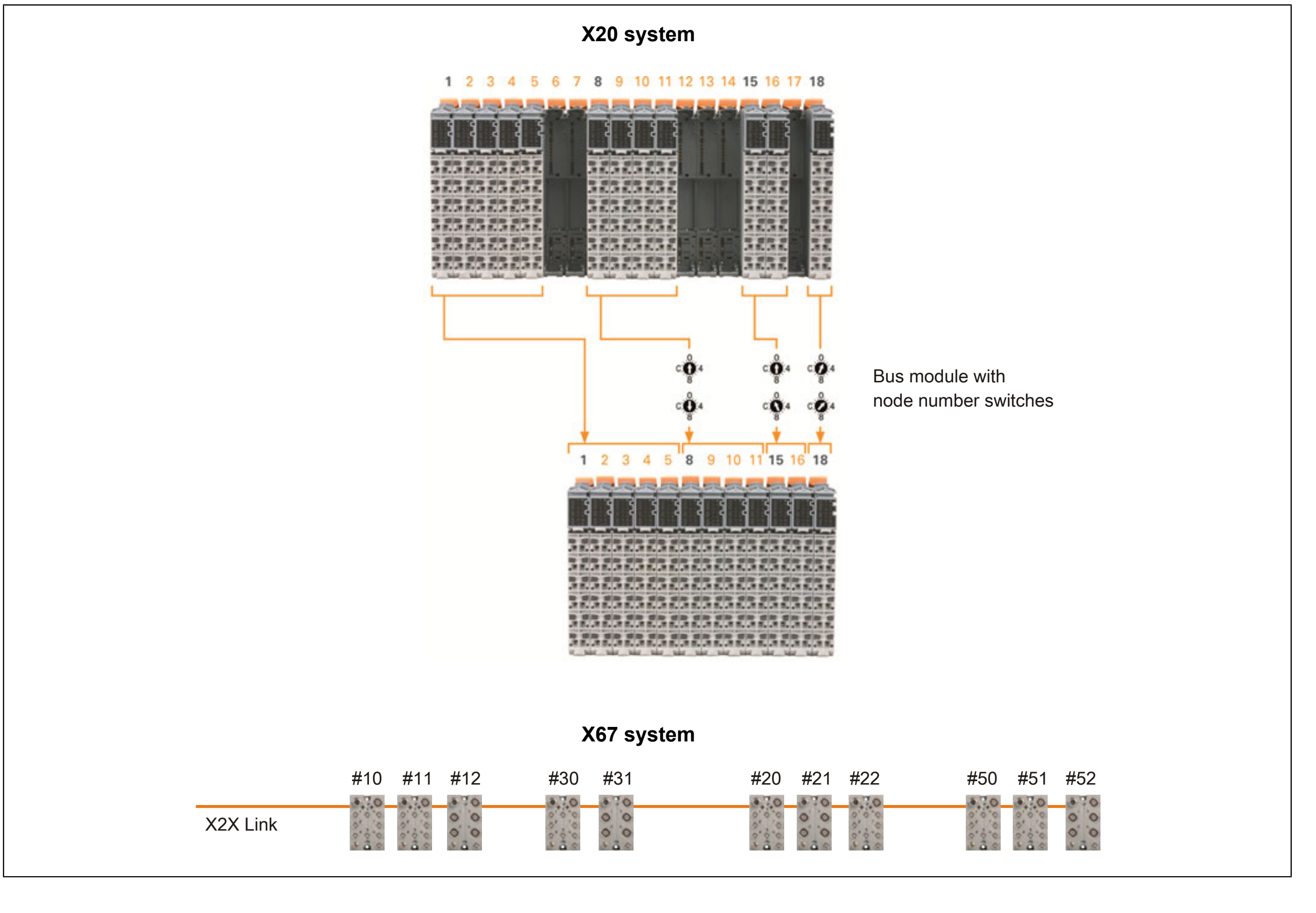


3.3.8 Configurable X2X Link address

The remote X2X Link backplane, which connects the individual I/O modules with each other, is set up to be self-addressing. Because of this, it is not necessary to set the node numbers. The module address is assigned according to its position in the X2X Link line.

In certain cases, e.g. when configurations of modular machines change, it is necessary to define specific module groups at a fixed address, regardless of the preceding modules in the line.

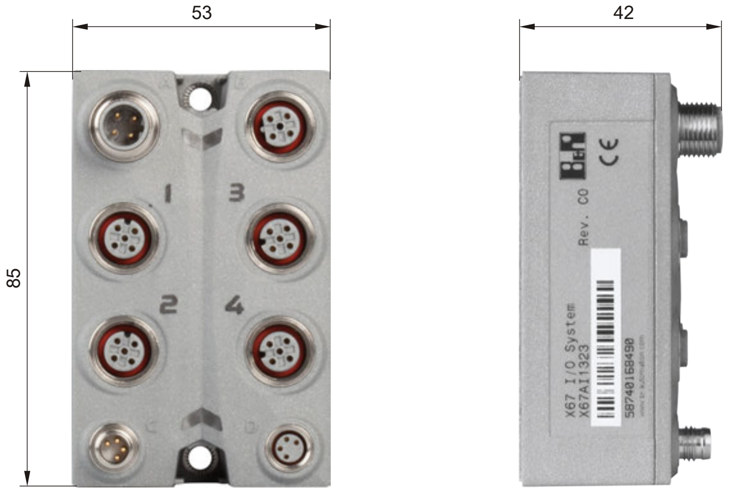
For this purpose, there are modules in both the X20 system and the X67 system with node number switches that allow you to set the X2X Link address. All subsequent modules refer to this offset and use it automatically for addressing purposes.



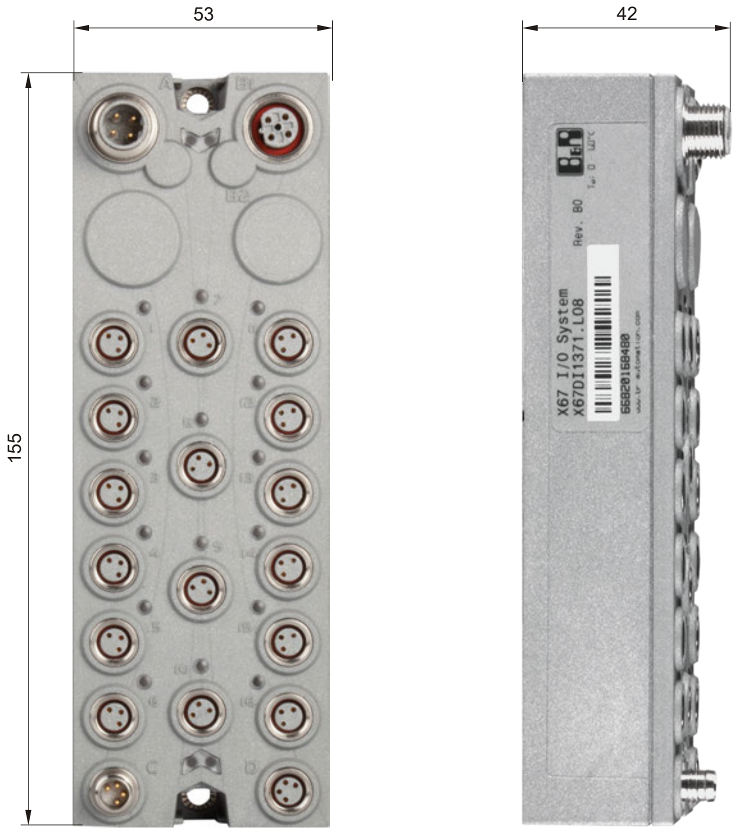
3.4 Mechanical and electrical configuration

3.4.1 Dimensions

X67 modules



X67 high-density modules



3.4.2 CAD support

To ensure CAD support, the dimensions are included in the ECAD macros in 2D. STEP data is available to allow 3D viewing.

The STEP data can be downloaded from the B&R website (www.br-automation.com) in the Downloads section for the respective module.

3.4.3 Installation

X67 modules can be installed in several different ways:

- "Installing on an aluminum frame" on page 830
- "Top-hat rail installation" on page 830
- "Installation on a mounting plate or directly on the machine " on page 831

Important!

Since electromagnetic disturbances are deflected via the base plate on the back, it is important to ensure that the mounting location has good conductivity!

The mounting location must also be connected with ground potential with good conductivity.

Information:

The following must be taken into consideration to ensure IP67 protection:

- **The union nuts on female/male connectors must be tightly secured with the specified tightening torque. The tightening torque value can be found in the module data sheet or the section "Connectors" on page 833.**
- **Female/Male connectors that are not being used must be closed with threaded caps!**
  - **Threaded caps M8, 50 pcs.: X67AC0M08**
  - **Threaded caps M12, 50 pcs.: X67AC0M12**

Information:

Shock and vibration resistance values (see "International and national certifications" on page 971) apply only if cables are installed securely.

Attaching an X67 module

The thickness of the base plate (1.5 mm) should be taken into consideration when defining the screw length.

The grooved imprint in the base plate ensures that the screws do not become loose, even without an additional retaining ring.

The recommended tightening torque for the M4 screw is 0.6 Nm.

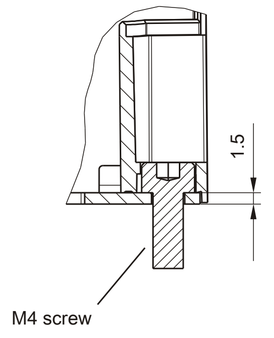


Figure 268: X67 system - Fastening an X67 module

3.4.3.1 Installing on an aluminum frame

Installation on an aluminum frame is done using 2 wedge nuts and M4 screws.

Figure 269: Installation on an aluminum frame

> **Important!**
>
> **For coated or anodized surfaces, the isolating coated or anodized layer in the area of the base plate for X67 modules must be removed.**

3.4.3.2 Top-hat rail installation

An X67 module can be installed on a top-hat rail using top-hat rail mounting plate X67ACTS35.

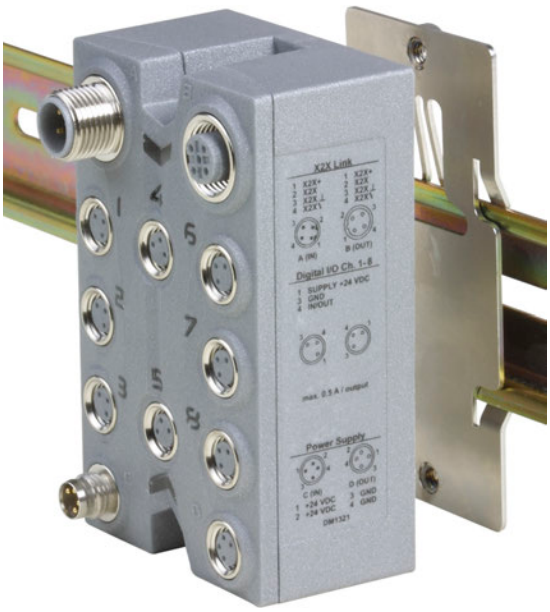


Figure 270: Top-hat rail installation

3.4.3.3 Installation on a mounting plate or directly on the machine

X67 modules can also be mounted on a mounting plate or directly on the machine.

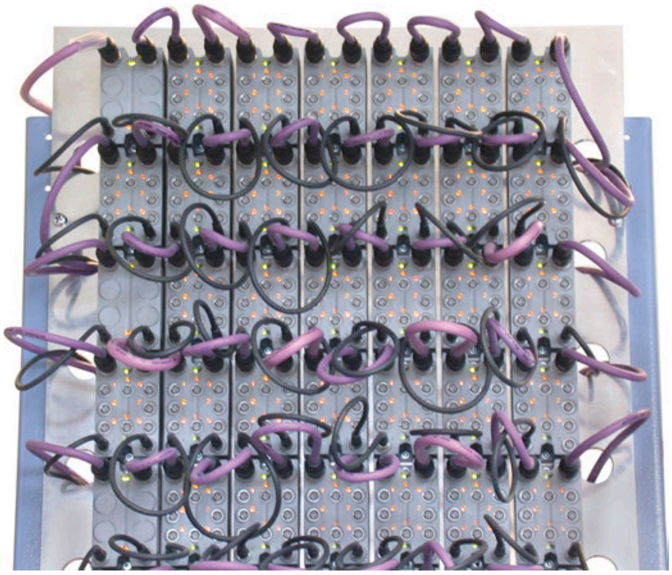
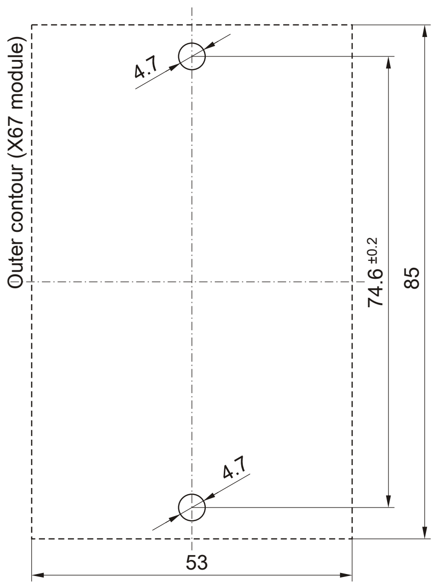
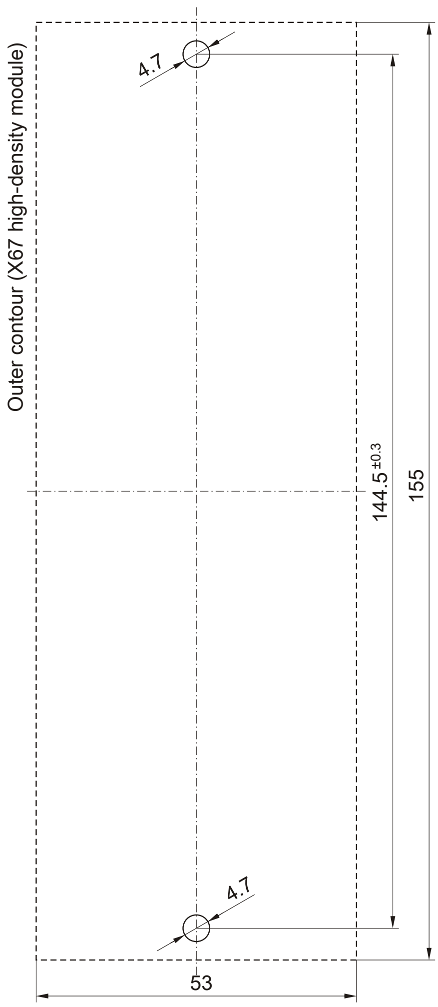


Figure 271: Installation on a mounting plate

3.4.3.3.1 Drilling template for X67 modules

The modules are mounted using M4 screws.

<table>
<tr><th>X67 modules</th><th>X67 high-density modules</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

3.4.4 Shielding and grounding

When using standard cables available from B&R, the cable shield is brought into the X67 module via the connector (complete 360° shielding). Inside the X67 housing, all components including the base plate make contact with the same ground. The final link in the chain is the mounting screw, which connects the base plate and the machine part / mounting plate and completes the seamless ground contact from the cable to the machine part / mounting plate. A contact with good conductivity between the X67 module base plate and the machine part / mounting plate is absolutely required.

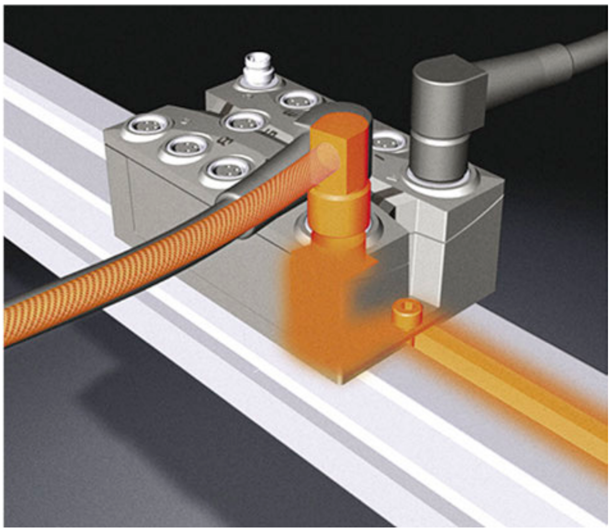


Figure 272: X67 system - Shielding and grounding

Important!

The shielding at both ends of the cable must be professionally grounded on field-assembled cables!

3.4.5 Connectors

The connectors for the X67 system are designed as circular connectors. In addition to field-assembled male connectors, B&R also offers pre-assembled cables for X2X Link, fieldbus and I/O functions.

The following connectors are used with the X67 system:

Threads	Tightening torque
M8	0.4 Nm
M12	0.6 Nm
M16	1.0 Nm

Information:

When using third-party connectors, we strongly recommend ensuring that the contacts are gold (Au) coated.

B&R connectors are designed for use with X67 components (see "General overview" on page 936).

3.4.6 The power supply concept

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

The X67's decentralized structure allows modules to be placed in different power supply groups as needed. This allows various modules to be connected to different voltage protection circuits or different emergency stop groups to be implemented.

The X2X Link network is operated totally independently of the I/O power supply. In addition to the communication lines, the connection cable contains 2 wires to supply power to the X2X Link electronics of each module. This is fully galvanically isolated from the I/O component. For this reason, power failures on the I/O side (e.g. due to short circuit, open circuit or emergency stop) only stop operation of the I/O section. The X2X Link network continues to work, and the corresponding status messages are sent to the CPU. This allows fault events to be quickly analyzed and corrected.

The X2X Link power supply is guaranteed by system supply modules.

A potential group involves multiple X67 modules that are supplied via a shared supply feed.

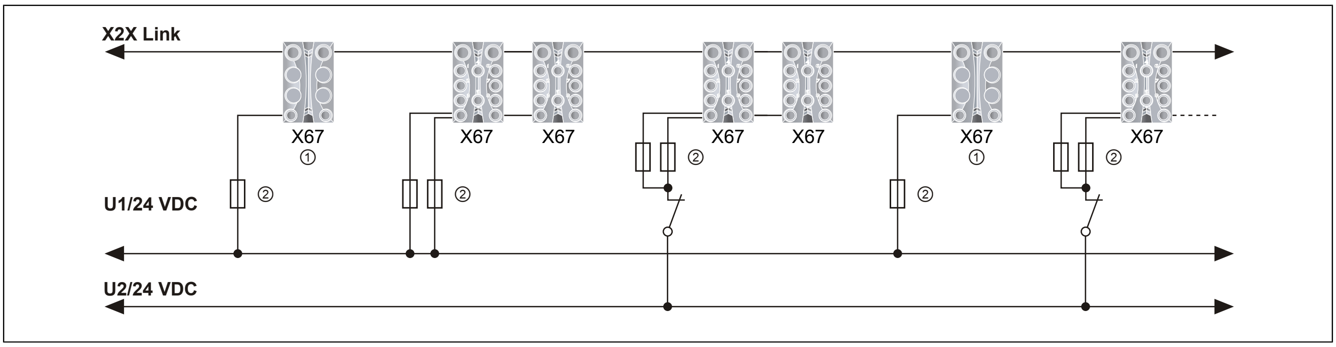


Figure 273: Power supply design with the help of 2 different potential groups

Legend

① System supply module
② Fuse, 4 A slow-blow

X67 I/O modules are power consumers on the X2X Link. System supply modules feed in the power. System supply modules should be planned according to the calculated power requirements. Since they supply voltage in both directions, they can be arranged either at the beginning or between the consumers. Redundant configurations are also possible by adding more system supply modules.

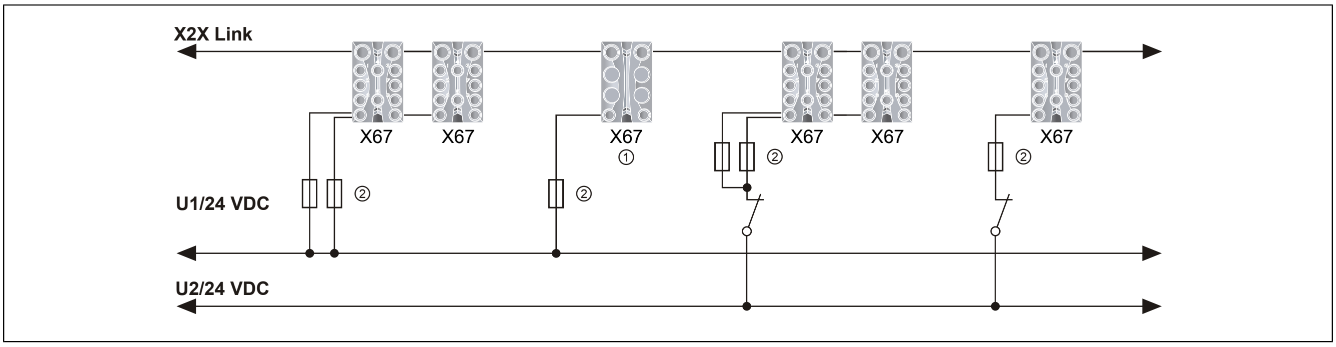


Figure 274: X2X Link power supply through flexible application of system supply modules

Legend

① System supply module
② Fuse, 4 A slow-blow

The bus controllers can supply several modules on the X2X Link without an additional system supply module.

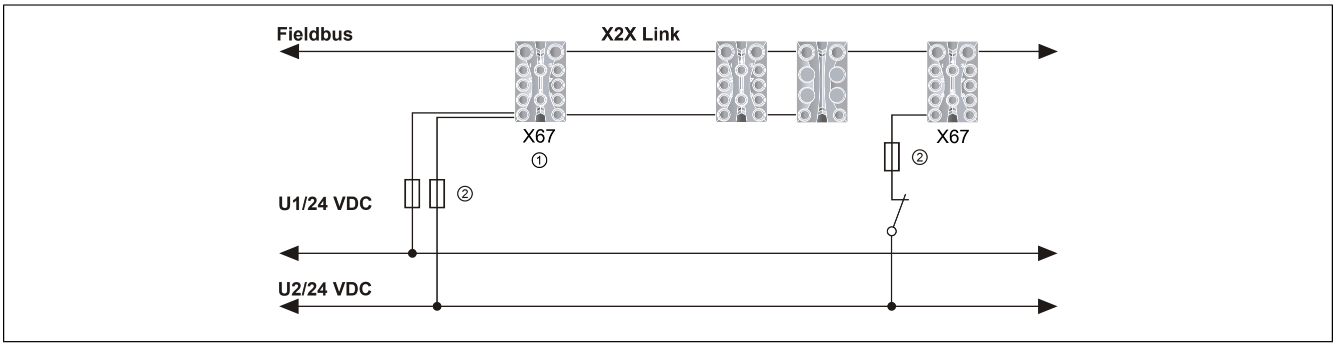


Figure 275: X2X Link power supply via bus controllers

Legend

① System supply module
② Fuse, 4 A slow-blow

3.4.6.1 Failure of I/O supply (ModuleOK)

Status "ModuleOK" consists of different module parameters and is available for monitoring the X67 modules. When the I/O supply voltage is lost, data point "ModuleOK" provides the value 0 (false).

3.4.7 Safe cutoff of a potential group

Information:

B&R keeps user's manuals as current as possible. From a safety standpoint however, the current certified version of the document must be used.

The current certified document is available for download under Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups.

3.4.7.1 Description of function

The operating principle "Safe cutoff of a potential group" enables the user to execute safety-related functions within a B&R system in combination with an external safety relay.
The safety function is limited to cutting off or interrupting the power to connected actuators.

Functionality

An external safety relay is connected to the I/O power supply for the potential group or an X20SP1130 power supply module is used. When the functional safe state is requested or state "Failsafe" occurs, then this feed cuts off the I/O power supply of the potential group. The power is then also cut off for all actuators connected to this potential group. However, module-internal energy storage devices (e.g. capacitors) remain charged and must be taken into account in the assessment of the safety function.

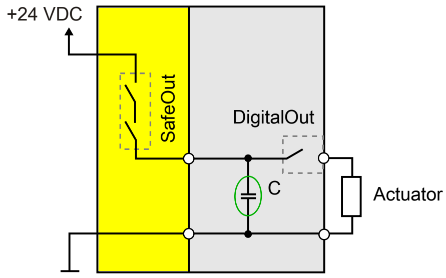


Figure 276: Functionality with internal energy storage

3.4.7.2 Scope of application / Standards referenced

The operating principle is confined to machine manufacturing applications and therefore implicitly to the following standards:

- EN ISO 13849-1:2015 / EN ISO 13849-2:2012

Requirements of other standards are not taken into consideration.

3.4.7.3 Intended use

Danger!

Danger from incorrect use of safety-related products/functions

Proper functionality is only ensured if the products/functions are used in accordance with their intended use by qualified personnel and the provided safety information is taken into account. The aforementioned conditions must be observed or covered by supplementary measures on your own responsibility in order to ensure the specified protective functions.

3.4.7.3.1 Qualified personnel

Use of safety-related products is restricted to the following persons:

- Qualified personnel who are familiar with relevant safety concepts for automation technology as well as applicable standards and regulations
- Qualified personnel who plan, develop, install and commission safety equipment in machines and systems

Qualified personnel in the context of this manual's safety guidelines are those who, because of their training, experience and instruction combined with their knowledge of relevant standards, regulations, accident prevention guidelines and operating conditions, are qualified to carry out essential tasks and recognize and avoid potentially dangerous situations.
In this regard, sufficient language skills are also required in order to be able to properly understand this manual.

3.4.7.3.2 Area of application

The safety-related B&R control components described in this manual were designed, developed and manufactured for special applications for machine and personnel protection. They are not suitable for any use involving serious risks or hazards that could lead to the injury or death of several people or serious environmental impact without the implementation of exceptionally stringent safety precautions. In particular, this includes the use of these devices to monitor nuclear reactions in nuclear power plants, flight control systems, air traffic control, the control of mass transport vehicles, medical life support systems and the control of weapon systems.

When using safety-oriented control components, the safety precautions applying to industrial control systems (e.g. the provision of safety devices such as emergency stop circuits, etc.) must be observed in accordance with applicable national and international regulations. The same applies for all other devices connected to the system, e.g. drives or light curtains.

The safety guidelines, information about connection conditions (nameplate and documentation) and limit values specified in the technical data must be read carefully before installation and commissioning and must be strictly observed.

3.4.7.3.3 Security concept

B&R products communicate via a network interface and were developed for integration into a secure network. The network and B&R products are affected by the following hazards (not a complete list):

- Unauthorized access
- Digital intrusion
- Data leakage
- Data theft
- A variety of other types of IT security breaches

It is the responsibility of the operator to provide and maintain a secure connection between B&R products and the internal network as well as other networks, such as the Internet, if necessary. The following measures and security solutions are suitable for this purpose:

- Segmentation of the network (e.g. separation of the IT and OT networks)
- Firewalls for the secure connection of network segments
- Implementation of a security-optimized user account and password concept
- Intrusion prevention and authentication systems
- Endpoint security solutions with modules for anti-malware, data leakage prevention, etc.
- Data encryption

It is the responsibility of the operator to take appropriate measures and to implement effective security solutions.

B&R Industrial Automation GmbH and its subsidiaries are not liable for damages and/or losses resulting from, for example, IT security breaches, unauthorized access, digital intrusion, data leakage and/or data theft.

Before B&R releases products or updates, they are subjected to appropriate functional testing. Independently of this, the development of customized test processes is recommended in order to be able to check the effects of changes in advance. Such changes include, for example:

- Installation of product updates
- Notable system modifications such as configuration changes
- Import of updates or patches for third-party software (non-B&R software)
- Hardware replacement

These tests should ensure that implemented security measures remain effective and that systems behave as expected.

3.4.7.3.4 Safety technology disclaimer

The proper use of all B&R products must be guaranteed by the customer through the implementation of suitable training, instruction and documentation measures. The guidelines set forth in system user's manuals must be taken into consideration here as well. B&R has no obligation to provide verification or warnings with regard to the customer's purpose of using the delivered product.

Changes to the devices are not permitted when using safety-related components. Only certified products are permitted to be used. Currently valid product versions in each case are listed in the corresponding certificates. Current certificates are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product. The use of non-certified products or product versions is not permitted.

All relevant information regarding these safety products must be read in the latest version of the related data sheet and the corresponding safety notices observed before the safety products are permitted to be operated. Certified data sheets are available on the B&R website (www.br-automation.com) in the Downloads section for the respective product.

B&R and its employees are not liable for any damages or loss resulting from the incorrect use of these products. The same applies to misuse that may result from specifications or statements made by B&R in connection with sales, support or application activities. It is the sole responsibility of the user to check all specifications and statements made by B&R for proper application as it pertains to safety-related applications. In addition, the user assumes sole responsibility for the proper design of the safety function as it pertains to safety-related applications.

3.4.7.3.5 Installation notes for X20 safety modules

Products must be protected against impermissible dirt and contaminants. Products are protected from dirt and contaminants up to pollution degree II as specified in the IEC 60664 standard.

Pollution degree II can usually be achieved in an enclosure with IP54 protection, but uncoated modules are NOT permitted to be operated in condensing relative humidity and temperatures under 0°C.

The operation of coated modules is allowed in condensing relative humidity.

Danger!

Pollution levels higher than specified by pollution degree II in standard IEC 60664 can result in dangerous failures. It is extremely important that you ensure a proper operating environment.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

The supply of X20 potential groups must generally be protected using a fuse with a maximum of 10 A.
For more information, see chapter "Mechanical and electrical configuration" of the X20 or X67 user's manual.

3.4.7.3.6 Installation notes for X67 safety modules

Danger!

The following points must be taken into account to ensure IP67 protection:

- **The union nuts on female/male connectors must be tightly secured with the specified tightening torque. For the tightening torque, see the X67 system user's manual.**
- **Female/Male connectors that are not being used must be closed with threaded caps!**
  - **Threaded caps M8, 50 pcs.: X67AC0M08**
  - **Threaded caps M12, 50 pcs.: X67AC0M12**

Danger!

The shock and vibration resistance values (see chapter "International and national certifications" of the X67 system user's manual) apply if cables are laid solidly.

Danger!

In order to guarantee a specific voltage supply, a SELV power supply that conforms to IEC 60204 must be used to supply the bus, SafeIO and SafeLOGIC controller. This also applies to all digital signal sources that are connected to the modules.

If the power supply is grounded (PELV system), then only a GND connection is permitted for grounding. Grounding types that have ground connected to +24 VDC are not permitted.

Danger!

Unused female connectors must be covered with threaded caps (X67AC0M08 or X67AC0M12 accessory). Otherwise, hazardous conditions may arise if the module fails to function properly.

3.4.7.3.7 Safe state

If an error is detected by the module (internal or wiring error), the modules enable the safe state. The safe state is structurally designed as a low state or cutoff and cannot be modified.

Applications in which the safe state must actively switch on an actuator cannot be implemented with this module. In these cases, other measures must be taken to meet this safety-related requirement (e.g. mechanical brakes for hanging load that engage on power failure).

3.4.7.4 System-specific information

The operating principle applies to a potential group.

All potential groups are generally only permitted to be supplied by 1 power supply module. The possible further processing of the power supply on the module is not permitted to result in multiple supply instances.

In the X20 system, only modules of type X20BM01, X20BM23 and X20BM26 that ensure the interruption of the internal I/O power supply to the left are permitted to be used as bus modules for power supply modules.
On modules X20PS9400 and X20PS3300, only the I/O power supply (+24 V I/O) is permitted to be switched with the safety relay. The bus power supply (+24 V BC/X2X L.) must be isolated.

When using module X67PS1300 to supply power to the X67 potential group, only the I/O power supply (+24 V I/O) is permitted to be switched with the safety relay. The bus power supply (+24 V BC/X2X L.) must be isolated.

The operating principle is limited to the modules listed in the following certificate.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

3.4.7.5 Safety guidelines

This section provides a summary of safety notices for the user.

Danger!

Failure of the safety function due to misuse

Observe the following safety guidelines. Failure to observe any of the following notices can lead to the failure of the safety function and may result in serious injury.

- **When using the operating principle, it is the user's responsibility to adhere to the relevant standards and safety directives. In addition, the guidelines for proper use must be observed.**
- **For all potentials supplying the modules, SELV/PELV power supplies must be used.**
- **The potential groups for which the operating principle is applied are only permitted to contain modules listed from certificate "Safe cutoff of potential groups".**
- **Uncoated X20 modules in which the operating principle is used are not permitted to be operated in condensing air humidity or at temperatures below 0°C.**
- **It is not permitted to mix modules from different systems (X20, X67, 7XV) within a potential group.**
- **It is not permitted to install multiple power supplies in a potential group (particularly with regard to power supply modules that also supply the bus supply).**
- **Ensure that the upstream safety relay is wired properly.**
- **Ensure that ALL sensors and actuators connected to the potential group are wired properly.**
- **Note possible impairments of the safety function due to the internal energy storage devices. If this is sufficient to enable a connected actuator and subsequently leads to a dangerous state, the protection objective is not given and alternatives or supplementary measures must be installed.**
- **The switch-off time must be verified by a control measurement!**
- **For modules with isolated I/O potential for sensors and actuators, the upstream safety relay must shut off the supply for both the sensors and actuators.**
- **The ground connections should be used as functional ground and not as protective ground and must not be connected to the 24 V supply voltage (GND is permitted). In addition, no protective components are permitted to be used between the ground and the 24 V supply voltage.**

3.4.7.5.1 Capacitances within the potential group

The internal capacitances of the module remain charged at the time of shutdown. The total capacitance of the potential group results from the sum of the capacitances of the individual modules, upstream external safety relay and actuator.

$$C_{total} = \sum_{i=1}^{n} C_i$$

The capacitances of the corresponding B&R modules are listed in the certificate.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

At the time a safety function is requested, it is not ensured that the standard outputs are enabled. If an output is disabled at the time of the request, the affected module-internal capacitances remain permanently charged. If the output is enabled by the standard application, an unexpected voltage peak occurs on the output.

In connection with the supply voltage, the total capacitance present in the system results in a charge that must be taken into account during cutoff. In the worst case scenario, it can be assumed that the total capacitance present in the system buffers each output present in the potential group. This behavior is not permitted to lead to a safety-critical state due to actuators in the potential group; alternatives or supplementary corrective measures must be installed.

3.4.7.5.2 Potential group structure

The potential group is only permitted to be made up of modules listed in the following certificate. Modules not listed in this certificate endanger the "absence of feedback" of the external cutoff and therefore the safety function.

Certificate
Website > Downloads > Certificates > Safety technology > X20, X67 > Safe cutoff of potential groups

To ensure clarity and that the external cutoff is triggered when a fault occurs, installing multiple power supply sources in a potential group is not permitted.

SELV/PELV power supplies must be used for both the bus supply (X2X) and the I/O power supply; otherwise, safety-related malfunctions can occur due to overvoltages.

For modules with isolated I/O potential for sensors and actuators, the upstream safety relay must shut off the supply for both the sensors and actuators; otherwise, feedback cannot be excluded.

3.4.7.5.3 Circuit examples

Single-channel without feedback

The following example shows a load being cutoff using the emergency stop safety function. Only safe actuators such as motors or input "Enable" of an ACOPOS/ACOPOSmulti drive are permitted to be used as the load in this case.

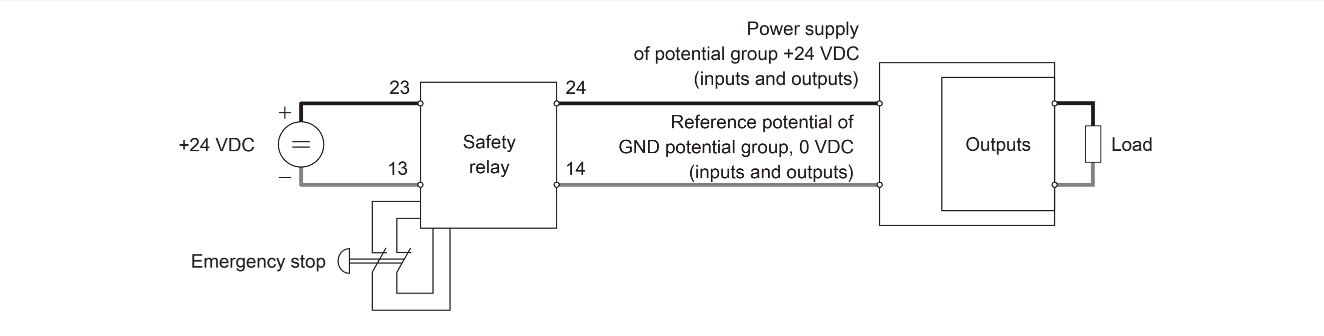


Figure 277: Circuit example: 1-channel without feedback

Provided that the external components being used (emergency stop switch, safety relay, load) satisfy the respective requirements, this example can achieve PL e (performance level as specified in EN ISO 13849-1:2015).

Dual-channel with feedback

The following example shows a load being cutoff using the emergency stop safety function. Feedback allows errors in the actuator to also be detected, and a cutoff is also possible if a fault event occurs due to the full dual-channel design. Whether or not 2 fully isolated potential groups – as shown in the example – are necessary depends on the application and how the safety solution is designed.

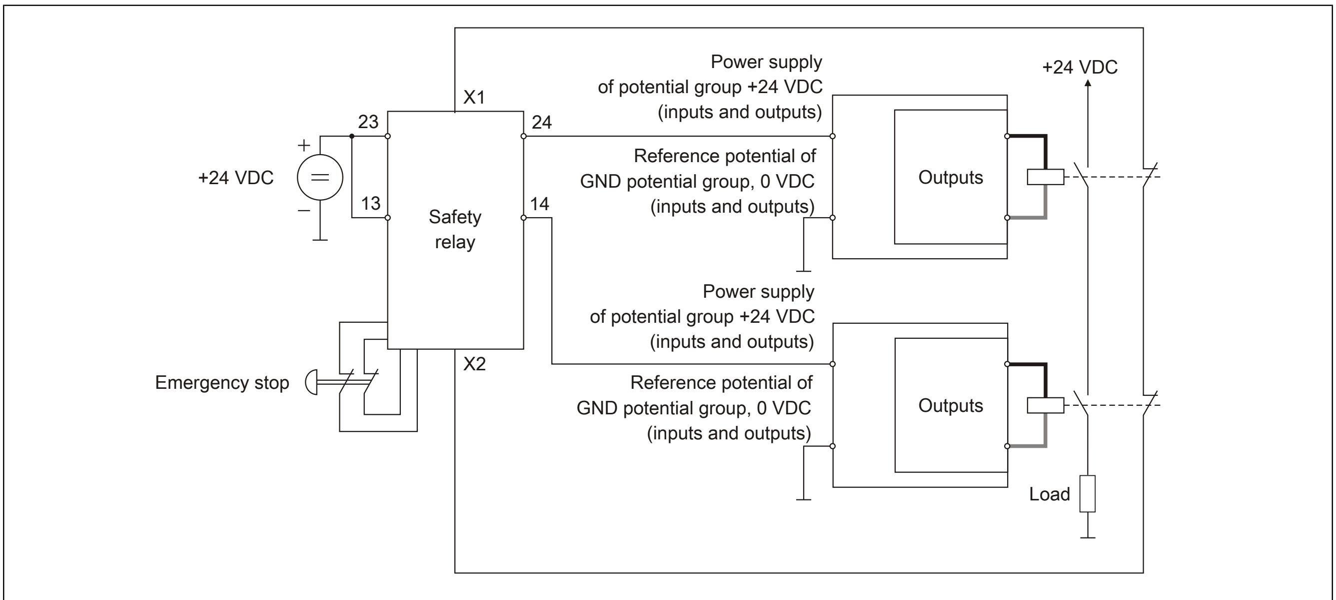


Figure 278: Circuit example: 2-channel with feedback

Provided that the external components being used (emergency stop switch, safety relay, load) satisfy the respective requirements, this example can achieve PL e.

Example with power supply module X20SP1130

The following examples show a load being cut off using safe power supply module X20SP1130 along with safe input module X20SI4100 and the "emergency stop" safety function.

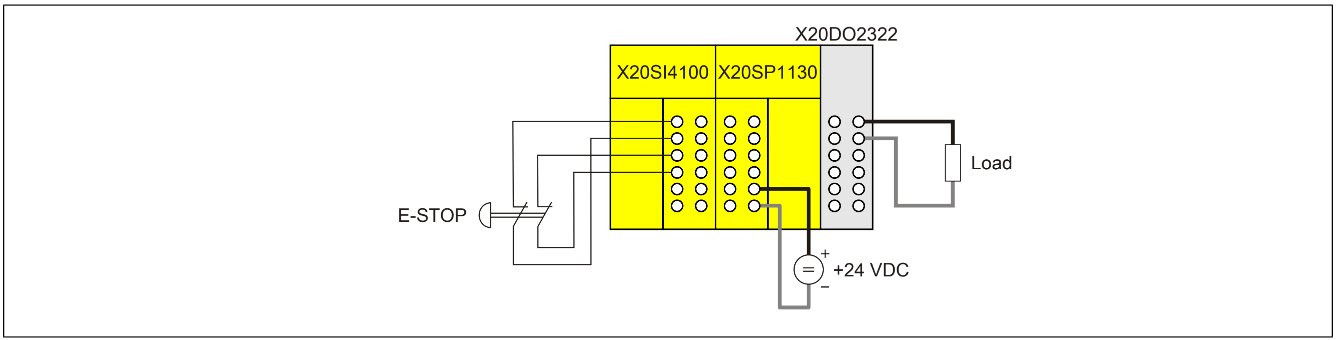


Figure 279: Circuit example with power supply module X20SP1130

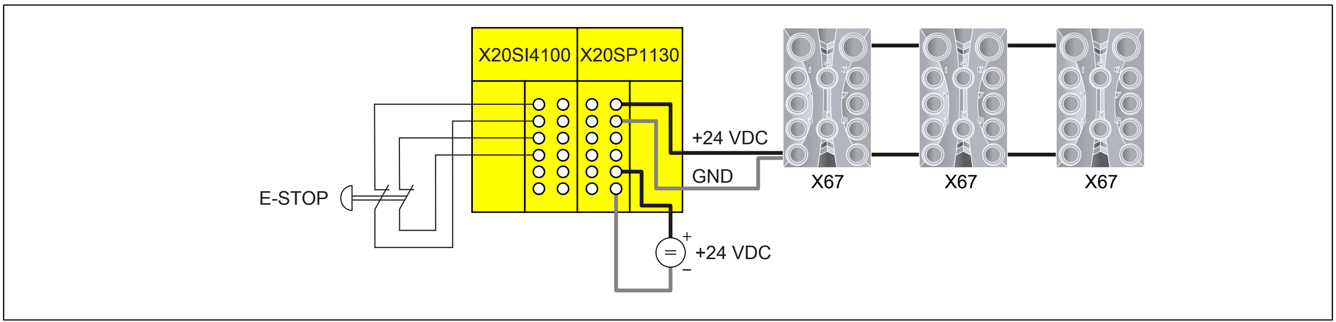


Figure 280: Circuit example with power supply module X20SP1130 and X67

Provided that the external components being used (emergency stop switch, load) satisfy the respective requirements, these examples can achieve PL e.

3.4.7.5.4 Wiring information

The operating principle "Safe cutoff of a potential group" only applies to the B&R modules being used. All other parts of the safety chain such as the application, upstream sensors or downstream actuators are NOT included in this principle.

For this reason, it is important to take the following points into consideration:

- Ensure proper wiring of the safety relay with the I/O supply. A short circuit between the output of the safety relay and an external 24 V voltage source can cause an unintended supply of 24 V to the internal supply voltage of the potential group. As a result, the safety function can no longer be guaranteed, which means that **ALL** of the channels in the potential group can no longer be cut off by the upstream safety relay.
- Make sure that **ALL** of the potential group's input and output channels and the connected sensors and actuators are wired properly. A short circuit between an input or output of the potential group and an external 24 V voltage source can cause the unintended feedback of 24 V to the internal supply voltage of the potential group. As a result, the safety function can no longer be guaranteed, which means that **ALL** of the output channels in the potential group can no longer be cut off by the upstream safety relay.
- In accordance with EN ISO 13849-2:2012, appendix D.2, table D.4, a short circuit between any 2 conductors can be excluded, provided that:
    - They are permanently installed and protected against external damage (e.g. using a cable duct or armored conduit)
    - OR they are in separate plastic-sheathed cables
    - OR they are installed within an electrical enclosure. This requires that the wiring as well as the area for electrical equipment meet the respective requirements [see EN 60204-1]
    - OR they are individually shielded with a ground connection

3.4.8 X67 system cabling

Due to the high degree of flexibility offered by the X67 system, a few things must be taken into consideration when wiring:

- Maximum number of X67 modules in one X2X line (253)
- Maximum distance between X67 modules
- Distance between the system supply modules
- Station number assignment
- Permissible current consumption
- Suitable connectors must be used for the X67 system (see section "Connectors" on page 833).

Possibilities for cabling the X67 system:

- X2X Link power supply with X67 system supply, X67 bus controller or X20 bus transmitter
- Isolated X2X Link and I/O power supply
- Creating electrical potential groups

3.4.8.1 X2X Link cabling

X2X Link connections:

- X67 module: M12, B-keyed connections (A → Input, B → Output)
- Interface module / master system: 4-pin terminal block

The maximum distance between 2 X67 stations is 100 meters.

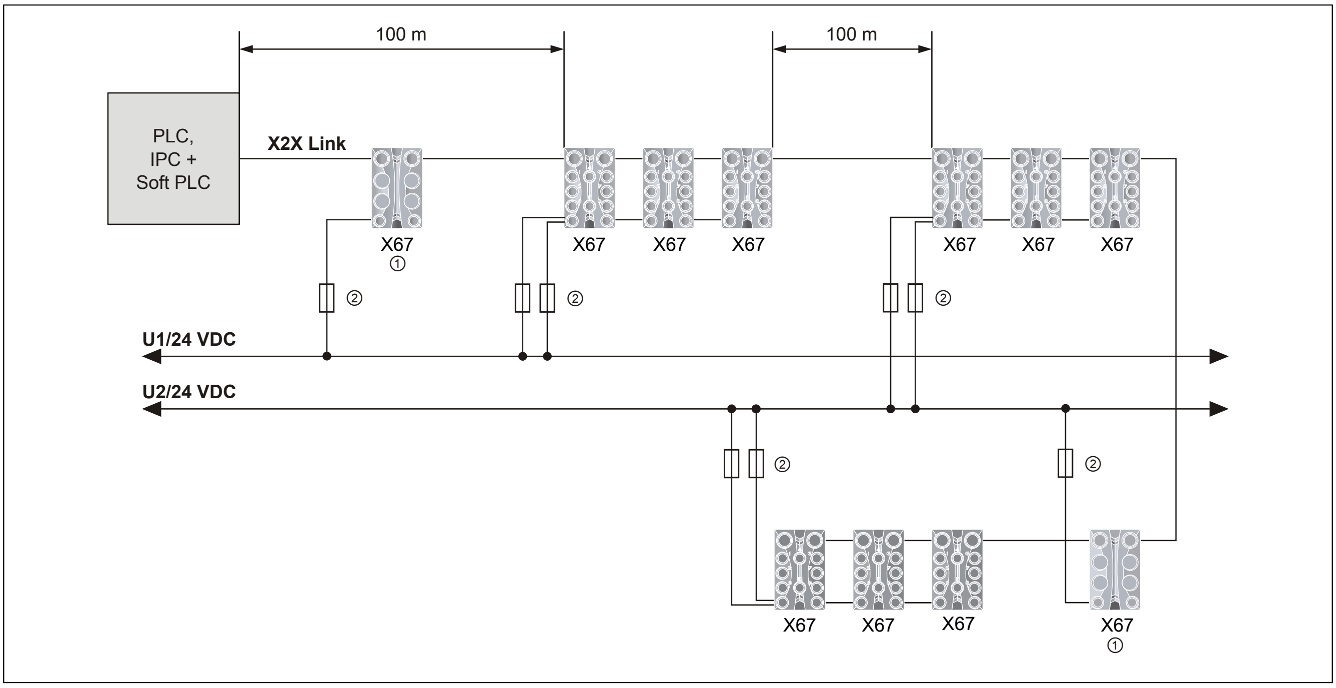
Information:

System supply module X67PS1300 cannot be used to bridge distances since it does not regenerate the signal.

Station numbers are automatically assigned according to the order of X67 modules (cable sequence).

Information:

All subsequent station numbers are shifted when an X67 station is connected/removed. The system supply module is not included in this calculation, however, and does not receive its own station number!



Legend

① System supply module
② Fuse, 4 A slow-blow

The supply voltage for the X2X Link is reduced by line resistance (line length).

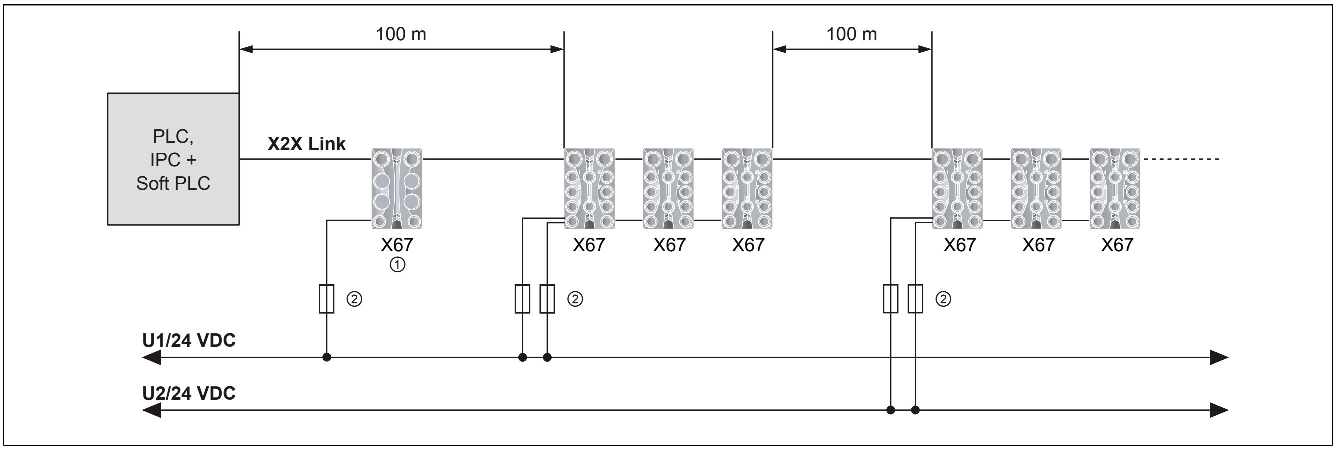
Information:

Be aware of voltage drops in the line!

Depending on the power consumption and type of modules being used, 15 or more X67 modules can be supplied by system supply module X67PS1300. However, this does not mean that 100 m distance between each station is possible (i.e. a total length of n x 100 m).

Information:

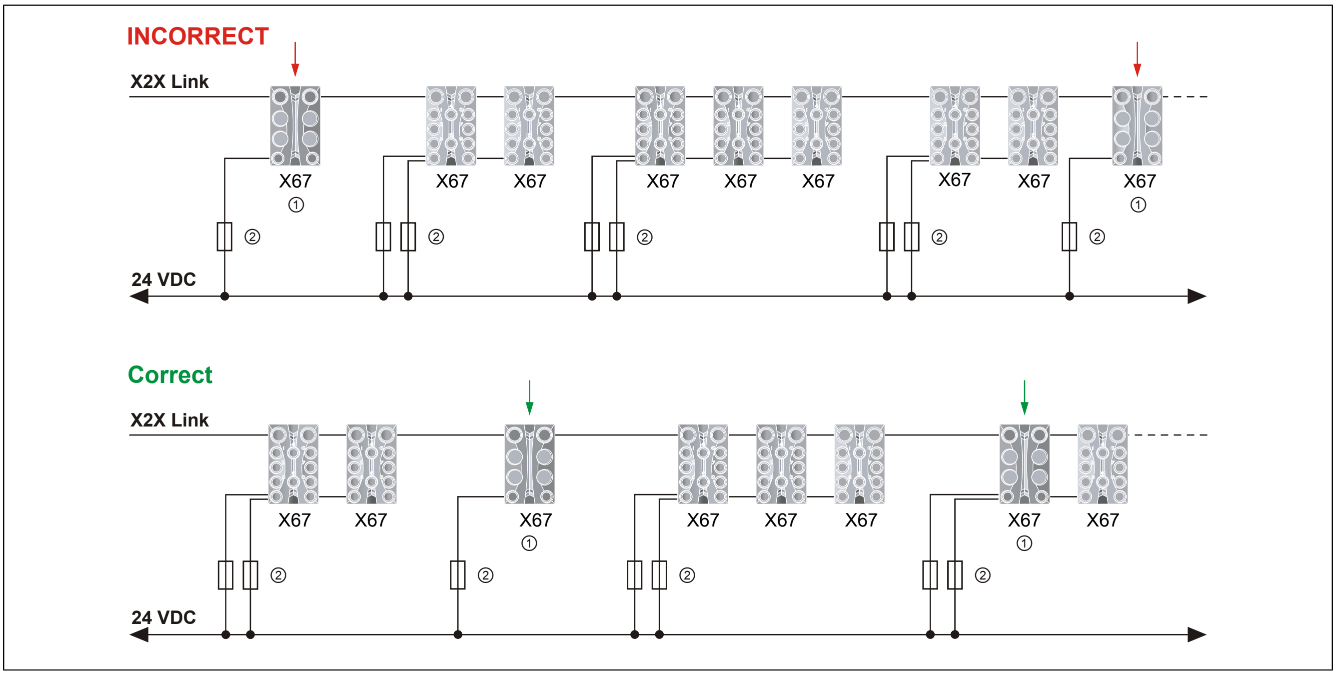
Regardless of the number of stations, the supply voltage is affected at total lengths over 100 m. Therefore, additional system supply modules must be added when necessary.



Legend

① System supply module
② Fuse, 4 A slow-blow

All system supply modules integrated in an X67 system must be distributed accordingly along the length of the line to avoid excessive voltage drops caused by the line length!



Legend

① System supply module
② Fuse, 4 A slow-blow

Information:

All system supply modules must be distributed evenly throughout the system!

3.4.8.2 X2X Link cabling on the bus controller

X2X Link connections:

- X67 module: M12, B-keyed connections (A → Input, B → Output)
- Bus controller: M12, B-keyed connection (B → Output)

Additional X67 stations can be connected without system supply module X67PS1300 depending on the power output of the bus controller.

The I/O image is made according to the order (cable sequence) of the X67 modules.

Information:

All subsequent I/O slots are shifted when an X67 station is connected/removed.

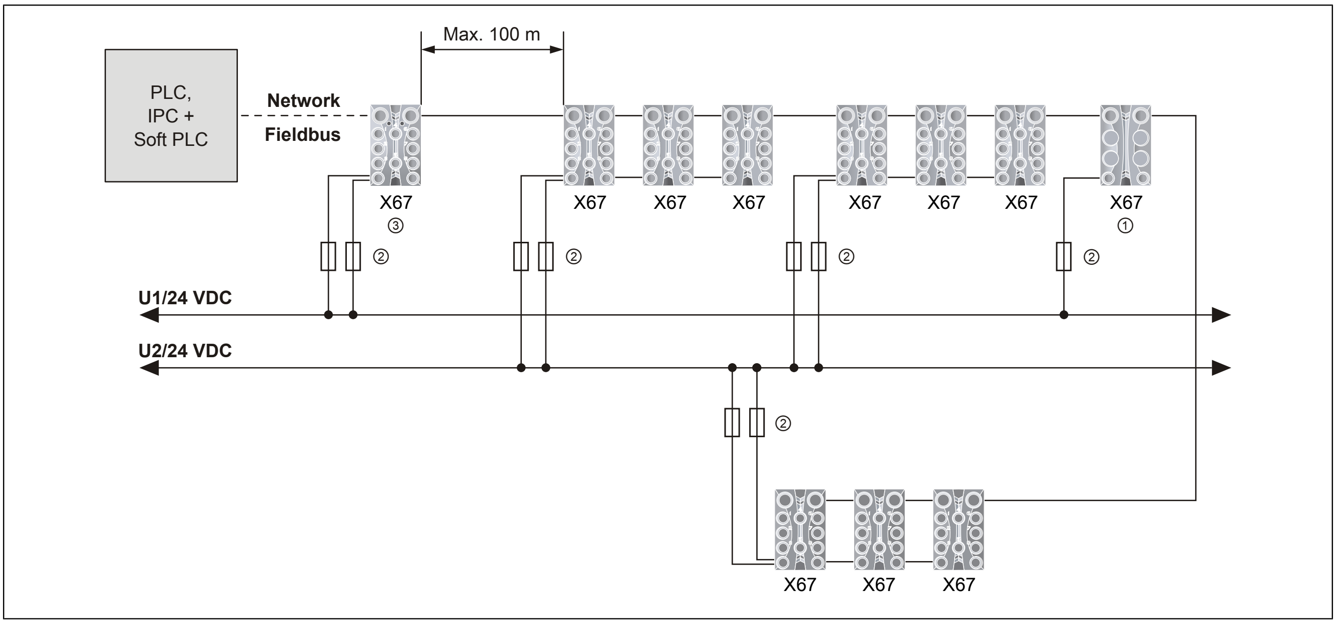


Figure 281: X2X Link cabling - Supply via bus controller

Legend

① System supply module
② Fuse, 4 A slow-blow
③ Bus controller

3.4.8.3 X2X Link cabling on bus transmitter X20BT9400

X2X Link connections:

- X67 module: M12, B-keyed connections (A → Input, B → Output)
- X20BT9400: X20 terminal block

Depending on the mounting orientation of the X20 system, 8 (horizontal installation) or 6 (vertical installation) X67 stations can be connected without system supply module X67PS1300.

Information:

All subsequent station numbers are shifted when an X67 station is connected/removed.

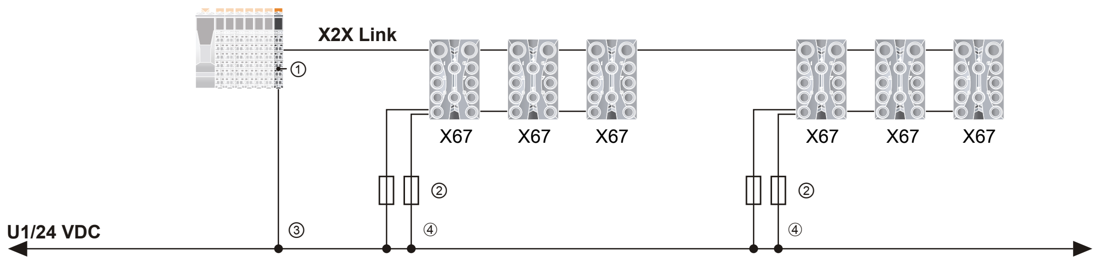


Figure 282: X2X Link cabling - Supply via bus transmitter X20BT9400

Legend
① X20 bus transmitter X20BT9400
② Fuse, 4 A slow-blow
③ X2X Link power supply
④ I/O power supply

If more than 8 or 6 X67 stations are connected to bus transmitter X20BT9400, then only the X67PS1300 system supply modules being used can be included when calculating the power requirements.

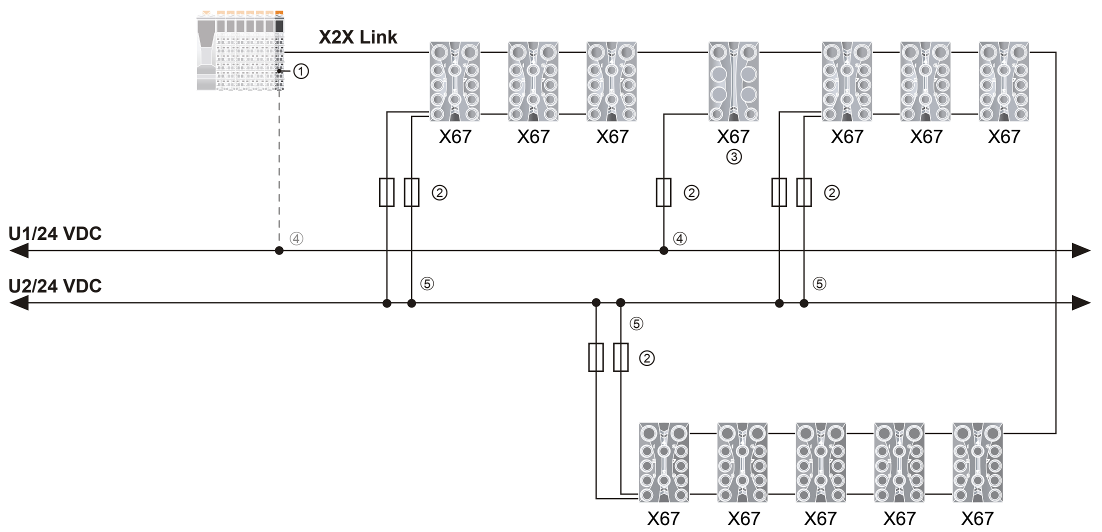


Figure 283: X2X Link cabling - Supply via X20 bus transmitter and X67 system supply

Legend
① X20 bus transmitter X20BT9400
② Fuse, 4 A slow-blow
③ System supply module
④ X2X Link power supply
⑤ I/O power supply

3.4.8.4 I/O power supply cabling

X67 module power supply connectors[1]:

- I/O modules, system supply:
  Connections C (input) and D (routing) are equal (pins connected).
- Bus controller:
  Connection C: 1 pair for I/O power supply, 1 pair for X2X Link supply
  Connector D: Routing of I/O power supply

Permissible [1] current consumption

- I/O modules: 8 A (without temperature derating)
- Bus controller: 4 A (without temperature derating)

Without the I/O power supply, the application has no access to the data points (see section "Failure of I/O supply (ModuleOK)" on page 834)! Only the node number is secured by an intact X2X Link power supply. The application must account for this accordingly if the I/O power supply is integrated in the emergency stop design (see section "The power supply concept" on page 833).

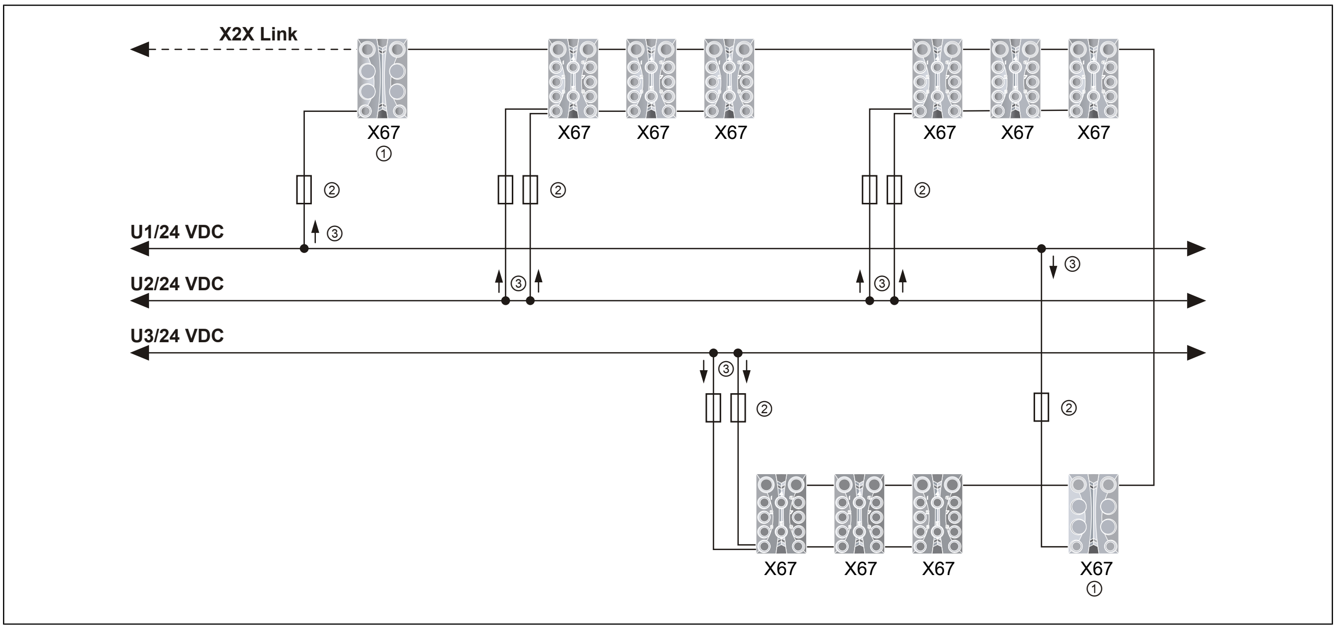


Figure 284: I/O power supply cabling - Isolated X2X Link and I/O power supply

Legend

① System supply module
② Fuse, 4 A slow-blow
③ Maximum 4 A

[1] For exact or possibly deviating technical data, please refer to the documentation for the corresponding X67 module.

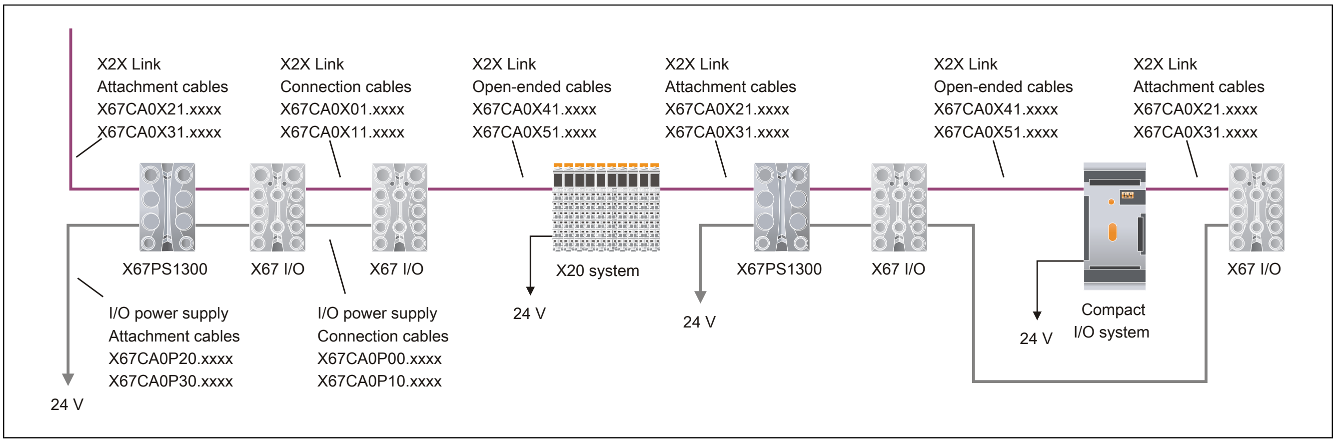
3.4.9 Combining X2X Link systems

The X2X Link provides a complete remote backplane, which is used for communicating between bus modules and over the X2X Link cable. Systems based on X2X Link can be combined with one another as needed.

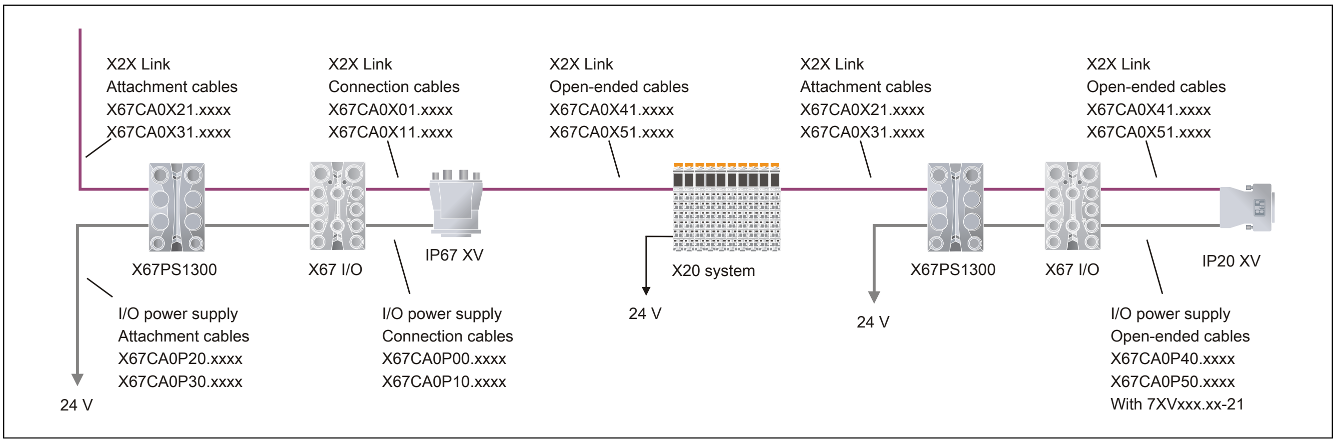
3.4.9.1 Connection overviews

The following connection overviews illustrate combinations of different systems that are based on X2X Link. The model numbers indicate which standard cables available from B&R can be used to connect with one another.

Combining X20, X67 and compact I/O system



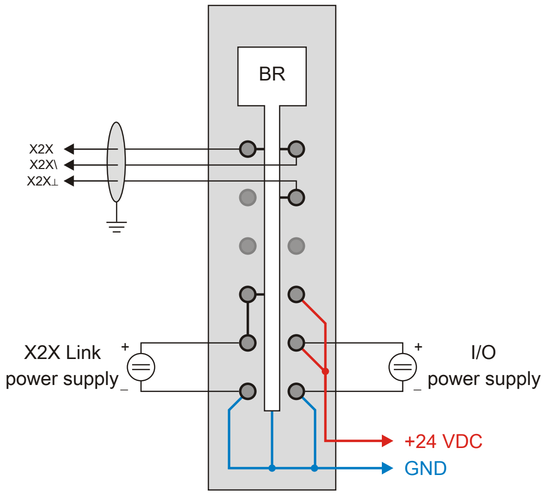
Combining X20, X67 and valve terminal connections



3.4.9.2 Connection examples

3.4.9.2.1 X20 system

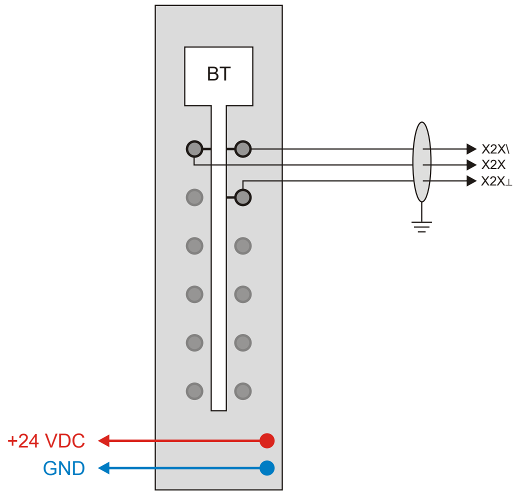
Connection example with bus receiver X20BR9300



Signal	Cable type	Model number
X2X Link In	Open-ended cables 1)	X67CA0X41.xxxx
		X67CA0X51.xxxx
	Cable for custom assembly	X67CA0X99.xxxx

1) In connection with X67 modules.

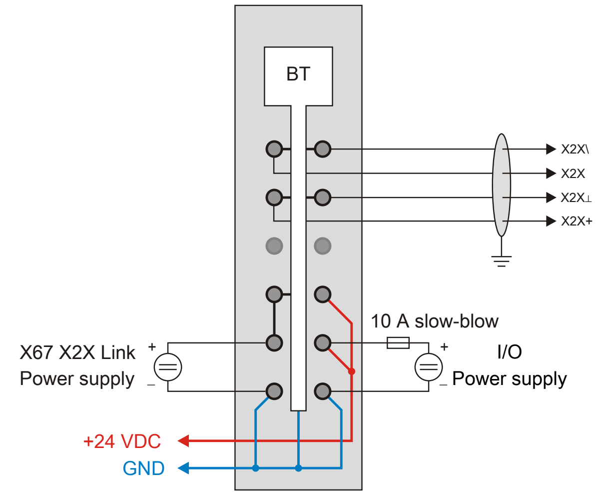
Connection example with bus transmitter X20BT9100



Signal	Cable type	Model number
X2X Link Out	Attachment cables 1)	X67CA0X21.xxxx
		X67CA0X31.xxxx
	Cable for custom assembly	X67CA0X99.xxxx

1) In connection with X67 modules.

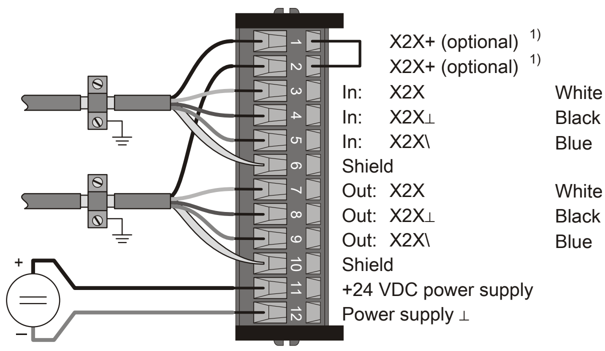
Connection example with bus transmitter X20BT9400



Signal	Cable type	Model number
X2X Link Out	Attachment cables [1]	X67CA0X21.xxxx
		X67CA0X31.xxxx
	Cable for custom assembly	X67CA0X99.xxxx

1) In connection with X67 modules.

3.4.9.2.2 Compact I/O system



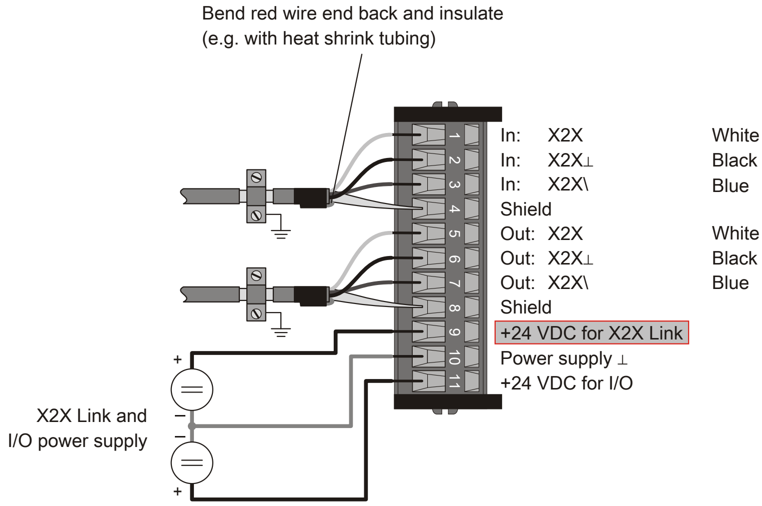
1) Used to forward the X2X Link power supply when using IP67 modules.

Signal	Cable type	Model number
X2X Link In	Open-ended cables[1]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.xxxx

1) Bridge for X2X+ in connection with X67 modules.

3.4.9.2.3 Valve connection

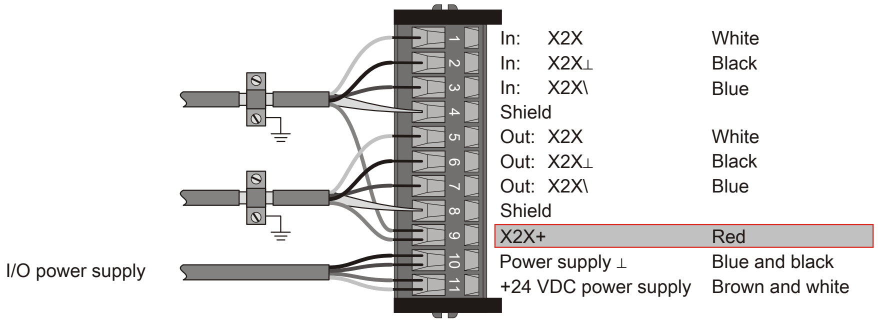
Connection example with 7XVxxx.xx-11/-12



Signal	Cable type	Model number
X2X Link In	Open-ended cables[1)]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1)]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.xxxx

1) In connection with X67 modules.

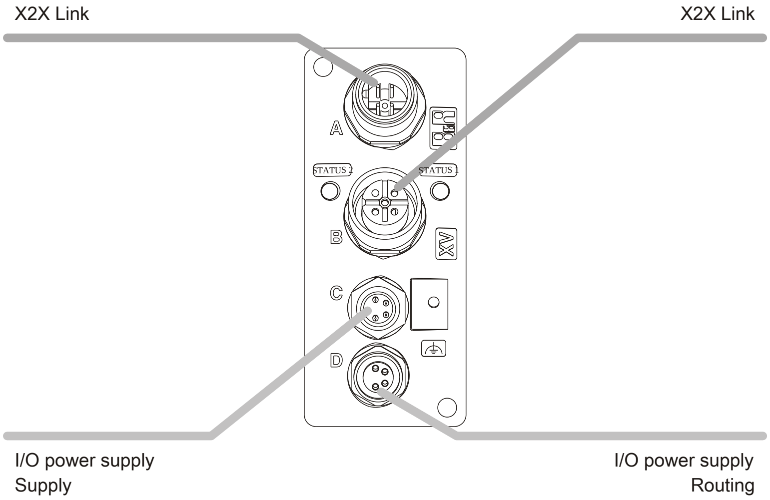
Connection example with 7XVxxx.xx-21



Signal	Cable type	Model number
X2X Link In	Open-ended cables[1)]	X67CA0X41.xxxx
		X67CA0X51.xxxx
X2X Link Out	Attachment cables[1)]	X67CA0X21.xxxx
		X67CA0X31.xxxx
X2X Link in/out	Cable for custom assembly	X67CA0X99.xxxx
I/O power supply	Open-ended cables[1)]	X67CA0P40.xxxx
		X67CA0P50.xxxx

1) In connection with X67 modules.

Connection example with 7XVxxx.xx-51/-62



Signal	Cable type	Model number
X2X Link	Connection cables[1]	X67CA0X01.xxxx
		X67CA0X11.xxxx
I/O power supply	Connection cables[1]	X67CA0P00.xxxx
		X67CA0P10.xxxx

1) In connection with X67 modules.

3.4.9.2.4 Connection of X2X Link interfaces with internal power supply

Information:

Power is supplied to the X2X Link network via the red wire. Power is supplied to the X2X Link network internally on the IF789 or LS189. An external power supply is therefore not needed.

In order to prevent short circuits on the housing, braided shield or cable shield, the red wire end must be insulated, e.g. using a heat shrink tubing.

To improve EMC immunity, the cable shield must be grounded on both sides. The supply line to the IF789 and LS189 should be grounded over a wide area close to the shield. The grounding on the X67 side is done via the connector on pre-assembled cables.

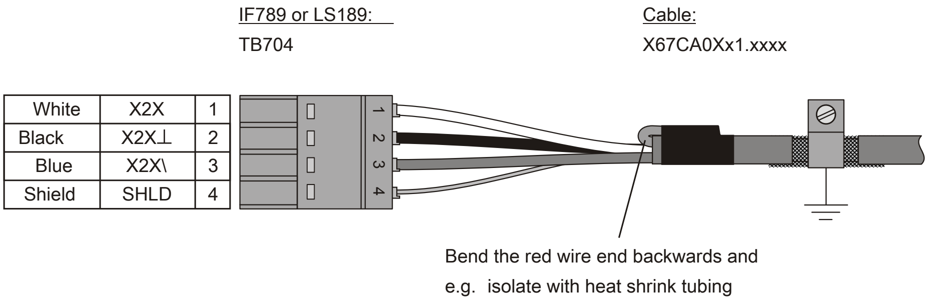


Figure 285: X2X Link attachment cable - Installation diagram

Information:

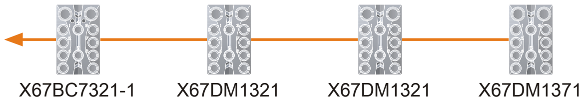
If the cable shield is not grounded, transmission errors may occur when strong electromagnetic interference occurs.

3.4.10 Calculating the power requirements

The power provided by the bus controllers and system supply modules is shown with a "+" sign. The power required by modules is shown with a "-" sign. To calculate the power requirements, the positive and negative power values must be added together. The sum is not permitted to be less than zero.

3.4.10.1 Example 1

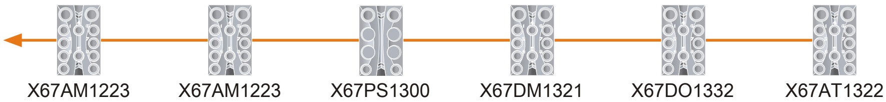
Calculation of the X2X Link power requirements and the module power required internally on the basis of the given hardware configuration. The X2X Link power requirements are balanced. X67PS1300 is not necessary. The internal module power consumption and the sensor/actuator power supply must come from the external power supply.



Module	X2X Link performance [W]	Internal module power [W]
X67BC7321-1	+3.00	10.30
X67DM1321	-0.75	2.50
X67DM1321	-0.75	2.50
X67DI1371	-0.75	1.00
Total	**+0.75**	**+16.30**

3.4.10.2 Example 2

Calculation of the X2X Link power requirements and the module power required internally on the basis of the given hardware configuration. The X2X Link power requirements calculation produces a surplus of +11.25 W. Therefore, one X67PS1300 is sufficient. The internal module power consumption and the sensor/actuator power supply must come from the external power supply.



Module	X2X Link performance [W]	Internal module power [W]
X67AM1223	-0.75	3.00
X67AM1223	-0.75	3.00
X67PS1300	+15.00	3.00 + 15.00
X67DM1321	-0.75	2.50
X67DO1332	-0.75	2.00
X67AT1322	-0.75	1.50
Total	**+11.25**	**+30.00**

3.5 Module overviews

3.5.1 Module overview X67 safety: Alphabetical

Model number	Short description	Page
X67SC4122.L12	X67 safe digital mixed module, 8 safe digital inputs, configurable input filter, 8 pulse outputs, 24 VDC, 4 safe type B1 digital outputs, 24 VDC, 2 A, OSSD < 500 µs, M12 connectors, high-density module	895
X67SI8103	X67 safe digital input module, 2x M12 interface each with 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC, 2x standardized 8-pin M12 device interface each with 1 digital input without safety function and 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC and 1 digital output without safety function, 24 VDC, 0.6 A and 1 device power supply, 24 VDC, 2 A	862

3.5.2 Module overview X67 safety: Grouped

Digital input modules

Model number	Short description	Page
X67SI8103	X67 safe digital input module, 2x M12 interface each with 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC, 2x standardized 8-pin M12 device interface each with 1 digital input without safety function and 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC and 1 digital output without safety function, 24 VDC, 0.6 A and 1 device power supply, 24 VDC, 2 A	862

Digital mixed modules

Model number	Short description	Page
X67SC4122.L12	X67 safe digital mixed module, 8 safe digital inputs, configurable input filter, 8 pulse outputs, 24 VDC, 4 safe type B1 digital outputs, 24 VDC, 2 A, OSSD < 500 µs, M12 connectors, high-density module	895

3.5.3 Digital input modules

3.5.3.1 Overview

Model number	Short description	Page
X67SI8103	X67 safe digital input module, 2x M12 interface each with 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC, 2x standardized 8-pin M12 device interface each with 1 digital input without safety function and 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC and 1 digital output without safety function, 24 VDC, 0.6 A and 1 device power supply, 24 VDC, 2 A	862

3.5.3.2 X67SI8103

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 1.1.4 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 395: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 396: Organization of general notices

3.5.3.2.1 General information

This module is equipped with 8 safe digital inputs. They are designed for a nominal voltage of 24 VDC.

The module can be used to read in digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. The module also provides pulse signals for diagnosing the sensor line.

- 8 safe digital inputs, sink circuit
- 2 pulse outputs - available on all 4 female connectors
- 2 standard inputs, sink circuit
- 2 standard outputs, source circuit
- Device supply
- Software input filter configurable for each channel
- Standardized 8-pin M12 device interface

3.5.3.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

3.5.3.2.2 Overview

Module	X67SI8103
Safe digital inputs	
Number of safe inputs	8
Number of standard inputs	2
Nominal voltage	24 VDC
Input filter	
Hardware	≤150 µs
Software	Default 0 ms, configurable between 0 and 500 ms
Input circuit	Sink
Pulse outputs	
Design	Push-Pull
Switching voltage	I/O power supply minus residual voltage
Digital outputs	
Number of standard outputs	2
Nominal voltage	24 VDC
Nominal output current	0.6 A
Total nominal current	1.2 A
Output protection	Thermal shutdown of individual channels in the event of overcurrent or short circuit, Integrated protection for switching inductive loads

Table 397: Digital mixed modules

3.5.3.2.3 Order data

Model number	Short description	Figure
	Digital input modules	
X67SI8103	X67 safe digital input module, 2x M12 interface each with 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC, 2x standardized 8-pin M12 device interface each with 1 digital input without safety function and 2 safe digital inputs, configurable input filter and 2 pulse outputs, 24 VDC and 1 digital output without safety function, 24 VDC, 0.6 A and 1 device power supply, 24 VDC, 2 A	

Table 398: X67SI8103 - Order data

Required accessories:
An overview of cabling X67 modules and associated model numbers for cables can be found in the module's download section on the B&R website (www.br-automation.com).

3.5.3.2.4 Technical data

Model number	X67SI8103
Short description	
I/O module	2x M12 interface each with 2 safe digital inputs and 2 pulse outputs, 24 VDC, 2x standardized 8-pin M12 device interface each with 1 digital input without safety function and 2 safe digital inputs and 2 pulse outputs, 24 VDC and 1 digital output without safety function, 24 VDC, 0.6 A and 1 device power supply, 24 VDC, 2 A
General information	
B&R ID code	0xBB7C
System requirements	
Automation Studio	3.0.81.15 or later
Automation Runtime	3.00 or later
SafeDESIGNER	2.70 or later
Safety Release	1.2 or later
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using status LED and software
I/O function	Yes, using status LED and software
Blackout mode	
Scope	Module
Function	Module function
Standalone mode	No
Max. I/O cycle time	1 ms
Connection type	
X2X Link	M12, B-coded
Inputs/Outputs	M12 8-pin or M12 5-pin, A-coded
I/O power supply	M8, 4-pin
Power consumption	
Bus	0.9 W
Internal I/O	2.1 W
Electrical isolation	
Channel - Bus	Yes
Channel - Channel	No
Certifications	
CE	Yes
KC	Yes
EAC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA IIA T5 Gc IP67, Ta = 0 - Max. 60°C TÜV 05 ATEX 7201X
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3
Functional safety	EN 50156-1:2004
Safety characteristics	
EN ISO 13849-1:2015	
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels [1)]
PL	PL e
DC	>94%
MTTFD	2500 years
Mission time	Max. 20 years
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
SIL CL	SIL 3
SFF	>90%
PFH / PFH_d	
Module	$<1*10^{-10}$
openSAFETY wired	Negligible
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour
PFD	$<2*10^{-5}$
Proof test interval (PT)	20 years

Table 399: X67SI8103 - Technical data

Model number	X67SI8103
24 VDC output	
Output voltage	24 VDC -15% / +20%
Output current	2 A
I/O power supply	
Nominal voltage	24 VDC
Voltage range	18 to 30 VDC
Integrated protection	Reverse polarity protection
Digital inputs	
Nominal voltage	24 VDC
Input voltage	24 VDC -15% / +20%
Input current at 24 VDC	Max. 7.24 mA
Input characteristics per EN 61131-2	Type 1
Input filter	
Hardware	≤150 µs
Input circuit	Sink
Input resistance	Min. 3.3 kΩ
Switching threshold	
Low	<5 VDC
High	>15 VDC
Isolation voltage between channel and bus	500 V_{eff}
Safe digital inputs	
Nominal voltage	24 VDC
Input characteristics per EN 61131-2	Type 1
Input filter	
Hardware	≤150 µs
Software	Configurable between 0 and 500 ms
Input circuit	Sink
Input voltage	24 VDC -15% / +20%
Input current at 24 VDC	Max. 8.28 mA
Input resistance	Min. 2.9 kΩ
Error detection time	200 ms
Isolation voltage between channel and bus	500 V_{eff}
Switching threshold	
Low	<5 VDC
High	>15 VDC
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line
Digital outputs	
Variant	FET, positive switching, output level readable
Nominal voltage	24 VDC
Switching voltage	I/O power supply minus residual voltage
Nominal output current	0.6 A
Total nominal current	1.2 A
Output protection	Thermal shutdown of individual channels in the event of overcurrent or short circuit, integrated protection for switching inductive loads [2)]
Leakage current when switched off	<500 µA
Residual voltage	≤300 mVDC at nominal current
Peak short-circuit current	<12 A
Braking voltage when switching off inductive loads	50 VDC
Isolation voltage between channel and bus	500 V_{eff}
Max. capacitive load	100 nF
Peak output current	1 A
Pulse outputs	
Variant	Push-Pull
Nominal output current	40 mA
Output protection	Shutdown of individual channels in the event of overload or short circuit [2)]
Peak short-circuit current	25 A for 15 µs
Short-circuit current	100 mA_{eff}
Leakage current when switched off	0.1 mA
Residual voltage	3 VDC
Switching voltage	I/O power supply minus residual voltage
Total nominal current	80 mA
Operating conditions	
Mounting orientation	
Any	Yes
Installation elevation above sea level	0 to 2000 m, no limitation
Degree of protection per EN 60529	IP67
Ambient conditions	
Temperature	
Operation	-40 to 60°C [3)]
Storage	-40 to 85°C
Transport	-40 to 85°C

Table 399: X67SI8103 - Technical data

Model number	X67SI8103
Mechanical properties	
Dimensions	
Width	53 mm
Height	85 mm
Depth	42 mm
Weight	190 g
Torque for connections	
M8	Max. 0.4 Nm
M12	Max. 0.6 Nm

Table 399: X67SI8103 - Technical data

1) The related danger notices in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Up to firmware version < 325: 0 to 60°C, firmware version 325 and later and up to hardware upgrade < 1.10.1.1 and hardware revision < G0: -25 to 60°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For additional information about installation, see chapter "Installation notes for X67 modules" on page 24.

3.5.3.2.5 LED status indicators

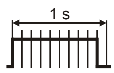
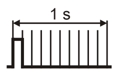
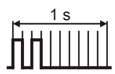
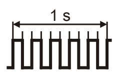
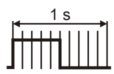
Figure	LED	Color	Status	Description
Status indicator re: Left: green (r), Right: red (e)  Status indicator SE Left: red (S); Right: red (E)	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1 2 3 4	Status of the corresponding device		
			Off	Without signal function: No error, all signals from female connector off ("low" state) Dual-channel evaluation: No error, dual-channel evaluation FALSE ("low" state)
		Green	On	Without signal function: All inputs on the female connector set ("high" state) Dual-channel evaluation: Dual-channel evaluation signal TRUE ("high" state)
			Blinking	Without signal function: Only one input on the female connector set ("high" state) Dual-channel evaluation: -
		Red	On	Without signal function: Error on all inputs of the female connector Dual-channel evaluation: Error in dual-channel evaluation
			Blinking	Without signal function: Error on only one input of the female connector, the signal is NOT set on the second input ("low" state) Dual-channel evaluation: -
		Red / Green	Blinking	Without signal function: Error on only one input of the female connector, the signal is set on the second input ("high" state) Dual-channel evaluation: -
	SE	Red	Off	Mode RUN or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 400: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

3.5.3.2.6 Connection elements

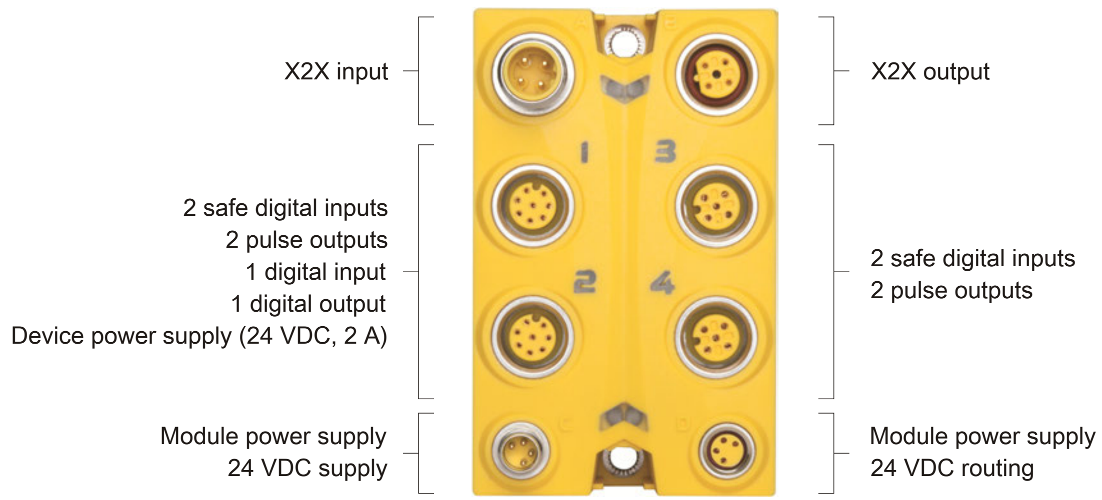


Figure 286: X67SI8103 - Connection elements

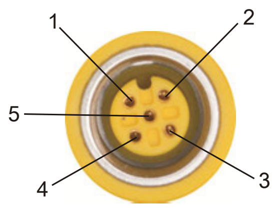
Pinout	Female connector	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5
	3 (IN)	Pulse 1	SI 5	GND	SI 6	Pulse 2
	4 (IN)	Pulse 1	SI 7	GND	SI 8	Pulse 2

Table 401: Pinout

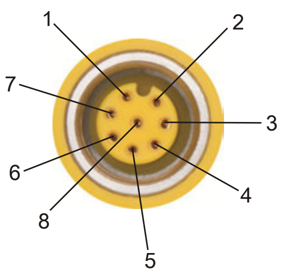
Pinout	Female connector	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
	1 (IN/OUT)	+24 VDC	Pulse 1	GND	SI 1	DI 1	Pulse 2	SI 2	DO 1
	2 (IN/OUT)	+24 VDC	Pulse 1	GND	SI 3	DI 2	Pulse 2	SI 4	DO 2

Table 402: Pinout

3.5.3.2.7 X2X Link

This module is connected to X2X Link using pre-assembled cables. The connection is made using a circular connector (2x M12, 4-pin).

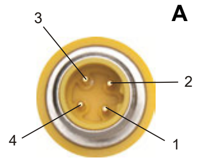
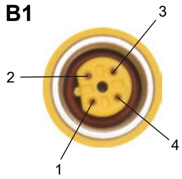
Connection	Pinout	
	Pin	**Name**
A  B1 	1	X2X+
	2	X2X
	3	X2X⊥
	4	X2X\
	A ... B-coded male connector on the module, input B1 ... B-coded female connector on the module, output SHLD ... Shielding provided by threaded insert in the module	

Table 403: X2X Link

3.5.3.2.8 24 VDC module supply

The module supply is connected using pre-assembled cables with circular connectors (2x M8, 4-pin). The supply is connected via the male C connector. Female connector D is used for routing the supply to other modules.

The maximum permissible current per supply is 4 A (in summation 8 A)!

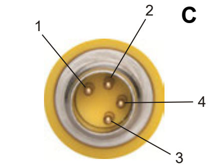
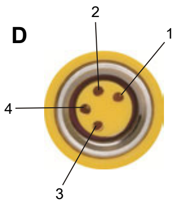
Connection	Pinout	
	Pin	**Name**
C  D 	1	24 VDC module supply [1)]
	2	24 VDC module supply [1)]
	3	GND
	4	GND
	C ... Male connector on the module, power supply D ... Female connector on the module, supply routing 1) Both supply pins must be supplied. It can only be ensured that the outputs are switched off if **both** pins are disconnected from the supply. If the summation current of the outputs is >4 A, current must also be supplied via female connector D, pin 2.	

Table 404: 24 VDC module supply

3.5.3.2.9 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

3.5.3.2.9.1 Connecting single-channel sensors with contacts

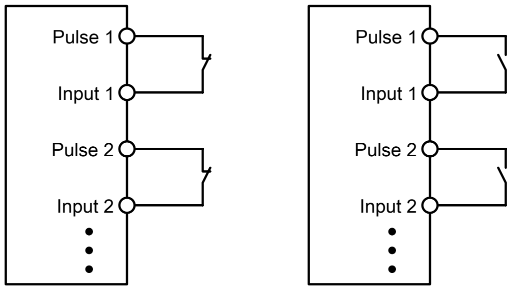


Figure 287: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

3.5.3.2.9.2 Connecting two-channel sensors with contacts

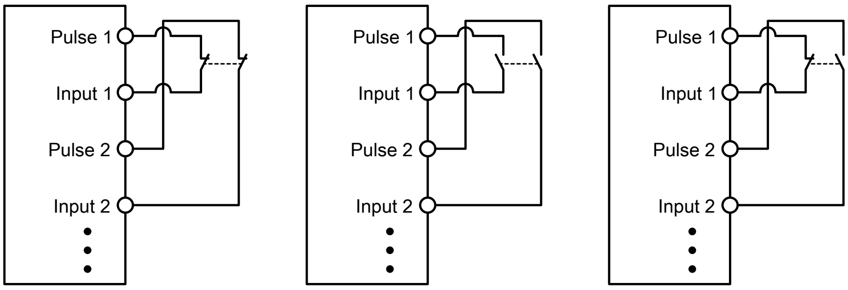


Figure 288: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

3.5.3.2.9.3 Connecting multi-channel sensors with contacts

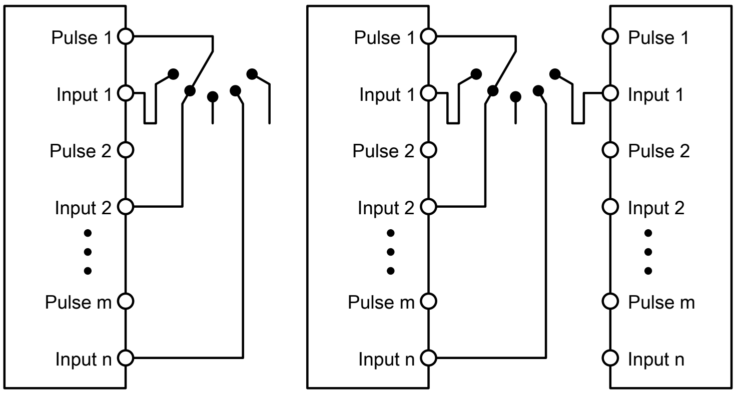


Figure 289: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

3.5.3.2.9.4 Connecting electronic sensors

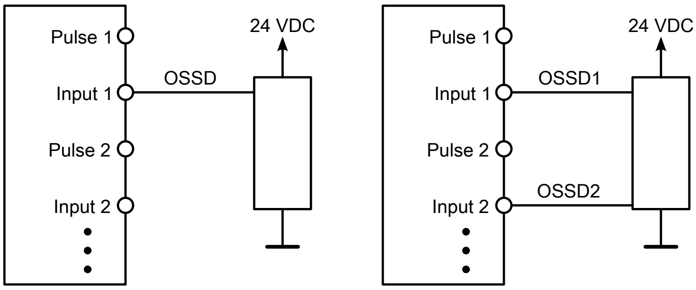


Figure 290: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

3.5.3.2.9.5 Using the same pulse signals

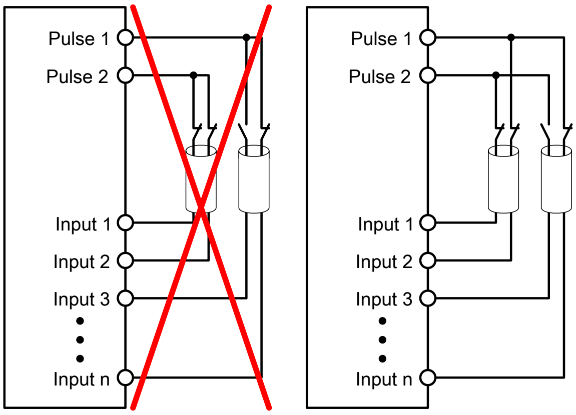


Figure 291: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

3.5.3.2.10 Error detection

3.5.3.2.10.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

3.5.3.2.10.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 405: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	**Closed**
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 406: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 407: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

3.5.3.2.11 Input circuit diagram

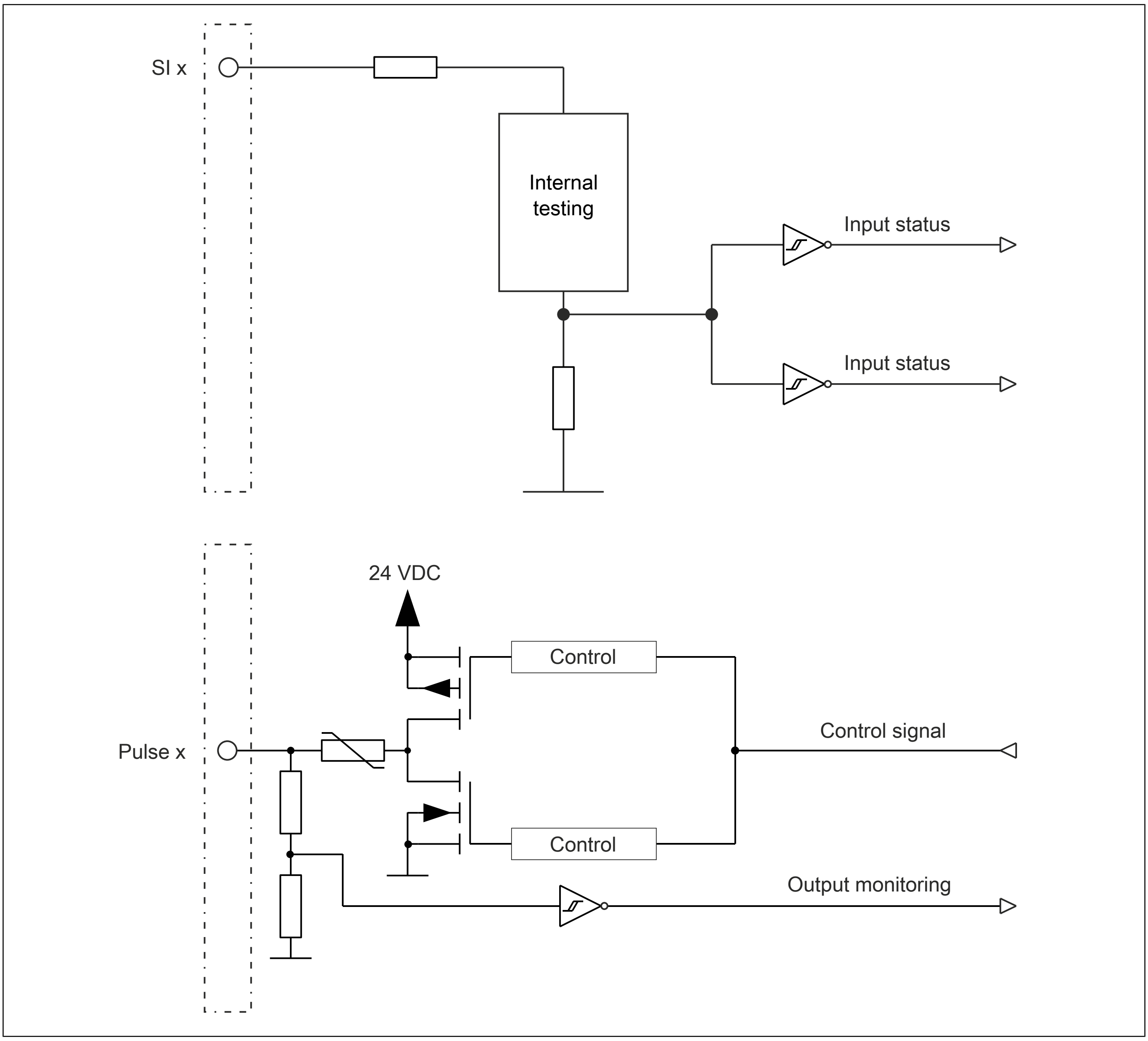


Figure 292: Input circuit diagram

3.5.3.2.12 Input circuit diagram - Standard input without safety function

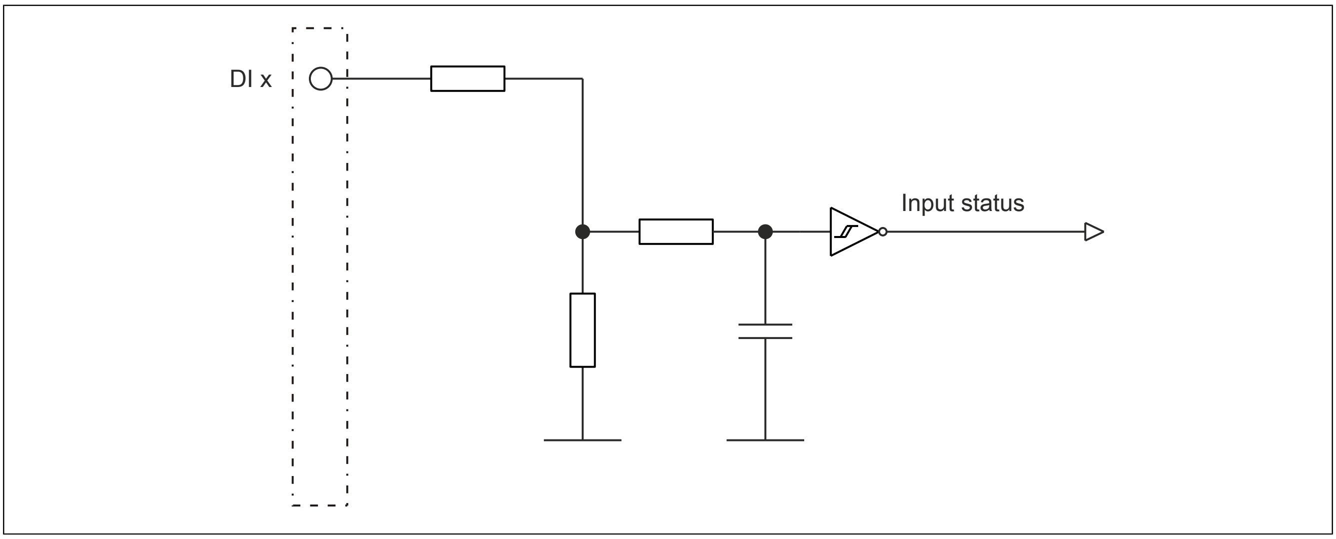


Figure 293: Input circuit diagram - Standard input without safety function

3.5.3.2.13 Output circuit diagram - Standard output without safety function

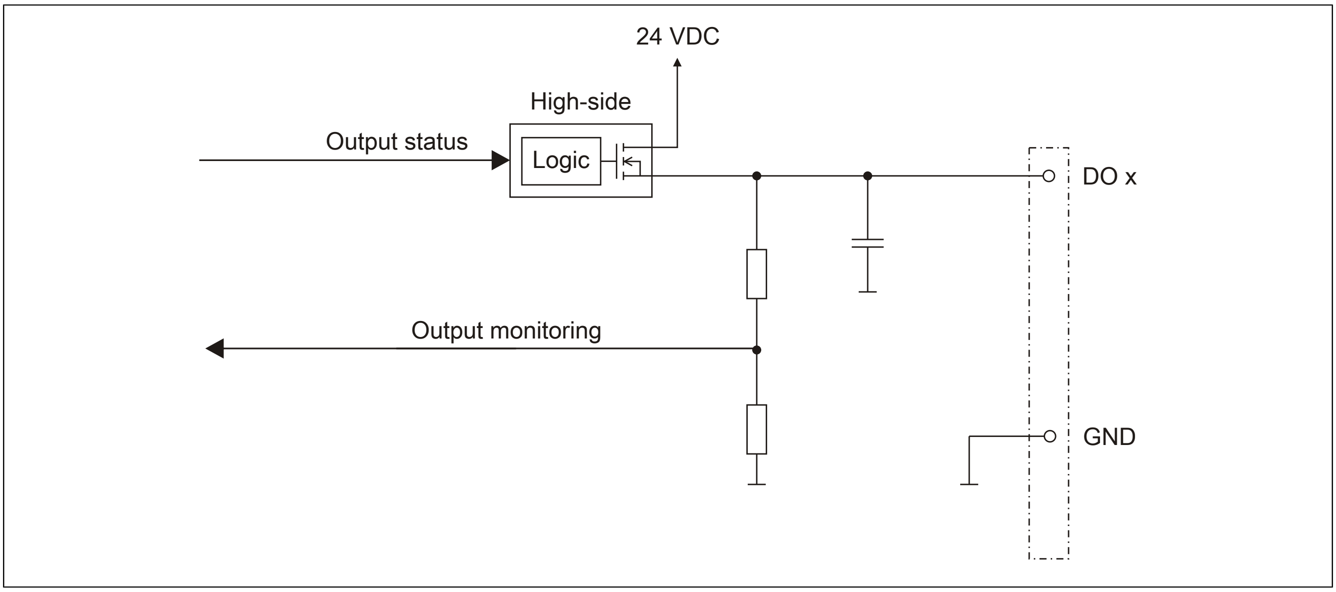


Figure 294: Output circuit diagram - Standard output without safety function

3.5.3.2.14 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 µs

3.5.3.2.15 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 µs

Maximum I/O update time
2150 µs + Filter time (see chapter "Filter")

3.5.3.2.16 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

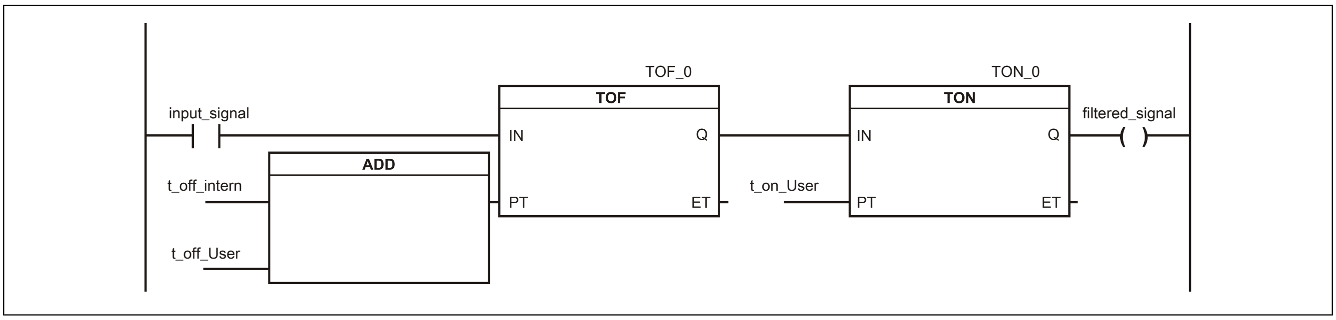


Figure 295: SI input filter - Diagram 1

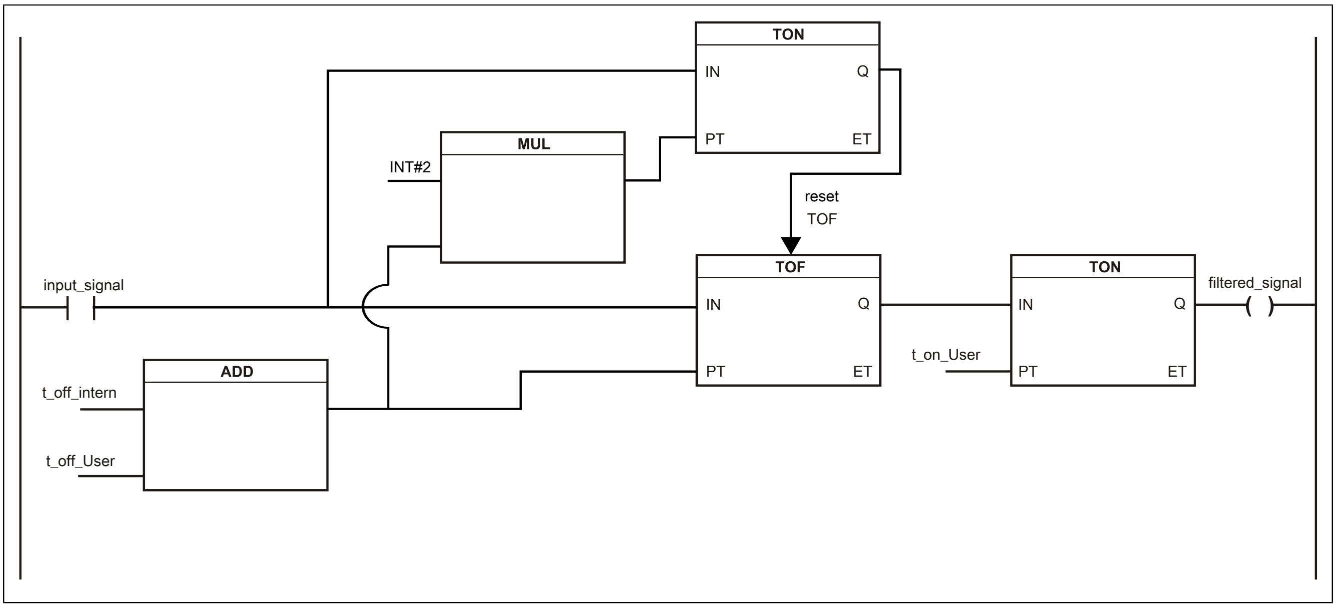


Figure 296: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

3.5.3.2.17 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

3.5.3.2.18 Register description

3.5.3.2.18.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 408: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.1.0 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Input status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
</table>

Table 409: I/O configuration parameters: General

Group: Output signal path - Release 1.10 and later

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutput0102</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 410: I/O configuration parameters: Output signal path

3.5.3.2.18.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 411: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr>
<tr><td>No</td><td>No requirement for synchronization of the networks</td></tr>
</table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 412: SafeDESIGNER parameters: Safety_Response_Time

Group: Connectorxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Pulse_Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.
<table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.</td></tr><tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr></table></td><td>Internal</td><td>-</td></tr>
<tr><td>SafeDigitalInputxx_Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>SafeDigitalInputxx_Filter_On_us</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>0</td><td>µs</td></tr>
<tr><td>TwoChannelProcessingMode</td><td>This parameter determines the type of dual-channel evaluation.
Permissible values:
• None
• Equivalent
• Antivalent</td><td>None</td><td></td></tr>
<tr><td>InvertDigitalInputxx
(Parameter only available on "Connector 1" and "2")</td><td>This parameter determines whether the value on the corresponding input is inverted.</td><td>No</td><td></td></tr>
<tr><td>InvertDigitalOutputxx
(Parameter only available on "Connector 1" and "2")</td><td>This parameter determines whether the value on the corresponding output is inverted.</td><td>No</td><td></td></tr>
</table>

Table 413: SafeDESIGNER parameters: Connectorxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

3.5.3.2.18.3 Parameters in SafeDESIGNER - Release 1.10 and higher

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 414: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr>
<tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr>
</table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 415: SafeDESIGNER parameters: Safety Response Time

Group: Connectorxx

Parameter	Description	Default value	Unit
SafeDigitalInputxx Pulse Mode	This parameter can be used to specify the pulse mode for the input channel. <table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Internal</td><td>The channel works exclusively with the associated pulse output.</td></tr><tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr></table>	Internal	-
SafeDigitalInputxx Filter Off	Switch-off filter for the channel to remove potentially disruptive signal low phases. • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	0	µs
SafeDigitalInputxx Filter On	Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short. • Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)	200000	µs
Discrepancy Time	This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error. • Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)	50000	µs
TwoChannelProcessingMode	This parameter determines the type of dual-channel evaluation. Permissible values: • None • Equivalent • Antivalent	None	
InvertDigitalInputxx (Parameter only available on "Connector 1" and "Connector 2")	This parameter determines whether the value on the corresponding input is inverted.	No	
InvertDigitalOutputxx (Parameter only available on "Connector 1" and "Connector 2")	This parameter determines whether the value on the corresponding output is inverted.	No	

Table 416: SafeDESIGNER parameters: Connectorxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

3.5.3.2.18.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:
Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>TwoChannelInputxxyy_state

Hardware upgrade 1.9.0.0 or later:
PLCopenFBKxxyy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
</table>

Table 417: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>PulseoutputErrors</td><td>(Read) [1)]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 9 = Channel 1 to 2</td><td>Bit no. 0 to 1 = Channel 1 to 2</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeTwoChannelOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of channel SI xx/yy</td></tr>
<tr><td>DigitalInputxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Physical channel DI xx</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Physical channel DO xx</td></tr>
<tr><td>DigitalOutputxxOK</td><td>Read</td><td>Read</td><td>BOOL</td><td>Status of channel DO xx</td></tr>
<tr><td>PhysicalStateOutputxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel DO xx</td></tr>
</table>

Table 417: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

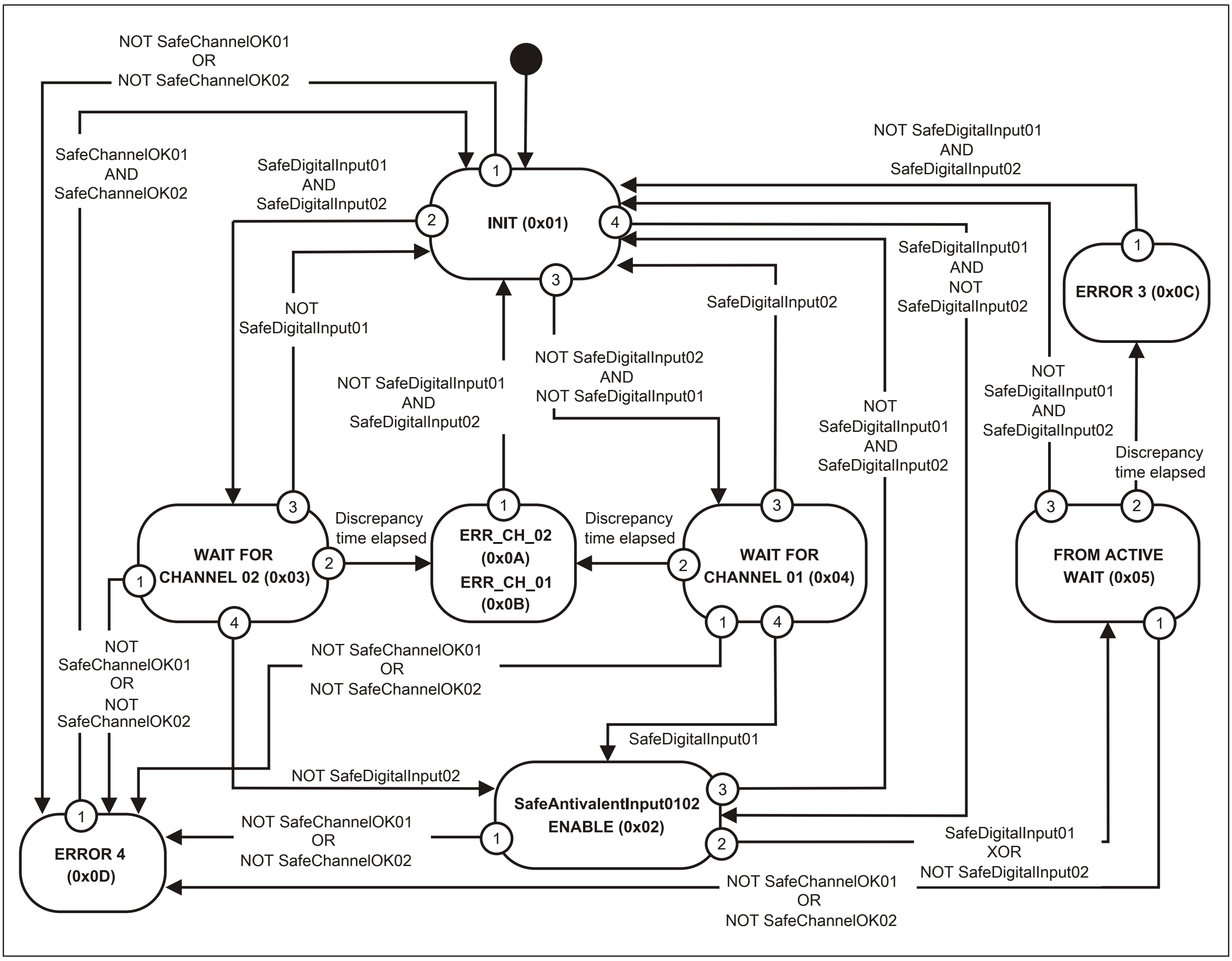


Figure 297: "Antivalent" function block - State diagram

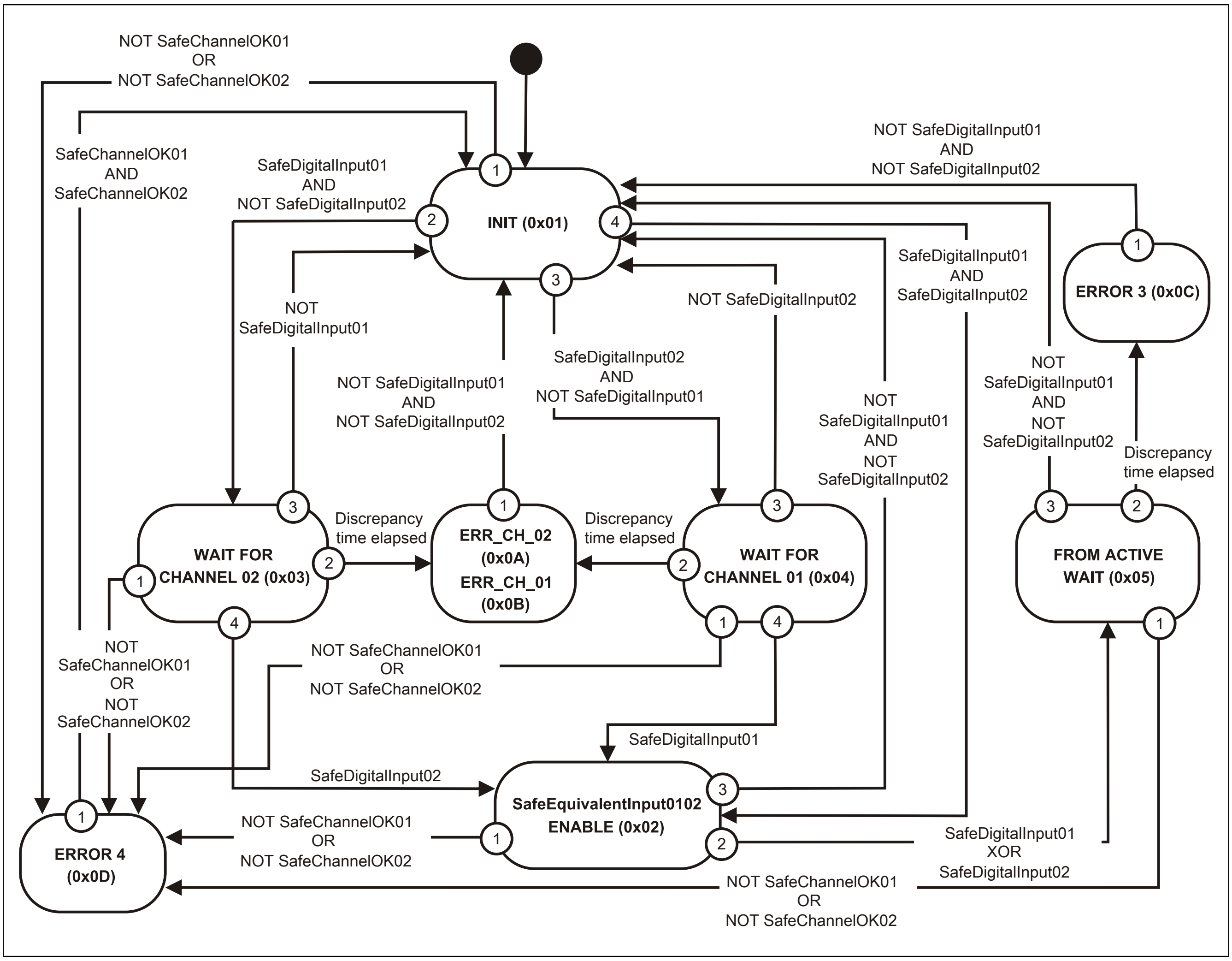


Figure 298: "Equivalent" function block - State diagram

3.5.4 Digital mixed modules

3.5.4.1 Overview

Model number	Short description	Page
X67SC4122.L12	X67 safe digital mixed module, 8 safe digital inputs, configurable input filter, 8 pulse outputs, 24 VDC, 4 safe type B1 digital outputs, 24 VDC, 2 A, OSSD < 500 μs, M12 connectors, high-density module	895

3.5.4.2 X67SC4122.L12

The module description included in this section is simply a non-certified excerpt from the module data sheet.

Version 1.141 of the data sheet is incorporated in this section.

The following sections in the user's manual are documented in a central location and therefore are not listed separately for the individual modules:

- 1.3.4 "Safety response time"
- 1.2 "Intended use"
- 1.1.2 "Release information"
- 1.1.4 "EC declaration of conformity"

Information:

B&R makes every effort to keep user's manuals as current as possible. From a safety point of view however, the current certified version of the data sheet must be used.

The current certified data sheet – including a detailed version history – is available for download from the B&R website at www.br-automation.com.

Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

Table 418: Organization of safety notices

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

Table 419: Organization of general notices

3.5.4.2.1 General information

The module is equipped with 8 safe digital inputs and 4 safe digital outputs. They are designed for a nominal voltage of 24 VDC.

The module can be used to read in digital signals and to control actuators in safety-related applications up to PL e or SIL 3.

The node number switch for setting the X2X Link address is a unique feature. When modular machine configurations change, it is required, for example, to define certain module groups at a fixed address that is independent of the preceding modules in the line. All subsequent standard modules refer to this offset and use it automatically for addressing purposes.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. The module also provides pulse signals for diagnosing the sensor line.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of switching cycles. The "high-side high-side" variant (output type B) is required for actuators with reference potential (e.g. enable inputs on frequency inverters). It is important to observe the special notices for the wiring in this case. Safe digital output modules are equipped with protection against automatic restart in the event of network errors.

- 8 safe digital inputs, sink circuit
- 8 pulse outputs
- Software input filter configurable for each channel
- 4 safe digital outputs, output type B with 2 A, source circuit
- Node number switches for setting the X2X Link address
- Integrated output protection

3.5.4.2.1.1 Function

Safe digital inputs

The module is equipped with safe digital input channels. It can be flexibly used for a wide range of tasks involving the reading of digital signals in safety-related applications up to PL e or SIL 3.

The module is equipped with filters that are individually configurable for switch-on and switch-off behavior. Switch-on filters are used to filter out signal disturbances. Switch-off filters are used to smooth testing gaps in external signal sources – i.e. OSSD signals – so that unintended cutoffs can be avoided.

The input signals of signal pairs (channels 1 and 2, 3 and 4, etc.) are monitored in the module for simultaneity. The maximum permitted discrepancy of inputs of a signal pair is configurable. Here, the signals of dual-channel evaluation directly represent the safe signal of a 2-channel sensor, such as from an E-stop button or safety light curtain.

The module provides pulse signals for diagnosing the sensor line. By default, each pulse signal provides a unique pulse pattern derived from the module's serial number and pulse channel number. This allows any pulse signals to be combined in one signal cable and still cover any cross fault combinations in the cable. The pulse check can also be disabled to connect electronic sensors with separate line monitoring (OSSD signals).

Safe digital outputs

The module is equipped with safe digital output channels. It can be flexibly used for controlling actuators in safety-related applications up to PL e or SIL 3.

The outputs are designed using semiconductor technology so that the safety-related characteristics do not depend on the number of operating cycles. In order to handle all situations involving actuators, there are basically 2 different types of outputs: the high-side - low-side variant (type A) and the high-side - high-side variant (type B). Type A outputs have safety-related advantages since the actuator can be cut off in its connection cable in all error scenarios. Type A outputs are limited to actuators without ground potential (e.g. relays, valves). For actuators with ground potential (e.g. enable inputs on frequency inverters), type B outputs are required. It is important to observe the special notices for the cabling in this case.

Safe digital output channels provide protection against automatic restart when network errors occur. Function blocks needed to fulfill additional requirements regarding protection against automatic restart are available in SafeDESIGNER. The outputs can also be controlled by the standard application. The combination of safety-related control and standard control is arranged such that the execution of a cutoff request always has top priority. For diagnostic purposes, the outputs are designed to be read back.

Depending on the product, the safe digital output channels are equipped with current measurement for detecting open circuits. This function can also be used to monitor muting lamps, for example.

The testing of the semiconductors that is necessary from a safety point of view results in what are known as OSSD low phases in many products. The effect of this is that when an output is active (high state), a switch-off situation (low state) occurs for a very brief amount of time. The test can be cut off if this behavior leads to problems in the application. Observe the associated safety-related notices!

openSAFETY

This module uses the protective mechanisms of openSAFETY when transferring data to the various bus systems. Because the data is encapsulated in the openSAFETY container in a fail-safe manner, the components on the network that are involved in the transfer do not require any additional safety-related features. At this point, only the safety-related characteristic values specified for openSAFETY in the technical data are to be consulted. The data in the openSAFETY container undergoes safety-related processing only when received by the remote station; for this reason, only this component is involved from a safety point of view. Read access to the data in the openSAFETY container for applications without safety-related characteristics is permitted at any point in the network without affecting the safety-related characteristics of openSAFETY.

open
SAFETY

3.5.4.2.2 Overview

Module	X67SC4122.L12
Safe digital inputs	
Number of inputs	8
Nominal voltage	24 VDC
Input filter Hardware Software	 ≤150 µs Default 0 ms, configurable between 0 and 500 ms
Input circuit	Sink
Safe digital outputs	
Number of outputs	4
Nominal voltage	24 VDC
Nominal output current	2 A
Total nominal current	5 A
Output protection	Thermal shutdown of individual channels in the event of overcurrent or short circuit, integrated protection for switching inductive loads
Pulse outputs	
Design	Push-Pull
Switching voltage	I/O power supply minus residual voltage

Table 420: Digital mixed modules

3.5.4.2.3 Order data

Model number	Short description	Figure
	Digital mixed modules	
X67SC4122.L12	X67 safe digital mixed module, 8 safe digital inputs, configurable input filter, 8 pulse outputs, 24 VDC, 4 safe type B1 digital outputs, 24 VDC, 2 A, OSSD <500 µs, M12 connectors, high-density module	

Table 421: X67SC4122.L12 - Order data

Required accessories:
An overview of cabling X67 modules and associated model numbers for cables can be found in the module's download section on the B&R website (www.br-automation.com).

3.5.4.2.4 Technical data

Model number	**X67SC4122.L12**
Short description	
I/O module	8 safe digital inputs, 8 pulse outputs, 24 VDC, 4 safe type B1 digital outputs, 24 VDC, 2 A, OSSD < 500 µs
General information	
B&R ID code	0xA7A6
System requirements	
Automation Studio	3.0.80 or later
Automation Runtime	3.00 or later
SafeDESIGNER	2.70 or later
Safety Release	1.2 or later
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using status LED and software
Outputs	Yes, using status LED and software
Inputs	Yes, using status LED and software
Blackout mode	
Scope	Module
Function	Module function
Standalone mode	No
Max. I/O cycle time	1 ms
Connection type	
X2X Link	M12, B-coded
Inputs/Outputs	M12, A-coded
I/O power supply	M8, 4-pin
Power consumption	
Bus	0.8 W
Internal I/O	1.8 W
Electrical isolation	
Channel - Bus	Yes
Channel - Channel	No
Certifications	
CE	Yes
KC	Yes
EAC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA IIA T5 Gc IP67, Ta = 0 - Max. 60°C TÜV 05 ATEX 7201X
Functional safety	cULus FSPC E361559 Energy and industrial systems Certified for functional safety ANSI UL 1998:2013
Functional safety	IEC 61508:2010, SIL 3 EN 62061:2013, SIL 3 EN ISO 13849-1:2015, Cat. 4 / PL e IEC 61511:2004, SIL 3
Functional safety	EN 50156-1:2004
Safety characteristics	
EN ISO 13849-1:2015	
MTTFD	2500 years
Mission time	Max. 20 years
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
PFH / PFH_d	
Module	$<1*10^{-10}$
openSAFETY wired	Negligible
openSAFETY wireless	$<1*10^{-14}$ * Number of openSAFETY packets per hour
PFD	$<2*10^{-5}$
Proof test interval (PT)	20 years

Table 422: X67SC4122.L12 - Technical data

Model number	X67SC4122.L12
Safe digital inputs	
EN ISO 13849-1:2015	
Category	Cat. 3 when using individual input channels, Cat. 4 when using input channel pairs (e.g. SI1 and SI2) or more than 2 input channels 1)
PL	PL e
DC	>94%
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
SIL CL	SIL 3
SFF	>90%
Safe digital outputs	
EN ISO 13849-1:2015	
Category	Cat. 3 if parameter "Disable OSSD = Yes-ATTENTION", Cat. 4 if parameter "Disable OSSD = No" 1)
PL	PL d if parameter "Disable OSSD = Yes-ATTENTION", PL e if parameter "Disable OSSD = No" 1)
DC	>60% if parameter "Disable OSSD = Yes-ATTENTION", >94% if parameter "Disable OSSD = No" 1)
IEC 61508:2010, IEC 61511:2004, EN 62061:2013	
SIL CL	SIL 2 if parameter "Disable OSSD = Yes-ATTENTION", SIL 3 if parameter "Disable OSSD = No" 1)
SFF	>60% if parameter "Disable OSSD = Yes-ATTENTION", >90% if parameter "Disable OSSD = No" 1)
I/O power supply	
Nominal voltage	24 VDC
Voltage range	18 to 30 VDC
Integrated protection	Reverse polarity protection
Safe digital inputs	
Nominal voltage	24 VDC
Input characteristics per EN 61131-2	Type 1
Input filter	
Hardware	≤150 µs
Software	Configurable between 0 and 500 ms
Input circuit	Sink
Input voltage	24 VDC -15% / +20%
Input current at 24 VDC	Max. 4.59 mA
Input resistance	Min. 5.23 kΩ
Error detection time	200 ms
Isolation voltage between channel and bus	500 V_{eff}
Switching threshold	
Low	<5 VDC
High	>15 VDC
Line length between pulse output and input	Max. 60 m with unshielded line Max. 400 m with shielded line
Safe digital outputs	
Variant	FET, 2x positive switching, type B1, output level readable
Nominal voltage	24 VDC
Nominal output current	2 A
Total nominal current	5 A
Output protection	Thermal shutdown of individual channels in the event of overcur- rent or short circuit, integrated protection for switching inductive loads 2)
Braking voltage when switching off inductive loads	Max. 45 VDC
Error detection time	1 s
Isolation voltage between channel and bus	500 V_{eff}
Peak short-circuit current	Max. 40 A < 1 ms
Leakage current when switched off	100 µA
Residual voltage	≤700 mVDC
Switching voltage	I/O power supply minus residual voltage
Max. switching frequency	1000 Hz
Test pulse length	Max. 1 ms
Max. capacitive load	100 nF
Peak output current	2.5 A (effective current ≤ 2 A)
Minimum load	12 mA
Current on loss of ground	
I_{OUT}	<3 mA, hardware revision B2 and later: <1 mA
I_{GND}	<110 mA
Pulse outputs	
Variant	Push-Pull
Nominal output current	50 mA
Output protection	Shutdown of individual channels in the event of overload or short circuit 2)
Peak short-circuit current	25 A for 5 ms

Table 422: X67SC4122.L12 - Technical data

Model number	X67SC4122.L12
Short-circuit current	1.4 A_{eff}
Leakage current when switched off	0.1 mA
Residual voltage	0.3 VDC
Switching voltage	I/O power supply minus residual voltage
Total nominal current	400 mA
Operating conditions	
Mounting orientation	
Any	Yes
Installation elevation above sea level	0 to 2000 m, no limitation
Degree of protection per EN 60529	IP67
Ambient conditions	
Temperature	
Operation	-40 to 60°C 3)
Storage	-40 to 85°C
Transport	-40 to 85°C
Mechanical properties	
Dimensions	
Width	53 mm
Height	155 mm
Depth	42 mm
Weight	350 g
Torque for connections	
M8	Max. 0.4 Nm
M12	Max. 0.6 Nm

Table 422: X67SC4122.L12 - Technical data

1) The related danger notices in the technical data sheet must also be observed.
2) The protective function is provided for max. 30 minutes for a continuous short circuit.
3) Up to hardware upgrade <1.10.1.1 and hardware revision <D0: 0 to 60°C

Danger!

Operation outside the technical data is not permitted and can result in dangerous states.

Information:

For additional information about installation, see chapter **"Installation notes for X67 modules" on page 24****.**

3.5.4.2.5 LED status indicators

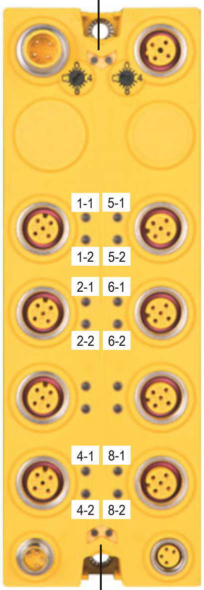
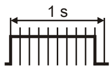
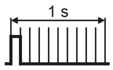
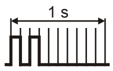
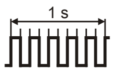
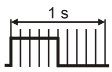
Figure	LED	Color	Status	Description
Status indicator re: Left: green (r), Right: red (e)  Status indicator SE Left: red (S); Right: red (E)	r	Green	Off	No power to module
			Single flash	Reset mode
			Double flash	Updating firmware
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Pulsating	Boot loader mode
			Triple flash	Updating safety-related firmware
			On	Error or I/O component not provided with voltage
	e + r	Red on / green single flash		Invalid firmware
	1-1 1-2 2-1 2-2 5-1 5-2 6-1 6-2	Input state of the corresponding digital input		
		Red	On	Warning/Error on an input channel
			Blinking	Error in dual-channel evaluation (synchronous blinking of 2 affected channels)
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Green	On	Input set
	4-1 4-2 8-1 8-2	Output status of the corresponding digital output		
		Red	On	Warning/Error on an output channel
			All on	Error on all channels, connection to the SafeLOGIC controller not OK or booting not yet completed
		Orange	On	Output set
	SE	Red	Off	Mode RUN or I/O component not provided with voltage
				Boot phase, missing X2X Link or defective processor
				Safety PREOPERATIONAL state Modules that are not used in the SafeDESIGNER application remain in the PREOPERATIONAL state.
				Safe communication channel not OK
				The firmware for this module is a non-certified pilot customer version.
				Boot phase, faulty firmware
			On	Safety state active for the entire module (= "FailSafe" state)
		The "SE" LEDs separately indicate the status of safety processor 1 ("S" LED) and safety processor 2 ("E" LED).		

Table 423: Status display

Danger!

Constantly lit "SE" LEDs indicate a defective module that must be replaced immediately.
It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

3.5.4.2.6 Connection elements

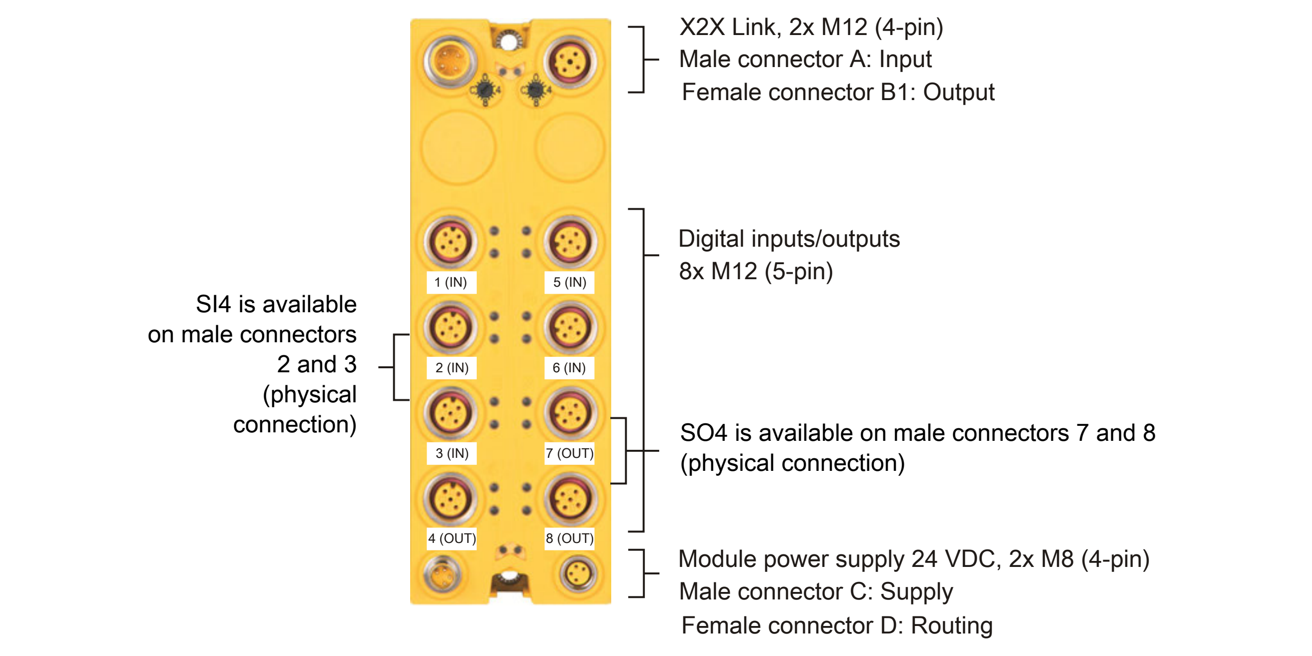


Figure 299: X67SC4122.L12 - Connection elements

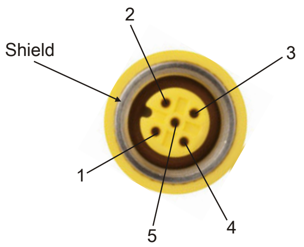
Pinout	Female connector	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5
	1 (IN)	Pulse 1	SI 1	GND	SI 2	Pulse 2
	2 (IN)	Pulse 3	SI 3	GND	SI 4	Pulse 4
	3 (IN)	NC	NC	GND	SI 4	Pulse 4
	5 (IN)	Pulse 5	SI 5	GND	SI 6	Pulse 6
	6 (IN)	Pulse 7	SI 7	GND	SI 8	Pulse 8
	4 (OUT)	GND	SO 1	GND	SO 2	GND
	7 (OUT)	GND	NC	GND	SO 4	GND
	8 (OUT)	GND	SO 3	GND	SO 4	GND

Table 424: Pinout

Information:

When using cables from B&R's line of accessories, cross faults between the two channels of a female connector cannot be ruled out in accordance with EN ISO 13849-2:2012. This is why shared error handling is implemented for both output channels of a female connector. This means that both output channels on this female connector are switched off as soon as an error has been detected on one of them. Comparable behavior applies to the acknowledgment of an error state. As soon as a channel error has been acknowledged, the error state on the other channel of the same female connector is also acknowledged.

Danger!

SI 4 is provided dually on female connectors 2 and 3 to make wiring easier. This makes it possible to use SI 4 for both one-channel sensors as well as two-channel sensors.

Two sensors cannot be connected to SI 4 in female connector 2 and SI 4 in female connector 3 because this would represent a parallel connection of two sensors on one input channel.

Information:

SO 4 is provided dually on female connectors 7 and 8 to make wiring easier. This makes it possible to use SO 4 for both one-channel actuators as well as two-channel actuators.

Connecting two actuators to SO 4 in female connector 7 and SO 4 in female connector 8 would cause a parallel connection of both actuators.

3.5.4.2.7 X2X Link

This module is connected to X2X Link using pre-assembled cables. The connection is made using a circular connector (2x M12, 4-pin).

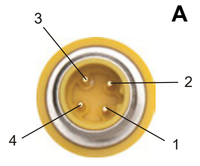
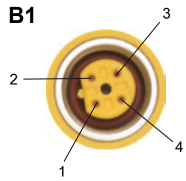
Connection	Pinout	
	Pin	Name
	1	X2X+
	2	X2X
	3	X2X⊥
	4	X2X\
	A ... B-coded male connector on the module, input B1 ... B-coded female connector on the module, output SHLD ... Shielding provided by threaded insert in the module	

Table 425: X2X Link

3.5.4.2.8 24 VDC module supply

The module supply is connected using pre-assembled cables with circular connectors (2x M8, 4-pin). The supply is connected via the male C connector. Female connector D is used for routing the supply to other modules.

The maximum permissible current per supply is 4 A (in summation 8 A)!

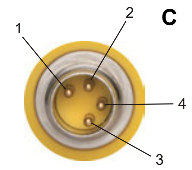
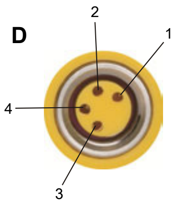
Connection	Pinout	
	Pin	Name
	1	24 VDC module supply 1)
	2	24 VDC module supply 1)
	3	GND
	4	GND
	C ... Male connector on the module, power supply D ... Female connector on the module, supply routing 1) Both supply pins must be supplied. It can only be ensured that the outputs are switched off if **both** pins are disconnected from the supply. If the summation current of the outputs is >4 A, current must also be supplied via female connector D, pin 2.	

Table 426: 24 VDC module supply

3.5.4.2.9 Node number switches

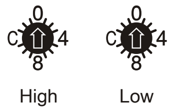


Figure 300: Node number switches for setting the X2X Link address

The decentralized X2X Link backplane, which connects individual X67 modules with one another, is set up to be self-addressing. Because of this, it is not necessary to set the node numbers. The module address is assigned according to its position in the X2X Link line.

In certain cases, e.g. when configurations of modular machines change, it is necessary to define specific module groups at a fixed address, regardless of the preceding modules in the line.

For this reason, the module is equipped with node number switches that can be used to set the X2X Link address. All subsequent modules refer to this offset and use it automatically for addressing purposes.

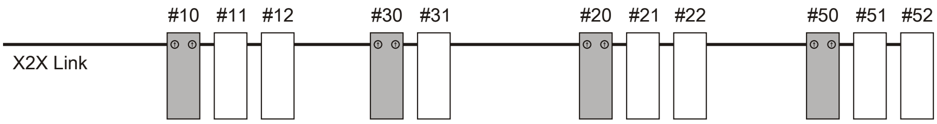


Figure 301: Sample configuration

If the node number on the module is set to 0x00, then the module address is assigned according to its position in the X2X Link line.

3.5.4.2.10 Connection examples

The typical connection examples in this section only represent a selection of the different wiring methods. The user must take error detection into account in each case.

Information:

For details about connection examples (such as circuit examples, compatibility class, max. number of supported channels, terminal assignments, etc.), see chapter Connection examples of the "Integrated safety technology" user's manual (MASAFETY-ENG).

3.5.4.2.10.1 Module behavior when GND connection is lost

In this section and all of its subsections, the term "connection element" is to be understood as follows for the respective system (X20, X67):

- X20: e.g. terminal block
- X67: e.g. M12, M8

A loss of GND on the module may cause current to flow from the module via the output or the GND connection of the connection element.

If power supplies, actuators or GND connections are grounded, the user must ensure that no grounding wires or any associated potential short circuits or open circuits will cause any additional impermissible GND connections.

The two currents I_{OUT} and I_{GND} are module-specific and must be taken from the technical data.

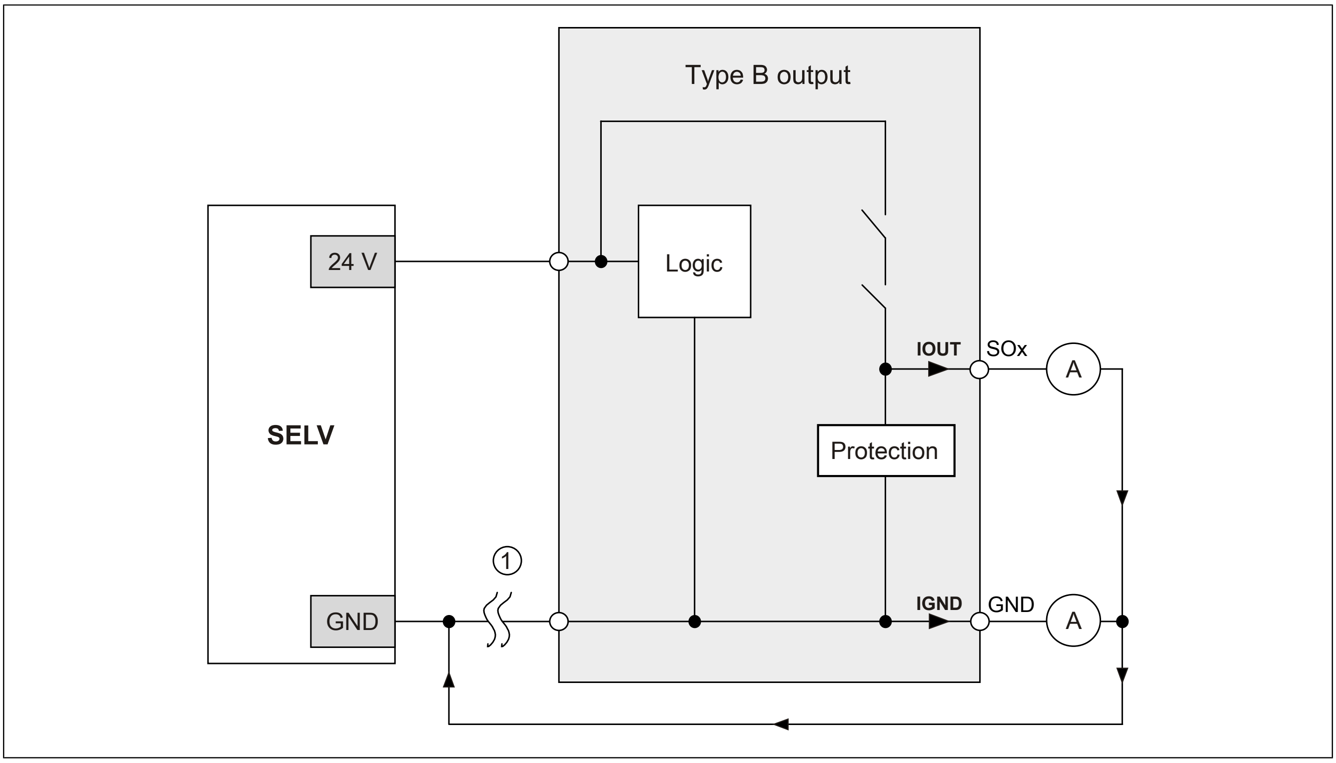


Figure 302: Module behavior when GND connection is lost

Danger!

The user is responsible for preventing any safety problems that could occur as a result of the I_{OUT} and I_{GND} currents specified in the technical data and the selected method of installation.

GND feedback to connection element, no external GND

If the module is used in the following wiring mode, then a loss of GND will not cause any problems because current is not able to flow via I_{OUT} or I_{GND}.

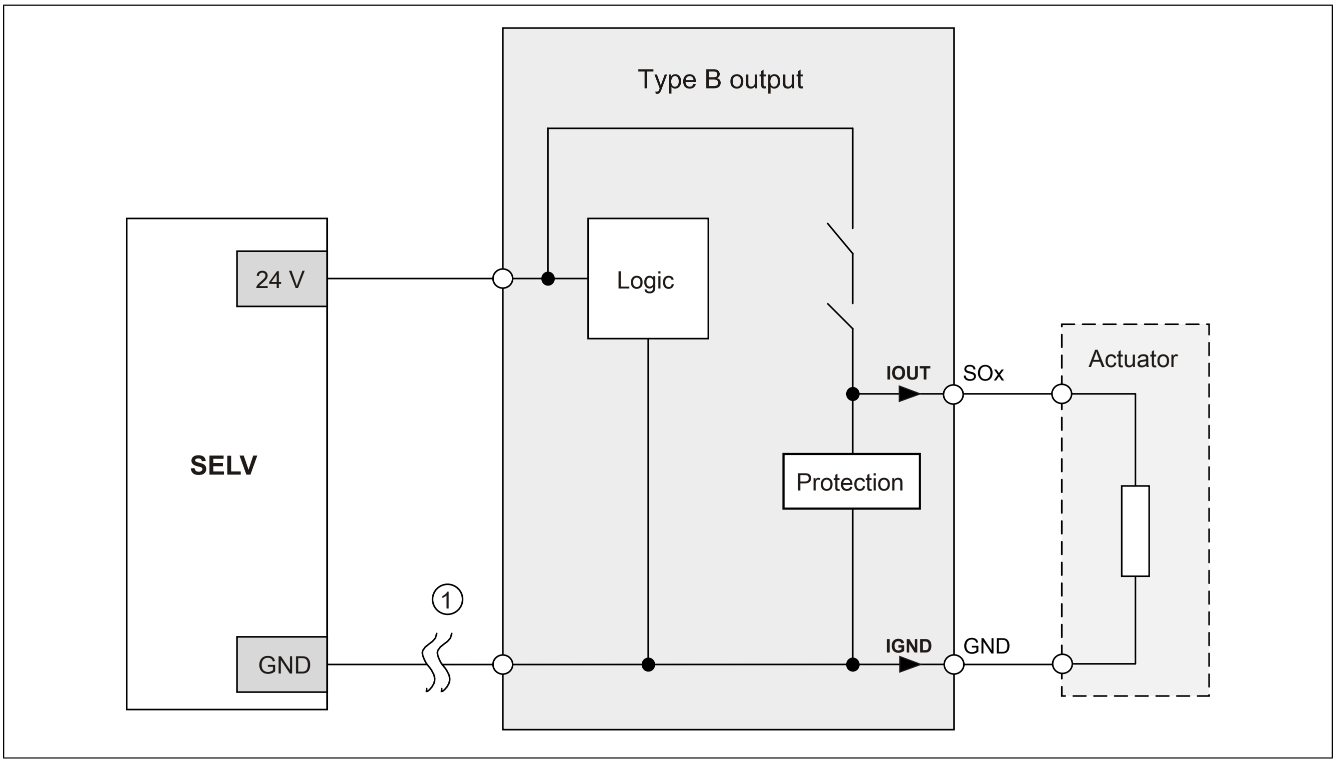


Figure 303: GND feedback to connection element

Danger!

Other wiring methods

If another wiring method is used, the user must ensure that a safety-critical state cannot occur if there are 2 external faults (open circuit, etc.). In addition, the current specifications for I_{OUT} and I_{GND} must be taken into consideration in the event that the GND connection is lost.

Using external GND without GND from connection element

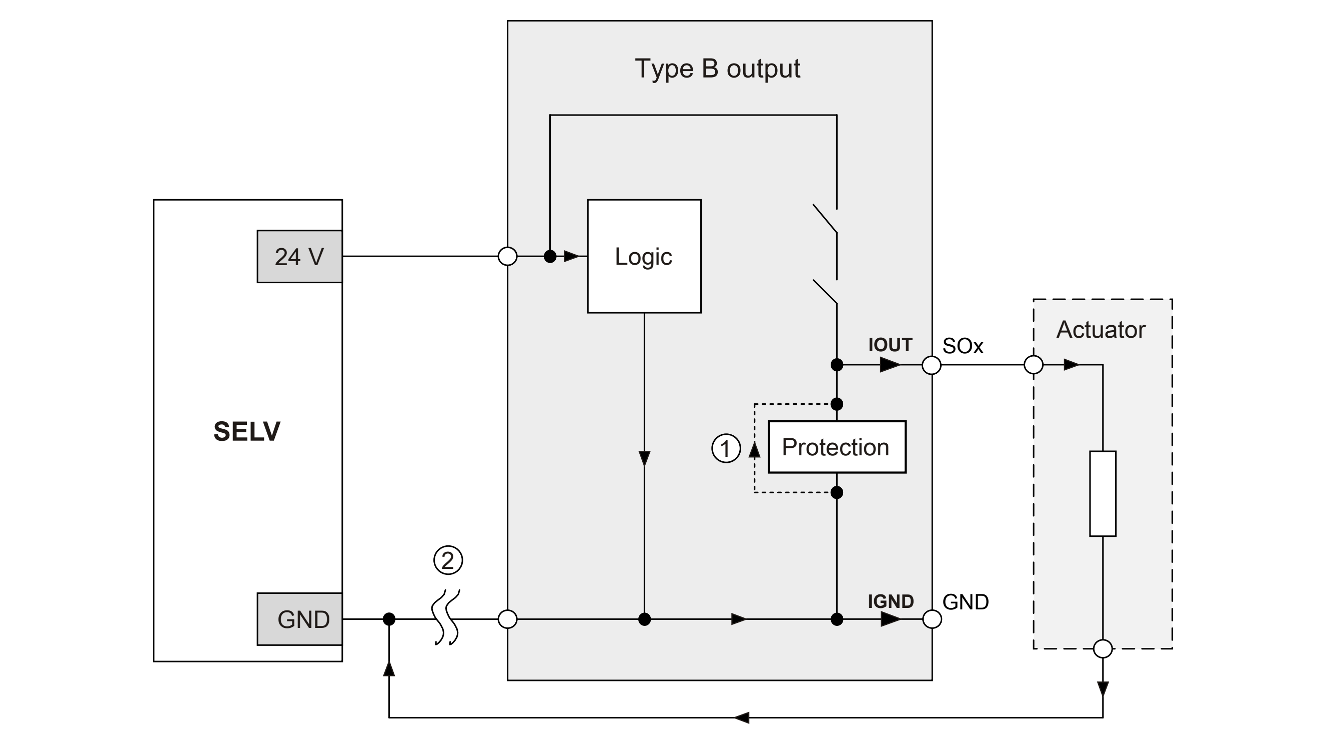


Figure 304: External GND only

Fault sequence:

- Fault ① (defective protective component):
  A component connected to GND on the output short circuits or behaves like an ohmic resistor. This fault is not always detected.
- Fault ② (open circuit on module GND):
  The module loses its direct connection to GND and current begins to flow through the defective protective component → I_{OUT} → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Using external GND and GND from connection element

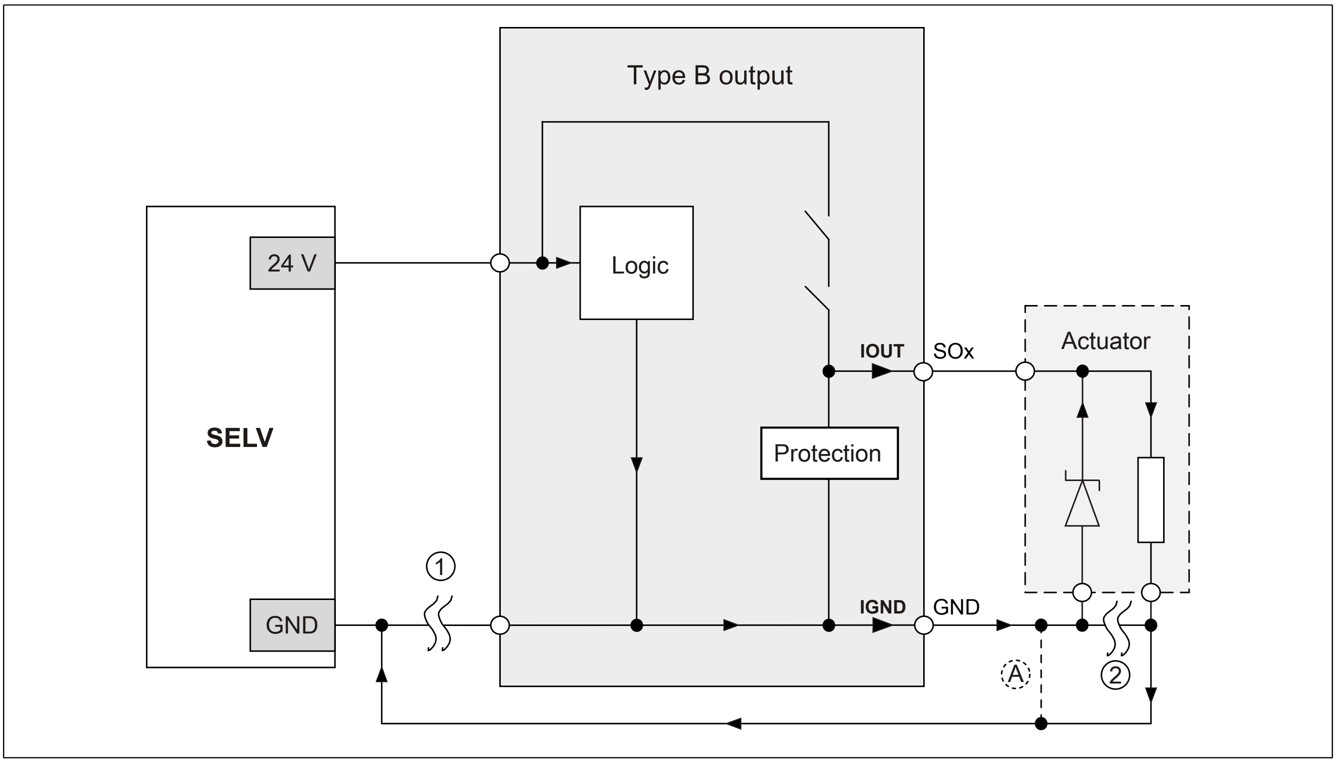


Figure 305: Possible connection error

Fault sequence:

- Fault ① (open circuit on module GND):
  No error is detected and the module continues to operate normally due to the additional external GND connection.
- Fault ② (open circuit on actuator's protective circuit):
  The module loses its direct connection to GND and current begins to flow through I_{GND} → damping diode → actuator.
  As a result, current above the maximum value permitted by the module is supplied to the actuator.

Danger!

This type of installation can cause hazardous situations and is therefore NOT permitted.

Possible remedies

This wiring method could be made possible, for example, by using two wires to complete the connection that experienced the open circuit fault in ② → see connection Ⓐ.

Information:

The diode in the actuator shown in the "Possible connection error" image is intended only to illustrate the error and is not mandatory.

3.5.4.2.10.2 Connecting single-channel sensors with contacts

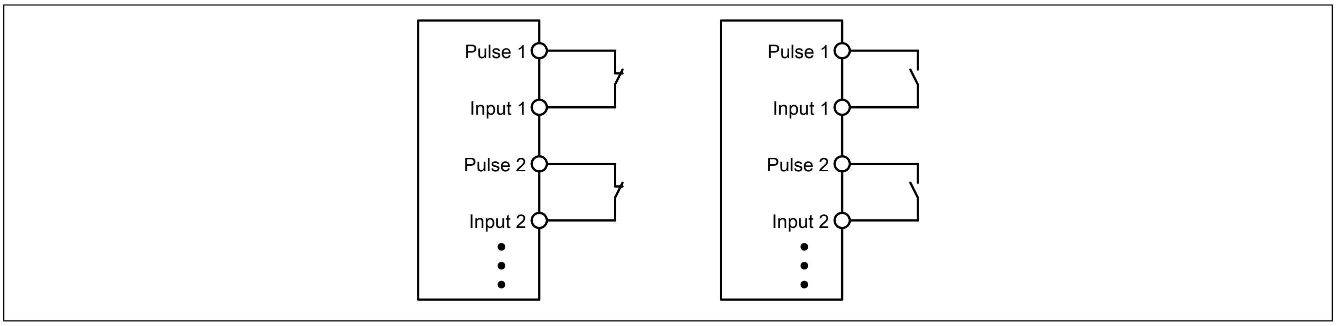


Figure 306: Connecting single-channel sensors with contacts

Single-channel sensors with contacts are the simplest connection.

With this connection, the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

3.5.4.2.10.3 Connecting two-channel sensors with contacts

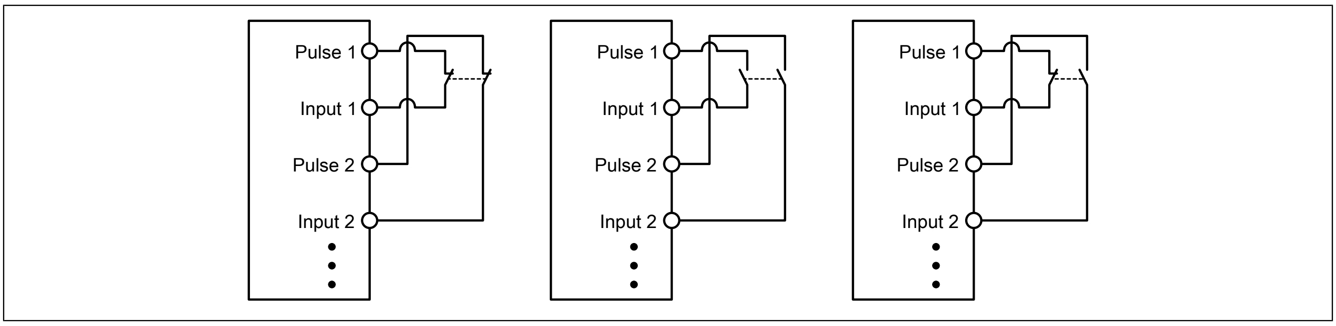


Figure 307: Connecting two-channel sensors with contacts

Sensors with contacts can be connected directly to a safe digital input module via two channels. Dual-channel evaluation is handled directly by the module.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the sensor according to the required category.

3.5.4.2.10.4 Connecting multi-channel sensors with contacts

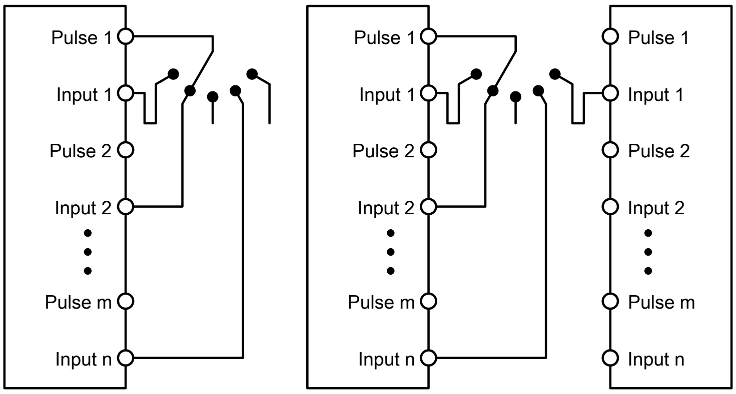


Figure 308: Connecting multi-channel sensors with contacts

Multi-channel switches (mode selector switches, switching devices with "shift key" capability) can be connected to multiple safe digital input modules.

If signals are evaluated internally in the module (see image to the left), the same pulse must be configured for all of the inputs being used. If signals are evaluated across all modules (see image to the right), all of the inputs must be configured to use an external pulse. In this type of application, pulse evaluation with the "default" pulse is not suitable; therefore, a separate pulse signal with approx. 4 ms low-phase is available.

In this case, multi-channel evaluation must be handled in the safety application (PLCopen function block "SF_ModeSelector"). The category achieved per EN ISO 13849-1:2015 in this way depends on the error models of the switching element (e.g. mode selector switch) and must be examined in combination with the error detection present in the PLCopen function block.

3.5.4.2.10.5 Connecting electronic sensors

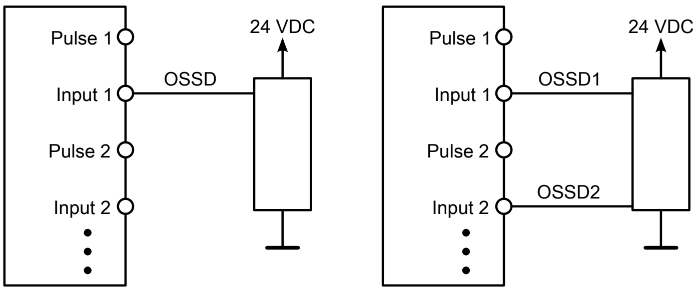


Figure 309: Connecting electronic sensors

Electronic sensors (light curtains, laser scanners, inductive sensors, etc.) can be connected directly to safe digital input modules. The switching thresholds of the input channels must be taken into account for these types of applications.

With single-channel wiring (see image on the left), the module satisfies Category 3 requirements in accordance with EN ISO 13849-1:2015. With two-channel wiring (see image on the right), the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not the wiring or connected electronic sensor. You are responsible for wiring the sensor in accordance with the required category and within the specifications set forth by the manufacturer of the electronic sensor.

3.5.4.2.10.6 Using the same pulse signals

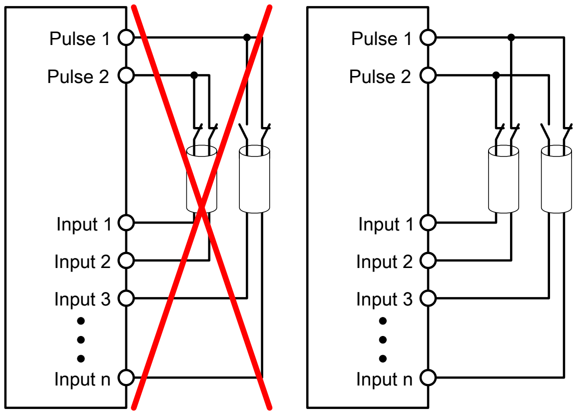


Figure 310: Using the same pulse signals

When using the same pulse signals for different inputs, they must be isolated from one another. Otherwise, damage to the cables may cause errors that are not detected by the module.

Danger!

If the same pulse signals are routed in the same cable, damage to the cable can cause cross faults between the signals to occur that are not detected by the module. This can result in dangerous situations.

For this reason, signal lines with the same pulse signal should be routed in different cables, or you should implement other error prevention measures in accordance with EN ISO 13849-2:2012.

Danger!

It is especially important to check the wiring when using the same pulse signal for two inputs that are located next to each other on the terminal. Pay special attention to ensure that poor wiring has not resulted in the two inputs being connected together.

3.5.4.2.10.7 Connecting safety-oriented actuators for Type B outputs

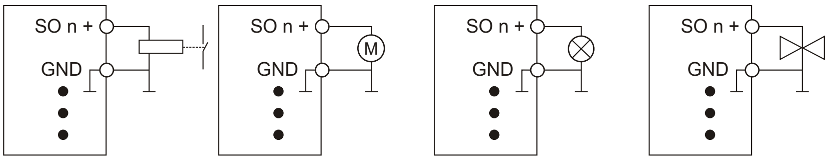


Figure 311: Connecting safety-oriented actuators for Type B outputs

Safety actuators (contactors, motors, muting lamps, valves) that are compatible with module performance data can be connected directly.

With this connection, the module satisfies Category 4 requirements in accordance with EN ISO 13849-1:2015. Be aware that this statement applies only to the module and not to the wiring shown. You are responsible for wiring the actuator in accordance with the required category and the characteristics of actuator.

If the actuators contain an inverse diode or electronic components, then the special instructions in section "Module behavior when GND connection is lost" must be followed.

3.5.4.2.11 Error detection

3.5.4.2.11.1 Internal module errors

The red "SE" LED makes it possible to evaluate the following error states:

- Module error, e.g. defective RAM, defective CPU, etc.
- Overtemperature/Undertemperature
- Overvoltage/Undervoltage
- Incompatible firmware version

Errors that occur within the module are detected according to the requirements of the standards listed in the certificate and within the minimum safety response time specified in the technical data. After this occurs, the module enters a safe state.

The internal module tests needed for this are only performed, however, if the module's firmware has been booted and the module is in either the PREOPERATIONAL state or the OPERATIONAL state. If this state is not achieved (for example, because the module has not been configured in the application), then the module will remain in the boot state.

BOOT mode on a module is clearly indicated by a slowly blinking SE LED (2 Hz or 1 Hz).

The error detection time specified in the technical data is relevant only for detecting external errors (i.e. wiring errors) in single-channel structures.

Danger!

Operating the safety module in BOOT mode is not permitted.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

3.5.4.2.11.2 Wiring errors

The wiring errors described in section "Error detection" are indicated by the red channel LED according to the application.

If a module detects an error, then:

- The channel LED is lit constantly red.
- Status signal (e.g. (Safe)ChannelOK, (Safe)InputOK, (Safe)OutputOK, etc.) is set to (SAFE)FALSE.
- Signal "SafeDigitalInputxx" or "SafeDigitalOutputxx" is set to SAFEFALSE.
- An entry is generated in the logbook.

Danger!

Recognizable errors (see the following chapters) are detected by the module within the error detection time. Errors not recognized by the module (or not recognized on time) that can lead to safety-critical states must be detected using additional measures.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

Type B output channels

Danger!

As illustrated in the following circuit examples, the connected actuators can be connected to GND on the load side. Connecting actuators on just one side without a GND supply is not permitted, however. This would cause a series connection of the actuators in the event of an open circuit, which could then cause a hazardous module error.

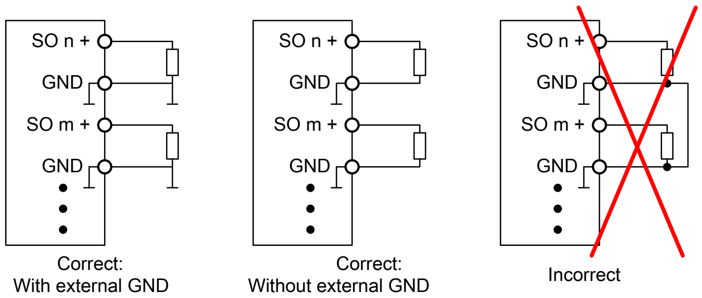


Figure 312: Invalid wiring

Connecting single-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels. The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this type of connection in combination with the configuration "Pulse Mode = Internal", the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	**Not detected**	**Not detected**
Open circuit	**Not detected**	**Not detected**

Table 427: SI error detection when "Pulse mode = Internal"

Connecting two-channel sensors with contacts

By default, every input channel is assigned a dedicated pulse output. This pulse output issues a specific signal that helps detect wiring problems, such as a short circuit to 24 VDC, GND or other signal channels.

The status of the connected switches is signaled via channel-specific LEDs, and the status of the dual-channel evaluation is signaled via the "OO" (for combinations with N.C./N.C. contacts) or "OC" LED (for combinations with N.C./N.O. contacts). On module types that do not have these LEDs, errors detected in the dual-channel evaluation are indicated by the respective channel LED blinking red.

With this type of connection in combination with the configuration "Pulse Mode = Internal" and combined with dual-channel evaluation in the module or in SafeDESIGNER, the modules can detect the following errors:

Error	Error on contact	
	Open	Closed
Ground fault on the pulse output	Detected	Detected
Pulse output shorted to 24 VDC	Detected	Detected
Cross fault between the pulse output and the other pulse signal	Detected	Detected
Ground fault on signal input	**Not detected**	Detected
Signal input shorted to 24 VDC	Detected	Detected
Cross fault between the signal input and the other pulse signal	Detected	Detected
Cross fault between the pulse output and the signal input	Detected[1)]	**Not detected**
Open circuit	**Not detected**	Detected[1)]

Table 428: SI error detection with "Pulse Mode = Internal" combined with dual-channel evaluation in the module or in SafeDESIGNER

1) Dual-channel evaluation of the module.

Connecting multi-channel sensors with contacts

The status of the connected switches is indicated by channel-specific LEDs. The LEDs "OO" and "OC" have no significance with this type of connection.

With this wiring, the following errors can be detected:

Error	
Ground fault on the pulse output	Detected
Pulse output shorted to 24 VDC	Detected
Cross fault between the pulse output and the other pulse signal	Detected[1)]
Ground fault on signal input (active signal)	Detected[1)]
Ground fault on signal input (inactive signal)	**Not detected**
Signal input shorted to 24 VDC	Detected
Cross fault between the signal input and the other pulse signal	Detected[1)]
Cross fault between the pulse output and the signal input (active signal)	**Not detected**
Open circuit (active signal)	Detected[1)]
Cross fault between the pulse output and the signal input (inactive signal)	Detected[1)]
Open circuit (inactive signal)	**Not detected**

Table 429: SI error detection when "Pulse Mode = External"

1) Detected by PLCopen function block "SF_ModeSelector" in the application.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

Information:

With the configuration "Pulse Mode = Internal", the pulses have a low phase of approximately 300 µs. This low phase is designed such that no additional degradation of the total response time can occur in the system. If line lengths exceed the max. line length (see technical data), problems may occur with this configuration. In these cases, configuration "Pulse Mode = External" can also be useful for normal sensors with contacts. The reduced error detection and extension of the total response time must be taken into account, however.

Connecting electronic sensors

A pulse pattern cannot be used with electronic sensors. The input channels must therefore be configured to "Pulse Mode = No Pulse".

Any gaps when testing the connected OSSD outputs must be masked out with the module's cutoff filter in order to avoid an unintended shutdown.

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Safety actuator connection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Ground fault on SOx+ (output type A) or SOx (output type B)				
All SO types	**Not detected**	Detected	**Not detected**	Detected
Ground fault on SOx- (output type A)				
X20SC0xxx	**Not detected**	Detected	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
SOx+ shorted to 24 VDC (output type A)				
X20SC0xxx	Detected	Detected	Detected	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
SOx shorted to 24 VDC (output type B)				
X20SC0xxx	Detected [1]	**Not detected**	Detected [1]	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SO6300				
X20SP1130				
X20SC2212		Detected [1]		
X67SC4122.L12				
SOx- shorted to 24 VDC (output type A)				
X20SC0xxx	Detected	Detected	Detected	Detected
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
GND shorted to 24 VDC				
X20SC0xxx	**Not detected**	**Not detected**	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SO6300				
X20SP1130				
X20SC2212				
X67SC4122.L12				
Cross fault between SOx+ (output type A) and the other signal (high)				
X20SC0xxx	Detected	Detected	Detected	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
Cross fault between SOx (output type B) and the other signal (high)				
X20SC0xxx	Detected [1]	**Not detected**	Detected [1]	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SO6300				
X20SP1130				
X20SC2212		Detected [1]		
X67SC4122.L12				
Cross fault between SOx- (output type A) and the other signal (high)				
X20SC0xxx	Detected	Detected	Detected	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0				
Cross fault between GND and the other signal (high)				
X20SC0xxx	**Not detected**	**Not detected**	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SO6300				
X20SP1130				
X20SC2212				
X67SC4122.L12				
Open circuit (output type A and B)				
X20SC0xxx	**Not detected**	**Not detected**	**Not detected**	**Not detected**
X20SLXxxx				
X20SRTxxx				
X20SOx1x0		**Not detected** [2]		**Not detected** [2]
X20SO6300		**Not detected**		**Not detected**
X20SP1130				
X20SC2212				
X67SC4122.L12				

Table 430: SO error detection

Error / module	Disable OSSD = No		Disable OSSD = Yes-ATTENTION	
	Error on output			
	Switched off	Switched on	Switched off	Switched on
Short circuit between SOx+ (output type A) and SOx- (output type A)				
X20SC0xxx X20SLXxxx X20SRTxxx X20SOx1x0	**Not detected**	Detected	**Not detected**	Detected

Table 430: SO error detection

1) If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel.
2) Open circuit can be detected via signal "CurrentOK". However, this signal cannot be used for safety purposes.

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Danger!

Possible error behavior of the actuators must be analyzed and avoided using corresponding responses (positively driven read-back contacts on a contactor, pressure switch on valves, etc.).

Danger!

This danger warning applies to all the modules listed in the "SO error detection" table with the exception of output channels of type A!

If SOx is shorted to high potentials, this will be detected by the module, but the connected actuator cannot be cut off due to the "only-plus-switching" design of the channel. Make sure that the wiring is correct in order to rule out SOx short circuits to high potentials (see EN ISO 13849-2:2012, Annex D.2.4, Table D.4).

3.5.4.2.12 Input circuit diagram

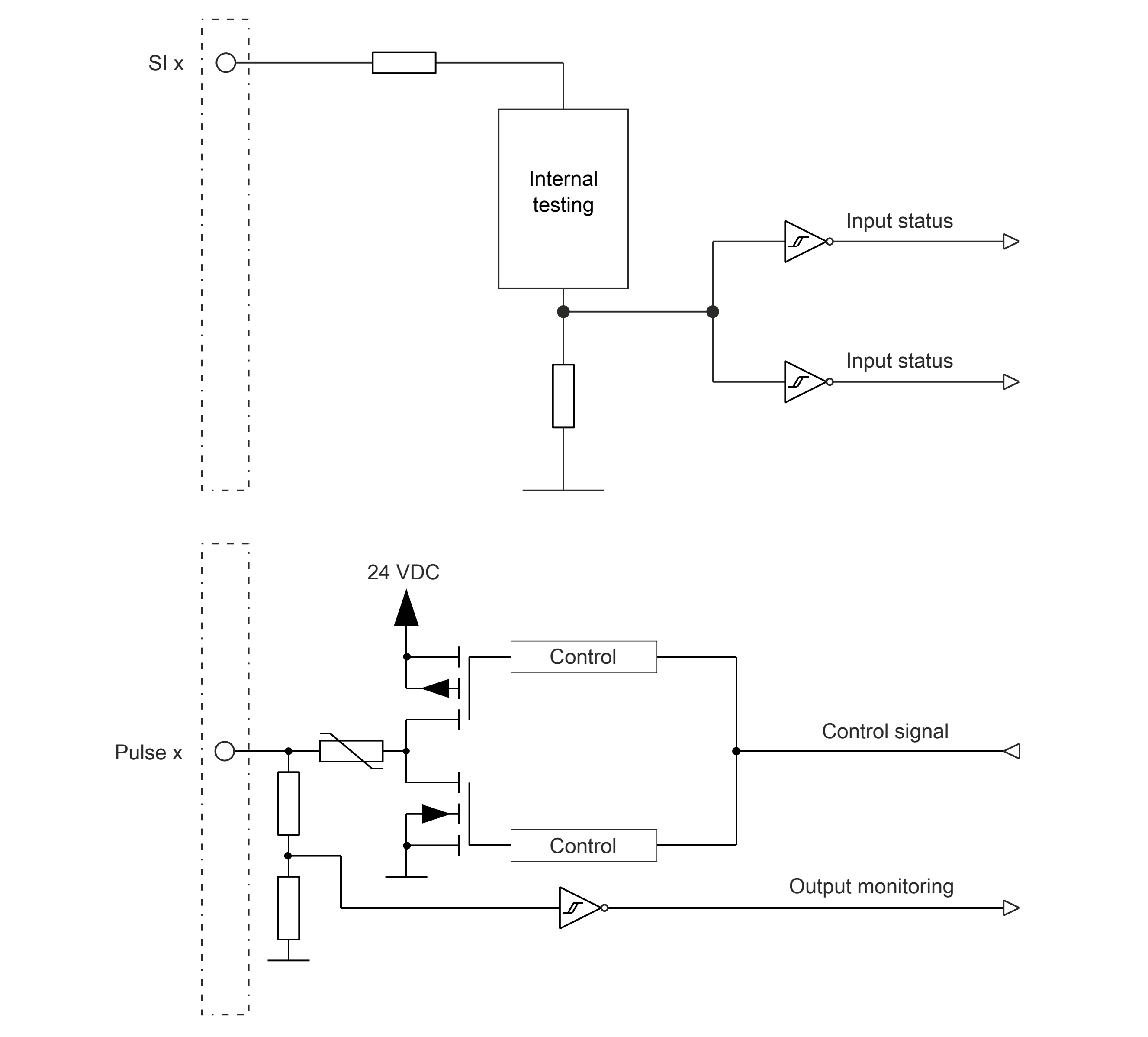


Figure 313: Input circuit diagram

3.5.4.2.13 Type B output circuit diagram

Type B digital output channels are designed for positive and positive switching inside the module.

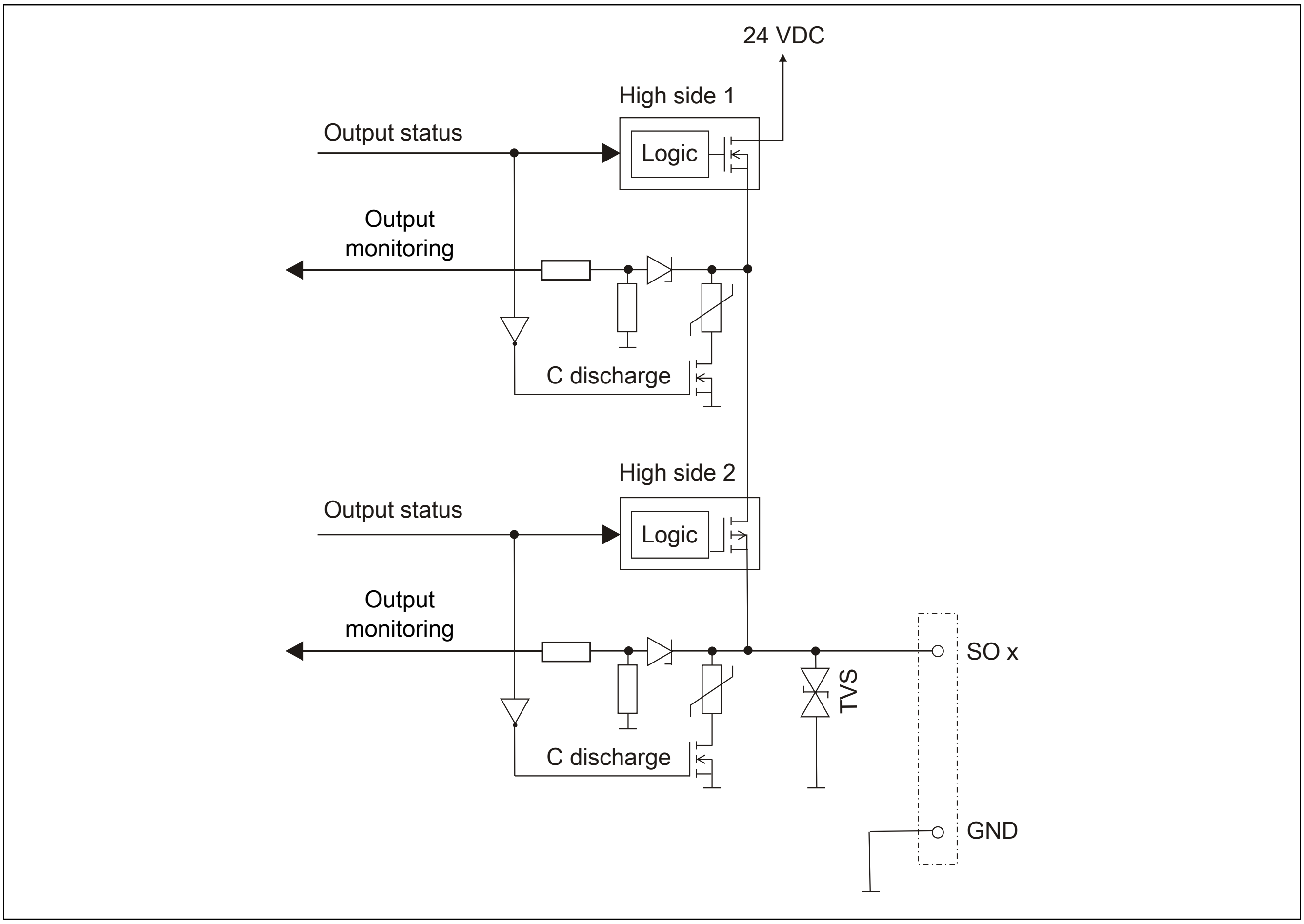


Figure 314: Type B output circuit diagram

3.5.4.2.14 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring.

Minimum cycle time
200 μs

3.5.4.2.15 I/O update time

The time needed by the module to generate a sample is specified by the I/O update time.

Minimum I/O update time
500 μs

Maximum I/O update time for input channels
2150 μs + Filter time (see chapter "Filter")

Maximum I/O update time for output channels
1800 μs

3.5.4.2.16 Filter

All safe digital input modules are equipped with separately configurable switch-on and switch-off filters. The functionality of the filters depends on the firmware version and is illustrated in the following table and figures:

Module type	Version	TOFF filter diagram	Filter time to be considered in addition to the total response time
I/O modules	<301	Diagram 1	2x TOFF filter time
SafeLOGIC-X	301, 311, 312	Diagram 1	2x TOFF filter time
I/O modules	≥301	Diagram 2	1x TOFF filter time
SafeLOGIC-X	302, ≥313	Diagram 2	1x TOFF filter time

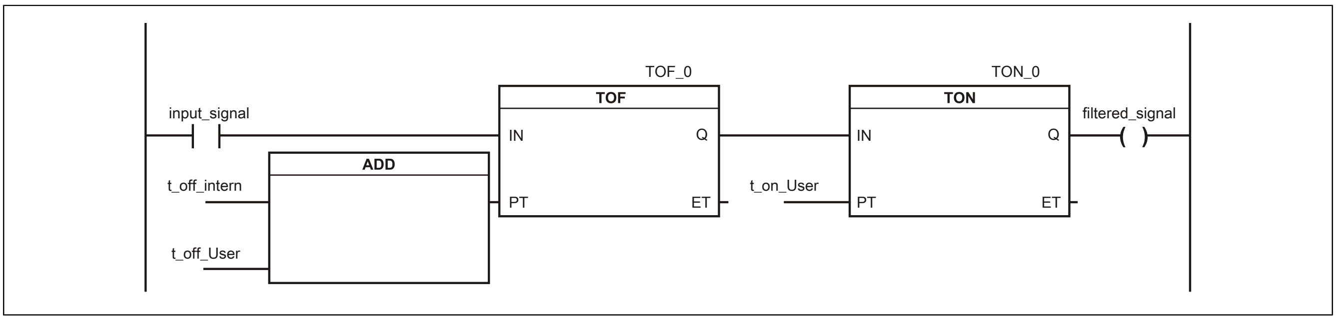


Figure 315: SI input filter - Diagram 1

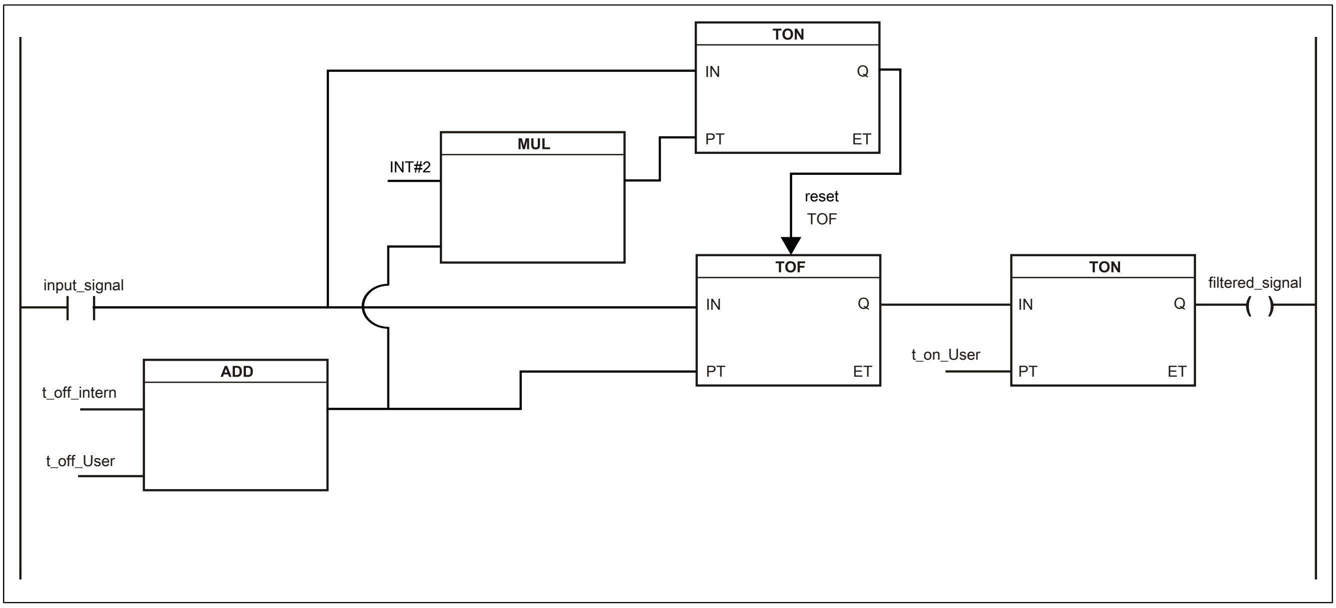


Figure 316: SI input filter - Diagram 2

Key:

- input_signal: Status of the input channel
- filtered_signal: Filtered status of the input channel. This is used as an input for the PLCopen function block and forwarded to the SafeLOGIC controller
- t_off_intern: Internal parameter (5 ms) for suppressing "external" test pulses (only with "Pulse Mode = External")
- t_off_User: Parameter for the switch-off filter
- t_on_User: Parameter for the switch-on filter

Unfiltered

The input state is collected with a fixed offset to the network cycle and transferred.

Switch-on filter

When switching from 0 to 1, the filtered status is collected with a fixed offset to the network cycle and transferred. The filter value can be configured (limit values are listed in the technical data).

Danger!

Errors that result from cross faults to other signals are detected by the module within the error detection time at the latest. By default, the switch-on filter is set to the error detection time value, which filters out faulty signals caused by possible cross faults. If the switch-on filter is set to a value smaller than the error detection time, faulty signals can cause temporary switch-on pulses to occur.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Switch-off filter

When switching from 1 to 0, the filtered status is collected with a fixed offset to the network cycle and transferred. The switch-off filter can be configured separately. This makes it possible to use the switch-off filter in actual applications (e.g. testing gaps of the light curtain) and to shorten response times. The filter value can be configured (limit values are listed in the technical data).

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

Information:

The actual effective filter depends on the I/O cycle time of the module. The actual effective filter can therefore deviate below the input value by the I/O cycle time (see the technical data for the module). If filter times are set less than the I/O cycle time of the module, no filter is effective.

Danger!

If "Pulse Mode = External" is used in the channel configuration, then an additional TOFF filter with 5 ms is enabled in the module. The corresponding information regarding the TOFF filter must also be considered when using the "Pulse Mode = External" setting.

3.5.4.2.17 Enabling principle

Each output channel has an additional standard switching signal that can be used to access the output channel from the standard application. As soon as the output channel has been enabled from a safety-related point of view (the setting of the channel is enabled from the point of view of the safety technology), the output channel can be set or cleared in the standard application independently of the additional safety-related runtime and jitter times.

Use of the enabling principle is specified in the I/O configuration in Automation Studio.

3.5.4.2.18 Restart behavior

Each digital input channel is not equipped with an internal restart interlock, which means that the associated channel data reverts back to the proper state automatically after an error situation on the module and/or network.
It is the responsibility of the user to connect the channel data of the safe input channels correctly and to provide them with a restart interlock. The restart interlocks of PLCopen function blocks can be used here, for example.
Using input channels without a correctly connected restart interlock can result in an automatic restart.

Each output channel is equipped with an internal restart interlock, which means that the following sequence must be followed in order to switch on a channel after an error situation on the module/network and/or after ending the safety function:

- Correct all module, channel or communication errors.
- Enable the safety-related signal for this channel (SafeOutput, etc.).
- Pause to ensure that the safety-related signal has been processed on the module (min. 1 network cycle).
- Positive edge on the release channel

For switching the release signal, the notes for manual reset function in EN ISO 13849-1:2015 must be observed.

The restart interlock functions independently of the enabling principle, which means that the behavior described above is not influenced by the parameter settings for the enabling principle or by the chronological position of the functional switching signal.

An automatic restart of the module can be configured by setting parameters. With this function, the output channel can be enabled using safety technology without an additional signal edge on the release channel. This function remains active as long as the release signal is TRUE and there is no error situation on the module/network.

Regardless of this parameter, a positive edge is required on the release channel for enabling the output channel in the following situations:

- After switching on
- After correcting an error on the safe communication channel
- After correcting a channel error
- After the release signal drops out

The automatic restart is configured in SafeDESIGNER using the channel parameters. If using an automatic restart, note the information in EN ISO 13849-1:2015.

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

3.5.4.2.19 Register description

3.5.4.2.19.1 Parameters in the I/O configuration

Group: Function model

Parameter	Description	Default value	Unit
Function model	This parameter is reserved for future functional expansions.	Default	-

Table 431: I/O configuration parameters: Function model

Group: General

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Module supervised</td><td>System behavior when a module is missing</td><td>On</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>A missing module triggers service mode.</td></tr><tr><td>Off</td><td>A missing module is ignored.</td></tr></table></td></tr>
<tr><td>Module information
(up to AS 3.0.90)</td><td>This parameter enables/disables the module-specific information in the I/O mapping:
• SerialNumber
• ModuleID
• HardwareVariant
• FirmwareVersion</td><td>Off</td><td>-</td></tr>
<tr><td rowspan="2">Blackout mode
(hardware upgrade 1.10.0.6 or later)</td><td>This parameter enables blackout mode (see section Blackout mode in Automation Help under: Hardware → X20 system → Additional information → Blackout mode).</td><td>Off</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>On</td><td>Blackout mode is enabled.</td></tr><tr><td>Off</td><td>Blackout mode is disabled.</td></tr></table></td></tr>
<tr><td>Channel status information</td><td>This parameter enables/disables the channel-specific status information in the I/O mapping.</td><td>On</td><td>-</td></tr>
<tr><td>State number of 2-channel evaluation</td><td>This parameter enables/disables the status information of dual-channel evaluation.</td><td>Off</td><td>-</td></tr>
<tr><td>Restart inhibit state numbers</td><td>This parameter enables/disables restart interlock status information.</td><td>Off</td><td>-</td></tr>
<tr><td>SafeLOGIC ID</td><td>In applications with multiple SafeLOGIC controllers, this parameter defines the module's association with a particular SafeLOGIC controller.
• Permissible values: 1 to 1024</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>SafeMODULE ID</td><td>Unique safety address of the module
• Permissible values: 2 to 1023</td><td>Assigned automatically</td><td>-</td></tr>
<tr><td>Max switching frequency channel x
(up to firmware version < 300)</td><td>Maximum switching frequency of the output channel.
• Permissible values: 1 Hz, 10 Hz, 100 Hz, 1000 Hz

This value specifies the max. switching frequency of the actuator connected to the output. It is especially important to adjust this parameter to the actual conditions for inductive or capacitive loads because the internal delay for checking the voltage to see if it is 0 V after a cutoff signal occurs is calculated using this parameter. Therefore, if this value is too high (e.g. 1000 Hz) and the voltage does not go to 0 within the corresponding time (in this example 500 µs) after a cutoff signal because of the connected actuator, then a channel error occurs.

If the output is controlled by the application using a higher switching frequency than configured, a channel-specific error may erroneously be detected on the module, which causes the channel to be cut off.</td><td>1</td><td>Hz</td></tr>
</table>

Table 432: I/O configuration parameters: General

Group: Output signal path

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">DigitalOutputxx</td><td>This parameter specifies the mode that can be used by the standard application to access the output channel.</td><td>Direct</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Direct</td><td>The output channel can be accessed directly by the standard application. Signals "DigitalOutputxx" are available in the I/O mapping accordingly.</td></tr><tr><td>Via SafeLOGIC</td><td>The output channel cannot be accessed directly by the standard application. Signals "DigitalOutputxx" are not available in the I/O mapping accordingly. It is only possible for the standard application to influence the output channel via the communication channels from the CPU to the SafeLOGIC controller.</td></tr></table></td></tr>
</table>

Table 433: I/O configuration parameters: Output signal path

3.5.4.2.19.2 Parameters in SafeDESIGNER - up to Release 1.9

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min_required_FW_Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is mandatory for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>The module is not required for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>Not_Present
(Release 1.9 and later)</td><td>The module is not required for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = Not_Present" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = Not_Present", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External_UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">Disable_OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr>
<tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr>
</table>
</td></tr>
</table>

Table 434: SafeDESIGNER parameters: Basic

Danger!

If function "External_UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Danger!

With "Disable_OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2010 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2010 or PL d per EN ISO 13849-1:2015, a daily check of the safety function by the user is necessary.

Group: Safety_Response_Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual_Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety_Response_Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety_Response_Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td rowspan="2">Synchronous_Network_Only</td><td>This parameter describes the synchronization characteristics of the network being used. They are defined in Automation Studio / Automation Runtime.</td><td>Yes</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>In order to calculate the safety response time, networks must be synchronous and their cycle times must either be the same or an integer ratio of the cycle times.</td></tr><tr><td>No</td><td>No requirement for synchronization of the networks</td></tr></table></td></tr>
<tr><td>Max_X2X_CycleTime_us</td><td>This parameter specifies the maximum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_Powerlink_CycleTime_us</td><td>This parameter specifies the maximum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Max_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the maximum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that a copy task is not included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>5000</td><td>µs</td></tr>
<tr><td>Min_X2X_CycleTime_us</td><td>This parameter specifies the minimum X2X cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_Powerlink_CycleTime_us</td><td>This parameter specifies the minimum POWERLINK cycle time used to calculate the safety response time.
Permissible values: 200 to 25,000 µs (corresponds to 0.2 to 25 ms)</td><td>200</td><td>µs</td></tr>
<tr><td>Min_CPU_CrossLinkTask_
CycleTime_us</td><td>This parameter specifies the minimum cycle time for the copy task on the CPU used to calculate the safety response time. The value 0 indicates that configurations without a copy task are also included for the response time.
Permissible values: 0 to 25,000 µs (corresponds to 0 to 25 ms)</td><td>0</td><td>µs</td></tr>
<tr><td>Worst_Case_Response_Time_us</td><td>This parameter specifies the limit value for monitoring the safety response time.
Permissible values: 3000 to 5,000,000 µs (corresponds to 3 ms to 5 s)</td><td>50000</td><td>µs</td></tr>
<tr><td>Node_Guarding_Lifetime</td><td>This parameter specifies the maximum number of attempts to be made during the time set with parameter "Node_Guarding_Timeout_s". The purpose of these attempts is to ensure that the module is available.
Permissible values: 1 to 255
Note
The larger the configured value, the greater the amount of asynchronous data traffic.
This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently using parameter "Worst_Case_Response_Time_us".</td><td>5</td><td>-</td></tr>
</table>

Table 435: SafeDESIGNER parameters: Safety_Response_Time

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse_Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="8">Possible "Pulse_Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>5</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>6</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Channel 5</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>7</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>8</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Channel 7</td><td>Default</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse_Source", then parameter "Pulse_Mode" must be set to "Internal" on the respective channel of the selected "Pulse_Source".</td></tr>
<tr><td rowspan="2">Pulse_Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is set for "Pulse_Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter_Off_us</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter_On_us</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy_Time_us</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for the "Dual-channel evaluation" function during which the state of both physical individual channels is permitted to be undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s) (up to Release 1.4: 0 to 500,000 µs - corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
</table>

Table 436: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

Parameter	Description	Default value	Unit
Auto_Restart	This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").	No	-
	Parameter value: Yes-ATTENTION – "Automatic restart" function is activated. Parameter value: No – "Automatic restart" function is not activated.		

Parameter value	Description
Yes-ATTENTION	"Automatic restart" function is activated.
No	"Automatic restart" function is not activated.

Table 437: SafeDESIGNER parameters: SafeDigitalOutputxx, SafeDigitalOutputxxyy

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

3.5.4.2.19.3 Parameters in SafeDESIGNER - Release 1.10 and later

Group: Basic

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td>Min required FW Rev</td><td>This parameter is reserved for future functional expansions.</td><td>Basic Release</td><td>-</td></tr>
<tr><td rowspan="2">Optional</td><td>This parameter can be used to configure the module as "optional". Optional modules do not have to be present, i.e. the SafeLOGIC controller will not indicate that these modules are not present. However, this parameter does not influence the module's signal or status data.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>No</td><td>This module is absolutely necessary for the application.

The module must be in OPERATIONAL mode after startup, and safe communication to the SafeLOGIC controller must be established without errors (SafeModuleOK = SAFETRUE). Processing of the safety application on the SafeLOGIC controller is delayed after startup until this state is achieved for all modules with "Optional = No".

After startup, module problems are indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is also made in the logbook.</td></tr>
<tr><td>Yes</td><td>This module is not necessary for the application.

The module is not taken into account during startup, which means the safety application is started regardless of whether the modules with "Optional = Yes" are in OPERATIONAL mode or if safe communication is properly established between these modules and the SafeLOGIC controller.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
<tr><td>Startup</td><td>This module is optional. The system determines how the module will proceed during startup.

If it is determined that the module is physically present during startup (regardless of whether it is in OPERATIONAL mode or not), then the module behaves as if "Optional = No" is set.

If it is determined that the module is not physically present during startup, then the module behaves as if "Optional = Yes" is set.</td></tr>
<tr><td>NotPresent</td><td>This module is not necessary for the application.

The module is ignored during startup, which means the safety application is started regardless of whether the modules with "Optional = NotPresent" are physically present.

Unlike when "Optional = Yes" is configured, the module is not started with "Optional = NotPresent", which optimizes system startup behavior.

After startup, module problems are NOT indicated by a quickly blinking "MXCHG" LED on the SafeLOGIC controller. An entry is NOT made in the logbook.</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">External UDID</td><td>This parameter enables the option on the module for the expected UDID to be specified externally by the CPU.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Yes-ATTENTION</td><td>The UDID is determined by the CPU. The SafeLOGIC controller must be restarted if the UDID is changed.</td></tr>
<tr><td>No</td><td>The UDID is specified by a teach-in procedure during startup.</td></tr>
</table>
</td></tr>
</table>

Table 438: SafeDESIGNER parameters: Basic

Danger!

If function "External UDID = Yes-ATTENTION" is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA (Failure Mode and Effects Analysis) in order to detect these situations and implement additional safety measures to handle them.

Group: Safety Response Time

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Manual Configuration</td><td>This parameter makes it possible to manually and individually configure the safety response time for the module.

The parameters for the safety response time are generally set in the same way for all stations involved in the application. For this reason, these parameters are configured for the SafeLOGIC controller in SafeDESIGNER. For application situations in which individual safety functions require optimal response time behavior, the parameters for the safety response time can be configured individually on the respective module.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes</td><td>Data from the module's "Safety Response Time" group is used to calculate the safety response time for the module's signals.</td></tr><tr><td>No</td><td>The parameters for the safety response time are taken from the "Safety Response Time" group on the SafeLOGIC controller.</td></tr></table></td></tr>
<tr><td>Safe Data Duration</td><td>This parameter specifies the maximum permissible data transmission time between the SafeLOGIC controller and SafeIO module.
For more information about the actual data transmission time, see section Diagnostics and service → Diagnostics tools → Network analyzer → Editor → Calculation of safety runtime of Automation Help. The cycle time of the safety application must also be added.
• Permissible values: 2000 to 10,000,000 µs (corresponds to 2 ms to 10 s)</td><td>20000</td><td>µs</td></tr>
<tr><td>Additional Tolerated Packet Loss</td><td>This parameter specifies the number of additional tolerated lost packets during data transfer.
• Permissible values: 0 to 10</td><td>0</td><td>Packets</td></tr>
<tr><td>Packets per Node Guarding</td><td>This parameter specifies the maximum number of packets used for node guarding.
• Permissible values: 1 to 255
Note
• The larger the configured value, the greater the amount of asynchronous data traffic.
• This setting is not critical to safety functionality. The time for safely cutting off actuators is determined independently of this.</td><td>5</td><td>Packets</td></tr>
</table>

Table 439: SafeDESIGNER parameters: Safety Response Time

Group: Module Configuration

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Disable OSSD</td><td>This parameter can be used to switch off automatic testing of the output driver for all of the module's channels.</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>Automatic testing of the output driver is switched off.</td></tr><tr><td>No</td><td>Automatic testing of the output driver is enabled.</td></tr></table></td></tr>
</table>

Table 440: SafeDESIGNER parameters: Module Configuration

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements for SIL 3 per EN 62061:2013 or PL e per EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2 per EN 62061:2013 or PL d per EN ISO 13849-1:2015, the user must check the safety function on a daily basis when using type B output channels.
For type B2 output channels, it is also important to ensure that all of the module's output channels are simultaneously in a switched-off state for at least 1 s during this test.

On X20SRTxxx modules, each output channel being used must be checked before the first safety request and every 24 hours. For this check, the corresponding channel must be switched on and off at least once.

Group: SafeDigitalInputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="3">Pulse Source</td><td>This parameter can be used to specify the pulse source for the input channel.</td><td>Default</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th></th><th colspan="8">Possible "Pulse Source"</th></tr>
<tr><th>Channel</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>2</td><td>Channel 1</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>3</td><td>Channel 1</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>4</td><td>Channel 1</td><td>-</td><td>Channel 3</td><td>Default</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>5</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>6</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>Channel 5</td><td>Default</td><td>-</td><td>-</td></tr>
<tr><td>7</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Default</td><td>-</td></tr>
<tr><td>8</td><td>Channel 1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>Channel 7</td><td>Default</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Note:
If a value other than "Default" is set for "Pulse Source", then the "Pulse Mode" parameter must be set to "Internal" on the respective channel of the selected "Pulse Source".</td></tr>
<tr><td rowspan="2">Pulse Mode</td><td>This parameter can be used to specify the pulse mode for the input channel.</td><td>Internal</td><td>-</td></tr>
<tr><td colspan="3">
<table>
<tr><th>Parameter value</th><th>Description</th></tr>
<tr><td>Internal</td><td>The channel works exclusively with the pulse output that is configured for "Pulse Source".</td></tr>
<tr><td>External</td><td>The channel works with any pulse output on a B&R input module as long as the pulse output is configured as "external".</td></tr>
<tr><td>No Pulse</td><td>The pulse check on the channel is disabled. Potential low phases of the signal must be removed using the switch-off filter in order to prevent unintended cutoff.</td></tr>
</table>
</td></tr>
<tr><td>Filter Off</td><td>Switch-off filter for the channel to remove potentially disruptive signal low phases.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>0</td><td>µs</td></tr>
<tr><td>Filter On</td><td>Switch-on filter for the channel that can be used to "debounce" the signals. This function also makes it possible for the module to lengthen a switch-off signal that would otherwise be too short.
• Permissible values: 0 to 500,000 µs (corresponds to 0 to 0.5 s)</td><td>200000</td><td>µs</td></tr>
<tr><td>Discrepancy Time</td><td>Parameter only available for odd-numbered channels.
This parameter specifies the maximum time for "dual-channel evaluation", during which the status of both physical individual channels can remain undefined without triggering an error.
• Permissible values: 0 to 10,000,000 µs (corresponds to 0 to 10 s)</td><td>50000</td><td>µs</td></tr>
</table>

Table 441: SafeDESIGNER parameters: SafeDigitalInputxx

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time.

Danger!

Signals with a low phase shorter than the safety response time can potentially be lost. Such signals should be lengthened accordingly using the "switch-on filter" function on the input module.

Danger!

Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0. Lengthening the low phase with a switch-on filter is not possible in these cases.

Group: SafeDigitalOutputxx

<table>
<tr><th>Parameter</th><th>Description</th><th>Default value</th><th>Unit</th></tr>
<tr><td rowspan="2">Auto Restart</td><td>This parameter can be used to configure an automatic restart on the module (see section "Restart behavior").</td><td>No</td><td>-</td></tr>
<tr><td colspan="3"><table><tr><th>Parameter value</th><th>Description</th></tr><tr><td>Yes-ATTENTION</td><td>"Automatic restart" function is activated.</td></tr><tr><td>No</td><td>"Automatic restart" function is not activated.</td></tr></table></td></tr>
</table>

Table 442: SafeDESIGNER parameters: SafeDigitalOutputxx

Danger!

Configuring an automatic restart can result in critical safety conditions. Take additional measures to ensure proper safety-related functionality.

3.5.4.2.19.4 Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>ModuleOk</td><td>Read</td><td>-</td><td>BOOL</td><td>Indicates if the module is OK</td></tr>
<tr><td>SerialNumber</td><td>Read</td><td>-</td><td>UDINT</td><td>Module serial number</td></tr>
<tr><td>ModuleID</td><td>Read</td><td>-</td><td>UINT</td><td>Module ID</td></tr>
<tr><td>HardwareVariant</td><td>Read</td><td>-</td><td>UINT</td><td>Hardware variant</td></tr>
<tr><td>FirmwareVersion</td><td>Read</td><td>-</td><td>UINT</td><td>Firmware version of the module</td></tr>
<tr><td>UDID_low</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>UDID, lower 4 bytes</td></tr>
<tr><td>UDID_high</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>UDID, upper 2 bytes</td></tr>
<tr><td>SafetyFWversion1</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 1</td></tr>
<tr><td>SafetyFWversion2</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Firmware version - Safety processor 2</td></tr>
<tr><td>SafetyFWcrc1
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 1</td></tr>
<tr><td>SafetyFWcrc2
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>CRC of firmware header on safety processor 2</td></tr>
<tr><td>Bootstate
(hardware upgrade 1.10.1.0 or later)</td><td>(Read) [1]</td><td>-</td><td>UINT</td><td>Startup state of the module.
Notes:

Some of the boot states do not occur during normal startup or are cycled through so quickly that they are not visible externally.
The boot states usually cycle through in ascending order. There are cases, however, in which a previous value is captured.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0003</td><td>Startup communication processor OK, no communication to the safety processors (check 24 V supply voltage!)</td></tr>
<tr><td>0x0010</td><td>FAILSAFE. At least one of the safety processors is in the safe state.</td></tr>
<tr><td>0x0020</td><td>Internal communication to safety processors started</td></tr>
<tr><td>0x0024</td><td>Firmware update of safety processors</td></tr>
<tr><td>0x0040</td><td>Firmware of safety processors started</td></tr>
<tr><td>0x0440</td><td>Firmware of safety processors running</td></tr>
<tr><td>0x0840</td><td>Waiting for openSAFETY "Operational" (loading SafeDESIGNER application or no valid application exists, waiting on acknowledgments such as module exchange)</td></tr>
<tr><td>0x1040</td><td>Evaluating the configuration according to the SafeDESIGNER application</td></tr>
<tr><td>0x3440</td><td>Stabilizing cyclic openSAFETY data exchange.
Note:
If the boot state remains here, check SafeDESIGNER parameters "(Default) Safe Data Duration", "(Default) Additional Tolerated Packet Loss".</td></tr>
<tr><td>0x4040</td><td>RUN. Final state, startup completed.</td></tr>
</table></td></tr>
<tr><td>Diag1_Temp</td><td>(Read) [1]</td><td>-</td><td>INT</td><td>Module temperature in °C</td></tr>
<tr><td>PLCopenFBKxy_state</td><td>Read</td><td>-</td><td>USINT</td><td>State number of dual-channel evaluation (PLCopen function block "Equivalent" or "Antivalent")</td></tr>
<tr><td>InputErrorStates</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th>Type of error</th></tr>
<tr><th>Inputs</th></tr>
<tr><th>Input stuck at high</th></tr>
<tr><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>PulseoutputErrors</td><td>(Read) [1]</td><td>-</td><td>UDINT</td><td>Channel status, additional information for channel error
<table>
<tr><th colspan="2">Type of error</th></tr>
<tr><th colspan="2">Pulse outputs</th></tr>
<tr><th>Feedback stuck at high (shorted to 24 VDC)</th><th>Feedback stuck at low (ground fault)</th></tr>
<tr><td>Bit no. 8 to 15 = Channel 1 to 8</td><td>Bit no. 0 to 7 = Channel 1 to 8</td></tr>
</table>
If a bit is set, the corresponding error has been detected on the respective channel.</td></tr>
<tr><td>SafeModuleOK</td><td>-</td><td>Read</td><td>SAFEBOOL</td><td>Indicates if the safe communication channel is OK</td></tr>
</table>

Table 443: Channel list

<table>
<tr><th>Channel name</th><th>Access via Automation Studio</th><th>Access via SafeDESIGNER</th><th>Data type</th><th>Description</th></tr>
<tr><td>SafeDigitalInputxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Physical channel SI xx</td></tr>
<tr><td>SafeEquivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentInputxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>SafeInputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of physical channel SI xx</td></tr>
<tr><td>SafeEquivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of equivalent channel SI xx/yy</td></tr>
<tr><td>SafeAntivalentOKxxyy</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of dual-channel evaluation of antivalent channel SI xx/yy</td></tr>
<tr><td>DigitalOutputxx</td><td>Write</td><td>-</td><td>BOOL</td><td>Enable signal - Channel SO xx</td></tr>
<tr><td>SafeDigitalOutputxx</td><td>-</td><td>Write</td><td>SAFEBOOL</td><td>Safe channel SO xx</td></tr>
<tr><td>SafeOutputOKxx</td><td>Read</td><td>Read</td><td>SAFEBOOL</td><td>Status of channel SO xx</td></tr>
<tr><td>ReleaseOutputxx</td><td>-</td><td>Write</td><td>BOOL</td><td>Release signal for the restart interlock of channel SO xx</td></tr>
<tr><td>PhysicalStateOutputxx</td><td>Read</td><td>Read</td><td>BOOL</td><td>Read-back value of physical channel SO xx</td></tr>
<tr><td>FBK_Status_1</td><td>Read</td><td>-</td><td>UINT</td><td>State number of the restart interlock of channel x. See "Restart interlock state diagram".
<table><tr><th>Bit 15 to 12</th><th>Bit 11 to 8</th><th>Bit 7 to 4</th><th>Bit 3 to 0</th></tr><tr><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table></td></tr>
</table>

Table 443: Channel list

1) This data is accessed in Automation Studio using the ASIOACC library.

PLCopen state diagrams "Antivalent" / "Equivalent"

The following state diagrams illustrate the effect of the "Antivalent" and "Equivalent" PLCopen function blocks integrated in the module.
The hexadecimal value in parentheses corresponds to the state number provided via the channels "PLCopenFBKxy_state" and "PLCopenFBKxxyy_state".

The following PLCopen state diagrams show the function for the "SafeAntivalentInput0102" and "SafeEquivalentInput0102" channels. The same diagrams are valid for the "SafeAntivalentInputxxyy" and "SafeEquivalentInputxxyy" channels, but "SafeDigitalInput01" and "SafeDigitalInput02" are to be replaced by the respective input.

In addition to the PLCopen specification, the SignalOK states of channels "SafeChannelOK01" and "SafeChannelOK02" are also checked.
If the SignalOK status of at least one of the two channels is not OK, the function block goes into an error state and the output signal is set to 0.
Error state "ERROR 4" is not taken from the PLCopen specification.

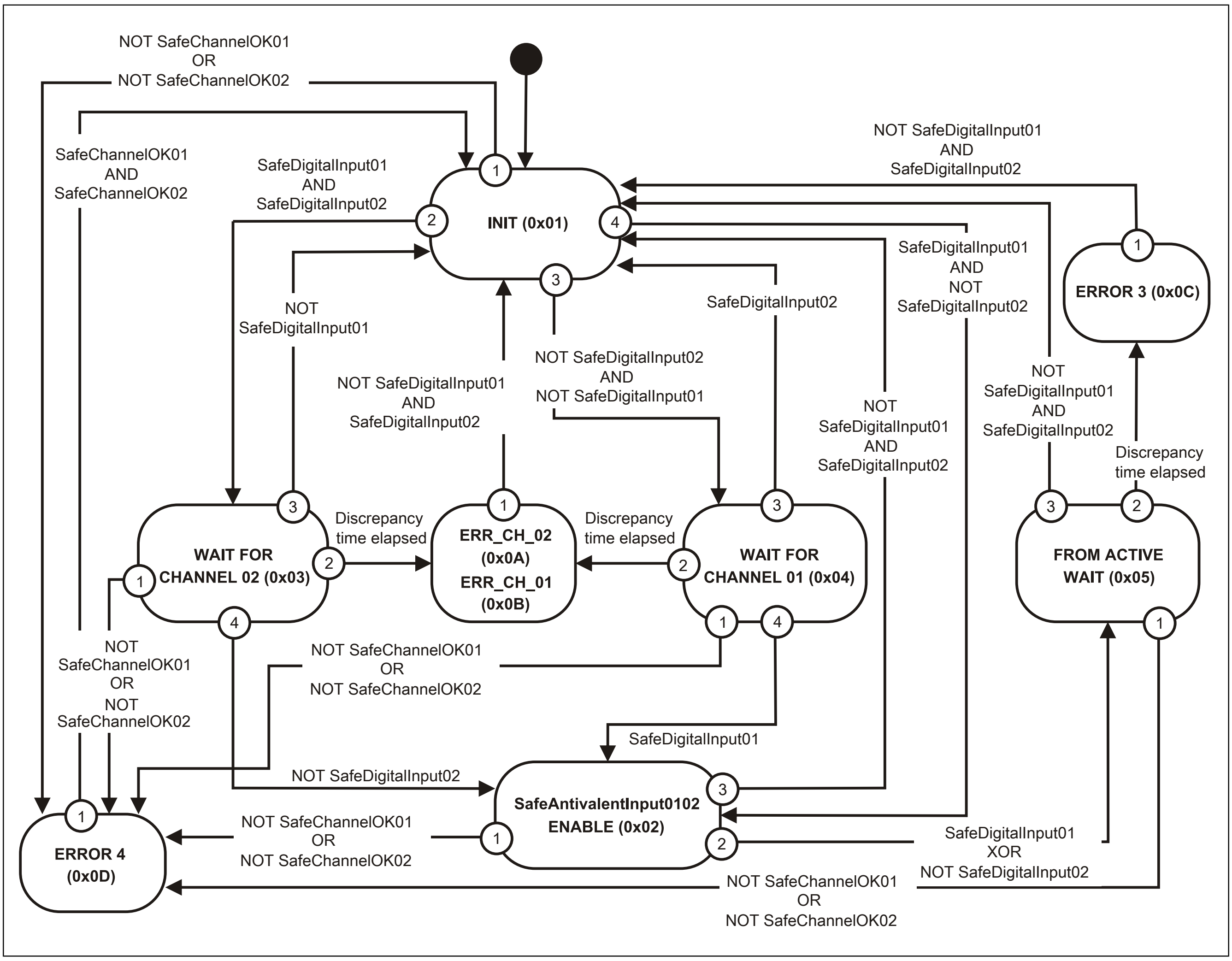


Figure 317: "Antivalent" function block - State diagram

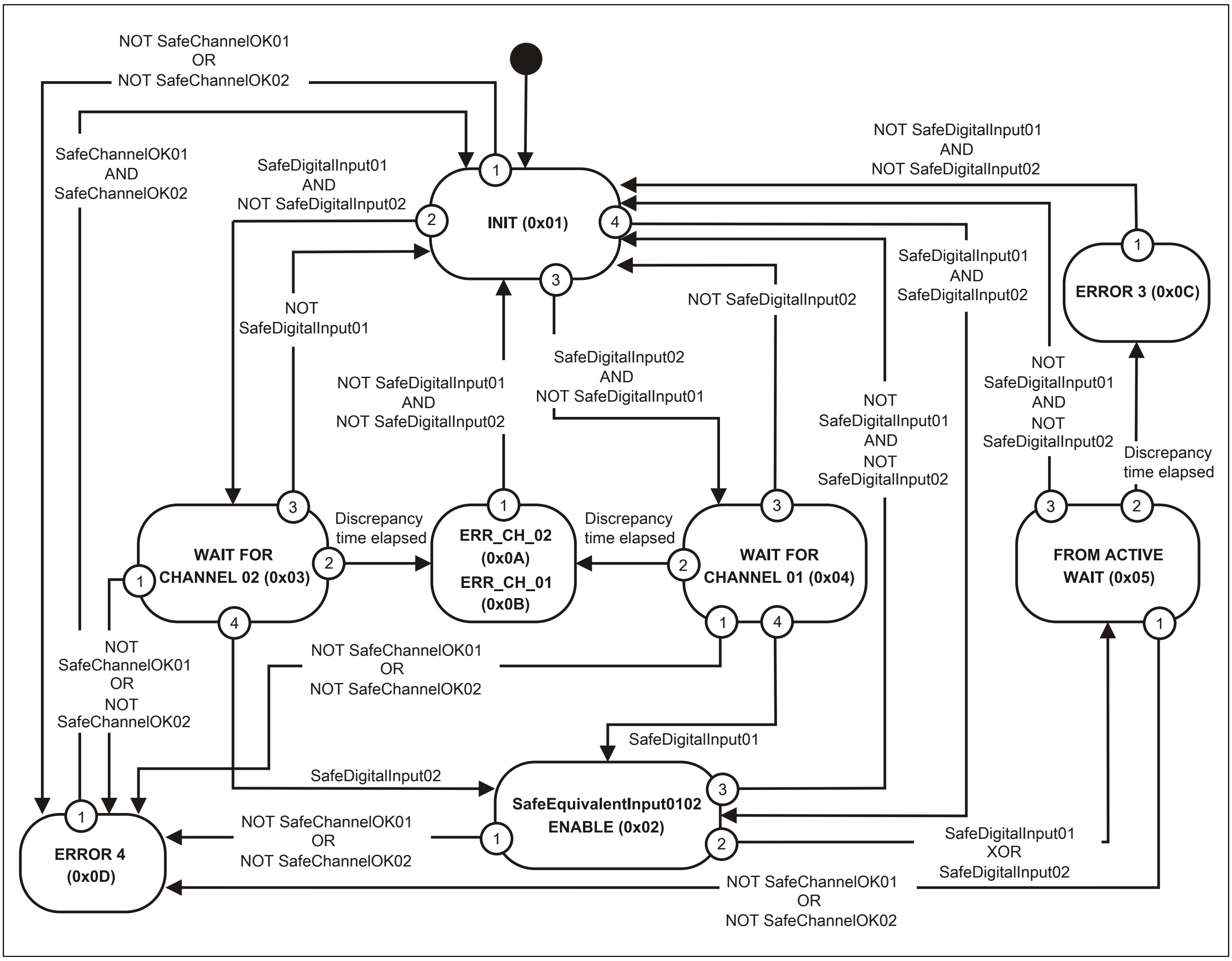


Figure 318: "Equivalent" function block - State diagram

Restart interlock state diagram

The following state diagram illustrates the effect of the restart interlock integrated in the module. The hexadecimal value in parentheses corresponds to the state number that is provided via the channel "FBK_Status_1".
For detailed information regarding restart interlock, see section "Restart behavior".

Information:

To set an output channel, a positive edge on signal "ReleaseOutput0x" is required after signal "SafeDigitalOutput0x". This edge must occur at least 1 network cycle after signal "SafeDigitalOutput0x". If this timing is not adhered to, the output channel remains inactive.

Information:

For the maximum switching frequency, see the technical data for the module.

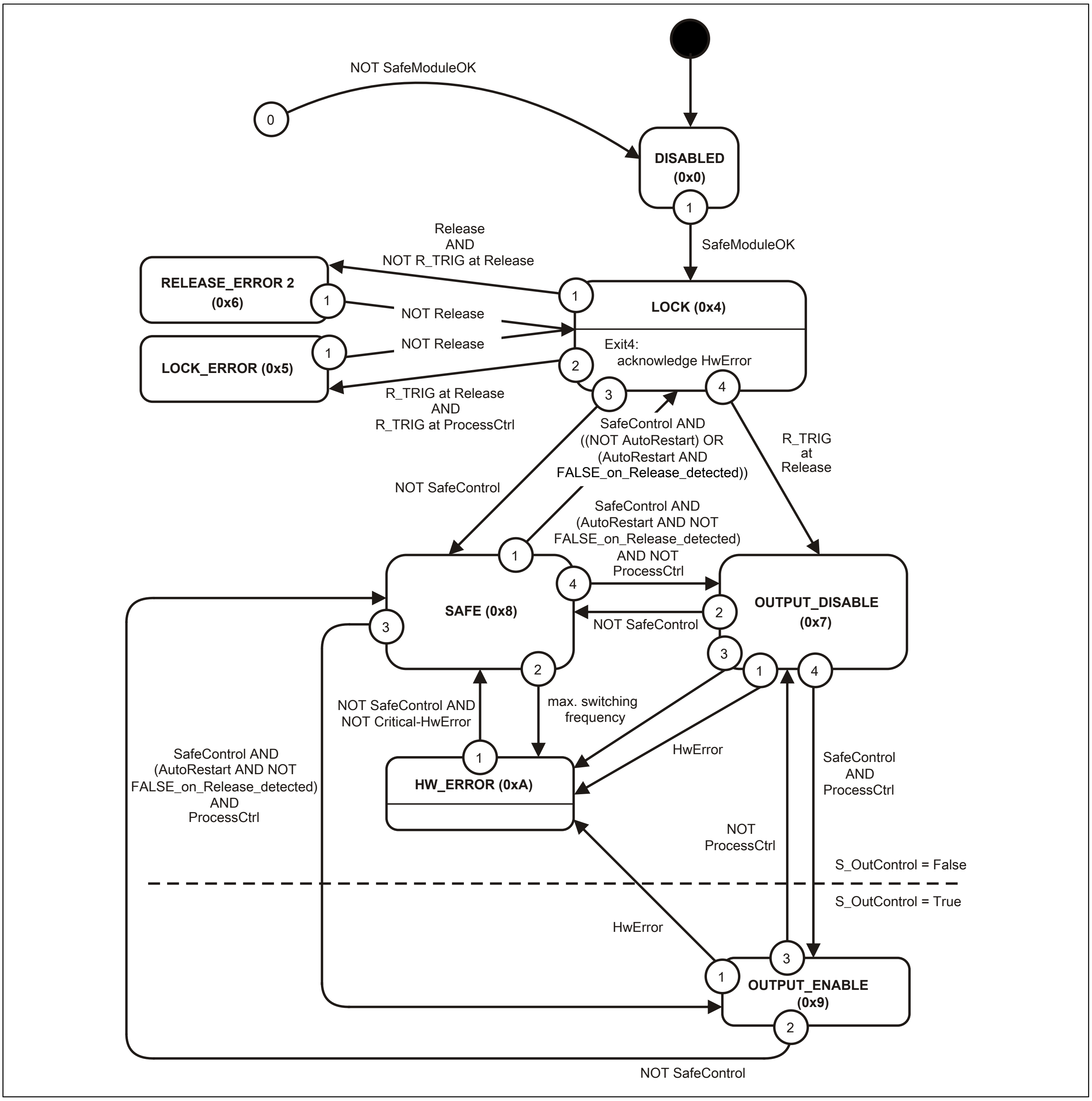


Figure 319: Restart interlock - State diagram

3.6 Accessories

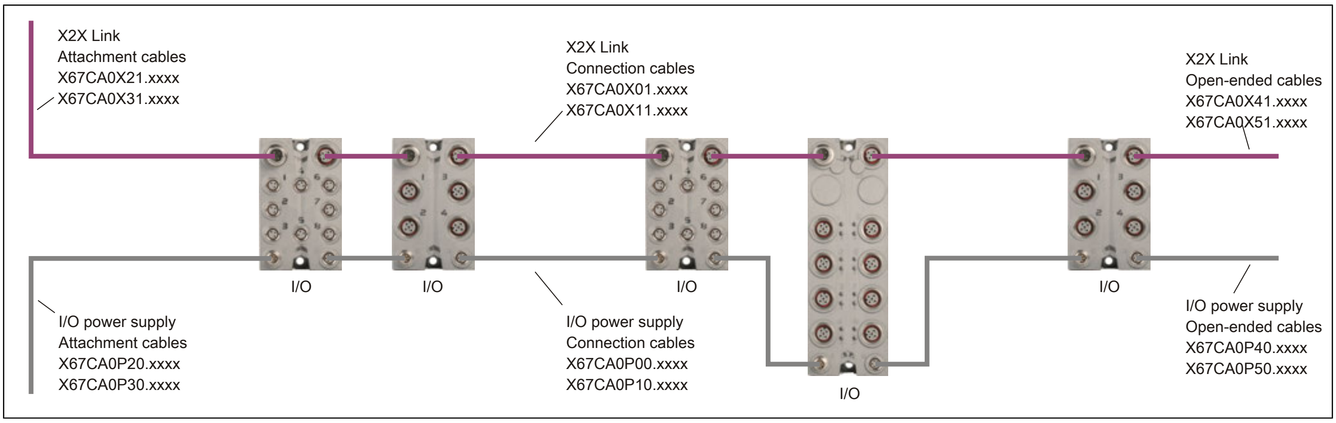
3.6.1 General overview

B&R offers pre-assembled cables, connectors and other accessories for the various connections on X67 modules.

> **Information:**
>
> **The color of the wires used in field-assembled cables may deviate from the standard. Make sure to check the proper pinout.**
>
> **Additional information about the various cables can be found in the respective documentation (see "Pre-assembled cables" on page 939) or the pinouts in the respective data sheets.**

The following overview shows all available accessories for each connector or fieldbus.

3.6.1.1 X2X Link and I/O power supply



X2X Link

M12 cable, 5-pin	Model number	Information
Attachment cables	X67CA0X21.xxxx	0.5 to 50 m; straight connector
	X67CA0X31.xxxx	2 to 25 m; angled connector
Connection cables	X67CA0X01.xxxx	0.25 to 50 m; straight connector
	X67CA0X11.xxxx	0.25 to 50 m; angled connector
Open-ended cables	X67CA0X41.xxxx	2 to 15 m; straight connector
	X67CA0X51.xxxx	2 to 5 m; angled connector
Free cable	X67CA0X99.xxxx	100 to 500 m
Field-assembled connectors	**Model number**	**Information**
Input	X67AC0X01	Cage clamp connection
	X67AC2X01	Screw clamp connection
Output	X67AC0X21	Cage clamp connection
	X67AC2X21	Screw clamp connection
Other	**Model number**	
Threaded caps	X67AC0M12	

I/O power supply

M8 cable, 4-pin	Model number	Information
Attachment cables	X67CA0P20.xxxx	0.25 to 50 m; straight connector
	X67CA0P30.xxxx	0.25 to 50 m; angled connector
Connection cables	X67CA0P00.xxxx	0.25 to 15 m; straight connector
	X67CA0P10.xxxx	0.25 to 15 m; angled connector
Open-ended cables	X67CA0P40.xxxx	0.25 to 5 m; straight connector
	X67CA0P50.xxxx	0.25 to 5 m; angled connector
Field-assembled connectors	**Model number**	**Information**
Input	X67AC0P00	Piercing connection
Output	X67AC0P20	Piercing connection
Other	**Model number**	
Threaded caps	X67AC0M08	

3.6.1.2 Module connections

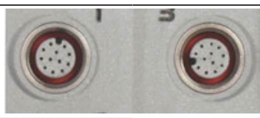
M8, 3-pin; Digital inputs/outputs

M8 cable, 3-pin	Model number	Information
Attachment cables	X67CA0D40.xxxx	2 to 20 m; straight connector
	X67CA0D50.xxxx	2 to 20 m; angled connector
Male connector	**Model number**	**Information**
Input	X67AC0D00	Piercing connection
Other	**Model number**	
Threaded caps	X67AC0M08	

M12, 5-pin; Analog and digital inputs and outputs, motor, communication

M12 cable, 5-pin	Model number	Information
Attachment cables	X67CA0A41.xxxx	2 to 20 m; straight connector
	X67CA0A51.xxxx	2 to 20 m; angled connector
Male connector	**Model number**	**Information**
Input	X67AC0A00	Cage clamp connection
	X67AC2A00	Screw clamp connection
	X67AC9A02	Thermocouple connector
Other	**Model number**	
Threaded caps	X67AC0M12	

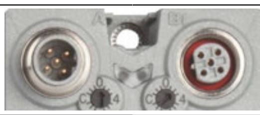
M12, 12-pin; counter, encoder



M12 cable, 12-pin	Model number	Information
Attachment cables	X67CA0I41.xxxx	2 to 10 m; straight connector
	X67CA0I51.xxxx	2 to 5 m; angled connector
Other	**Model number**	
Threaded caps	X67AC0M12	

3.6.1.3 Fieldbus systems

CAN bus / DeviceNet



M12 cable, 5-pin	Model number	Information
Attachment cables	X67CA0C22.xxxx	5 to 50 m; straight connector
	X67CA0C32.xxxx	5 to 50 m; angled connector
Connection cables	X67CA0C02.xxxx	2 to 40 m; straight connector
Field-assembled connectors	**Model number**	**Information**
Input	X67AC0C21	Cage clamp connection
	X67AC2C21	Screw clamp connection
Output	X67AC0C01	Cage clamp connection
	X67AC2C01	Screw clamp connection
Other	**Model number**	
Terminating resistor	X67AC9C03	
Y-connector	X67AC8C00	
Threaded caps	X67AC0M12	

PROFIBUS DP

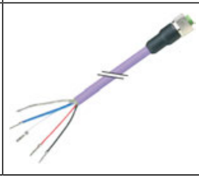
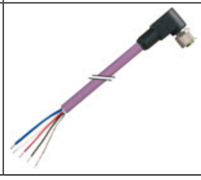
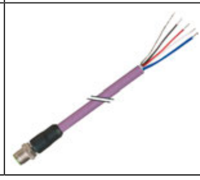
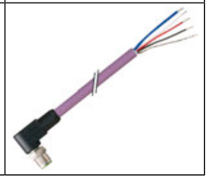
M8 cable, 4-pin	Model number	Information
Attachment cables	X67CA0B22.xxxx	5 to 50 m; straight connector
	X67CA0B32.xxxx	5 to 50 m; angled connector
Connection cables	X67CA0B12.xxxx	0.5 to 15 m; straight connector
Open-ended cables	X67CA0B52.xxxx	5 to 50 m; straight connector
Field-assembled connectors	**Model number**	**Information**
Input	X67AC0X01	Cage clamp connection
	X67AC2X01	Screw clamp connection
Output	X67AC0X21	Cage clamp connection
	X67AC2X21	Screw clamp connection
Other	**Model number**	
Terminating resistor	X67AC9B03	
Y-connector	X67AC8C00	
Threaded caps	X67AC0M12	

POWERLINK

M12 cable, 5-pin	Model number	Information
Attachment cables	X67CA0E41.xxxx	1 to 50 m; straight connector
	X67CA3E41.xxxx	15 m; straight connector; can be used in cable drag chains
Connection cables	X67CA0E61.xxxx	1 to 20 m; straight connector
Field-assembled connectors	**Model number**	**Information**
Inputs/Outputs	X67AC2E01	Insulation piercing connection
Other	**Model number**	
Threaded caps	X67AC0M12	

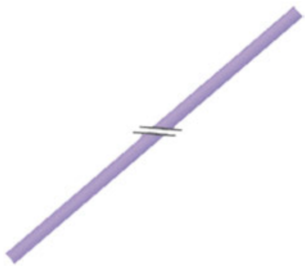
3.6.2 Pre-assembled cables

3.6.2.1 X2X Link cables

	Short description, model number					
Length	**X2X Link connection cable**		**X2X Link attachment cable**		**X2X Link open-ended cables**	
0.25 m	X67CA0X01.0002	X67CA0X11.0002				
0.3 m	X67CA0X01.0003					
0.5 m	X67CA0X01.0005	X67CA0X11.0005	X67CA0X21.0005			
1 m	X67CA0X01.0010	X67CA0X11.0010	X67CA0X21.0010		X67CA0X41.0010	
2 m	X67CA0X01.0020	X67CA0X11.0020		X67CA0X31.0020	X67CA0X41.0020	X67CA0X51.0020
5 m	X67CA0X01.0050	X67CA0X11.0050	X67CA0X21.0050	X67CA0X31.0050	X67CA0X41.0050	X67CA0X51.0050
10 m	X67CA0X01.0100	X67CA0X11.0100		X67CA0X31.0100	X67CA0X41.0100	
15 m	X67CA0X01.0150	X67CA0X11.0150	X67CA0X21.0150	X67CA0X31.0150		
20 m			X67CA0X21.0200			
25 m	X67CA0X01.0250	X67CA0X11.0250				
50 m	X67CA0X01.0500	X67CA0X11.0500	X67CA0X21.0500	X67CA0X31.0500		
						

1) Standard length = 0.2 m.

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

	Short description, model number
Length	X2X Link cable for custom assembly
100 m	X67CA0X99.1000
500 m	X67CA0X99.5000
	

3.6.2.1.1 Technical data

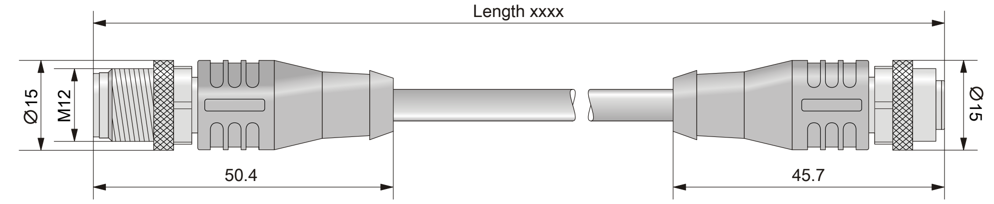
Product ID	X67CA0X01	X67CA0X11	X67CA0X21	X67CA0X31	X67CA0X41	X67CA0X51	X67CA0X99
General information							
Note	Halogen-free						
Durability	Flame resistant						
Connection	M12, 4-pin, straight	M12, 4-pin, angled	M12, 4-pin, straight	M12, 4-pin, angled	M12, 4-pin, straight	M12, 4-pin, angled	-
Type	Connection cables		Attachment cables		Open-ended cables		-
Cable cross section							
Data cables							
AWG	2x 24 AWG						
mm²	2x 0.25 mm²						
Supply lines							
AWG	2x 22 AWG						
mm²	2x 0.34 mm²						
Cable construction							
Signal lines							
Shield	Paired shield with aluminum foil						
Stranding	Twisted pair wires						
Cable stranding	0.35 mm² (22 AWG) with filler						
Complete shielding	Tinned copper braiding, coverage >85%						
Outer sheathing							
Material	Thermoplastic polyurethane (TPU)						
Color	Violet						
Labeling	B&R X67CA0Xxx.xxxx Rev. G0 ESCHA FC[1]						-
Lines							
Type	Tinned copper ETB1 Data line: fine stranded wire (19x 0.13 mm) Supply line: Fine stranded wire (19x 0.15 mm)						
Wire colors							
Data cables	Blue, white						
Supply lines	Red, black						
Wire insulation							
Data cables	Cell polyethylene (PE)						
Supply lines	Polyethylene (PE)						
Electrical characteristics							
Nominal current	Max. 4 A / contact at 40°C						
Operating voltage	Max. 250 V						
Degree of insulation	Category II in accordance with IEC 61076-2						
Conductor resistance	Data line: ≤78 Ω/km Supply line: ≤55 Ω/km						
Insulation resistance	≥100 MΩ						
Operating conditions							
EN 60529 protection							
Connector/Coupling	IP67, only when screwed in						-
Environmental conditions							
Temperature							
Transport	-40 to 80°C						
Fixed installation	-40 to 80°C						
Flexible installation[2]	-25 to 60°C						
Mechanical characteristics							
Dimensions							
Length	Various						
Diameter	6.9 mm ±0.2 mm						
Bend radius	≥15x outer diameter						
Drag chain data							
Acceleration	Max. 4 m/s²						
Flex cycles	Min. 2 million						
Speed	Max. 3 m/s						
Weight	0.063 kg/m						

Table 444: X67CA0Xxx - Technical data

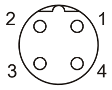
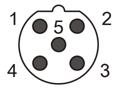
1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

3.6.2.1.2 X67CA0X01.xxxx

Dimensions



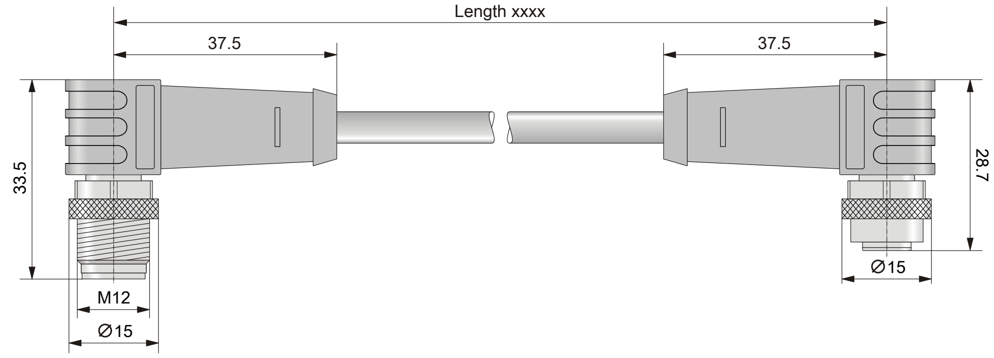
Pinout

Male connector	Pin	Description	Wire colors	Female connector
 B-coded	1	X2X+	Red	 B-coded
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	5	NC	-	
	M12 1)	SHLD	-	

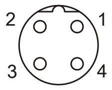
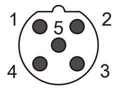
1) Shielding 360° around M12 knurled-head screw.

3.6.2.1.3 X67CA0X11.xxxx

Dimensions



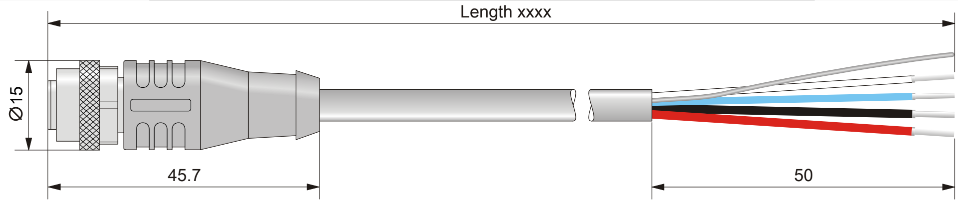
Pinout

Male connector	Pin	Description	Wire colors	Female connector
 B-coded	1	X2X+	Red	 B-coded
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	5	NC	-	
	M12 1)	SHLD	-	

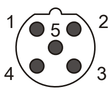
1) Shielding 360° around M12 knurled-head screw.

3.6.2.1.4 X67CA0X21.xxxx

Dimensions



Pinout

Female connector	Pin	Description	Wire colors	Open-ended
 B-coded	1	X2X+	Red	For custom Wiring
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	5	NC	-	
	M12 1)	SHLD	-	

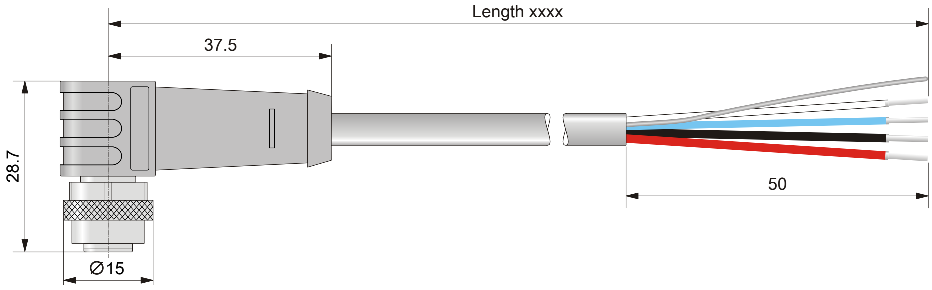
1) Shielding 360° around M12 knurled-head screw.

Information:

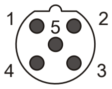
See note in section **"Connection of X2X Link interfaces with internal power supply" on page 856****.**

3.6.2.1.5 X67CA0X31.xxxx

Dimensions



Pinout

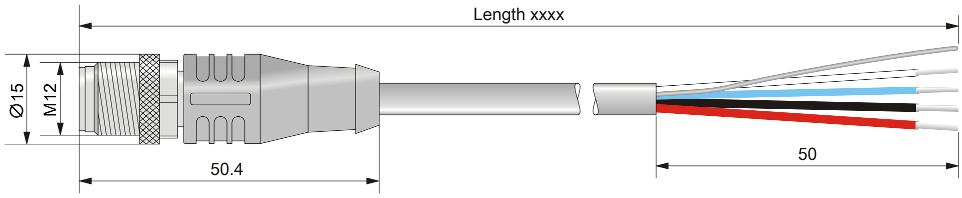
Female connector	Pin	Description	Wire colors	Open-ended
 B-coded	1	X2X+	Red	For custom Wiring
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	5	NC	-	
	M12 1)	SHLD	-	

1) Shielding 360° around M12 knurled-head screw.

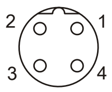
> **Information:**
>
> **See note in section "Connection of X2X Link interfaces with internal power supply" on page 856.**

3.6.2.1.6 X67CA0X41.xxxx

Dimensions



Pinout

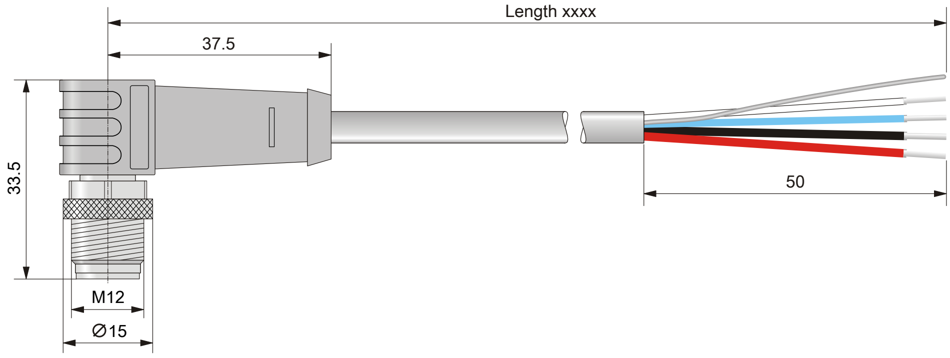
Male connector	Pin	Description	Wire colors	Open-ended
 B-coded	1	X2X+	Red	For custom Wiring
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	M12 1)	SHLD	-	

1) Shielding 360° around M12 knurled-head screw.

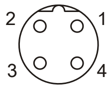
> **Information:**
>
> **See note in section "Connection of X2X Link interfaces with internal power supply" on page 856.**

3.6.2.1.7 X67CA0X51.xxxx

Dimensions



Pinout

Male connector	Pin	Description	Wire colors	Open-ended
 B-coded	1	X2X+	Red	For custom Wiring
	2	X2X	White	
	3	X2X⊥	Black	
	4	X2X\	Blue	
	M12 1)	SHLD	-	

1) Shielding 360° around M12 knurled-head screw.

Information:

See note in section **"Connection of X2X Link interfaces with internal power supply" on page 856****.**

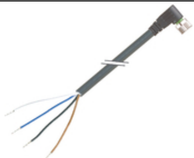
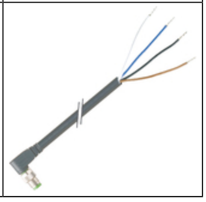
3.6.2.1.8 X67CA0X99.xxxx

Dimensions

Pinout

	Description	Wire colors	
For custom Wiring	X2X+	Red	For custom Wiring
	X2X	White	
	X2X⊥	Black	
	X2X\	Blue	
	SHLD	-	

3.6.2.2 I/O power supply cables

	Short description, model number					
Length	Power connection cable		Power attachment cable		Power open-ended cable	
0.25 m[1]	X67CA0P00.0002	X67CA0P10.0002	X67CA0P20.0002	X67CA0P30.0002	X67CA0P40.0002	X67CA0P50.0002
0.4 m						X67CA0P50.0004
0.5 m	X67CA0P00.0005					
1 m	X67CA0P00.0010	X67CA0P10.0010	X67CA0P20.0010			
1.5 m	X67CA0P00.0015					
2 m	X67CA0P00.0020	X67CA0P10.0020		X67CA0P30.0020	X67CA0P40.0020	X67CA0P50.0020
5 m	X67CA0P00.0050	X67CA0P10.0050	X67CA0P20.0050	X67CA0P30.0050	X67CA0P40.0050	X67CA0P50.0050
10 m	X67CA0P00.0100	X67CA0P10.0100	X67CA0P20.0100	X67CA0P30.0100		
15 m	X67CA0P00.0150	X67CA0P10.0150	X67CA0P20.0150	X67CA0P30.0150		
20 m	X67CA0P00.0200		X67CA0P20.0200	X67CA0P30.0200		
25 m			X67CA0P20.0250			
50 m			X67CA0P20.0500	X67CA0P30.0500		
						

1 Standard length = 0.2 m.

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

3.6.2.2.1 Technical data

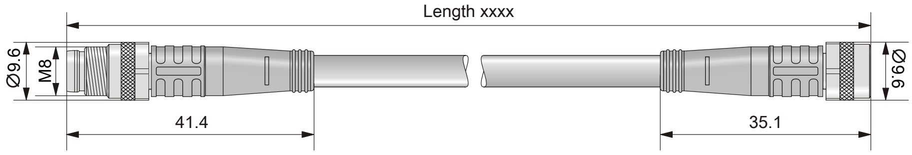
Product ID	X67CA0P00	X67CA0P10	X67CA0P20	X67CA0P30	X67CA0P40	X67CA0P50
General information						
Note	PVC- and silicone-free LABS- (PWIS-) and halogen-free					
Durability	Good chemical and oil resistance Flame resistant Good UV and ozone resistance					
Connection	M8, 4-pin, straight	M8, 4-pin, angled	M8, 4-pin, straight	M8, 4-pin, angled	M8, 4-pin, straight	M8, 4-pin, angled
Type	Connection cables		Attachment cables		Open-ended cables	
Cable cross section						
AWG	4x 22 AWG					
mm²	4x 0.34 mm²					
Cable construction						
Complete shielding	Not shielded					
Outer sheathing						
Material	Polyurethane (PUR)					
Color	Black					
Labeling	B&R X67CA0Pxx.xxxx Rev. G0 ESCHA FC[1)]					
Lines						
Wire insulation	Polypropylene (PP) 9Y					
Wire colors	Brown, black, blue, white					
Type	Uncoated copper ETP1 Fine stranded wire (42x 0.1 mm / 42x 38 AWG), class 5					
Stranding	4-wire twisted pair					
Electrical characteristics						
Nominal current	Max. 4 A in accordance with EN / contact at 40°C Max. 3 A in accordance with UL / contact					
Operating voltage	Max. 30 V					
Degree of insulation	Category II in accordance with IEC 61076-2					
Conductor resistance	≤57 Ω/km					
Insulation resistance	≥100 MΩ					
Operating conditions						
EN 60529 protection						
Connector/Coupling	IP67, only when screwed in					
Environmental conditions						
Temperature						
Transport	-40 to 90°C					
Fixed installation	-30 to 90°C					
Flexible installation[2)]	-25 to 60°C					
Mechanical characteristics						
Dimensions						
Length	Various					
Diameter	4.7 mm ±0.2 mm					
Bend radius	≥10x outer diameter					
Drag chain data						
Acceleration	Max. 5 m/s²					
Flex cycles	5 million					
Speed	Max. 3.3 m/s					

Table 445: X67CA0Pxx - Technical data

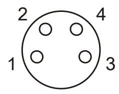
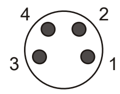
1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

3.6.2.2.2 X67CA0P00.xxxx

Dimensions

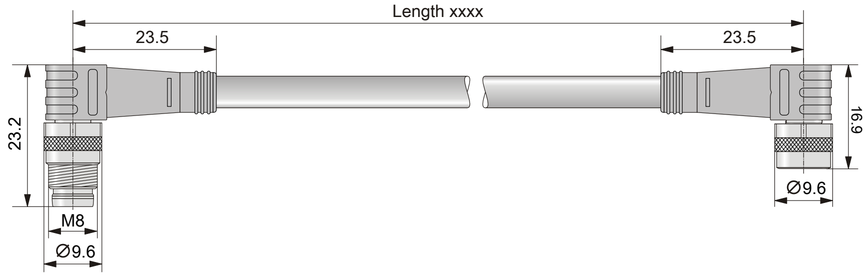


Pinout

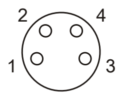
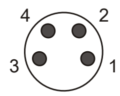
M8 male connector, 4-pin	Pin	Name	Wire colors	M8 female connector, 4-pin
	1	24 VDC	Brown	
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

3.6.2.2.3 X67CA0P10.xxxx

Dimensions

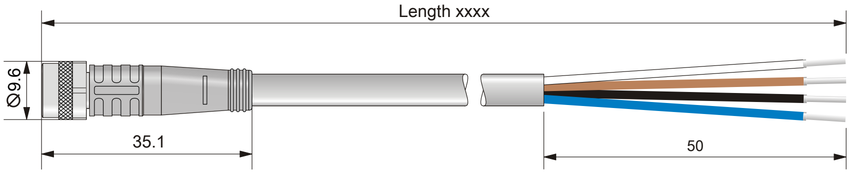


Pinout

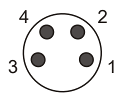
M8 male connector, 4-pin	Pin	Name	Wire colors	M8 female connector, 4-pin
	1	24 VDC	Brown	
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

3.6.2.2.4 X67CA0P20.xxxx

Dimensions



Pinout

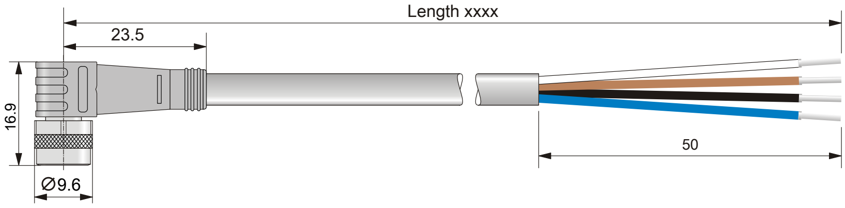
M8 female connector, 4-pin	Pin	Name	Wire colors	Open
	1	24 VDC	Brown	For custom wiring
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

Information:

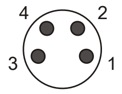
Both wires must be used since the current load is split.

3.6.2.2.5 X67CA0P30.xxxx

Dimensions



Pinout

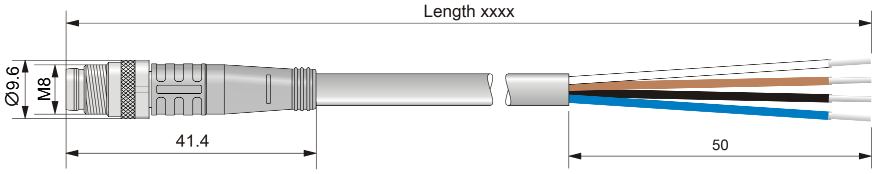
M8 female connector, 4-pin	Pin	Name	Wire colors	Open
	1	24 VDC	Brown	For custom wiring
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

Information:

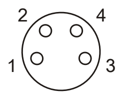
Both wires must be used since the current load is split.

3.6.2.2.6 X67CA0P40.xxxx

Dimensions



Pinout

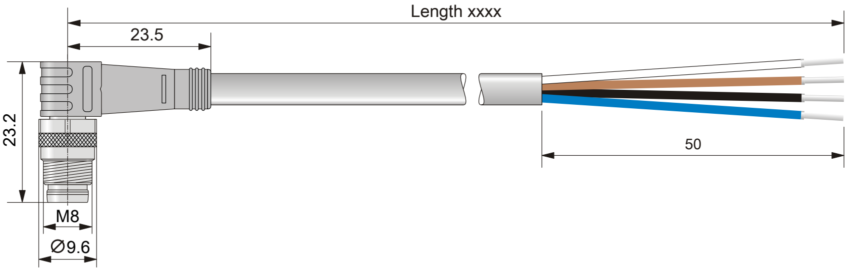
M8 male connector, 4-pin	Pin	Name	Wire colors	Open
	1	24 VDC	Brown	For custom wiring
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

Information:

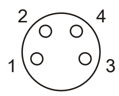
Both wires must be used since the current load is split.

3.6.2.2.7 X67CA0P50.xxxx

Dimensions



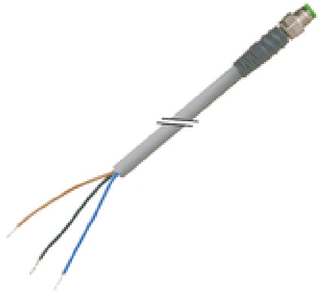
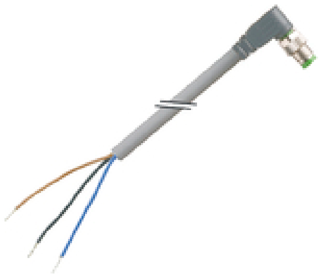
Pinout

M8 male connector, 4-pin	Pin	Name	Wire colors	Open
	1	24 VDC	Brown	For custom wiring
	2	24 VDC	White	
	3	GND	Blue	
	4	GND	Black	

Information:

Both wires must be used since the current load is split.

3.6.2.3 M8 sensor cables

	Short description	
Length	M8 sensor cables	
2 m	X67CA0D40.0020	X67CA0D50.0020
5 m	X67CA0D40.0050	X67CA0D50.0050
10 m	X67CA0D40.0100	X67CA0D50.0100
15 m	X67CA0D40.0150	X67CA0D50.0150
20 m	X67CA0D40.0200	X67CA0D50.0200
		

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

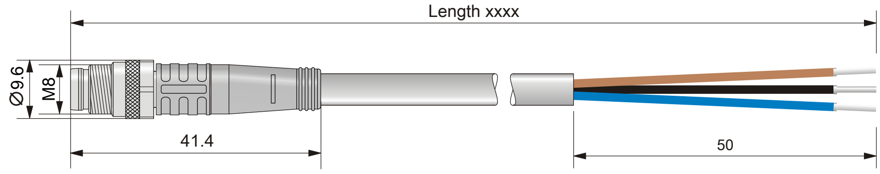
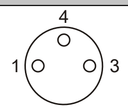
3.6.2.3.1 Technical data

Product ID	X67CA0D40	X67CA0D50
General information		
Note	PVC- and silicone-free LABS- (PWIS-) and halogen-free	
Durability	Good chemical and oil resistance Flame resistant Good UV and ozone resistance	
Connection	M8, 3-pin, straight	M8, 3-pin, angled
Type	Attachment cables	
Cable cross section		
AWG	3x 22 AWG	
mm²	3x 0.34 mm²	
Cable construction		
Complete shielding	Not shielded	
Outer sheathing		
Material	Polyurethane (PUR)	
Color	Gray	
Labeling	B&R X67CA0Dxx.xxxx Rev. G0 ESCHA FC[1]	
Lines		
Wire insulation	Polypropylene (PP) 9Y	
Wire colors	Brown, black, blue	
Type	Uncoated copper ETP1 Fine stranded wire (42x 0.1 mm / 42x 38 AWG), class 5	
Stranding	3-wire twisted pair	
Electrical characteristics		
Nominal current	Max. 4 A / contact at 40°C	
Operating voltage	Max. 60 V	
Degree of insulation	Category II in accordance with IEC 61076-2	
Conductor resistance	≤57 Ω/km	
Insulation resistance	≥100 MΩ	
Operating conditions		
EN 60529 protection		
Connector/Coupling	IP67, only when screwed in	
Environmental conditions		
Temperature		
Transport	-40 to 90°C	
Fixed installation	-30 to 90°C	
Flexible installation[2]	-25 to 60°C	
Mechanical characteristics		
Dimensions		
Length	Various	
Diameter	4.3 mm ±0.2 mm	
Bend radius	≥10x outer diameter	
Drag chain data		
Acceleration	Max. 5 m/s²	
Flex cycles	5 million	
Speed	Max. 3.3 m/s	

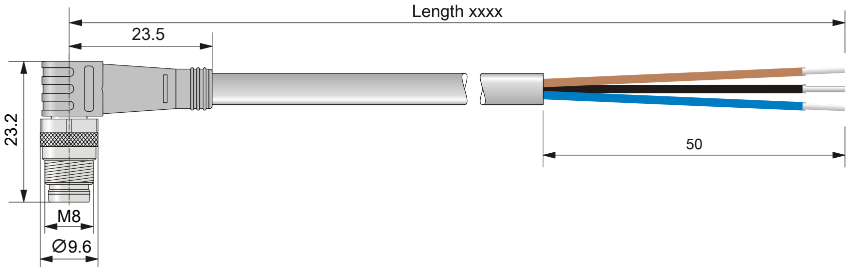
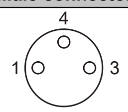
Table 446: X67CA0Dxx - Technical data

1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

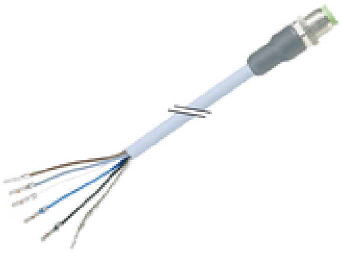
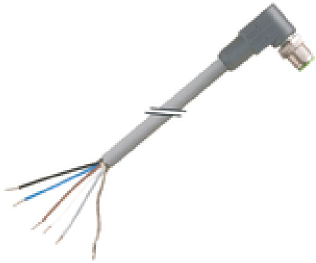
3.6.2.3.2 X67CA0D40.xxxx

Dimensions				
				
Pinout				
Male connector	**Pin**	**Name**	**Wire colors**	**Open**
	1	Sensor/actuator power supply 24 VDC	Brown	For custom wiring
	3	GND	Blue	
	4	Input/Output x	Black	

3.6.2.3.3 X67CA0D50.xxxx

Dimensions				
				
Pinout				
Male connector	**Pin**	**Name**	**Wire colors**	**Open**
	1	Sensor/actuator power supply 24 VDC	Brown	For custom wiring
	3	GND	Blue	
	4	Input/Output x	Black	

3.6.2.4 M12 sensor cables

	Short description	
Length	M12 sensor cables	
2 m	X67CA0A41.0020	X67CA0A51.0020
5 m	X67CA0A41.0050	X67CA0A51.0050
10 m	X67CA0A41.0100	X67CA0A51.0100
15 m	X67CA0A41.0150	X67CA0A51.0150
20 m	X67CA0A41.0200	X67CA0A51.0200
		

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

3.6.2.4.1 Technical data

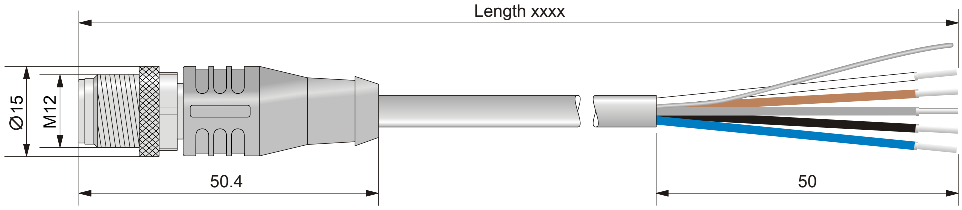
Product ID	X67CA0A41	X67CA0A51
General information		
Note	PVC- and silicone-free LABS- (PWIS-) and halogen-free	
Durability	Good chemical and oil resistance Flame resistant Good UV and ozone resistance	
Connection	M12, 5-pin, straight	M12, 5-pin, angled
Type	Attachment cables	
Cable cross section		
AWG	5x 22 AWG	
mm²	5x 0.34 mm²	
Cable construction		
Complete shielding	Tinned copper braiding, coverage 84%, 0.25 mm² with filler	
Outer sheathing		
Material	Polyurethane (PUR) UL	
Color	Gray	
Labeling	B&R X67CA0Axx.xxxx Rev. G0 ESCHA FC[1)]	
Lines		
Wire insulation	Polypropylene (PP) 9Y	
Wire colors	Brown, black, blue, white, gray	
Type	Uncoated copper ETP1 Fine stranded wire (42x 0.1 mm / 42x 38 AWG), class 5	
Stranding	5 wires stranded using filler	
Electrical characteristics		
Nominal current	Max. 4 A / contact at 40°C	
Operating voltage	Max. 60 V	
Degree of insulation	Category II in accordance with IEC 61076-2	
Conductor resistance	≤57 Ω/km	
Insulation resistance	≥100 MΩ	
Operating conditions		
EN 60529 protection		
Connector/Coupling	IP67, only when screwed in	
Environmental conditions		
Temperature		
Transport	-40 to 90°C	
Fixed installation	-30 to 90°C	
Flexible installation[2)]	-25 to 60°C	
Mechanical characteristics		
Dimensions		
Length	Various	
Diameter	5.6 mm ±0.2 mm	
Bend radius	≥12x outer diameter	
Drag chain data		
Acceleration	Max. 5m/s²	
Flex cycles	2 million	
Speed	Max. 1.6 m/s	

Table 447: X67CA0Axx - Technical data

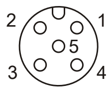
1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

3.6.2.4.2 X67CA0A41.xxxx

Dimensions



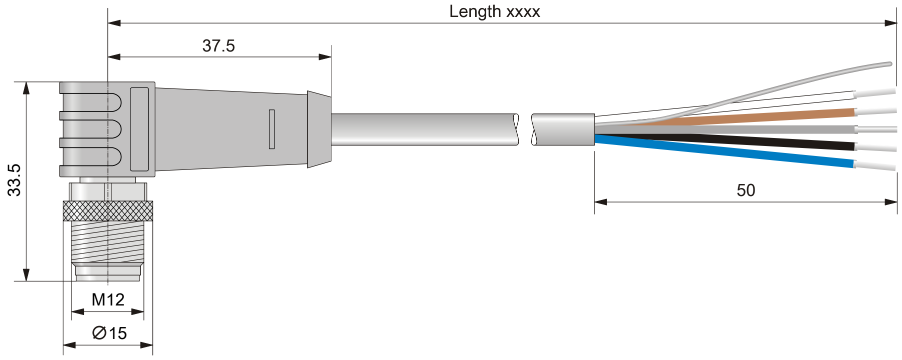
Pinout

Male connector	Pin	Name	Wire colors	Open
 A-keyed	1	Pinout, see module description	Brown	For custom wiring
	2		White	
	3		Blue	
	4		Black	
	5 1)		Gray	
	M12 2)	SHLD	-	

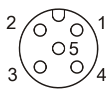
1) Do not use the gray connecting line in connection with X67 modules for which pin 5 is used as a shield connection. The cable shield for this cable is connected using a union nut.
2) Shield on M12 knurled-head screw in 360° design

3.6.2.4.3 X67CA0A51.xxxx

Dimensions

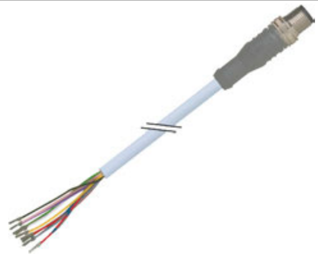


Pinout

Male connector	Pin	Name	Wire colors	Open
 A-keyed	1	Pinout, see module description	Brown	For custom wiring
	2		White	
	3		Blue	
	4		Black	
	5 1)		Gray	
	M12 2)	SHLD	-	

1) Do not use the gray connecting line in connection with X67 modules for which pin 5 is used as a shield connection. The cable shield for this cable is connected using a union nut.
2) Shield on M12 knurled-head screw in 360° design

3.6.2.5 Multifunction cables

	Short description	
Length	Multifunction attachment cable	
2 m	X67CA0I41.0020	X67CA0I51.0020
5 m	X67CA0I41.0050	X67CA0I51.0050
10 m	X67CA0I41.0100	
		

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

3.6.2.5.1 Technical data

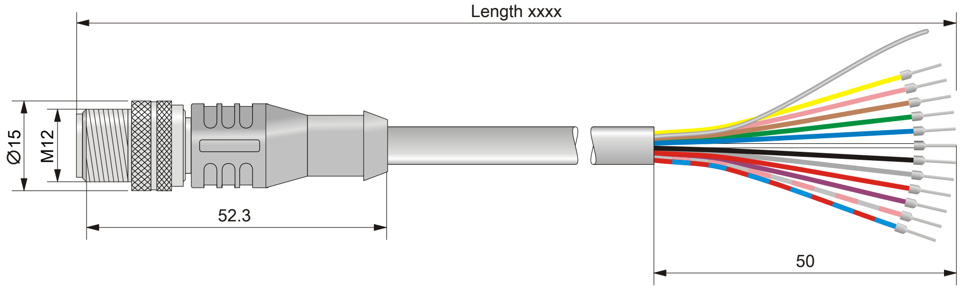
Product ID	X67CA0I41	X67CA0I51
General information		
Note	Halogen-free Free of CFCs and cadmium	
Durability	Oil resistant in accordance with VDE 0472, Part 803 Flame-retardant in accordance with VDE 0472, Part 804 / B Salt water resistant	
Connection	M12, 12-pin, straight	M12, 12-pin, angled
Type	Attachment cables	
Cable cross section		
AWG	12x 28 AWG	
mm²	12x 0.14 mm²	
Cable construction		
Complete shielding	Copper braiding, coverage >84%	
Outer sheathing		
Material	Polyether polyurethane (PUR)	
Color	Gray	
Labeling	B&R X67CA0Ixx.xxxx Rev. G0 yyyyyyy ESCHA FC[1)]	
Lines		
Wire insulation	Polypropylene (PP) 9Y	
Wire colors	Brown, black, blue, white, gray, green, pink, yellow, red, violet, gray/pink, red/blue	
Type	El copper bare conductors Fine stranded wire (72x 0.05 mm / 72x 44 AWG)	
Stranding	12-wire twisted pair	
Electrical characteristics		
Nominal current	1.5 A / Contact in accordance with IEC 60512-3	
Operating voltage	30 V	
Insulation resistance	$>10^9 \Omega$ in accordance with IEC 60512-2	
Operating conditions		
EN 60529 protection		
Connector/Coupling	IP67, only when screwed in, in accordance with IEC 60529	
Environmental conditions		
Temperature		
Transport	-40 to 90°C	
Fixed installation	-40 to 90°C	
Flexible installation	0 to 90°C	
Mechanical characteristics		
Dimensions		
Length	Various	
Diameter	6.2 mm ±0.15 mm	
Bend radius	≥10x outer diameter	

Table 448: X67CA0Ixx - Technical data

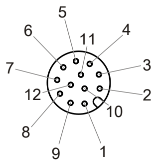
1) xx.xxxx: Group number and length of the cable;
yyyyyyy: Cable number

3.6.2.5.2 X67CA0I41.xxxx

Dimensions

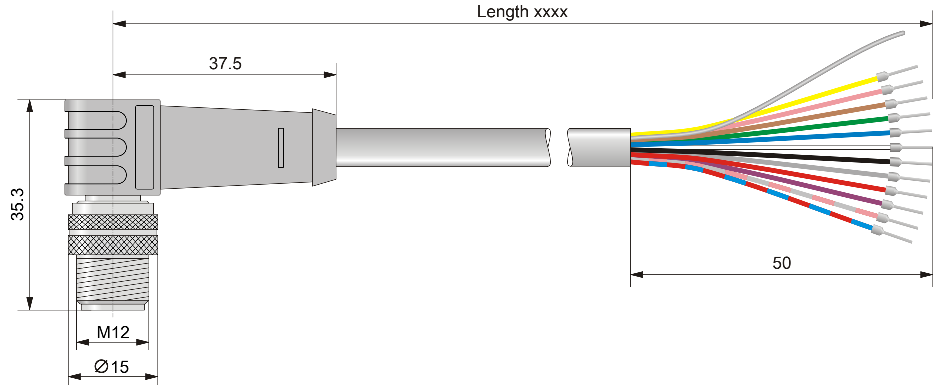


Pinout

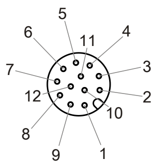
Male connector	Pin	Name	Wire colors	Open
 A-keyed	1	Pinout, see module description	Brown	For custom wiring
	2		Blue	
	3		White	
	4		Green	
	5		Pink	
	6		Yellow	
	7		Black	
	8		Gray	
	9		Red	
	10		Purple	
	11		Gray/Pink	
	12		Red/Blue	

3.6.2.5.3 X67CA0I51.xxxx

Dimensions



Pinout

Male connector	Pin	Name	Wire colors	Open
 A-keyed	1	Pinout, see module description	Brown	For custom wiring
	2		Blue	
	3		White	
	4		Green	
	5		Pink	
	6		Yellow	
	7		Black	
	8		Gray	
	9		Red	
	10		Purple	
	11		Gray/Pink	
	12		Red/Blue	

3.6.2.6 CAN bus / DeviceNet cables

	Short description, model number		
Length	Connection cables	Attachment cables	
2 m	X67CA0C02.0020		
5 m	X67CA0C02.0050	X67CA0C22.0050	X67CA0C32.0050
10 m	X67CA0C02.0100		
15 m	X67CA0C02.0150	X67CA0C22.0150	X67CA0C32.0150
20 m	X67CA0C02.0200		
35 m	X67CA0C02.0350		
40 m	X67CA0C02.0400		
50 m		X67CA0C22.0500	X67CA0C32.0500
			

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

3.6.2.6.1 Technical data

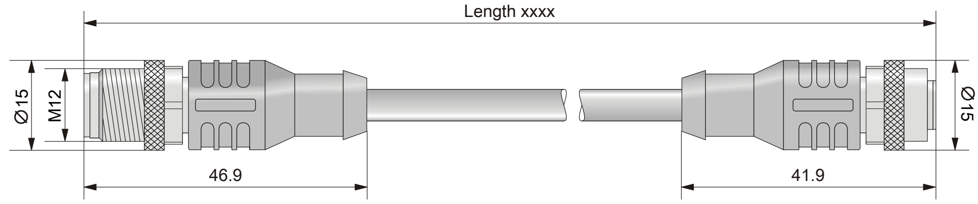
Product ID	X67CA0C02	X67CA0C22	X67CA0C32
General information			
Note	Halogen-free		
Durability	Flame resistant		
Connection	M12, 4-pin, straight	M12, 4-pin, straight	M12, 4-pin, angled
Type	Connection cables	Attachment cables	
Cable cross section			
Data cables			
AWG	2x 24 AWG		
mm²	2x 0.25 mm²		
Supply lines			
AWG	2x 22 AWG		
mm²	2x 0.34 mm²		
Cable construction			
Signal lines			
Shield	Paired shield with aluminum foil		
Stranding	Twisted pair wires		
Cable stranding	0.35 mm² (22 AWG) with filler		
Complete shielding	Tinned copper braiding, coverage >85%		
Outer sheathing			
Material	Thermoplastic polyurethane (TPU)		
Color	Violet		
Labeling	B&R X67CA0Cxx.xxxx Rev. G0 ESCHA FC[1]		
Lines			
Type	Tinned copper ETB1 Data line: fine stranded wire (19x 0.13 mm) Supply line: Fine stranded wire (19x 0.15 mm)		
Wire colors			
Data cables	Blue, white		
Supply lines	Red, black		
Wire insulation			
Data cables	Cell polyethylene (PE)		
Supply lines	Polyethylene (PE)		
Electrical characteristics			
Nominal current	Max. 4 A / contact at 40°C		
Operating voltage	Max. 250 V		
Degree of insulation	Category II in accordance with IEC 61076-2		
Conductor resistance	Data line: ≤78 Ω/km Supply line: ≤55 Ω/km		
Insulation resistance	≥100 MΩ		
Operating conditions			
EN 60529 protection			
Connector/Coupling	IP67, only when screwed in		
Environmental conditions			
Temperature			
Transport	-40 to 80°C		
Fixed installation	-40 to 80°C		
Flexible installation[2]	-25 to 60°C		
Mechanical characteristics			
Dimensions			
Length	Various		
Diameter	6.9 mm ±0.2 mm		
Bend radius	≥15x outer diameter		
Drag chain data			
Acceleration	Max. 4 m/s²		
Flex cycles	2 million		
Speed	Max. 3 m/s		
Weight	0.063 kg/m		

Table 449: X67CA0Cxx - Technical data

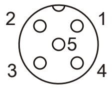
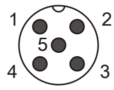
1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

3.6.2.6.2 X67CA0C02.xxxx

Dimensions

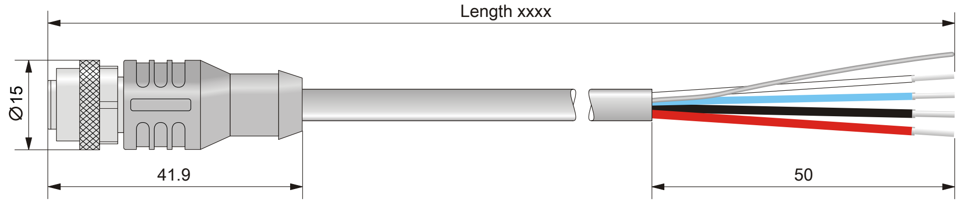


Pinout

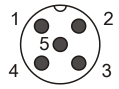
Male connector	Pin	Name	Wire colors	Female connector
 A-keyed	1	SHLD	Tracer wire / Shield	 A-keyed
	2	V+	Red	
	3	CAN GND	Black	
	4	CAN_H	White	
	5	CAN_L	Light blue	

3.6.2.6.3 X67CA0C22.xxxx

Dimensions

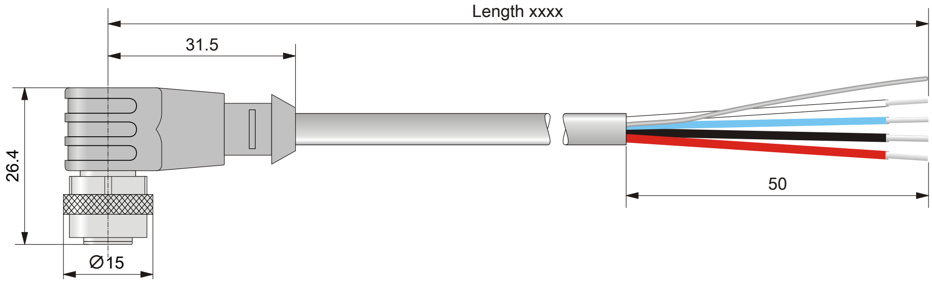


Pinout

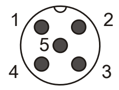
Female connector	Pin	Name	Wire colors	Open
 A-keyed	1	SHLD	Tracer wire / Shield	For custom wiring
	2	V+	Red	
	3	CAN GND	Black	
	4	CAN_H	White	
	5	CAN_L	Light blue	

3.6.2.6.4 X67CA0C32.xxxx

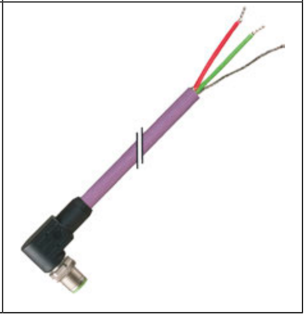
Dimensions



Pinout

Female connector	Pin	Name	Wire colors	Open
 A-keyed	1	SHLD	Tracer wire / Shield	For custom wiring
	2	V+	Red	
	3	CAN GND	Black	
	4	CAN_H	White	
	5	CAN_L	Light blue	

3.6.2.7 PROFIBUS DP cables

	Short description, model number			
Length	Connection cables	Attachment cables		Open-ended cables
0.5 m	X67CA0B12.0005			
2 m	X67CA0B12.0020			
5 m	X67CA0B12.0050	X67CA0B22.0050	X67CA0B32.0050	X67CA0B52.0050
10 m	X67CA0B12.0100			
15 m	X67CA0B12.0150	X67CA0B22.0150	X67CA0B32.0150	X67CA0B52.0150
50 m		X67CA0B22.0500	X67CA0B32.0500	X67CA0B52.0500
				

Length	Tolerances for cable lengths
0 to <1 m	+2 cm
1 m to <10 m	+5 cm
10 m to xx m	+10 cm

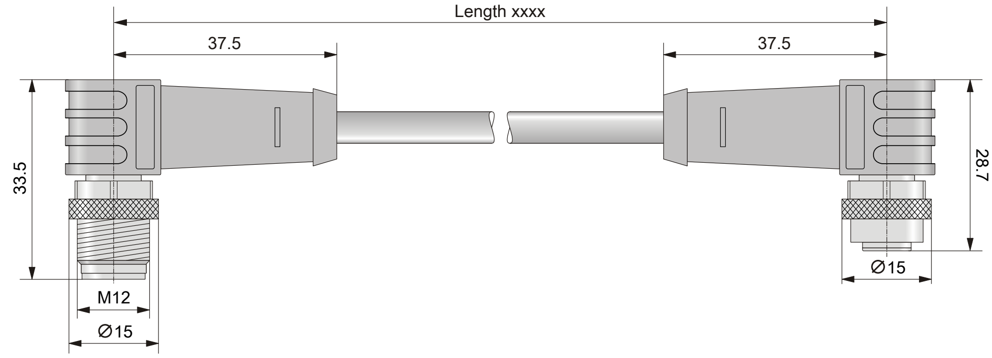
3.6.2.7.1 Technical data

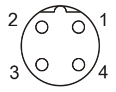
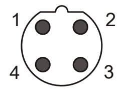
Product ID	X67CA0B12	X67CA0B22	X67CA0B32	X67CA0B52
General information				
Note	PVC- and silicone-free LABS- (PWIS-) and halogen-free			
Durability	Good chemical and oil resistance Flame resistant			
Connection	M12, 4-pin, angled	M12, 4-pin, straight	M12, 4-pin, angled	M12, 4-pin, angled
Type	Connection cables	Attachment cables		Open-ended cables
Cable cross section				
AWG	2x 22 AWG			
mm²	2x 0.34 mm²			
Cable construction				
Complete shielding	Tinned copper braiding, coverage >85%			
Outer sheathing				
Material	Polyurethane (PUR)			
Color	Violet			
Labeling	B&R X67CA0Bxx.xxxx Rev. G0 ESCHA FC[1]			
Lines				
Wire insulation	Foam polyethylene (PE) with skin layer			
Wire colors	Red, green			
Type	Uncoated copper ETB1 Fine stranded wire (19x 0.15 mm), class 5			
Stranding	2-wire twisted pair			
Electrical characteristics				
Nominal current	Max. 4 A / contact at 40°C			
Operating voltage	Max. 60 V			
Degree of insulation	Category II in accordance with IEC 61076-2			
Conductor resistance	≤55 Ω/km			
Insulation resistance	≥100 MΩ			
Operating conditions				
EN 60529 protection				
Connector/Coupling	IP67, only when screwed in			
Environmental conditions				
Temperature				
Transport	-40 to 80°C			
Fixed installation	-25 to 80°C			
Flexible installation[2]	-25 to 60°C			
Mechanical characteristics				
Dimensions				
Length	Various			
Diameter	7.6 mm ±0.3 mm			
Bend radius	≥10x outer diameter			
Drag chain data				
Acceleration	Max. 5m/s²			
Flex cycles	Min. 1 million			
Speed	Max. 3.3 m			

Table 450: X67CA0Bxx - Technical data

1) xx.xxxx: Group number and cable length.
2) In cable drag chain operation.

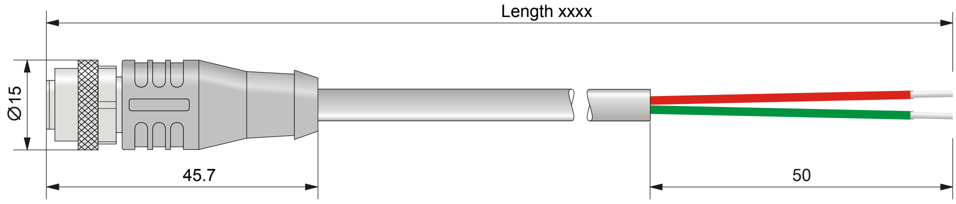
3.6.2.7.2 X67CA0B12.xxxx

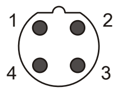
Dimensions


Pinout				
Male connector	**Pin**	**Name**	**Wire colors**	**Female connector**
	1	NC	-	
	2	A	Green	
	3	NC	-	
	4	B	Red	
B-keyed	M12[1)]	SHLD	-	B-keyed

1) Shield on M12 knurled-head screw in 360° design

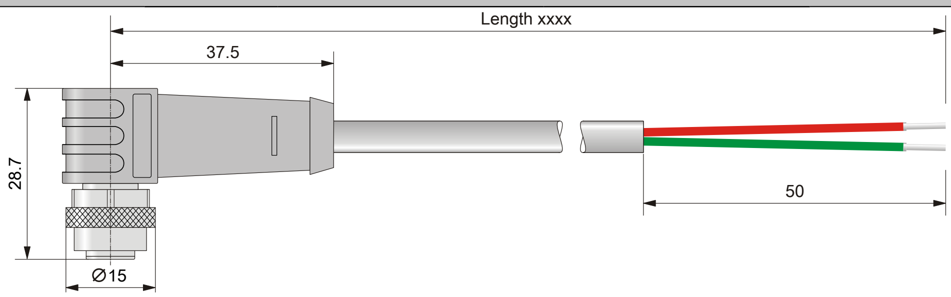
3.6.2.7.3 X67CA0B22.xxxx

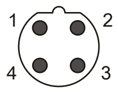
Dimensions


Pinout				
Female connector	**Pin**	**Name**	**Wire colors**	**Open**
	1	NC	-	For custom wiring
	2	A	Green	
	3	NC	-	
	4	B	Red	
B-keyed	M12[1)]	SHLD	-	

1) Shield on M12 knurled-head screw in 360° design

3.6.2.7.4 X67CA0B32.xxxx

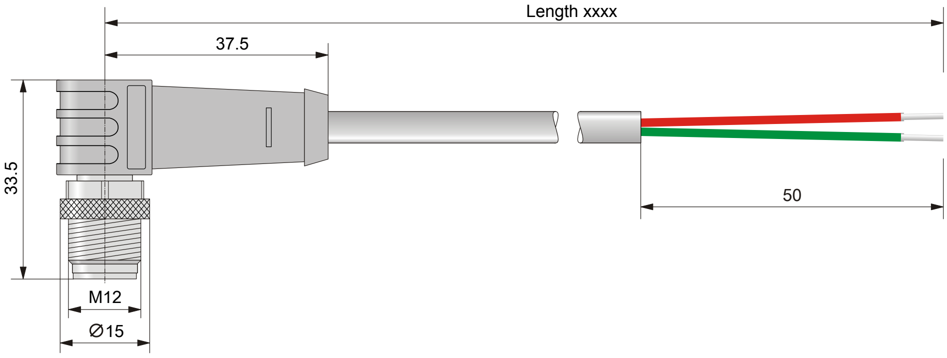
Dimensions


Pinout				
Female connector	**Pin**	**Name**	**Wire colors**	**Open**
	1	NC	-	For custom wiring
	2	A	Green	
	3	NC	-	
	4	B	Red	
B-keyed	M12	SHLD	-	

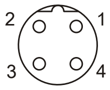
1) Shield on M12 knurled-head screw in 360° design

3.6.2.7.5 X67CA0B52.xxxx

Dimensions



Pinout

Male connector	Pin	Name	Wire colors	Open
	1	NC	-	For custom wiring
	2	A	Green	
	3	GND	-	
	4	B	Red	
B-keyed	M12[1)]	SHLD	-	

1) Shield on M12 knurled-head screw in 360° design

3.6.2.8 X67 POWERLINK/Ethernet cable

	Short description, model number	
Length	Attachment cables - RJ45 to M12	Connection cables - M12 to M12
1 m	X67CA0E41.0010	X67CA0E61.0010
2 m	X67CA0E41.0020	X67CA0E61.0020
3 m	X67CA0E41.0030	
5 m	X67CA0E41.0050	X67CA0E61.0050
10 m		X67CA0E61.0100
15 m	X67CA0E41.0150 X67CA3E41.0150	X67CA0E61.0150
20 m		X67CA0E61.0200
50 m	X67CA0E41.0500	
		

Length	**Tolerances for cable lengths**
0 to <10 m	+10 cm
10 m to <50 m	+2% of the length

3.6.2.8.1 Technical data

Product ID	X67CA0E41	X67CA0E61	X67CA3E41
General information			
Note	Halogen-free		
Durability	Flame-retardant in accordance with IEC 60332-1-2		Oil resistant in accordance with EN 60811-2-1 Flame-retardant in accordance with IEC 60332-1-2 UV resistant in accordance with UL 2556
Connection	RJ45 to M12; 4-pin	M12 to M12, 4-pin	RJ45 to M12; 4-pin
Type	Attachment cables	Connection cables	
Cable cross section			
AWG	4x 22 AWG		
mm²	4x 0.34 mm²		
Cable construction			
Complete shielding	Aluminum-clad foil (overlapping), tinned copper braiding, 85% covering		
Outer sheathing			
Material	Polyurethane (PUR)		
Color	Green		
Labeling	B&R X67CA0Exx.xxxx Rev. C0 [1]		
Lines			
Wire insulation	Polyethylene (PE)		
Wire colors	White, yellow, blue, orange		
Type	Tinned copper stranded wire Fine stranded wire (7x 0.25 mm / 7x 30 AWG)		
Stranding	4-wire twisted pair		
Electrical characteristics			
Conductor resistance	≤120 Ω/km at 20°C		
Transfer properties	Category 5 / Class D up to 100 MHz in accordance with ISO/IEC 11801 (EN50173-1), ISO/IEC 24702 (EN 50173-3)		
Transfer rate	10/100 Mbit/s		
Insulation resistance	≥500 MΩ/km at 20°C		
Operating conditions			
EN 60529 protection			
Cables	IP67		
Male M12 connector	IP67, only when screwed in		
RJ45 connector	IP20, only when connected properly		
Environmental conditions			
Temperature			
Transport	-40 to 70°C		
Fixed installation	-40 to 70°C		
Flexible installation	-20 to 60°C		
Mechanical characteristics			
Dimensions	Various		
Length			
Diameter	6.5 mm ±0.2 mm		
Bend radius	≥7x outer diameter		
Drag chain data			
Acceleration	-		4 m/s²
Flex cycles	-		Min. 3 million
Speed	-		4 m/s
Weight	0.062 kg/m		0.061 kg/m

Table 451: X67CAxExx - Technical data

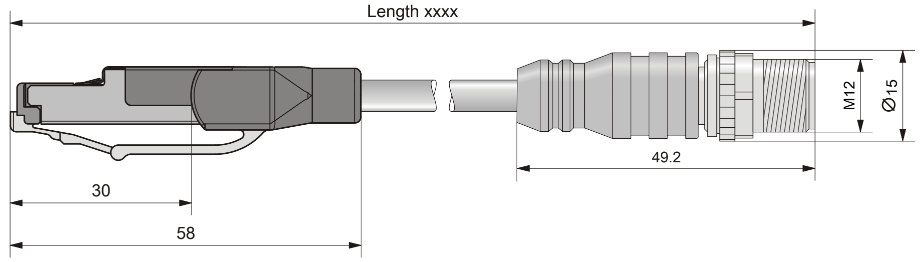
1) xx.xxxx: Group number and cable length.

3.6.2.8.2 X67CA0E41.xxxx and X67CA3E41.xxxx

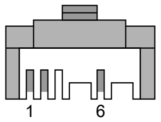
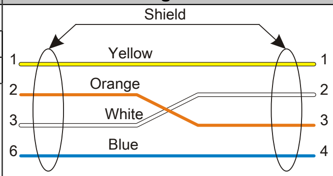
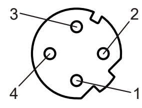
This cable is offered in 2 variants:

- X67CA0Exx: Standard variant
- X67CA3Exx: Can be used in cable drag chains

Dimensions

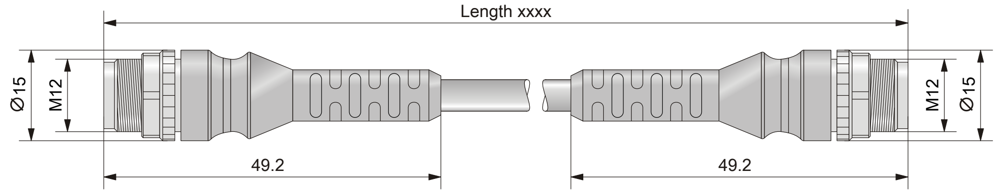


Pinout

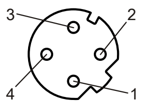
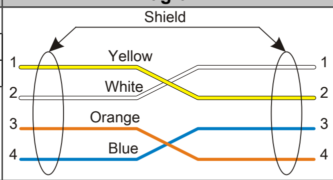
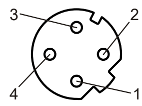
Male RJ45 connector	Pin	Description	Diagram	M12 connector
	1 - 1	TXD		 D-coded
	2 - 3	RXD		
	3 - 2	TXD\		
	6 - 4	RXD\		

3.6.2.8.3 X67CA0E61.xxxx

Dimensions



Pinout

Male connector	Pin	Description	Diagram	Male connector
 D-coded	1 - 2	TXD		 D-coded
	2 - 1	RXD		
	3 - 4	TXD\		
	4 - 3	RXD\		

3.6.3 Field-assembled connectors

3.6.3.1 I/O power supply

Model number	Short description	Figure
	I/O supply	
X67AC0P00	X67 male M8 connector, 4-pin, piercing connection	

Table 452: X67AC0P00 - Order data

Model number	Short description	Figure
	I/O supply	
X67AC0P20	X67 female M8 connector, 4-pin, piercing connection	

Table 453: X67AC0P20 - Order data

3.6.3.2 Sensors/Actuators

Model number	Short description	Figure
	Sensors / actuators	
X67AC0D00	X67 male M8 connector, 3-pin, piercing connection	

Table 454: X67AC0D00 - Order data

Model number	Short description	Figure
	Sensors / actuators	
X67AC0A00	X67 male M12 connector, 5-pin, A-keyed, cage clamp connection	

Table 455: X67AC0A00 - Order data

Model number	Short description	Figure
	Sensors / actuators	
X67AC2A00	X67 male M12 connector, 5-pin, A-keyed, screw clamp connection	

Table 456: X67AC2A00 - Order data

3.6.3.3 Special-purpose connectors

Model number	Short description	Figure
	Special plugs	
X67AC9A02	X67 M12 thermocouple connector, for temperature compensation at measurement points, screw clamp connections	

Table 457: X67AC9A02 - Order data

3.6.3.4 CAN bus / DeviceNet

Model number	Short description	Figure
	CAN bus / DeviceNet	
X67AC0C01	X67 male M12 connector, 5-pin, A-keyed, shielded, cage clamp connection	

Table 458: X67AC0C01 - Order data

Model number	Short description	Figure
	CAN bus / DeviceNet	
X67AC2C01	X67 male M12 connector, 5-pin, A-keyed, shielded, screw clamp connection	

Table 459: X67AC2C01 - Order data

Model number	Short description	Figure
	CAN bus / DeviceNet	
X67AC0C21	X67 female M12 connector, 5-pin, A-keyed, shielded, cage clamp connection	

Table 460: X67AC0C21 - Order data

Model number	Short description	Figure
	CAN bus / DeviceNet	
X67AC2C21	X67 female M12 connector, 5-pin, A-keyed, shielded, screw clamp connection	

Table 461: X67AC2C21 - Order data

3.6.3.5 PROFIBUS DP / X2X Link

Model number	Short description	Figure
	PROFIBUS DP / X2X Link	
X67AC0X01	X67 male M12 connector, 5-pin, B-keyed, shielded, cage clamp connection	

Table 462: X67AC0X01 - Order data

Model number	Short description	Figure
	PROFIBUS DP / X2X Link	
X67AC2X01	X67 male M12 connector, 5-pin, B-keyed, shielded, screw clamp connection	

Table 463: X67AC2X01 - Order data

Model number	Short description	Figure
	PROFIBUS DP / X2X Link	
X67AC0X21	X67 female M12 connector, 5-pin, B-keyed, shielded, cage clamp connection	

Table 464: X67AC0X21 - Order data

Model number	Short description	Figure
	PROFIBUS DP / X2X Link	
X67AC2X21	X67 female M12 connector, 5-pin, B-keyed, shielded, screw clamp connection	

Table 465: X67AC2X21 - Order data

3.6.3.6 POWERLINK/Ethernet

Model number	Short description	Figure
	POWERLINK/Ethernet	
X67AC2E01	X67 male M12 connector, 4-pin, D-keyed, shielded, insulation piercing connection	

Table 466: X67AC2E01 - Order data

3.6.4 Other accessories

3.6.4.1 Terminating resistor

Model number	Short description	Figure
	Terminating resistor	
X67AC9C03	X67 CAN M12 terminating resistor	

Table 467: X67AC9C03 - Order data

Model number	Short description	Figure
	Terminating resistor	
X67AC9B03	X67 PROFIBUS DP M12 terminating resistor	

Table 468: X67AC9B03 - Order data

3.6.4.2 Connectors

Model number	Short description	Figure
	Connectors	
X67AC8C00	X67 CAN Y-connector	

Table 469: X67AC8C00 - Order data

Model number	Short description	Figure
	Connectors	
X67AC8B01	X67 PROFIBUS DP Y-connector	

Table 470: X67AC8B01 - Order data

3.6.4.3 Threaded caps

Model number	Short description	Figure
	Threaded caps	
X67AC0M08	X67 M8 threaded caps, 50 pcs.	
X67AC0M12	X67 M12 threaded caps, 50 pcs.	

Table 471: X67AC0M08, X67AC0M12 - Order data

3.6.4.4 Plain text tags

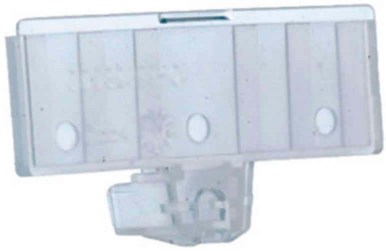
Model number	Short description	Figure
	Plain text tag	
X67AC0SH1	X67 plain text tag	

Table 472: X67AC0SH1 - Order data

3.6.4.5 Mounting plates for top-hat rails

Model number	Short description	Figure
	Mounting plates for DIN rails	
X67ACTS35	X67 top-hat rail mounting plate	
X67ACTS35.0010	X67 top-hat rail mounting plate, 10 pcs.	

Table 473: X67ACTS35, X67ACTS35.0010 - Order data

3.6.4.6 Installation tool

The connectors and couplings of pre-assembled X67 cables have additional width across flats on the knurled-head screw that can be used for an installation tool. A torque wrench (M8 or M12) is available as an accessory to make installation easy. It can be used to ensure the absolute safety of the connection to the X67 module.

Model number	Short description	Figure
	Mounting tools	
X67ACTQMX	X67 torque wrench set, for X67 M8 and M12 connectors, for hex-head connectors	

Table 474: X67ACTQMX - Order data

3.7 International and national certifications

Products and services from B&R comply with applicable standards. This includes international standards from organizations such as ISO, IEC and CENELEC, as well as national standards from organizations such as UL, CSA, DNV GL, etc. We are committed to ensuring the reliability of our products in an industrial environment.

Information:

Certifications that apply to a particular module are available at the following places:

- **The data sheet's technical data under "Certifications".**
- **At www.br-automation.com in the technical data under "Products" in section "Certifications"**
- **On the side of the module housing**

3.7.1 Overview of certifications

Markings	Function	certificate authority	Region
	CE marking	Notified bodies	Europe (EU)
	Functional safety (CE)	Notified bodies	Europe (EU)
	Explosion protection (CE)	Notified bodies	Europe (EU)
	Underwriters Laboratories Inc. (UL) (certification for US and Canada)	UL	Canada USA
	Canadian Standards Association (CSA) (certification for US and Canada)	CSA	Canada USA
	GOST-R	Federal agency on technical regulating and metrology	Former Russian Federation
	Eurasian Conformity (EAC)	Federal agency on technical regulating and metrology	Eurasian Trade Union
	Korean Conformity (KC)	Radio Research Agency (RRA)	Korea
	Regulatory Compliance Mark (RCM)	ACMA	Australia Oceania

3.7.2 EU directives and standards (CE)

CE markings

Europe (EU)

The respective product complies with all applicable EU directives and relevant harmonized standards.

Certification of these products is performed in cooperation with accredited testing laboratories.

EMC directive 2014/30/EU

All devices satisfy the protection requirements of the "EMC directive" and are designed for industrial use:

Applicable standards from this directive:

EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing
EN 61000-6-2	Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity standard for industrial environments
EN 61000-6-4	Electromagnetic compatibility (EMC) - Part 6-4: Generic standards ; Generic standards - Emission standard for industrial environments

Low voltage directive 2014/35/EU

The low voltage directive applies to electrical equipment with a nominal voltage from 50 to 1000 VAC and from 75 to 1500 VDC.
All devices within the area of application of this directive satisfy the its protection requirements.

Applicable standard from this directive:

EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing

The corresponding declaration of conformity is available for download from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity PLC

Machinery directive 2006/42/EC

Functional safety

open SAFETY certified product

Europe (EU)

In accordance with the machinery directive, safety technology products are designed, developed, tested and labeled for special applications providing protection to machinery and personnel.

Certification of these products is performed exclusively in cooperation with EU-authorized bodies (Notified Bodies).

Applicable standards from this directive:

SIL 3:

IEC 61508-1	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 1: General requirements
IEC 61508-2	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
IEC 61508-3	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements
IEC 61508-4	Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations
EN 62061	Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems
IEC 61511-1	Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements

PL e, Cat. 4:

EN ISO 13849-1	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design
EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing

Declarations of conformity, certificates and any other safety-related documentation can be downloaded from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity FS PLC

Certificate

Homepage > Downloads > Certificates > Safety technology > X20, X67

User's manual

Homepage > Downloads > Safety technology > Integrated Safety Technology User's Manual

ATEX directive 2014/34/EU

ATEX / Zone 2

II 3G Ex nA nC IIA T5 Gc

Europe (EU)

Products with this marking are suitable for use in potentially explosive environments. The X20 system is certified for use in environments with explosive gases with a normal level of safety (Zone 2).

Certification of these products is performed exclusively in cooperation with EU-authorized bodies (Notified Bodies).

Each module is also accompanied by an information sheet providing detailed installation and safety guidelines.

Applicable standards from this directive:

EN 60079-0	Explosive atmospheres Part 0: Equipment - General requirements
EN 60079-15	Explosive atmospheres - Part 15: Equipment protection by type of protection "n"

The declaration of conformity and certificate can be downloaded from the B&R website. For information about the versions of applicable standards, see the declaration of conformity.

Declaration of conformity

Homepage > Downloads > Certificates > Declarations of conformity > Declaration of conformity ATEX X67

Certificate

Homepage > Downloads > Certificates > ATEX > X67 > TÜV 05 ATEX 7201 X

3.7.2.1 Overview of standards

Standard	Description
EN ISO 13849-1	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design
EN 55011 (CISPR 11)	Industrial, scientific and medical equipment - Radio frequency disturbance characteristics - Limits and methods of measurement
EN 55016-2-1 (CISPR 16-2-1)	Specification for radio disturbance and immunity measuring apparatus and methods - Part 2-1: Methods of measurement of disturbances and immunity - Conducted disturbance measurements
EN 55016-2-3 (CISPR 16-2-3)	Specification for radio disturbance and immunity measuring apparatus and methods - Part 2-3: Methods of measurement of disturbances and immunity - Radiated disturbance measurements
EN 55022 (CISPR 22)	Information technology equipment - Radio disturbance characteristics - Limits and methods of measurement
EN 60068-2-6	Environmental testing - Part 2-6: Procedures - Test Fc: Vibration (sinusoidal)
EN 60068-2-27	Environmental testing - Part 2-27: Test procedure - Test Ea and guidance: Shock
EN 60068-2-31[1)]	Environmental testing - Part 2-31: Test procedure - Test Ec: Rough handling shocks, mainly for devices
EN 60079-0	Explosive atmospheres Part 0: Equipment - General requirements
EN 60079-15	Explosive atmospheres - Part 15: Equipment protection by type of protection "n"
EN 60529	Degrees of protection provided by enclosures (IP code)
EN 60664-1	Insulation coordination for equipment within low-voltage systems - Part 1: Principles, requirements and tests
EN 60721-3-2	Classification of environmental conditions - Part 3: Classification of groups of environmental parameters and their severities - Section 2: Transport and handling
EN 60721-3-5	Classification of environmental conditions - Part 3: Classification of groups of environmental parameters and their severities - Section 5: Usage on and in land vehicles
EN 61000-4-2	Electromagnetic compatibility (EMC) - Part 4-2: Testing and measurement techniques - Electrostatic discharge immunity test
EN 61000-4-3	Electromagnetic compatibility (EMC) - Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test
EN 61000-4-4	Electromagnetic compatibility (EMC) - Part 4-4: Testing and measurement techniques - Electrical fast transient/burst immunity test
EN 61000-4-5	Electromagnetic compatibility (EMC) - Part 4-5: Testing and measuring techniques - Surge immunity test
EN 61000-4-6	Electromagnetic compatibility (EMC) - Part 4-6: Testing and measurement techniques - Immunity to conducted disturbances, induced by radio-frequency fields
EN 61000-4-8	Electromagnetic compatibility (EMC) - Part 4-8: Testing and measuring techniques - Power frequency magnetic field immunity test
EN 61000-4-11	Electromagnetic compatibility (EMC) - Part 4-11: Testing and measuring techniques - Voltage dips, short interruptions and voltage variations
EN 61000-4-29	Electromagnetic compatibility (EMC) - Part 4-29: Testing and measurement techniques - Voltage dips, short interruptions and voltage variations on d.c. input power port immunity tests
EN 61000-6-2	Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity standard for industrial environments
EN 61000-6-4	Electromagnetic compatibility (EMC) - Part 6-4: Generic standards - Emission standard for industrial environments
EN 61131-2	Programmable logic controllers - Part 2: Guidance for inspection and routine testing
IEC 61508-1	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 1: General requirements
IEC 61508-2	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems
IEC 61508-3	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 3: Software requirements
IEC 61508-4	Functional safety of electrical / electronic / programmable electronic safety-related systems - Part 4: Definitions and abbreviations
IEC 61511-1	Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements
EN 62061	Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems

1) Replacement for EN 60068-2-32

3.7.2.2 Requirements for immunity to disturbances

Immunity	Test carried out in accordance with	Requirements in accordance with
Electrostatic discharge (ESD)	EN 61000-4-2	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
High-frequency electromagnetic fields (HF field)	EN 61000-4-3	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
High-speed transient electrical disturbances (Burst)	EN 61000-4-4	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Surge voltages (Surge)	EN 61000-4-5	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Conducted disturbances	EN 61000-4-6	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Magnetic fields with electrical frequencies	EN 61000-4-8	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Voltage dips (AC) Short-term interruptions (AC) Voltage fluctuations (AC)	EN 61000-4-11	EN 61131-2: Product standard - Programmable logic controllers
		EN 61000-6-2: Generic standard - Immunity to disturbances in industrial sectors
Short-term interruptions (DC) Voltage fluctuations (DC)	EN 61000-4-29	EN 61131-2: Product standard - Programmable logic controllers

Evaluation criteria for performance

Criteria	During testing	After testing
A	The PLC system shall continue to operate as intended. No loss of function or performance.	The PLC system shall continue to operate as intended.
B	Degradation of performance accepted. No change of operating mode. No irreversible loss of stored data.	The PLC system shall continue to operate as intended. Temporary degradation of performance must be self-recoverable.
C	Loss of functions accepted, but no destruction of hardware or software (program or data).	The PLC system shall continue to operate as intended automatically, after manual restart or power off/power on.
D	Degradation or failure of functionality that can no longer be restored.	PLC system permanently damaged or destroyed.

Electrostatic discharge (ESD)

Test carried out in accordance with EN 61000-4-2	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Contact discharge (CD) to conductive external parts	±4 kV Criteria B	
Air discharge (AD) to insulating external parts	±8 kV Criteria B	

High-frequency electromagnetic fields (HF field)

Test carried out in accordance with EN 61000-4-3	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Housing, completely wired	80 MHz to 1 GHz, 10 V/m 1.4 GHz to 2 GHz, 3 V/m 2 GHz to 2.7 GHz, 1 V/m Criteria A	

High-speed transient electrical disturbances (Burst)

Test carried out in accordance with EN 61000-4-4	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	±2 kV / 5 kHz Criteria B	
AC power outputs	±2 kV / 5 kHz[1)] Criteria B	±2 kV / 5 kHz Criteria B
AC other I/Os	±2 kV / 5 kHz[1)] Criteria B	-
DC mains inputs/outputs	±2 kV / 5 kHz[1)] Criteria B	
Other I/Os and interfaces	±1 kV / 5 kHz[1)] Criteria B	

1) Only for connections with a permitted line length >3 m.

Surge voltages (Surge)

Test carried out in accordance with EN 61000-4-5	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC mains inputs/outputs Line / line	±1 kV Criteria B	
AC mains inputs/outputs Line / ground	±2 kV Criteria B	
DC mains inputs/outputs Line / line	±0.5 kV[1)] Criteria B	±0.5 kV Criteria B
DC power inputs Line / ground	±0.5 kV[1)] Criteria B	±0.5 kV Criteria B
DC power outputs Line / ground	±0.5 kV[1)] Criteria B	±0.5 kV Criteria B
Signal connections, unshielded Line / ground	±1 kV[1)] Criteria B	
All shielded lines Line / ground	±1 kV[1)] Criteria B	-

1) Only for connections with a permitted line length >30 m.

Conducted disturbances

Test carried out in accordance with EN 61000-4-6	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC mains inputs/outputs	10 V 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	
DC mains inputs/outputs	10 V 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	
Other I/Os and interfaces	10 V[1)] 150 kHz to 80 MHz 80% AM (1 kHz) Criteria A	

1) Only for connections with a permitted line length >3 m.

Magnetic fields with electrical frequencies

Test carried out in accordance with EN 61000-4-8	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
Housing, completely wired	30 A/m 3 axes (x, y, z) 50/60 Hz[1)] Criteria A	

1) Mains frequency per manufacturer data

Voltage dips

Test carried out in accordance with EN 61000-4-11	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	0% residual voltage 250/300 periods (50/60 Hz)[1] 20 attempts Criteria C	
	40% residual voltage 10/12 periods (50/60 Hz)[1] 20 attempts Criteria C	
	70% residual voltage 25/30 periods (50/60 Hz)[1] 20 attempts Criteria C	

1) Mains frequency per manufacturer data

Short-term interruptions

Test carried out in accordance with EN 61000-4-11 / EN 61000-4-29	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	0% residual voltage 0.5 periods (50/60 Hz)[1] 20 interruptions Criteria A	0% residual voltage 1 period (50/60 Hz)[1] 3 interruptions Criteria B
DC power inputs	0% residual voltage ≥10 ms (PS2) 20 interruptions Criteria A	-

1) Mains frequency per manufacturer data

Voltage fluctuations

Test carried out in accordance with EN 61000-4-11 / EN 61000-4-29	Requirements in accordance with EN 61131-2 / Zone B	Requirements in accordance with EN 61000-6-2
AC power inputs	-15% / +10% Test duration per 30 minutes Criteria A	-
DC power inputs	-15% / +20% Test duration per 30 minutes Criteria A	-

3.7.2.3 Emission requirements

<table>
<tr><th>Phenomenon</th><th>Test carried out in accordance with</th><th>Limits in accordance with</th></tr>
<tr><td rowspan="2">Emissions related to lines</td><td rowspan="2">EN 55011 / EN 55022
EN 55016-2-1</td><td>EN 61131-2: Product standard -
Programmable logic controllers</td></tr>
<tr><td>EN 61000-6-4: Generic standard -
Emissions in industrial sectors</td></tr>
<tr><td rowspan="2">Radiated emissions</td><td rowspan="2">EN 55011 / EN 55022
EN 55016-2-3</td><td>EN 61131-2: Product standard -
Programmable logic controllers</td></tr>
<tr><td>EN 61000-6-4: Generic standard -
Emissions in industrial sectors</td></tr>
</table>

Emissions related to lines

<table>
<tr><th>Test carried out in accordance with
EN 55011 / EN 55022 / EN 55016-2-1</th><th>Limits in accordance with
EN 61131-2 / Zone B</th><th>Limits in accordance with
EN 61000-6-4</th></tr>
<tr><td rowspan="2">AC mains connection
150 kHz to 30 MHz</td><td colspan="2">150 kHz to 500 kHz
79 dB (µV) quasi-peak value
66 dB (µV) average value</td></tr>
<tr><td colspan="2">500 kHz to 30 MHz
73 dB (µV) quasi-peak value
60 dB (µV) average value</td></tr>
<tr><td rowspan="2">Telecommunications / network connection
150 kHz to 30 MHz</td><td>-</td><td>150 kHz to 500 kHz
97 to 87 dB (µV) quasi-peak value
53 to 40 dB (µA) quasi-peak value
84 to 74 dB (µV) average value
40 to 30 dB (µA) average value</td></tr>
<tr><td>-</td><td>500 kHz to 30 MHz
87 dB (µV) quasi-peak value
43 dB (µA) quasi-peak value
74 dB (µV) average value
30 dB (µA) average value</td></tr>
</table>

Radiated emissions

<table>
<tr><th>Test carried out in accordance with
EN 55011 / EN 55022 / EN 55016-2-3</th><th>Limits in accordance with
EN 61131-2 / Zone B</th><th>Limits in accordance with
EN 61000-6-4</th></tr>
<tr><td rowspan="2">Electric field / Measured from 10 m
30 MHz to 1 GHz</td><td colspan="2">30 MHz to 230 MHz
40 dB (µV/m) quasi-peak value</td></tr>
<tr><td colspan="2">230 MHz to 1 GHz
47 dB (µV/m) quasi-peak value</td></tr>
<tr><td rowspan="2">Electric field / Measured from 3 m
1 GHz to 6 GHz[1)]</td><td>-</td><td>1 GHz to 3 GHz[1)]
76 dB (µV/m) peak value
56 dB (µV/m) average value</td></tr>
<tr><td>-</td><td>3 GHz to 6 GHz[1)]
80 dB (µV/m) peak value
60 dB (µV/m) average value</td></tr>
</table>

1) Depending on highest internal frequency

3.7.2.4 Mechanical conditions

Testing	Test carried out in accordance with	Requirements in accordance with
Vibration (sinusoidal) / Operation	EN 60068-2-6	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-5 / Class 5M3
Shock / Operation	EN 60068-2-27	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-5 / Class 5M3
Vibration (sinusoidal) / Transport (packaged)	EN 60068-2-6	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
		EN 60721-3-2 / Class 2M3
Shock / Transport (packaged)	EN 60068-2-27	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
Free fall / Transport (packaged)	EN 60068-2-31[1)]	EN 61131-2: Product standard - Programmable logic controllers
		EN 60721-3-2 / Class 2M1
Toppling / Transport (packaged)	EN 60068-2-31	EN 60721-3-2 / Class 2M1
		EN 60721-3-2 / Class 2M2
		EN 60721-3-2 / Class 2M3

1) Replacement for EN 60068-2-32

Vibration (sinusoidal) / Operation

Test carried out in accordance with EN 60068-2-6	Requirements in accordance with EN 61131-2		Requirements in accordance with EN 60721-3-5 / Class 5M3	
Vibration (sinusoidal) / Operation[1)]	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**
	5 to 8.4 Hz	Deflection 3.5 mm	2 to 8 Hz	Deflection 7.5 mm
	8.4 to 150 Hz	Acceleration 1 g[2)]	8 to 200 Hz	Acceleration 2 g[2)]
	-	-	200 to 500 Hz	Acceleration 4 g[2)]
	20 sweeps for each axis[3)]			

1) Uninterrupted duty with movable frequency in all 3 axes (x, y, z); 1 octave per minute
2) 1 g = 10 m/s²
3) 2 sweeps = 1 frequency cycle (fmin → fmax → fmin)

Shock / Operation

Test carried out in accordance with EN 60068-2-27	Requirements in accordance with EN 61131-2	Requirements in accordance with EN 60721-3-5 / Class 5M3
Shock / Operation[1)]	Acceleration 15 g Duration 11 ms 18 shocks	Acceleration 30 g Duration 11 ms 18 shocks

1) Pulse (half-sine) stress in all 3 axes (x, y, z)

Vibration (sinusoidal) / Transport (packaged)

Test carried out in accordance with EN 60068-2-6	Requirements in accordance with EN 60721-3-2 / Class 2M1		Requirements in accordance with EN 60721-3-2 / Class 2M2		Requirements in accordance with EN 60721-3-2 / Class 2M3	
Vibration (sinusoidal) / Transport (packaged)[1)]	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**	**Frequency**	**Amplitude**
	2 to 9 Hz	Deflection 3.5 mm	2 to 9 Hz	Deflection 3.5 mm	2 to 8 Hz	Deflection 7.5 mm
	9 to 200 Hz	Acceleration 1 g[2)]	9 to 200 Hz	Acceleration 1 g[2)]	8 to 200 Hz	Acceleration 2 g[2)]
	200 to 500 Hz	Acceleration 1.5 g[2)]	200 to 500 Hz	Acceleration 1.5 g[2)]	200 to 500 Hz	Acceleration 4 g[2)]
	20 sweeps for each axis[3)]					

1) Uninterrupted duty with movable frequency in all 3 axes (x, y, z); 1 octave per minute
2) 1 g = 10 m/s²
3) 2 sweeps = 1 frequency cycle (fmin → fmax → fmin)

Shock / Transport (packaged)

Test carried out in accordance with EN 60068-2-27	Requirements in accordance with EN 60721-3-2 / Class 2M1	Requirements in accordance with EN 60721-3-2 / Class 2M2
Shock / Transport (packaged)[1)]	Type I Acceleration 10 g Duration 11 ms 18 shocks	
	Type II -	Type II Acceleration 30 g Duration 6 ms 18 shocks

1) Pulse (half-sine) stress in all 3 axes (x, y, z)

Free fall / Transport (packaged)

Tests in accordance with EN 60068-2-31	Requirements in accordance with EN 61131-2 with shipping packaging		Requirements in accordance with EN 61131-2 with shipping packaging		Requirements in accordance with EN 60721-3-2 / Class 2M1	
Free fall / Transport (packaged)	**Weight**	**Height**	**Weight**	**Height**	**Weight**	**Height**
	<10 kg	1.0 m	<10 kg	0.3 m	<20 kg	0.25 m
	10 to 40 kg	0.5 m	10 to 40 kg	0.3 m	20 to 100 kg	0.25 m
	>40 kg	0.25 m	>40 kg	0.25 m	>100 kg	0.1 m
	5 attempts					

1) Replacement for EN 60068-2-32

Toppling / Transport (packaged)

Test carried out in accordance with EN 60068-2-31	Requirements in accordance with EN 60721-3-2 / Class 2M1		Requirements in accordance with EN 60721-3-2 / Class 2M2		Requirements in accordance with EN 60721-3-2 / Class 2M3	
Toppling / Transport (packaged)	**Weight**	**Required**	**Weight**	**Required**	**Weight**	**Required**
	<20 kg	Yes	<20 kg	Yes	<20 kg	Yes
	20 to 100 kg	-	20 to 100 kg	Yes	20 to 100 kg	Yes
	>100 kg	-	>100 kg	-	>100 kg	Yes
	Topple on all edges					

3.7.2.5 Electrical safety

Overvoltage category

Requirement per EN 61131-2	Definition per EN 60664-1
Overvoltage category II	Equipment of "Overvoltage category II" is energy-consuming equipment to be supplied by the fixed installation.

Pollution degree

Requirement per EN 61131-2	Definition per EN 60664-1
Pollution degree 2	Only non-conductive pollution occurs. Occasionally, however, temporary conductivity caused by condensation is to be expected.

Protection rating provided by enclosure (IP code)

Requirement per manufacturer	Meaning of codes per EN 60529	Meaning for the protection of equipment	Meaning for the protection of personnel
IP67	First number IP**6**x	Dust-proof	Protected against touching dangerous parts with conductor.
	Second number IPx**7**	Protected against the effects of temporary submersion in water.	

3.7.3 UL / CSA

UL

Ind. Cont. Eq.
E115267

Canada / USA

Underwriters Laboratories (UL)

Products with this marking have been tested by Underwriters Laboratories and are listed as "Industrial Control Equipment" in category NRAQ (programmable controllers) with file number E115267.

This marking is valid for the USA and Canada and simplifies the certification of your machines and systems in these regions.

Standards applied:

UL 508	Standard for industrial control equipment
UL 61010-1	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements
UL 61010-2-201	Standard for Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use - Part 2-201: Particular Requirements for Control Equipment
CSA C22.2 No. 142-M1987	Process control equipment
CSA C22.2 No. 61010-1	Safety requirements for electrical equipment for measurement, control and laboratory use - Part 1: General requirements
CSA C22.2 No. 61010-2-201	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 2-201: Particular requirements for control equipment

Certificate

Homepage > Downloads > Certificates > UL > X67 > E115267 UL Certificate of Compliance X67

CSA HazLoc

HazLoc Cl. I, Div. 2
Group ABCD
0-60°C T5

Canada / USA

Canadian Standards Association (CSA)

Products with this marking have been tested by the Canadian Standards Association and are suitable for use in potentially explosive environments.
The products are listed in CLASS 2258 (Process control equipment - For hazardous locations) with file number 244665.
The X20 system is certified for Hazardous Locations Class I, Division 2.
Each certified module is accompanied by an information sheet providing detailed installation and safety guidelines.

This marking is valid for the USA and Canada and simplifies the certification of your machines and systems in these regions.

Standards applied:

CSA C22.2 No. 0-M1991	General Requirements - Canadian Electrical Code Part II
CSA C22.2 No. 142-M1987	Process control equipment
CSA C22.2 No. 213-M1987	Non-incendive electrical equipment for use in Class I, Division 2 hazardous locations
UL Std No. 916:2007	Energy Management Equipment
ANSI/ISA 12.12.01:2007	Nonincendive Electrical Equipment for Use in Class I and II, Division 2 and Class III, Division 1 and 2 Hazardous (Classified) Locations

Certificate

Homepage > Downloads > Certificates > HazLoc > CSA > X20, X67 > 244665 CSA HazLoc Certificate of Compliance X20, X67

3.7.4 Other certifications

GOST-R

GOST-R

Products with this marking have been tested by an accredited testing laboratory and approved for import to the Russian Federation (based on EU compliance).

EAC

Eurasian Conformity (EAC)

Products with this marking have been tested by an accredited testing laboratory and approved for import (based on EU compliance) to the newly founded Eurasian Economic Union (Russia, Belarus, Kazakhstan, etc.).

KC

Korean Conformity (KC)

Products with this marking have been tested by an accredited testing laboratory and approved for import to the Korean market (based on EU compliance).

RCM

Regulatory Compliance Mark (RCM)

Products with this marking have been tested by an accredited testing laboratory and certified by the ACMA. This marking is valid in Australia/Oceania and simplifies the certification of your machines and systems in these areas (based on EU compliance).

4 Connection examples

4.1 Introduction

This chapter describes how to connect sensors and actuators with safe B&R input or output channels.

The connection examples in this chapter should be considered as assistance when using B&R safety technology. Other connection variants are certainly possible according to the normative requirements and technical specifications and your own discretion.

The entire contents of this chapter have been prepared assiduously and thoroughly. The information contained herein is provided as assistance to the user. Nevertheless, the user must assume responsibility for any checks that must be made, especially concerning suitability in individual cases. It is therefore the sole responsibility of the user to check current standards with regard to contents and applicability to the respective field of activity. Quoted passages from and interpretations of standards as well as the examples are meant solely as explanations and in no way claim to be complete. They are therefore not permitted to be generalized or used without verification. B&R Industrial Automation GmbH and its subsidiaries (B&R in the following) assume no liability for recommendations that may arise from or be implied by the following descriptions. Guarantee, warranty or liability claims that go beyond the general B&R terms of delivery cannot be derived from the following descriptions.

4.2 Connecting drive systems

This section looks at connecting drive systems to a safe output channel. Particular attention is given to the STO connection on the drive system. Information about all other connections is included in the documentation for the respective drive system.

4.2.1 Circuit examples

4.2.1.1 Drive system on type A safe output channel (high-side / low-side) with OSSD

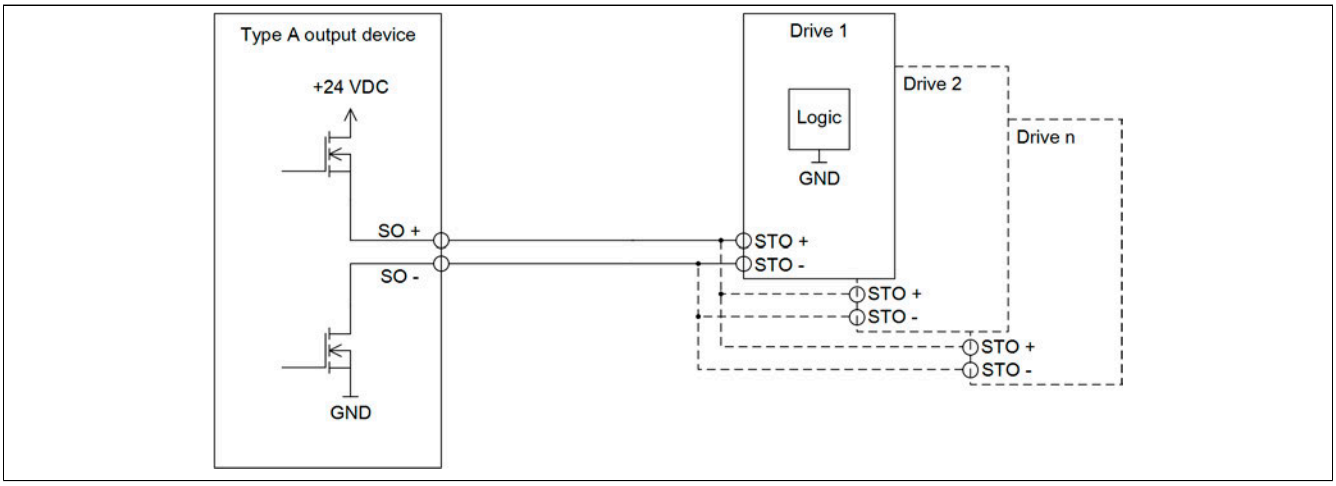


Figure 320: Drive system on type A safe output channel (high-side / low-side) with OSSD - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	No

Table 475: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

4.2.1.2 Drive system on type A safe output channel (high-side / low-side) without OSSD

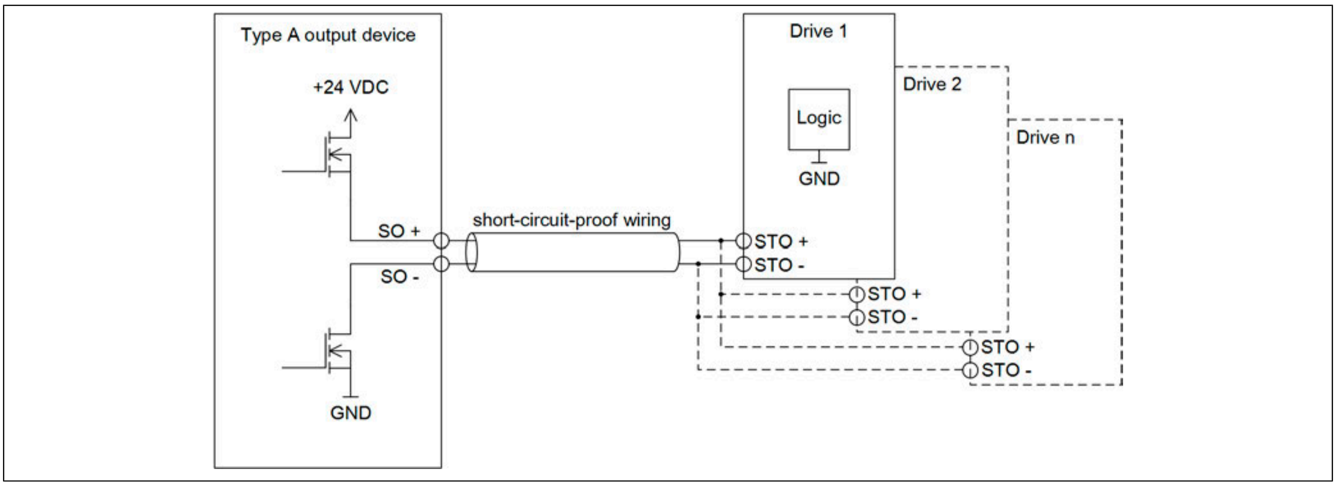


Figure 321: Drive system on type A safe output channel (high-side / low-side) without OSSD - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 3.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	Yes-ATTENTION

Table 476: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements of SIL 3 in accordance with EN IEC 62061:2010, or PL e according to EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2, in accordance with EN IEC 62061:2010, or PL d according to EN ISO 13849-1:2015, an additional check of the safety function by the user may be necessary. For details, see the data sheet for the corresponding module.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

4.2.1.3 Drive system on type B safe output channel (high-side / high-side) with OSSD

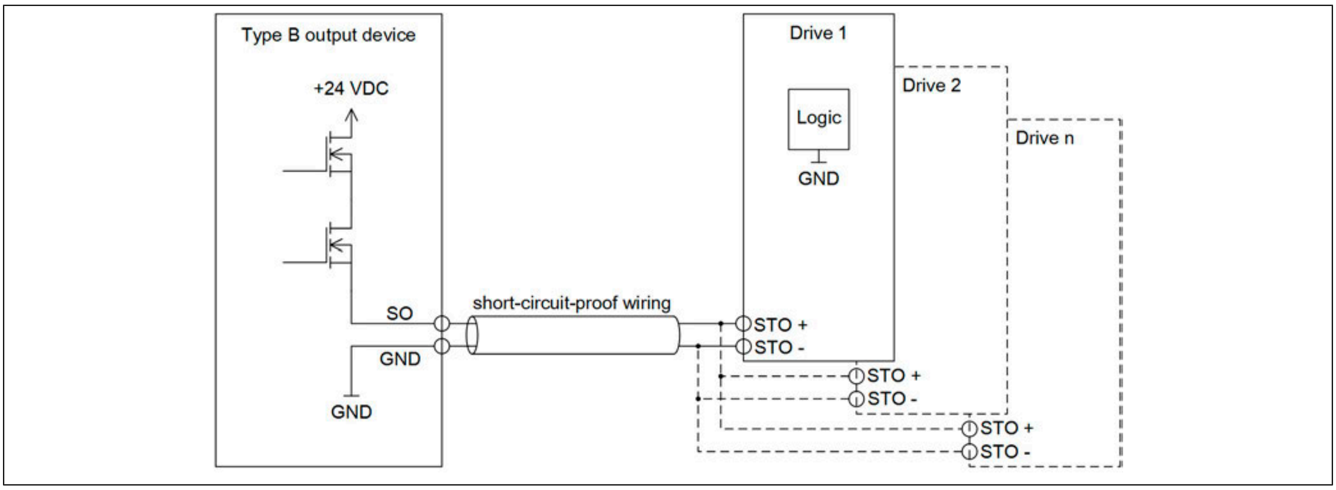


Figure 322: Drive system on type B safe output channel (high-side / high-side) with OSSD - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	No

Table 477: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

4.2.1.4 Drive system on type B safe output channel (high-side / high-side) without OSSD

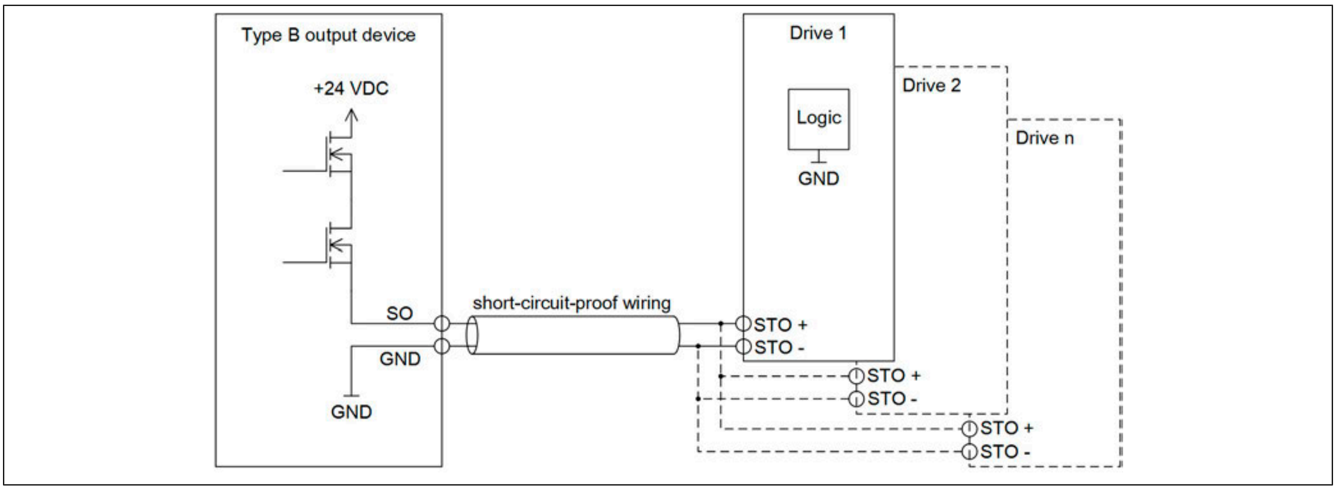


Figure 323: STO for drive system on type B safe output channel (high-side / high-side) without OSSD - Circuit example

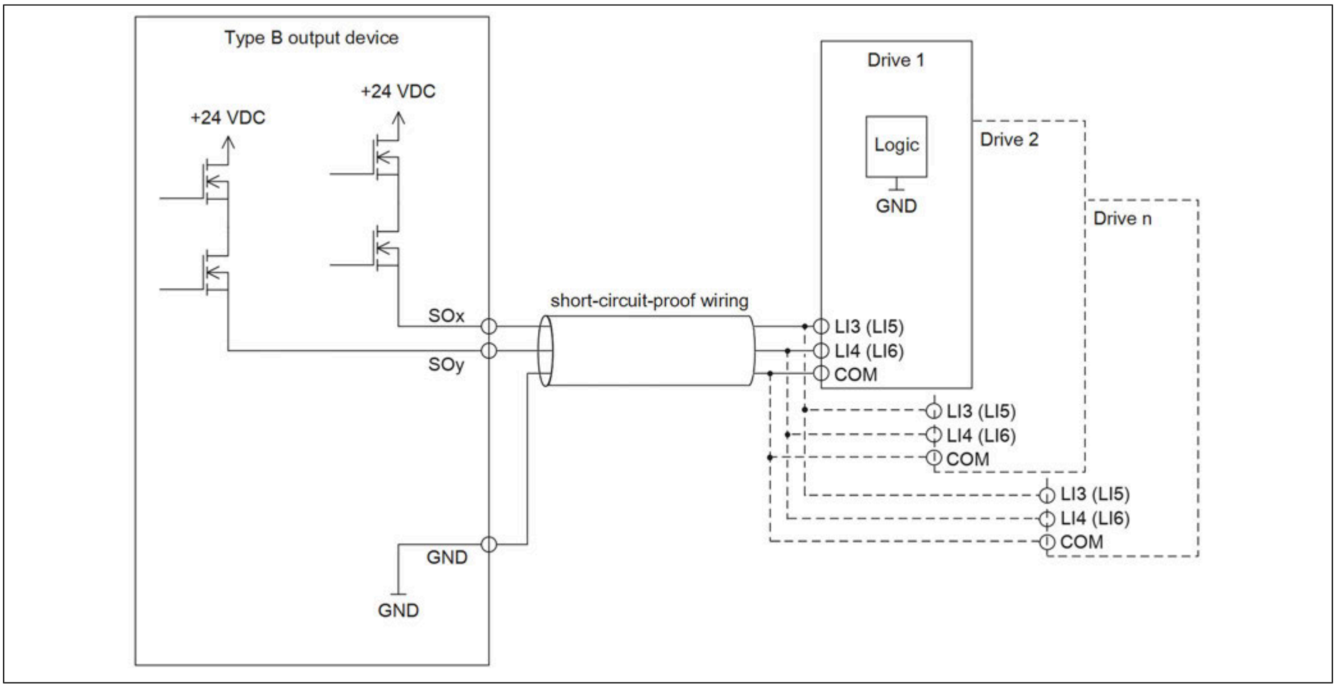


Figure 324: SLS for ACOPOSinverter P74 on type B safe output channel (high-side / high-side) without OSSD - Circuit example

These circuits meet the requirements of EN ISO 13849-1:2015 CAT 3.

When using these circuits, the following parameters must be taken into account in Automation Studio / SafeDESIGNER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	Yes-ATTENTION

Table 478: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements of SIL 3 in accordance with EN IEC 62061:2010, or PL e according to EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2, in accordance with EN IEC 62061:2010, or PL d according to EN ISO 13849-1:2015, an additional check of the safety function by the user may be necessary. For details, see the data sheet for the corresponding module.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

4.2.1.5 Drive system on safe relay output channel (high-side / high-side)

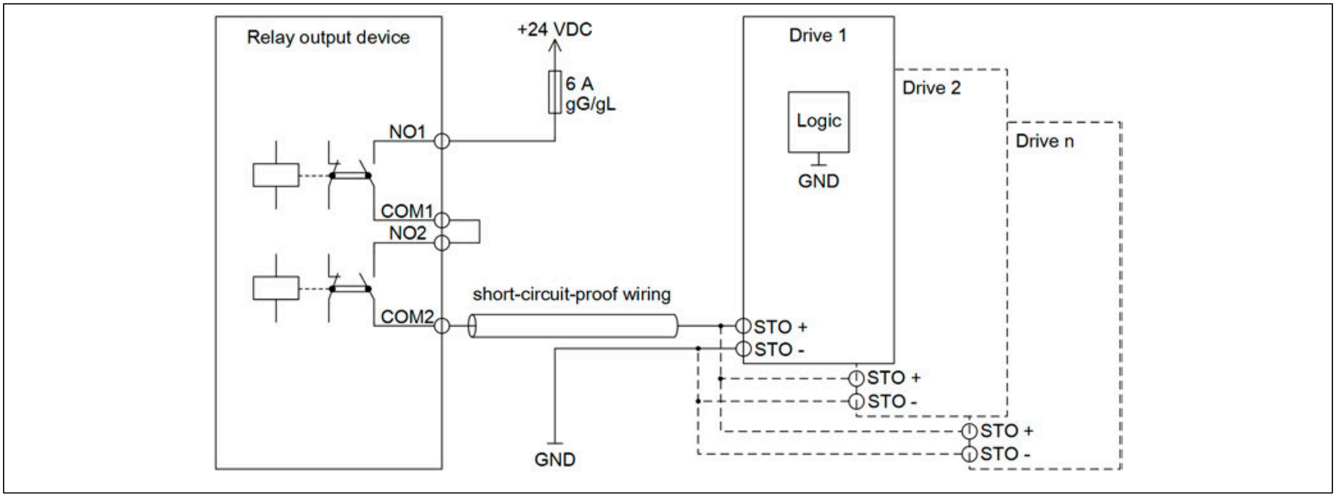


Figure 325: STO for drive system on safe relay output channel (high-side / high-side) - Circuit example

The STO connection meets the requirements of EN ISO 13849-1:2015 CAT 4.

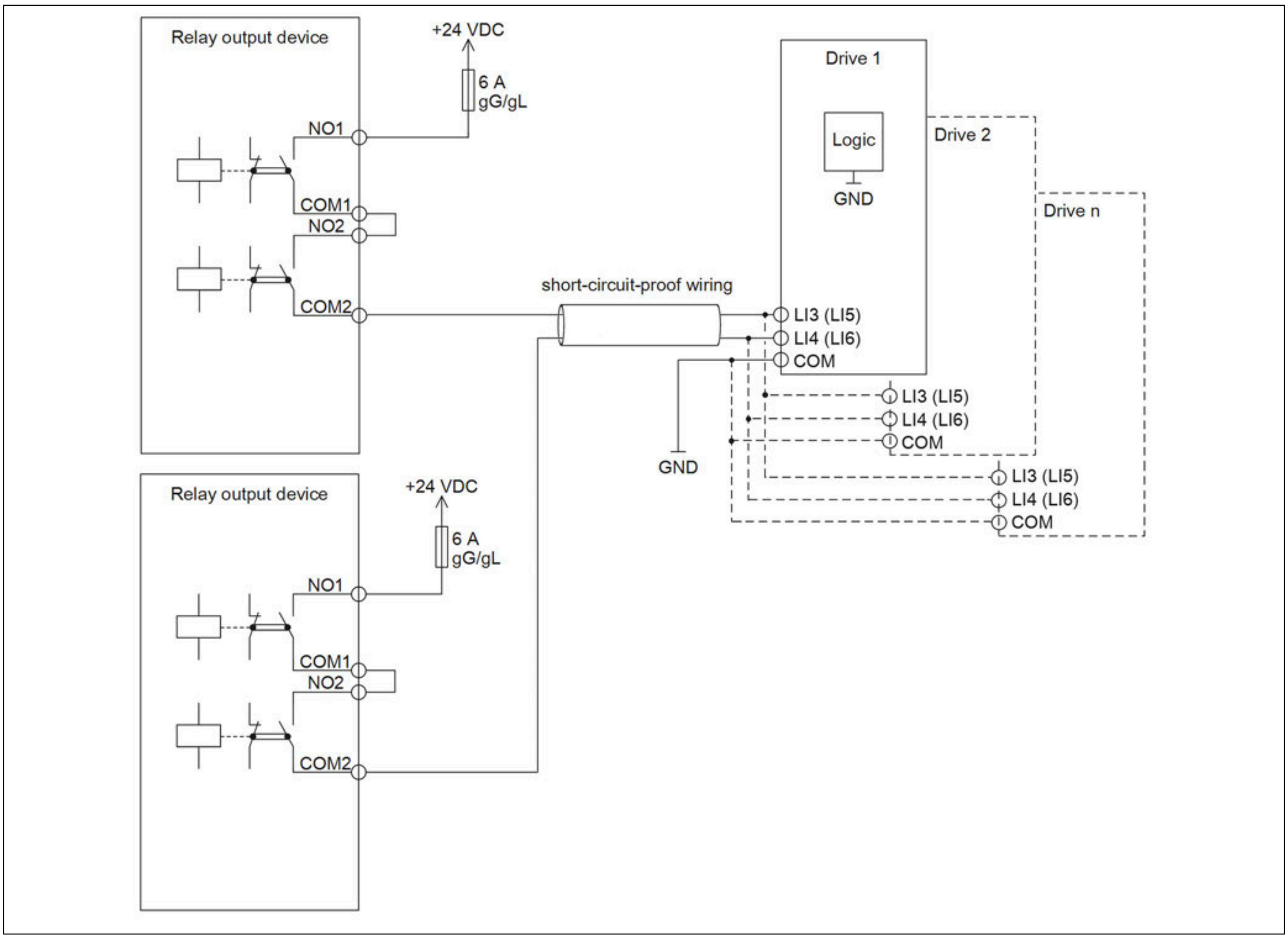


Figure 326: SLS for ACOPOSinverter P74 on safe relay output channel (high-side / high-side) - Circuit example

The SLS connection meets the requirements of EN ISO 13849-1:2015 CAT 3.

Danger!

Make sure that a proper protective circuit is used for the relay contacts (see technical data for the module). Also consider that operation outside of the specification is not permitted.

Operating outside of the specification or not using a protective circuit can cause the relay contacts to melt simultaneously, resulting in a loss of safety functionality.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Danger!

The user is responsible for ensuring that each relay channel is cut off at least 1x per week so that the appropriate internal tests can be completed.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

Danger!

A relay channel does not have error detection for wiring problems. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures or by the connected device.

4.2.1.6 Drive system on safe relay output channel (high-side / low-side)

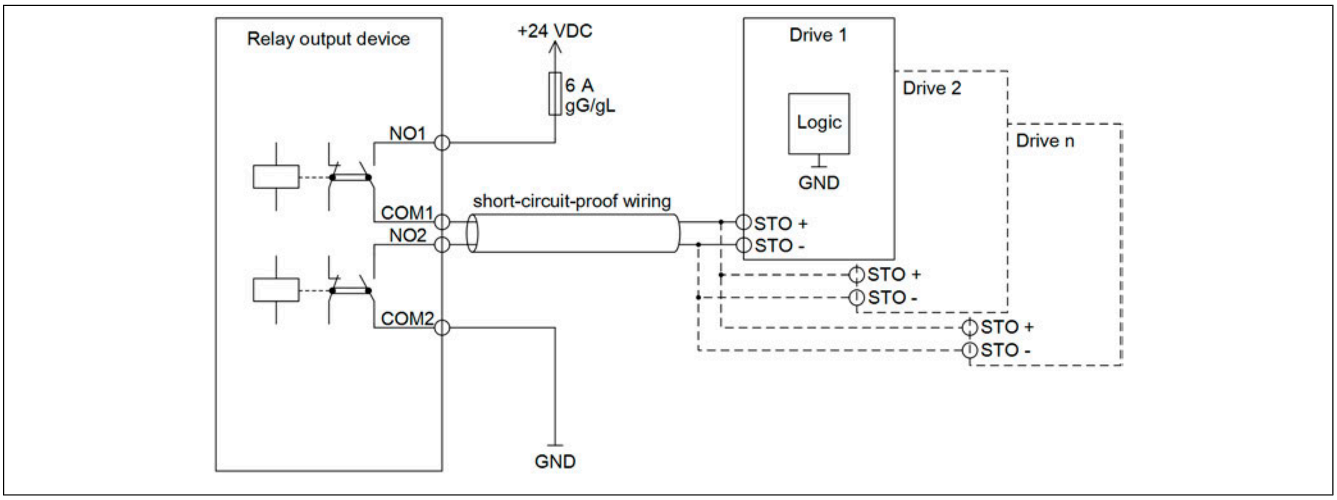


Figure 327: Drive system on safe relay output channel (high-side / low-side) - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

Danger!

This connection is only permitted if the GND-switching relay is not bypassed by a protective circuit in the drive (e.g. to ground).

Danger!

Make sure that a proper protective circuit is used for the relay contacts (see technical data for the module). Also consider that operation outside of the specification is not permitted.

Operating outside of the specification or not using a protective circuit can cause the relay contacts to melt simultaneously, resulting in a loss of safety functionality.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

Danger!

For applications above category 1 per EN ISO 13849-1:2015, the two relay contacts of both relays must be connected in series. In this type of application, the two relays must be controlled using signal "SafeDigitalOutputxxyy".

Controlling the two relay contacts using only the single signals "SafeDigitalOutputxx" is not permitted for applications above category 1 per EN ISO 13849-1:2015 since certain operating states can cause the two relay contacts to melt simultaneously in this case.

Danger!

The user is responsible for ensuring that each relay channel is cut off at least 1x per week so that the appropriate internal tests can be completed.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

Danger!

A relay channel does not have error detection for wiring problems. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures or by the connected device.

4.2.1.7 Drive system on safe relay output channel (high-side)

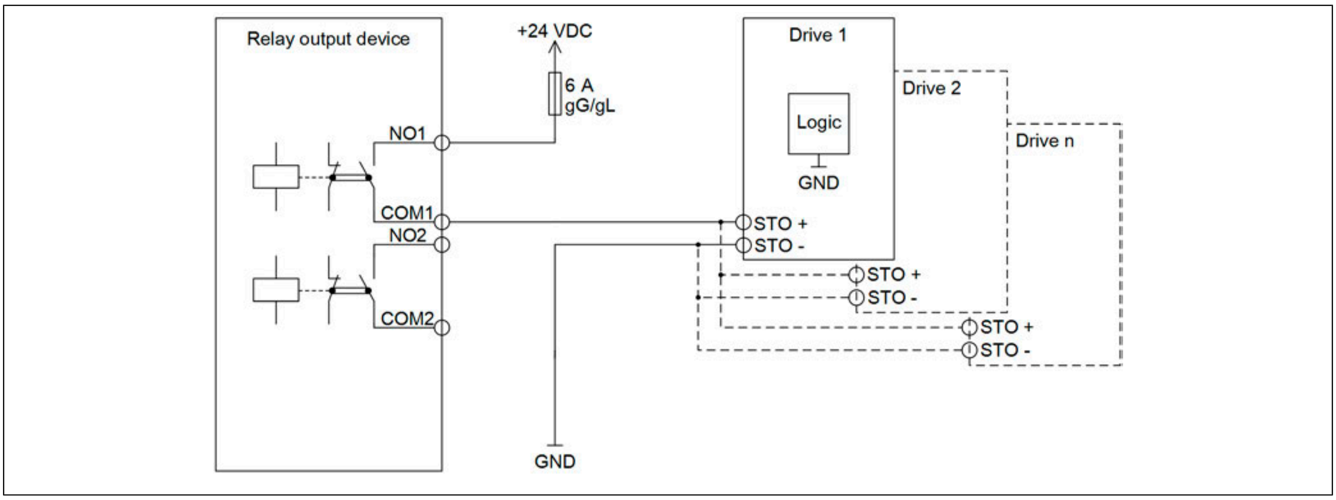


Figure 328: STO for drive system on safe relay output channel (high-side) - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 1.

Danger!

Make sure that a proper protective circuit is used for the relay contacts (see technical data for the module). Also consider that operation outside of the specification is not permitted.

Operating outside of the specification or not using a protective circuit can cause the relay contacts to melt simultaneously, resulting in a loss of safety functionality.

Danger!

The user is responsible for ensuring that each relay channel is cut off at least 1x per week so that the appropriate internal tests can be completed.

Danger!

A safety-related output channel is only permitted to be switched off for a maximum of 24 hours. The channel must be switched on by the end of this period so that the module's internal channel test can be performed.

Danger!

A relay channel does not have error detection for wiring problems. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures or by the connected device.

4.2.1.8 ACOPOSmicro on X20SO6300, X20SC0806 and X20SC0402 with OSSD

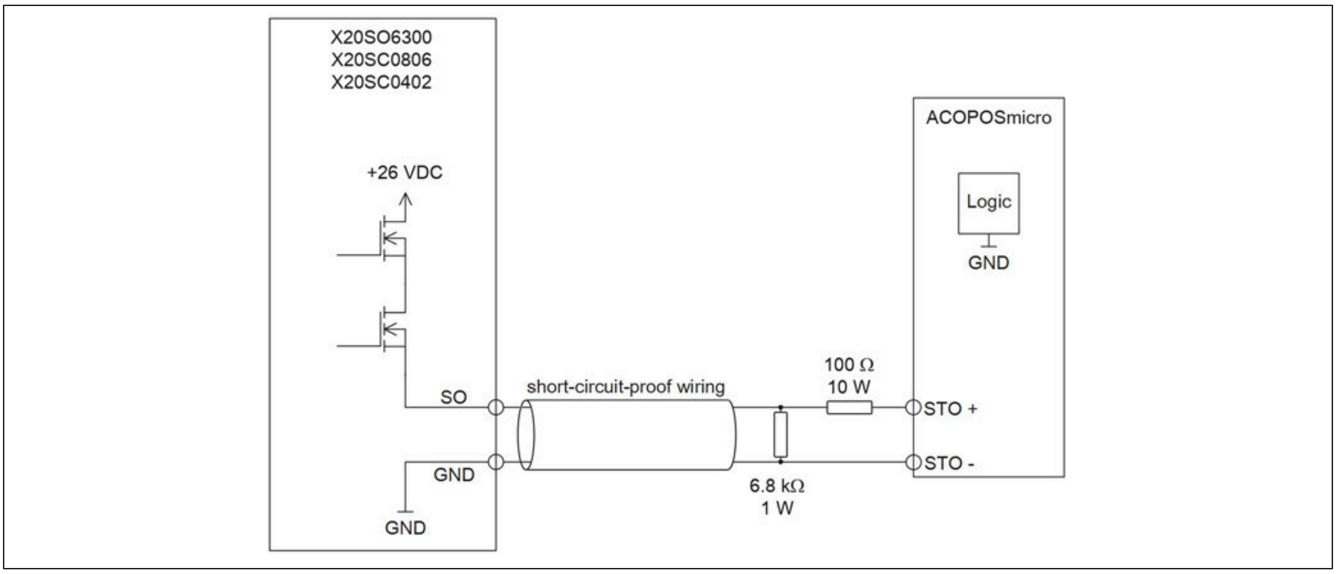


Figure 329: ACOPOSmicro on X20SO6300, X20SC0806 and X20SC0402 with OSSD - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 3.

The output module must be supplied with at least 26 VDC.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGNER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	No

Table 479: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

4.2.1.9 ACOPOSmicro on X20SO6300, X20SC0806 and X20SC0402 without OSSD

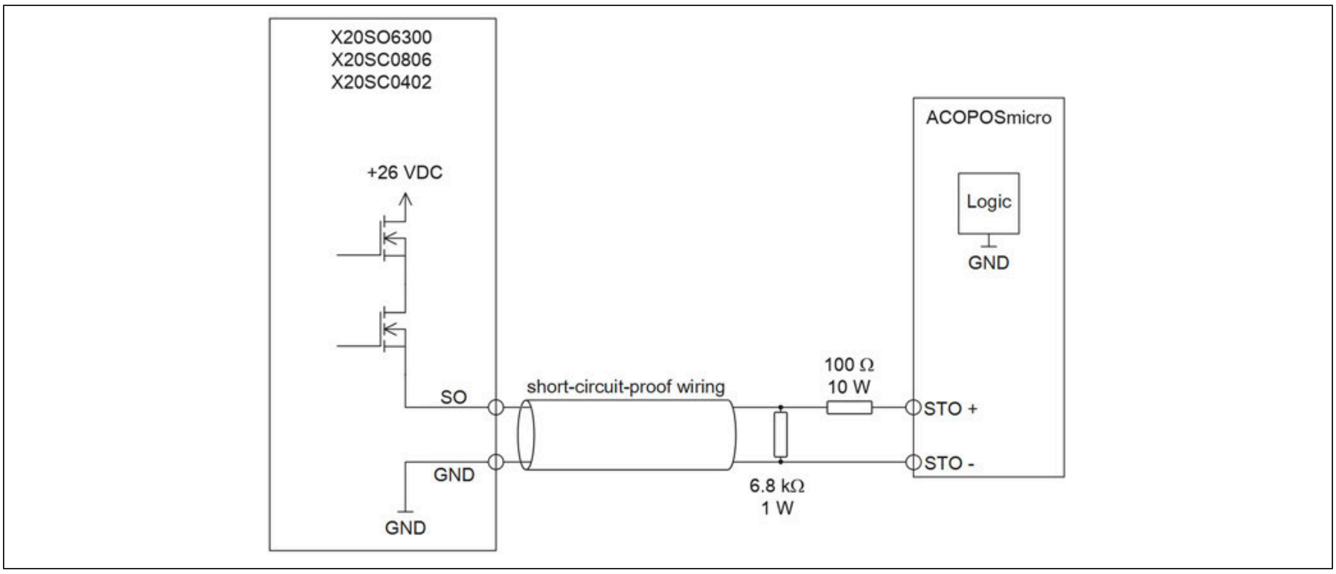


Figure 330: ACOPOSmicro on X20SO6300, X20SC0806 and X20SC0402 without OSSD - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 3.

The output module must be supplied with at least 26 VDC.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	Yes-ATTENTION

Table 480: Parameters in Automation Studio / SafeDESIGNER for the safe output channel

Danger!

With "Disable OSSD = Yes-ATTENTION", the module has reduced error detection capabilities and no longer meets the requirements of SIL 3 in accordance with EN IEC 62061:2010, or PL e according to EN ISO 13849-1:2015.

In order to meet the requirements for applications up to SIL 2, in accordance with EN IEC 62061:2010, or PL d according to EN ISO 13849-1:2015, an additional check of the safety function by the user may be necessary. For details, see the data sheet for the corresponding module.

Danger!

To prevent possible faults caused by short circuits to other voltage levels, wiring that protects against short circuits is needed for the actuator connection. The measures referenced in EN ISO 13849-2:2012, appendix D.2.4, table D.4 must be selected.

4.2.2 Tested products

4.2.2.1 B&R

Danger!

In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.

ACOPOS servo drives

Model number	Compatibility class	Category
8V1010.*	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 3, PLd.
8V1016.*	C2 - Compatible, construction identical to a C1 device	
8V1022.*		
8V1045.*		
8V1064.*		
8V1090.*		
8V1180.*		
8V128M.*		
8V1320.*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 13	Without OSSD: see Fig. 323.	CAT 3	SO → STO+ GND → STO-	= =	SOx → X1.Enable x ... 1, 2 GND → X1.COM (8, 9)
X20SO2110 X20SO4110		Max. 13	Without OSSD: see Fig. 321.	CAT 3	SO+ → STO+ SO- → STO-	= =	SOx+ → X1.Enable x ... 1, 2, 3, 4 SOx- → X1.COM (8, 9) x ... 1, 2, 3, 4
X20SO2120 X20SO4120 X20SC0842		Max. 20	Without OSSD: see Fig. 321.	CAT 3	SO+ → STO+ SO- → STO-	= =	SOx+ → X1.Enable x ... 1, 2, 3, 4 SOx- → X1.COM (8, 9) x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402		Max. 5	Without OSSD: see Fig. 323.	CAT 3	SO → STO+ GND → STO-	= =	SOx → X1.Enable x ... 1, 2, 3, 4, 5, 6 GND → X1.COM (8, 9)
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 3	see Fig. 325.	CAT 3	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → X1.Enable GND (ext. supply) → X1.COM (8, 9)
			see Fig. 327.	CAT 3	COM1 → STO+ NO2 → STO-	= =	COM1 → X1.Enable NO2 → X1.COM (8, 9)
			see Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → X1.Enable GND (ext. supply) → X1.COM (8, 9)
X67SC4122.L12	Not compatible						

ACOPOSmulti inverter modules

Model number	Compatibility class	Category
8BVI0014H*	C2 - Compatible, construction identical to a C1 device	The product is suitable for applications up to CAT 4, PLe.
8BVI0028H*		
8BVI0055H*	C1 - Compatible and physically tested	
8BVI0110H*	C2 - Compatible, construction identical to a C1 device	
8BVI0220H*		
8BVI0330H*		
8BVI0440H*		
8BVI0660H*		
8BVI0880H*		
8BVI1650H*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel [1]	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example: Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 13	With OSSD: see Fig. 322. Without OSSD: see Fig. 323.	CAT 4 CAT 3	SO → STO+ GND → STO-	= =	SOx → X1.Enable1 SOx → X1.Enable2 x ... 1, 2 GND → X1.COM (1) GND → X1.COM (3)
X20SO2110 X20SO4110		Max. 13	With OSSD: see Fig. 320. Without OSSD: see Fig. 321.	CAT 4 CAT 3	SO+ → STO+ SO- → STO-	= =	SOx+ → X1.Enable1 SOx+ → X1.Enable2 x ... 1, 2, 3, 4 SOx- → X1.COM (1) SOx- → X1.COM (3) x ... 1, 2, 3, 4
X20SO2120 X20SO4120 X20SC0842		Max. 20	With OSSD: see Fig. 320. Without OSSD: see Fig. 321.	CAT 4 CAT 3	SO+ → STO+ SO- → STO-	= =	SOx+ → X1.Enable1 SOx+ → X1.Enable2 x ... 1, 2, 3, 4 SOx- → X1.COM (1) SOx- → X1.COM (3) x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402		Max. 2	With OSSD: see Fig. 322. Without OSSD: see Fig. 323.	CAT 4 CAT 3	SO → STO+ GND → STO-	= =	SOx → X1.Enable1 SOx → X1.Enable2 x ... 1, 2, 3, 4, 5, 6 GND → X1.COM (1) GND → X1.COM (3)
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 18	see Fig. 325.	CAT 4	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → X1.Enable1 COM2 → X1.Enable2 GND (ext. supply) → X1.COM (1) GND (ext. supply) → X1.COM (3)
			see Fig. 327.	CAT 4	COM1 → STO+ NO2 → STO-	= =	COM1 → X1.Enable1 COM1 → X1.Enable2 NO2 → X1.COM (1) NO2 → X1.COM (3)
			see Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → X1.Enable1 COM1 → X1.Enable2 GND (ext. supply) → X1.COM (1) GND (ext. supply) → X1.COM (3)
X67SC4122.L12	See table in circuit example.	Max. 10	With OSSD: see Fig. 322. Without OSSD see Fig. 323.	CAT 4 CAT 3	SO → STO+ GND → STO-	= =	SOx → X1.Enable1 SOx → X1.Enable2 x ... 1, 2, 3, 4 GND → X1.COM (1) GND → X1.COM (3)

1) The number of enable inputs refers to individual enable inputs (X1.Enable1) and not to enable pairs (X1.Enable1 & X1.Enable2).

ACOPOSremote inverter modules

Model number	Compatibility class	Category
8CVI045E*	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
8CVI045H*	C2 - Compatible, construction identical to a C1 device	
8CVI045S*		
8CVI088E*		
8CVI088H*		
8CVI088S*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel [1]	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 13	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)
X20SO2110 X20SO2120 X20SO4110 X20SO4120		Max. 7	With OSSD: see Fig. 320.	CAT 4	SO+ → STO+	=	SOx+ → X3A.Enable1 SOx+ → X3A.Enable2 x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321.	CAT 3	SO- → STO-	=	SOx- → X3A.COM (5) SOx- → X3A.COM (2) x ... 1, 2, 3, 4
X20SC0842		Max. 20	With OSSD: see Fig. 320.	CAT 4	SO+ → STO+	=	SOx+ → X3A.Enable1 SOx+ → X3A.Enable2 x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321.	CAT 3	SO- → STO-	=	SOx- → X3A.COM (5) SOx- → X3A.COM (2) x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402		Max. 2	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2, 3, 4, 5, 6
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 18	see Fig. 325.	CAT 4	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → X3A.Enable1 COM2 → X3A.Enable2 GND (ext. supply) → X3A.COM (5) GND (ext. supply) → X3A.COM (2)
			see Fig. 327.	CAT 4	COM1 → STO+ NO2 → STO-	= =	COM1 → X3A.Enable1 COM1 → X3A.Enable2 NO2 → X3A.COM (5) NO2 → X3A.COM (2)
			see Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → X3A.Enable1 COM1 → X3A.Enable2 GND (ext. supply) → X3A.COM (5) GND (ext. supply) → X3A.COM (2)
X67SC4122.L12	See table in circuit example.	Max. 10	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2, 3, 4
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)

1) The number of enable inputs refers to individual enable inputs (X3A.Enable1) and not to enable pairs (X3A.Enable1 & X3A.Enable2).

ACOPOSmotor

Model number	Compatibility class	Category
8DI440.D50227100-1	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
8DI330.xxxxxxx0x-1	C2 - Compatible, construction identical to a C1 device	
8DI340.xxxxxxx0x-1		
8DI440.xxxxxxx0x-1		
8DI450.xxxxxxx0x-1		
8DI460.xxxxxxx0x-1		
8DI540.xxxxxxx0x-1		
8DI550.xxxxxxx0x-1		
8DI560.xxxxxxx0x-1		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel 1)	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example: Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 8	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)
X20SO2110 X20SO4110		Max. 8	With OSSD: see Fig. 320.	CAT 4	SO+ → STO+	=	SOx+ → X3A.Enable1 SOx+ → X3A.Enable2 x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321.	CAT 3	SO- → STO-	=	SOx- → X3A.COM (5) SOx- → X3A.COM (2) x ... 1, 2, 3, 4
X20SO2120 X20SO4120 X20SC0842		Max. 20	With OSSD: see Fig. 320.	CAT 4			
			Without OSSD: see Fig. 321.	CAT 3			
X20SO6300 X20SC0806 X20SC0402		Max. 2	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2, 3, 4, 5, 6
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 20	see Fig. 325.	CAT 4	COM2 → STO+	=	COM2 → X3A.Enable1 COM2 → X3A.Enable2
					GND (ext. supply) → STO-	=	GND (ext. supply) → X3A.COM (5) GND (ext. supply) → X3A.COM (2)
			see Fig. 327.	CAT 4	COM1 → STO+	=	COM1 → X3A.Enable1 COM1 → X3A.Enable2
					NO2 → STO-	=	NO2 → X3A.COM (5) NO2 → X3A.COM (2)
			see Fig. 328.	CAT 1	COM1 → STO+	=	COM1 → X3A.Enable1 COM1 → X3A.Enable2
					GND (ext. supply) → STO-	=	GND (ext. supply) → X3A.COM (5) GND (ext. supply) → X3A.COM (2)
X67SC4122.L12	See table in circuit example.	Max. 17	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X3A.Enable1 SOx → X3A.Enable2 x ... 1, 2, 3, 4
			Without OSSD see Fig. 323.	CAT 3	GND → STO-	=	GND → X3A.COM (5) GND → X3A.COM (2)

1) The number of enable inputs refers to individual enable inputs (X3A.Enable1) and not to enable pairs (X3A.Enable1 & X3A.Enable2).

ACOPOSmicro

Model number	Compatibility class	Category
80VD100PD.C022-01	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 3, PLd.
80VD100PD.*	C2 - Compatible, construction identical to a C1 device	
80VD100PS.*		
80SD100XD.*		
80SD100XS.*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	**=**	**Product label**
X20SC2212	See table in circuit example.	Max. 1	With OSSD: see Fig. 322	CAT 3	SO → STO+	=	SOx → X2.24 V Enable x ... 1, 2
			Without OSSD: see Fig. 323	CAT 3	GND → STO-	=	GND → X2.COM Enable
X20SO2110 X20SO2120 X20SO4110 X20SO4120	X2X cycle time: 700 µs < X2X cycle < 800 µs, 1400 µs < X2X cycle < 1600 µs, 2800 µs < X2X cycle < 3200 µs; See table in circuit example.	Max. 1	With OSSD: see Fig. 320	CAT 3	SO+ → STO+	=	SOx+ → X2.24 V Enable x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321	CAT 3	SO- → STO-	=	SOx- → X2.COM Enable x ... 1, 2, 3, 4
X20SC0842	See table in circuit example.	Max. 1	With OSSD: see Fig. 320	CAT 3	SO+ → STO+	=	SOx+ → X2.24 V Enable x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321	CAT 3	SO- → STO-	=	SOx- → X2.COM Enable x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402		Max. 1	With OSSD: see Fig. 329	CAT 3	SO → 100 Ω; 10 W → STO+	=	SOx → 100 Ω; 10 W → X2.24 V Enable x ... 1, 2, 3, 4, 5, 6
			Without OSSD: see Fig. 330	CAT 3	GND → STO-	=	GND → X2.COM Enable
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 2	see Fig. 325	CAT 3	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → X2.24 V Enable GND (ext. supply) → X2.COM Enable
			see Fig. 327	CAT 3	COM1 → STO+ NO2 → STO-	= =	COM1 → X2.24 V Enable NO2 → X2.COM Enable
			see Fig. 328	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → X2.24 V Enable GND (ext. supply) → X2.COM Enable
X67SC4122.L12	See table in circuit example.	Max. 1	With OSSD: see Fig. 322	CAT 3	SO → STO+	=	SOx → X2.24 V Enable x ... 1, 2, 3, 4
			Without OSSD: see Fig. 323	CAT 3	GND → STO-	=	GND → X2.COM Enable

ACOPOSinverter - STO function

Model number	Compatibility class	Category
8I74T400037.01P-1	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 3, PLd.
8I84T400075.01P-1		
8I74T40*	C2 - Compatible, construction identical to a C1 device	
8I74S20*		
8I84T40*		
8I84T20*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 1	Without OSSD: see Fig. 323.	CAT 3	SO → STO+ GND → STO-	= =	SOx → STO x ... 1, 2 GND → COM
X20SO2110 X20SO2120 X20SO4110 X20SO4120 X20SC0842	Not compatible						
X20SO6300 X20SC0806 X20SC0402	See table in circuit example.	Max. 1	With OSSD: see Fig. 322.	CAT 3	SO → STO+ GND → STO-	= =	SOx → STO x ... 1, 2, 3, 4, 5, 6 GND → COM
			Without OSSD: see Fig. 323.	CAT 3			
X20SC0842		Max. 1	Without OSSD: see Fig. 323.	CAT 3	SO → STO+ GND → STO-	= =	SOx → STO x ... 5, 6 GND → COM
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 1	see Fig. 325.	CAT 3	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → STO GND (ext. supply) → COM
			see Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → STO GND (ext. supply) → COM
X67SC4122.L12	Not compatible						

ACOPOSinverter - SLS function

Model number	Compatibility class	Category
8I74T400037.01P-1	C1 - Compatible and physically tested	When using the SLS function the product is suitable for applications up to CAT 3, PLd.
8I74T40*	C2 - Compatible, construction identical to a C1 device	
8I74S20*		

Model number of compatible products	Compatibility class	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel		Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
			LI3 / LI4	LI5 / LI6			Figure label	=	Product label
X20SC2212	C1	See table in circuit example.	Max. 20	Max. 20	Without OSSD: see Fig. 324.	CAT 3	SOx → LI3 (LI5) SOy → LI4 (LI6) GND → COM	= = =	SO1 → LI3 (LI5) SO2 → LI4 (LI6) GND → COM
X20SO2110 X20SO2120 X20SO4110 X20SO4120	Not compatible								
X20SO6300	C1	See table in circuit example.	Max. 18	0	Without OSSD: see Fig. 324.	CAT 3	SOx → LI3 (LI5) SOy → LI4 (LI6) GND → COM	= = =	SOx → LI3 (LI5) x ... 1, 2, 3, 4, 5, 6 SOy → LI4 (LI6) y ... 1, 2, 3, 4, 5, 6 **Note:** y must be different than x GND → COM
X20SC0806 X20SC0402	C2								
X20SC0842	C1		Max. 4	Max. 2	Without OSSD: see Fig. 324.	CAT 3	SOx → LI3 (LI5) SOy → LI4 (LI6) GND → COM	= = =	SO5 → LI3 (LI5) SO6 → LI4 (LI6) GND → COM
X20SP1130	Not compatible								
X20SC2432	C1	-	Max. 20	Max. 16	see Fig. 326.	CAT 3	COM2 (device 1) → LI3 (LI5) COM2 (device 2) → LI4 (LI6) GND (ext. supply) → COM	= = =	COM2 (device 1) → LI3 (LI5) COM2 (device 2) → LI4 (LI6) GND (ext. supply) → COM
X20SO2530	C2								
X67SC4122.L12	C1	See table in circuit example.	Max. 20	Max. 20	Without OSSD: see Fig. 324.	CAT 3	SOx → LI3 (LI5) SOy → LI4 (LI6) GND → COM	= = =	SOx → LI3 (LI5) x ... 1, 2, 3, 4 SOy → LI4 (LI6) y ... 1, 2, 3, 4 **Note:** y must be different than x GND → COM

ACOPOS P3

Model number	Compatibility class	Category
8EI8X8HWS10.0400-1	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
8EI1X6*	C2 - Compatible, construction identical to a C1 device	
8EI2X2*		
8EI4X5*		
8EI8X8*		

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel [1]	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 20	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X7.Enable1 SOx → X7.Enable2 x ... 1, 2
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X7.COM (1) GND → X7.COM (2)
X20SO2110 X20SO2120 X20SO4110 X20SO4120 X20SC0842		Max. 20	With OSSD: see Fig. 320.	CAT 4	SO+ → STO+	=	SOx+ → X7.Enable1 SOx+ → X7.Enable2 x ... 1, 2, 3, 4
			Without OSSD: see Fig. 321.	CAT 3	SO- → STO-	=	SOx- → X7.COM (1) SOx- → X7.COM (2) x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402		Max. 16	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X7.Enable1 SOx → X7.Enable2 x ... 1, 2, 3, 4, 5, 6
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X7.COM (1) GND → X7.COM (2)
X20SC0842		Max. 4	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X7.Enable1 SOx → X7.Enable2 x ... 5, 6
			Without OSSD: see Fig. 323.	CAT 3	GND → STO-	=	GND → X7.COM (1) GND → X7.COM (2)
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 20	see Fig. 325.	CAT 4	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → X7.Enable1 COM2 → X7.Enable2 GND (ext. supply) → X7.COM (1) GND (ext. supply) → X7.COM (2)
			see Fig. 327.	CAT 4	COM1 → STO+ NO2 → STO-	= =	COM1 → X7.Enable1 COM1 → X7.Enable2 NO2 → X7.COM (1) NO2 → X7.COM (2)
			see Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → X7.Enable1 COM1 → X7.Enable2 GND (ext. supply) → X7.COM (1) GND (ext. supply) → X7.COM (2)
X67SC4122.L12	See table in circuit example.	Max. 20	With OSSD: see Fig. 322.	CAT 4	SO → STO+	=	SOx → X7.Enable1 SOx → X7.Enable2 x ... 1, 2, 3, 4
			Without OSSD see Fig. 323.	CAT 3	GND → STO-	=	GND → X7.COM (1) GND → X7.COM (2)

1) The number of enable inputs refers to individual enable inputs (X7.Enable1) and not to enable pairs (X7.Enable1 & X7.Enable2).

4.2.2.2 ABB

Danger!

In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.

ABB ACS880-01

Model number	Compatibility class	Category
ACS880-01-02A4-3	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 3, PLe.

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel [1]	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 2	With OSSD: See Fig. 322. Without OSSD: See Fig. 323.	CAT 3 CAT 3	SO → STO+ GND → STO-	= =	SOx → XSTO.IN1 SOx → XSTO.IN2 x ... 1, 2 GND → XSTO.SGND
X20SO2110 X20SO2120 X20SO4110 X20SO4120	Not compatible						
X20SO6300 X20SC0806 X20SC0402	See table in circuit example.	Max. 2	With OSSD: See Fig. 322. Without OSSD: See Fig. 323.	CAT 3 CAT 3	SO → STO+ GND → STO-	= =	SOx → XSTO.IN1 SOx → XSTO.IN2 x ... 1, 2, 3, 4, 5, 6 GND → XSTO.SGND
X20SC0842	Not compatible						
X20SP1130	Not compatible						
X20SC2432 X20SO2530	-	Max. 2	See Fig. 325.	CAT 3	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → XSTO.IN1 COM2 → XSTO.IN2 GND (ext. supply) → XSTO.SGND
			See Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → XSTO.IN1 COM1 → XSTO.IN2 GND (ext. supply) → XSTO.SGND
X67SC4122.L12	See table in circuit example.	Max. 2	With OSSD: See Fig. 322. Without OSSD: See Fig. 323.	CAT 3 CAT 3	SO → STO+ GND → STO-	= =	SOx → XSTO.IN1 SOx → XSTO.IN2 x ... 1, 2 GND → XSTO.SGND

1) The number of enable inputs refers to the individual enable inputs (XSTO.IN1) and not enable pairs (XSTO.IN1 & XSTO.IN2).

4.2.2.3 Phoenix

Danger!

In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.

Motor starter

Model number	Compatibility class	Category
ELR H3-IES-PT-24DC/500AC-2	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 3, PLe.

Compatibility for control supply voltage Us

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example: Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 2	Without OSSD: See Fig. 323.	CAT 3	SO → STO+ GND → STO-	= =	SOx → Us x ... 1, 2 GND → ⊥
X20SO2110 X20SO2120 X20SO4110 X20SO4120		Max. 4	Without OSSD: See Fig. 321.	CAT 3	SO+ → STO+ SO- → STO-	= =	SOx+ → Us x ... 1, 2, 3, 4 SOx- → ⊥ x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402 X20SLX806 X20SLX402	Not compatible						
X20SC0842 X20SLX842	See table in circuit example.	Max. 20	Without OSSD: See Fig. 321.	CAT 3	SO → STO+ GND → STO-	= =	SOx → Us x ... 1, 2, 3, 4 GND → ⊥
X20SP1130 X67SC4122.L12	Not compatible						
X20SC2432 X20SO2530	-	Max. 2	See Fig. 325.	CAT 3	COM2 → STO+ GND (ext. supply) → STO-	= =	COM2 → Us GND (ext. supply) → ⊥
			See Fig. 327.	CAT 3	COM1 → STO+ NO2 → STO-	= =	COM1 → Us NO2 → ⊥
			See Fig. 328.	CAT 1	COM1 → STO+ GND (ext. supply) → STO-	= =	COM1 → Us GND (ext. supply) → ⊥

Compatibility for control input ON

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Enable inputs per output channel	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC2212	See table in circuit example.	Max. 20	With OSSD: See Fig. 322.	CAT 3	SO → STO+	=	SOx → ON x ... 1, 2
			Without OSSD: See Fig. 323.	CAT 3	GND → STO-	=	GND → ⊥E
X20SO2110 X20SO2120 X20SO4110 X20SO4120 X20SC0842 X20SLX842		Max. 20	With OSSD: See Fig. 320.	CAT 3	SO+ → STO+	=	SOx+ → ON x ... 1, 2, 3, 4
			Without OSSD: See Fig. 321.	CAT 3	SO- → STO-	=	SOx- → ⊥E x ... 1, 2, 3, 4
X20SO6300 X20SC0806 X20SC0402 X20SLX806 X20SLX402		Max. 20	With OSSD: See Fig. 322.	CAT 3	SO → STO+	=	SOx → ON x ... 1, 2, 3, 4, 5, 6
			Without OSSD: See Fig. 323.	CAT 3	GND → STO-	=	GND → ⊥E
X20SC0842 X20SLX842		Max. 6	With OSSD: See Fig. 322.	CAT 3	SO → STO+	=	SOx → ON x ... 5, 6
			Without OSSD: See Fig. 323.	CAT 3	GND → STO-	=	GND → ⊥E
X20SP1130 X20SC2432 X20SO2530 X67SC4122.L12	Not compatible						

4.3 Connecting devices with OSSD signals

This section looks at connecting light grids, laser scanners, transponders, etc. to a safe input channel. Particular attention is given to the OSSD outputs on the device. Information about all other connections is included in the documentation for the respective product.

The following general circuit examples assume a 2-channel output on the product (i.e. 2 separate OSSD outputs).

4.3.1 Circuit example

4.3.1.1 OSSD signals on a safe input channel

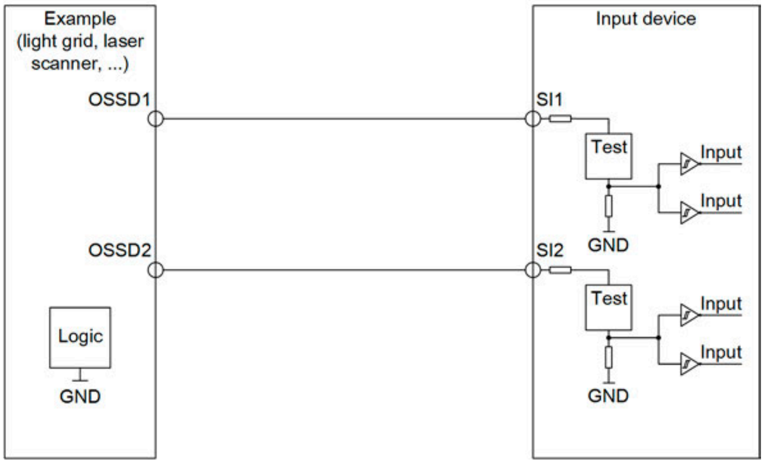


Figure 331: Circuit example - OSSD signals on a safe input channel

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Pulse Mode (SafeDigitalInput01)	See table under "Tested products"
Pulse Mode (SafeDigitalInput02)	
Filter Off (SafeDigitalInput01)	
Filter Off (SafeDigitalInput02)	
Discrepancy Time (SafeDigitalInput01)	

Table 481: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.3.2 Tested products - Light grids

4.3.2.1 Leuze

Danger!

In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.

MLC light grids

Model number	Compatibility class	Category
MLC100*	C2 - Compatible, construction identical to a C1 device	The product is suitable for applications up to CAT 4, PLe.
MLC310* (in combination with MLC300*)	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 2, PLc.
MLC320*	C2 - Compatible, construction identical to a C1 device	The product is suitable for applications up to CAT 2, PLc.
MLC510*		The product is suitable for applications up to CAT 4, PLe.
MLC511*		
MLC520*		
MLC530*		

<table>
<tr><th rowspan="2">Model number of compatible products</th><th rowspan="2">Parameters in Automation Studio / SafeDESIGNER</th><th rowspan="2">Circuit example</th><th rowspan="2">EN ISO 13849-1:2015 Category</th><th colspan="3">Terminal assignments for circuit example</th></tr>
<tr><th>Figure label</th><th>=</th><th>Product label</th></tr>
<tr><td>X20SC2212</td><td rowspan="10">SafeDigitalInputxx:
• Pulse Mode = No pulse
• Filter Off = 2400 µs
• Discrepancy Time = 2800 µs

SafeDigitalInput(xx+1):
• Filter Off = 2400 µs

xx ... 1, 3, 5, 7, ... 19</td><td rowspan="10">See Fig. 331.</td><td rowspan="10">CAT 2:
MLC3xx

CAT 4:
MLC100,
MLC5xx;</td><td rowspan="10">OSSD1 → SI1

OSSD2 → SI2</td><td rowspan="10">=

=</td><td rowspan="10">Receiver.OSSD1 → SIx
x ... 1, 3, 5, 7, ... 19

Receiver.OSSD2 → SIx
x ... 2, 4, 6, 8, ... 20</td></tr>
<tr><td>X20SC2432</td></tr>
<tr><td>X20SI2100</td></tr>
<tr><td>X20SI4100</td></tr>
<tr><td>X20SI9100</td></tr>
<tr><td>X20SLX210</td></tr>
<tr><td>X20SLX410</td></tr>
<tr><td>X20SLX910</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td>X67SI8103</td></tr>
</table>

MLD light grids

Model number	Compatibility class	Category
MLD310*	C2 - Compatible, construction identical to a C1 device	The product is suitable for applications up to CAT 3, PLd.
MLD320*		
MLD330*		
MLD335*		
MLD510* (in combination with MLD500*)	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
MLD520*	C2 - Compatible, construction identical to a C1 device	
MLD530*		
MLD535*		

<table>
<tr><th rowspan="2">Model number of compatible products</th><th rowspan="2">Compatibility class</th><th rowspan="2">Parameters in Automation Studio / SafeDESIGNER</th><th rowspan="2">Circuit example</th><th rowspan="2">EN ISO 13849-1:2015 Category</th><th colspan="3">Terminal assignments for circuit example</th></tr>
<tr><th>Figure label</th><th>=</th><th>Product label</th></tr>
<tr><td>X20SC0402</td><td rowspan="6">C2</td><td rowspan="18">SafeDigitalInputxx:
• Pulse Mode = No pulse
• Filter Off = 2200 µs
• Discrepancy Time = 2200 µs

SafeDigitalInput(xx+1):
• Pulse Mode = No pulse
• Filter Off = 2200 µs

xx ... 1, 3, 5, 7, ... 19</td><td rowspan="18">See Fig. 331.</td><td rowspan="18">CAT 3:
MLD3xx

CAT 4:
MLD5xx</td><td rowspan="18">OSSD1 → SI1

OSSD2 → SI2</td><td rowspan="18">=

=</td><td rowspan="18">Receiver.OSSD1 → SIx
x ... 1, 3, 5, 7, ... 19

Receiver.OSSD2 → SIx
x ... 2, 4, 6, 8, ... 20</td></tr>
<tr><td>X20SC0806</td></tr>
<tr><td>X20SC0842</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X20SC2432</td></tr>
<tr><td>X20SI2100</td></tr>
<tr><td>X20SI4100</td><td>C1</td></tr>
<tr><td>X20SI8110</td><td rowspan="11">C2</td></tr>
<tr><td>X20SI9100</td></tr>
<tr><td>X20SLX210</td></tr>
<tr><td>X20SLX402</td></tr>
<tr><td>X20SLX410</td></tr>
<tr><td>X20SLX806</td></tr>
<tr><td>X20SLX811</td></tr>
<tr><td>X20SLX842</td></tr>
<tr><td>X20SLX910</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td>X67SI8103</td></tr>
</table>

COMPACTplus light grids

<table>
<tr><th>Model number</th><th>Compatibility class</th><th>Category</th></tr>
<tr><td>CPRTxxx/Tx
(in combination with passive mirror PM2-500V)</td><td>C1 - Compatible and physically tested</td><td rowspan="2">The product is suitable for applications up to CAT 4, PLe.</td></tr>
<tr><td>CPRxx-xxxx/Tx</td><td>C2 - Compatible, construction identical to a C1 device</td></tr>
</table>

<table>
<tr><th rowspan="2">Model number of compatible products</th><th rowspan="2">Compatibility class</th><th rowspan="2">Parameters in Automation Studio / SafeDESIGNER</th><th rowspan="2">Circuit example</th><th rowspan="2">EN ISO 13849-1:2015 Category</th><th colspan="3">Terminal assignments for circuit example</th></tr>
<tr><th>Figure label</th><th>=</th><th>Product label</th></tr>
<tr><td>X20SC0402</td><td rowspan="6">C2</td><td rowspan="18">SafeDigitalInputxx:
• Pulse Mode = No pulse
• Filter Off = 2200 µs
• Discrepancy Time = 22000 µs

SafeDigitalInput(xx+1):
• Pulse Mode = No pulse
• Filter Off = 2200 µs

xx ... 1, 3, 5, 7, ... 19</td><td rowspan="18">See Fig. 331.</td><td rowspan="18">CAT 4</td><td rowspan="18">OSSD1 → SI1

OSSD2 → SI2</td><td rowspan="18">=

=</td><td rowspan="18">T1.OSSD1 output → SIx
or T2.OSSD1 output → SIx
or T3.OSSD1 output → SIx
or T4.OSSD1 output → SIx
x ... 1, 3, 5, 7, ... 19

T1.OSSD2 output → SIx
or T2.OSSD2 output → SIx
or T3.OSSD2 output → SIx
or T4.OSSD2 output → SIx
x ... 2, 4, 6, 8, ... 20</td></tr>
<tr><td>X20SC0806</td></tr>
<tr><td>X20SC0842</td></tr>
<tr><td>X20SC2212</td></tr>
<tr><td>X20SC2432</td></tr>
<tr><td>X20SI2100</td></tr>
<tr><td>X20SI4100</td><td>C1</td></tr>
<tr><td>X20SI8110</td><td rowspan="11">C2</td></tr>
<tr><td>X20SI9100</td></tr>
<tr><td>X20SLX210</td></tr>
<tr><td>X20SLX402</td></tr>
<tr><td>X20SLX410</td></tr>
<tr><td>X20SLX806</td></tr>
<tr><td>X20SLX811</td></tr>
<tr><td>X20SLX842</td></tr>
<tr><td>X20SLX910</td></tr>
<tr><td>X67SC4122.L12</td></tr>
<tr><td>X67SI8103</td></tr>
</table>

4.3.3 Tested products - Laser scanners

4.3.3.1 Leuze

> **Danger!**
>
> **In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.**

RS4 laser scanners

Model number	Compatibility class	Category
RS4-2E	C2 - Compatible, construction identical to a C1 device	The product is suitable for applications up to CAT 3, PLd.
RS4-2M		
RS4-4		
RS4-4E		
RS4-4M	C1 - Compatible and physically tested	
RS4-6E	C2 - Compatible, construction identical to a C1 device	
RS4-6M		

Model number of compatible products	Compatibility class	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC0402	C2	**SafeDigitalInputxx:** • Pulse Mode = No pulse • Filter Off = 2000 µs • Discrepancy Time = 2000 µs **SafeDigitalInput(xx+1):** • Pulse Mode = No pulse • Filter Off = 2000 µs xx ... 1, 3, 5, 7, ... 19	See Fig. 331.	CAT 3	OSSD1 → SI1 OSSD2 → SI2	= =	X1.OSSD1 → SIx x ... 1, 3, 5, 7 ... 19 X1.OSSD2 → SIx x ... 2, 4, 6, 8 ... 20
X20SC0806							
X20SC0842							
X20SC2212							
X20SC2432							
X20SI2100							
X20SI4100	C1						
X20SI8110	C2						
X20SI9100							
X20SLX210							
X20SLX402							
X20SLX410							
X20SLX806							
X20SLX811							
X20SLX842							
X20SLX910							
X67SC4122.L12							
X67SI8103							

4.3.4 Tested products - Transponders

4.3.4.1 Leuze

> **Danger!**
>
> **In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.**

RD800 transponders

Model number	Compatibility class	Category
RD800* (in combination with RD800* receiver)	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.

Model number of compatible products	Compatibility class	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC0402	C2	**SafeDigitalInputxx:** • Pulse Mode = No pulse • Filter Off = 2000 µs • Discrepancy Time = 2000 µs **SafeDigitalInput(xx+1):** • Pulse Mode = No pulse • Filter Off = 2000 µs xx ... 1, 3, 5, 7, ... 19	See Fig. 331.	CAT 4	OSSD1 → SI1 OSSD2 → SI2	= =	M12-Connector.OS1 → SIx x ... 1, 3, 5, 7 ... 19 M12-Connector.OS2 → SIx x ... 2, 4, 6, 8 ... 20
X20SC0806							
X20SC0842							
X20SC2212							
X20SC2432							
X20SI2100							
X20SI4100	C1						
X20SI8110	C2						
X20SI9100							
X20SLX210							
X20SLX402							
X20SLX410							
X20SLX806							
X20SLX811							
X20SLX842							
X20SLX910							
X67SC4122.L12							
X67SI8103							

4.4 Connecting devices without OSSD signals

This section looks at connecting devices without OSSD signals to safe input channels.

4.4.1 Circuit examples

4.4.1.1 OSSD signals via light grid on a safe input channel pair

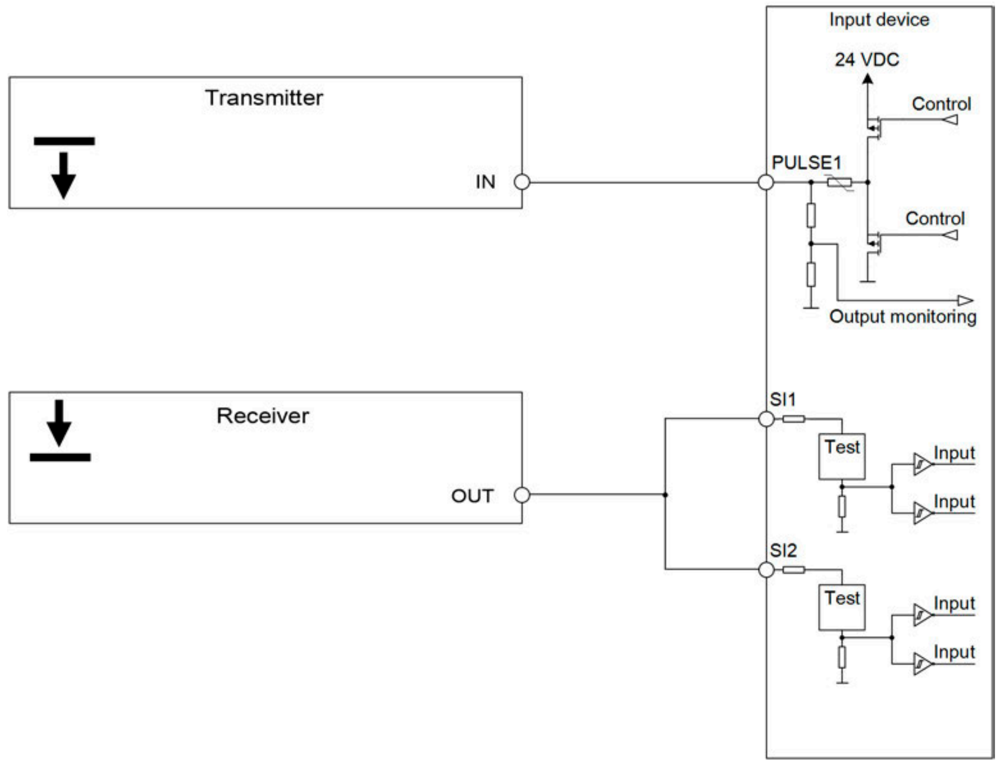


Figure 332: OSSD signals via light grid on a safe input channel pair - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

The lines from the input module to the transmitter (pulse 1 → IN) or from the receiver to the input module (OUT → SI1 + SI2) must be isolated in separate plastic-sheathed cables.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Filter Off (SafeDigitalInput01)	See table under "Tested products"
Filter Off (SafeDigitalInput02)	

Table 482: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.4.1.2 OSSD signals via light grid on a safe input channel

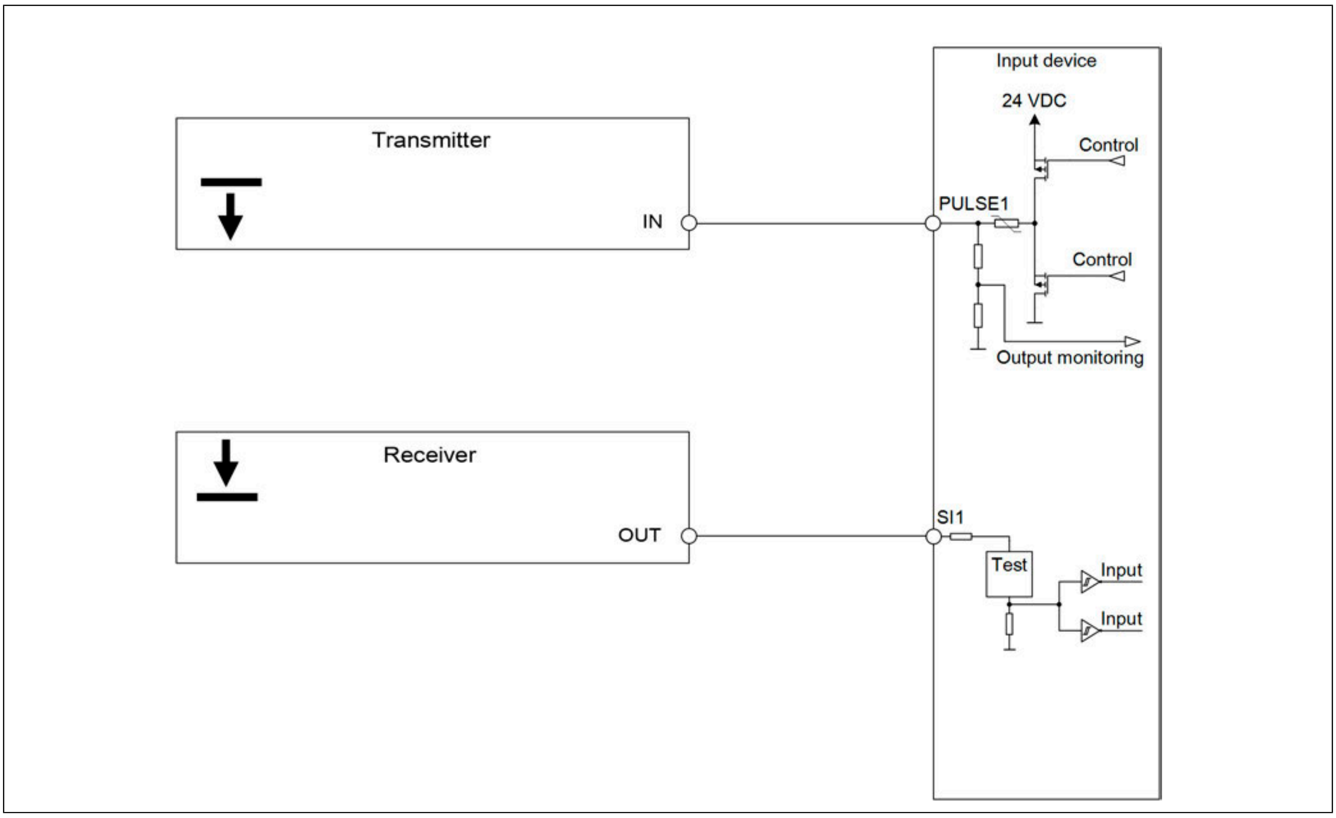


Figure 333: OSSD signals via light grid on a safe input channel - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 3.

The lines from the input module to the transmitter (pulse 1 → IN) or from the receiver to the input module (OUT → SI1) must be isolated in separate plastic-sheathed cables.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Filter Off (SafeDigitalInput01)	See table under "Tested products"

Table 483: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.4.1.3 OSSD signals via cascaded light grid on a safe input channel pair

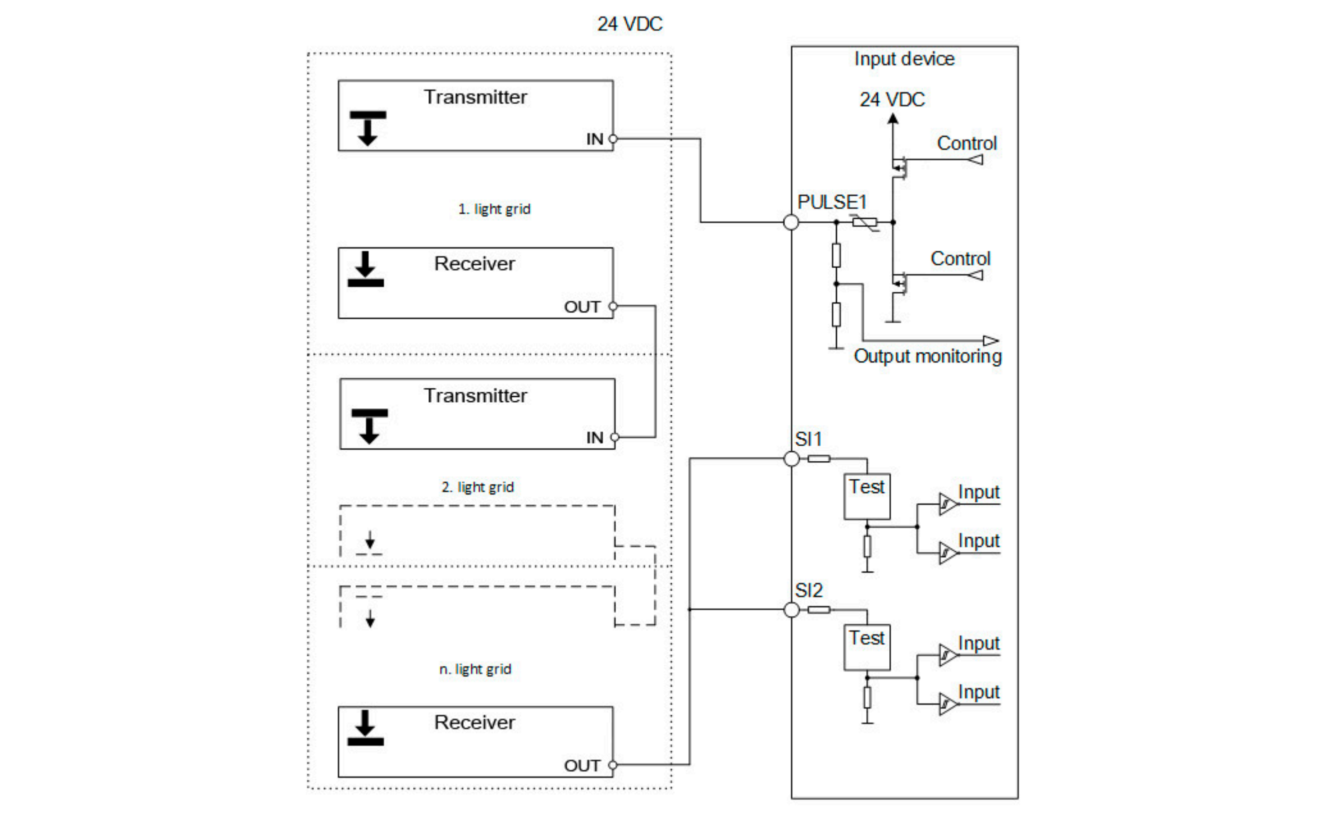


Figure 334: OSSD signals via cascaded light grid on a safe input channel pair - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

The lines from the input module to the transmitter (pulse 1 → IN) or from the receiver to the input module (OUT → SI1 + SI2) must be isolated in separate plastic-sheathed cables.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGNER.

Parameter	Value
Pulse Mode (SafeDigitalInput01)	See table under "Tested products"
Pulse Mode (SafeDigitalInput02)	
Filter Off (SafeDigitalInput01)	
Filter Off (SafeDigitalInput02)	

Table 484: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.4.1.4 OSSD signals via cascaded light grid on a safe input channel

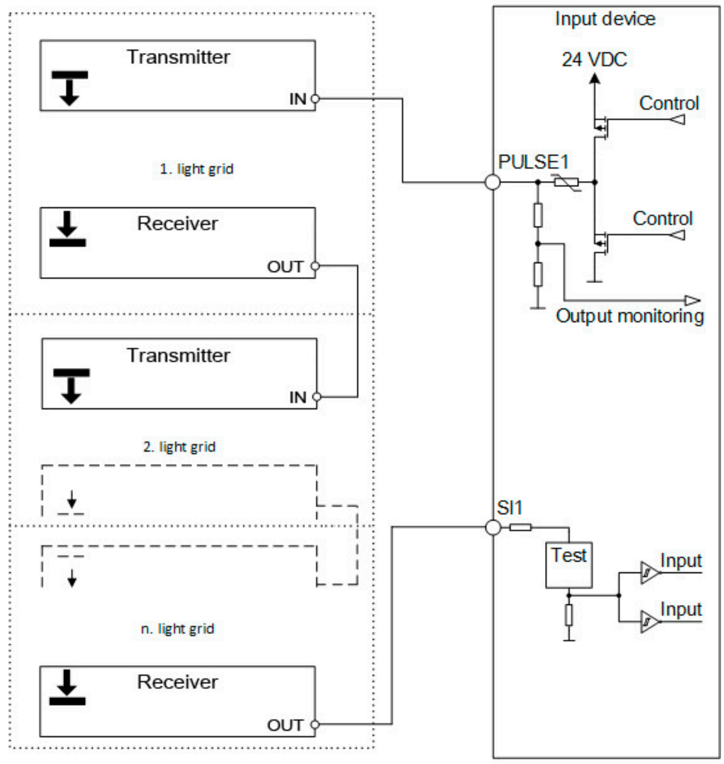


Figure 335: OSSD signals via cascaded light grid on a safe input channel - Circuit example

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 3.

The lines from the input module to the transmitter (pulse 1 → IN) or from the receiver to the input module (OUT → SI1) must be isolated in separate plastic-sheathed cables.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Pulse Mode (SafeDigitalInput01)	See table under "Tested products"
Filter Off (SafeDigitalInput01)	

Table 485: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.4.2 Tested products

4.4.2.1 SICK

> **Danger!**
>
> **In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.**

Light grid - L41

Model number	Compatibility class	Category
L41S-11MA1A + L41E-11MA1A	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
L41S-21KA1A + L41E-21KA1A	C2 - Compatible, construction identical to a C1 device	
L41S-21MA1A + L41E-21MA1A		
L41S-33MA2A + L41E-33MA2A		

4.4.2.1.1 Pulse Mode = internal

CAT 4 - Connection examples

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example: Figure label	=	Product label
X20SI2100 X20SI4100	X2X cycle time: 500 µs < X2X cycle < 800 µs, 1000 µs < X2X cycle < 1600 µs, 2000 µs < X2X cycle < 3200 µs **SafeDigitalInputxx:** • Filter Off = 2400 µs **SafeDigitalInput(xx+1):** • Pulse Source = Pulse xx • Filter Off = 2400 µs xx ... 1, 3	See Fig. 332.	CAT 4	IN → Pulse 1 OUT → SI1 + SI2	= =	L41S.TE → Pulse x x ... 1, 3, 5, 7 L41E.Q → SIx + SI(x+1) x ... 1, 3, 5, 7
X20SC2212 X20SC2432 X67SC4122.L12	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 3000 µs **SafeDigitalInput(xx+1):** • Pulse Source = Pulse xx • Filter Off = 3000 µs xx ... 1, 3, 5, 7					
X20SLX210 X20SLX410	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 4800 µs **SafeDigitalInput(xx+1):** • Pulse Source = Pulse xx • Filter Off = 4800 µs xx ... 1, 3					
X20SLX811 X20SLX842 X20SLX806 X20SLX402	X2X cycle time: No limitations **SafeDigitalInputxx:** • Pulse Source = Pulse z • Filter Off = 2400 µs **SafeDigitalInput(xx+1):** • Pulse Source = Pulse z • Filter Off = 2400 µs xx ... 1, 2, 3, 4, 5, 6, 7 z ... 1, 2, 3, 4			IN → Pulse 1 OUT → SI1 + SI2	= =	L41S.TE → Pulse z z ... 1, 2, 3, 4 L41E.Q → SIx + SI(x+1) x ... 1, 2, 3, 4, 5, 6, 7

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
				Figure label	**=**	**Product label**
X20SI9100 X20SLX910	X2X cycle time: No limitations **SafeDigitalInputxx:** • Pulse Source = Pulse z • Filter Off = 4800 µs **SafeDigitalInput(xx+1):** • Pulse Source = Pulse z • Filter Off = 4800 µs xx ... 1, 3, 5, 7, ... 19 z ... 1, 2, 3, 4	See Fig. 332.	CAT 4	IN → Pulse 1 OUT → SI1 + SI2	= =	L41S.TE → Pulse z z ... 1, 2, 3, 4 L41E.Q → SIx + SI(x+1) x ... 1, 3, 5, 7, ... 19
X67SI8103	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 3000 µs **SafeDigitalInput(xx+2):** • Filter Off = 3000 µs xx ... 1, 2, 3, 4, 5, 6 Instead of variables "SafeTwoChannelInput" and "SafeTwoChannelOK", function block "SF_Equivalent" must be used in SafeDESIGNER; the inputs being used must be connected manually.			IN → Pulse 1 OUT → SI1 + SI3	= =	L41S.TE → Pulse z z ... 1, 2 L41E.Q → SIxx + SI(xx+2) xx ... 1, 2, 3, 4, 5, 6

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

CAT 3 - Connection examples

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
				Figure label	=	Product label
X20SI2100 X20SI4100	X2X cycle time: 500 µs < X2X cycle < 800 µs, 1000 µs < X2X cycle < 1600 µs, 2000 µs < X2X cycle < 3200 µs **SafeDigitalInputxx:** • Filter Off = 2400 µs xx ... 1, 2, 3, 4	See Fig. 333.	CAT 3	IN → Pulse 1 OUT → SI1	= =	L41S.TE → Pulse x x ... 1, 2, 3, 4 L41E.Q → SIx x ... 1, 2, 3, 4
X20SC2212 X20SC2432 X67SC4122.L12	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 3000 µs xx ... 1, 2, 3, 4, ... 8					
X20SLX210 X20SLX410	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 4800 µs xx ... 1, 2, 3, 4					
X20SLX811 X20SLX842 X20SLX806 X20SLX402	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 2400 µs xx ... 1, 2, 3, 4, 5, 6, 7, 8			IN → Pulse 1 OUT → SI1	= =	L41S.TE → Pulse z z ... 1, 2, 3, 4 L41E.Q → SIx x ... 1, 2, 3, 4, 5, 6, 7, 8
X20SI9100 X20SLX910	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 4800 µs xx ... 1, 2, 3, 4, ... 20			IN → Pulse 1 OUT → SI1	= =	L41S.TE → Pulse z z ... 1, 2, 3, 4 L41E.Q → SIx x ... 1, 2, 3, 4, ... 20
X67SI8103	X2X cycle time: No limitations **SafeDigitalInputxx:** • Filter Off = 3000 µs xx ... 1, 2, 3, 4, ... 8			IN → Pulse 1 OUT → SI1	= =	L41S.TE → Pulse z z ... 1, 2 L41E.Q → SIxx xx ... 1, 2, 3, 4, ... 8

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

4.4.2.1.2 Pulse Mode = external

CAT 4 - Connection examples

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
				Figure label	=	Product label
X20SI2100 X20SI4100 X20SLX210 X20SLX410 X20SC2432 X67SC4122.L12	**SafeDigitalInputxx:** • Pulse Mode = external • Filter Off = n*500 µs **SafeDigitalInput(xx+1):** • Pulse Mode = external • Filter Off = n*500 µs xx ... 1, 3, 5, 7 n ... Number of light grids (max. 10)	See Fig. 334.	CAT 4	IN → Pulse 1 OUT → SI1 + SI2	= =	L41S.TE → Pulse x x ... 1, 2, 3, 4, 5, 6, 7 L41E.Q → SIx + SI(x+1) x ... 1, 2, 3, 4, 5, 6, 7

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Danger!

With the configuration "Pulse Mode = external", 5 ms must be added to the total response time.

CAT 3 - Connection examples

Model number of compatible products	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
				Figure label	=	Product label
X20SI2100 X20SI4100 X20SLX210 X20SLX410 X20SC2432 X67SC4122.L12	**SafeDigitalInputxx:** • Pulse Mode = external • Filter Off = n*500 µs xx ... 1, 2, 3, 4, ... 8 n ... Number of light grids (max. 10)	See Fig. 335.	CAT 3	IN → Pulse 1 OUT → SI1	= =	L41S.TE → Pulse x x ... 1, 2, 3, 4, ... 8 L41E.Q → SIx x ... 1, 2, 3, 4, ... 8

Danger!

Configuring a switch-off filter lengthens the safety response time. The configured filter value must be added to the total response time.

Danger!

With the configuration "Pulse Mode = external", 5 ms must be added to the total response time.

4.5 Connecting a safe output channel to a safe input channel

This section looks at connecting a safe output channel to a safe input channel. Particular attention is given to the SO connections of the output channel. Information about all other connections is included in the documentation for the respective module.

4.5.1 Circuit example

4.5.1.1 Safe output channel (high-side / high-side) to safe input channels

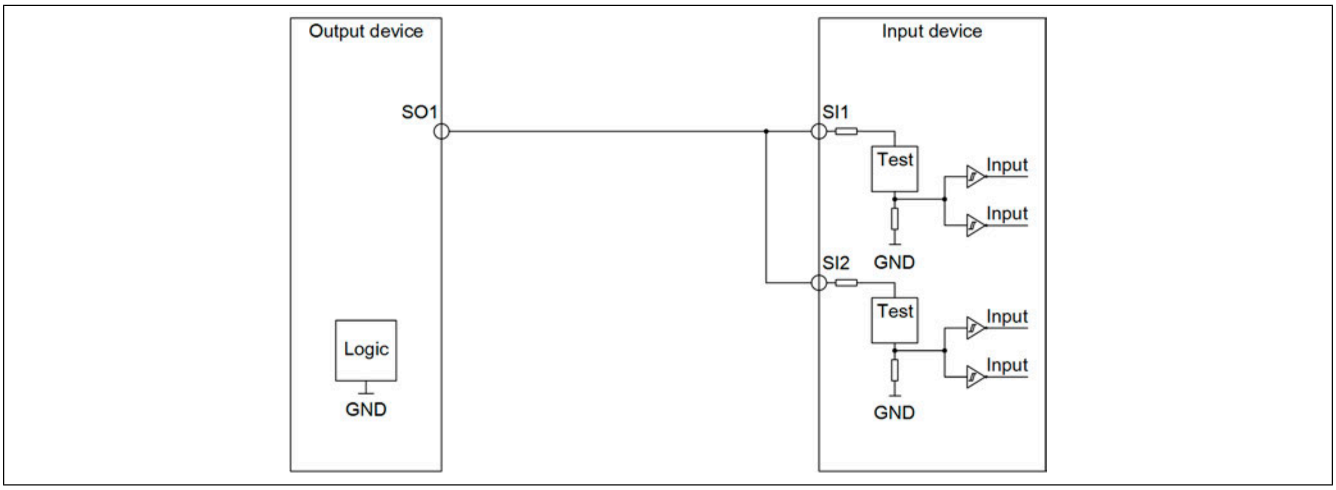


Figure 336: Circuit example: Safe output channel (high-side / high-side) to safe input channels

This circuit meets the requirements of EN ISO 13849-1:2015 CAT 4.

When using this circuit, the following parameters must be taken into account in Automation Studio / SafeDESIGN-ER.

Parameter	Value
Pulse Mode (SafeDigitalInput01)	See table under "Tested products"
Filter Off (SafeDigitalInput01)	
Discrepancy Time (SafeDigitalInput01)	
Pulse Mode (SafeDigitalInput02)	
Filter Off (SafeDigitalInput02)	
Max. switching frequency channel x [Hz] (prior to firmware version 300)	1 Hz
Disable OSSD	No

Table 486: Parameters in Automation Studio / SafeDESIGNER for the safe input channel

Danger!

With the configuration "Pulse Mode = No Pulse", the module itself is not able to detect wiring errors. Internal errors are still detected, however. All errors resulting from incorrect or faulty wiring must be handled through supplementary measures per EN ISO 13849-2:2012 or by the connected device.

Danger!

Configuring a switch-off filter lengthens the safety response time!
The configured filter value must be added to the total response time once or twice depending on the firmware version (for details, see the chapter "Filters" in the technical data sheet).
Configuring a switch-off filter causes signals with a low phase shorter than the switch-off filter to be filtered out. If this results in a problem concerning safety functionality, then the switch-off filter must be set to 0.
To minimize the effect of EMC interference, the max. line lengths between the pulse output and input specified in the technical data must be taken into account.
When connecting devices with OSSD signals (signals with test pulses), you must select a switch-off filter in each case that is substantially smaller than the repeat rate of the test pulses.

4.5.2 Tested products

4.5.2.1 B&R

Danger!

In addition to the information in this chapter, the notes in the current edition of the technical documentation for each used product must be observed.

X20SO

Model number	Compatibility class	Category
X20SC2212	C1 - Compatible and physically tested	The product is suitable for applications up to CAT 4, PLe.
X20SC0806		
X20SC0842		
X20SO6300		
X20SC0402	C2 - Compatible, construction identical to a C1 device	
X20SLX402		
X20SLX806		
X20SLX842		
X20SO2110	Not compatible	
X20SO2120	Not compatible	
X20SO4110	Not compatible	
X20SO4120	Not compatible	
X20SP1130	Not compatible	
X67SC4122.L12	Not compatible	

Model number of compatible products	Compatibility class	Parameters in Automation Studio / SafeDESIGNER	Circuit example	EN ISO 13849-1:2015 Category	Terminal assignments for circuit example		
					Figure label	=	Product label
X20SC0402	C2	**SafeDigitalInputxx:** • Pulse Mode = No pulse • Filter Off = 2100 µs • Discrepancy Time = 100 µs **SafeDigitalInput(xx+1):** • Pulse Mode = No pulse • Filter Off = 2100 µs xx ... 1, 3, 5, 7, ... 19	See Fig. 336.	CAT 4	SO1 → SI1 SO1 → SI2	= =	SOx_1→ SIx_2 x_1 ... 1 to 6 x_2 ... 1, 3, 5, 7, ... 19 SOx_1→ $SI(x_2+1)$ x_1 ... 1 to 6 x_2 ... 1, 3, 5, 7, ... 19
X20SC0806	C1						
X20SC0842							
X20SC2212	C2						
X20SC2432							
X20SI2100							
X20SI4100	C1						
X20SI8110							
X20SI9100	C2						
X20SLX210							
X20SLX402							
X20SLX410							
X20SLX806							
X20SLX811							
X20SLX842							
X20SLX910							
X67SC4122.L12							
X67SI8103							

Danger!

Only outputs SO5 and SO6 are permitted to be used on modules X20SC0842 and X20SLX842.

5 SafeDESIGNER

5.1 Introduction

5.1.1 General information

SafeDESIGNER is software used to develop safety applications for safety controllers.

It is based on the IEC 61131-3 standard and meets the safety requirements for the development process defined in IEC 61508.

The software contains a code editor for developing the program for the safety controller using the visual programming languages FBD and LD and text-based programming language ST. It also includes a table-based variable editor for managing variables, a cross-reference window and many other functions for the various development phases of a safety application such as editing, compiling, transmitting, printing, controlling the safety controller, debugging the safety application, etc.

The system allows libraries to be included. Defined in cooperation with PLCopen, standard safety-related function blocks such as emergency stop, safety door control and two-hand control are available as basic elements.

User management makes it possible to restrict the access rights for project changes to authorized programmers and record which changes have been made by which user.

The Safety View shows connected safe devices with their associated descriptions, safety identifiers and I/O signals. The user can easily map the I/O signals of the safe device to the global I/O variables when adding them to the code using drag-and-drop.

The connected safe devices are configured in the device configuration editor.

5.1.2 System requirements

The system requirements correspond to the version of Automation Studio being used (see Automation software → Software installation → Automation Studio → System requirements).

5.1.3 SafeDESIGNER maintenance version

In addition to the full version of SafeDESIGNER, a maintenance version is also available with a reduced range of functions designed specifically for commissioning safety applications. The maintenance version includes all of the functions from the full version; the limitation is that no changes can be made to the safety-related application or parameters.

Main functions of the maintenance version:

- Configuring machine options
- Debugging the safety application including variable watch and force functions
- Creating a machine log including the individually configured machine options

5.2 Developing a safety function

The following sections contain only the most important points for creating an example project. To learn how to use SafeDESIGNER in its entirety as well as the procedure and aspects involved with a safety application, refer to training module "TM510 - Working with SafeDESIGNER".

Danger!

For the correct implementation of a safety application, applicable laws and standards must be observed in all phases of the life cycle. This user's manual is limited to the use of SafeDESIGNER only. This user's manual is therefore no substitute for sound safety-related training.

5.2.1 Configuration in Automation Studio

From the point of view of safety technology, Automation Studio manages all of the modules for the safety-related components.

From the point of view of Automation Studio, SafeIO modules behave like standard input or output modules but with hidden complexity. Variables can be connected to the individual available I/O channels, which are used like normal I/Os in the standard application.

Management involves the following points:

- Adding the SafeLOGIC controller
- Naming the safety application
- Configuring the data exchange between the standard CPU and SafeLOGIC controller
- Adding the SafeIO modules
- Assigning the SafeIO modules to the SafeLOGIC controller

5.2.2 SafeDESIGNER software

SafeDESIGNER constitutes the heart of safety programming.

SafeDESIGNER is used to create the safety application and to configure the individual modules. For this purpose, all safety-related components assigned to the corresponding SafeLOGIC controller are taken automatically from the Automation Studio configuration.

The development of a safety application in SafeDESIGNER is divided into the following points:

- Programming the safety function
- Linking the I/O channels
- Configuring the modules
- Performing a download to the SafeLOGIC controller
- Testing the application
- Creating the documentation

5.3 IEC 61131-3 and the SafeDESIGNER software

Due to special safety requirements, only some of the features defined in IEC 61131-3 are implemented in the SafeDESIGNER software.

The following list shows the implemented IEC features:

- Variables must be declared (similar to variable declarations in high-level programming languages).
- A distinction is made between global and local data.
- Programming means symbolic programming.
- The source code of a program for a safety controller is structured using program organizational units (POUs). User-defined function blocks can be programmed and instantiated.
- The programming languages Function Block Diagram (FBD), Ladder Diagram (LD) and Structured Text (ST) are available for developing the program code.
- Use of specially developed, specific libraries for the safety controller.

5.3.1 Libraries in SafeDESIGNER

Per the IEC, functions and function blocks from included libraries can be reused in projects.

Due to the special safety requirements, only specially developed, safety-related libraries can be used in the safe programming system. These libraries contain reusable functions and function block POUs.

Subtree "Libraries" is available in the project tree for using libraries (i.e. including and removing them). Each included library is displayed in this subtree with its own icon.

After including a library, the functions and function blocks it contains can be added to a code worksheet by dragging and dropping them from the Editor Wizard.

5.3.2 Program organization units (POUs)

Program organizational units, or POUs for short, are the language elements of a PLC program. These are small, independent software units containing program code. The name of a POU must be unique, i.e. it is only permitted to be assigned once within a project.

2 types of POUs are available in SafeDESIGNER:

- 1 program
- Any number of user-defined function blocks (FBs).

Each POU consists of 2 different parts: the variable declaration section and the code section. Both are called "worksheets". All local variables are declared in the declaration section. The instruction or code section of a POU contains the statements programmed in the FBD, LD and ST programming languages.

Function block POUs

Function blocks are POUs with multiple input/output parameters and internal memory. The value that a function block returns as a result depends on the current value of its internal memory. Additional function blocks or functions can be called in a function block. Programs cannot be called. Recursive calls are not permitted. The abbreviation for function blocks in this manual is "FB".

The following factors influence the size of a POU and must be taken into account:

- The memory limit per POU is 64 kB.
- Data types are initialized according to their idle current principle (SAFEBOOL/BOOL with FALSE, numerical values with 0). Initialization must only be carried out if the initial values deviate from these base values.
- A POU should be kept as compact as possible in its function. It is important to ensure that both readability and reusability are provided in accordance with the corresponding software guidelines.
- Too many interface variables should also be avoided with regard to testability and code size. A maximum of 16 channels should be used for orientation purposes. More channels are permitted, but they increase the memory required for POU conversion.
- Any number of global variables can be used in POUs. It is important to note that this may limit reusability, however.

Like all IEC-defined function blocks and specific function blocks for a safety controller (stored in a library), user-defined function block POUs are also available in the Edit wizard after editing, saving and compiling the associated worksheets. This makes it possible to simply drag and drop the call of a user-defined function block into the code of the calling POU.

Calling a function block in another POU is called instantiation.

Program POUs (programs)

Programs contain a logical combination of functions and function blocks according to the requirements of the controller process. The behavior and use of programs is similar to that of function blocks. Programs have input and output parameters and can have internal memory.

In SafeDESIGNER, only one program is permitted per project. This program is added automatically when a new project is created. The program name "Main" cannot be changed, and the program cannot be copied or deleted.

5.3.3 Function blocks instantiation

Per IEC 61131-3, SafeDESIGNER offers the possibility of instantiation. Instantiation means that a function block is defined once and can then be used several times. This applies equally to all function blocks: user-defined function blocks (created in a function block POU), IEC-defined function blocks and function blocks in libraries.

Since function blocks always have internal memory, their values must be stored in a different memory area each time the function block is called. Instance names are used for this purpose. The instance name is declared in the variable declaration of the POU in which the function block is used. Each instance has an identifier (the instance name) as well as input and output parameters.

5.3.4 Variables and data types

SafeDESIGNER supports 2 types of variables: Local and global variables.

Per IEC 61131-3, programming is carried out using variables instead of directly addressing inputs and outputs or using flags. Variables are automatically declared in the variables worksheets when they are added to the code.

These variables worksheets are implemented as variable tables. The declarations are therefore not specified in plain text (as described in the IEC standard) but in the form of a table, which makes it much easier to handle the declarations. Each table row contains a variable or instance declaration; each table column represents a variable property (i.e. an element of the declaration). In this way, the table fully reflects the declaration syntax per IEC 61131.

The scope of a variable determines in which POU a variable can be used. Possible scopes are "Local" and "Global". The scope of each variable is defined by where the variable is declared (local or global variables worksheet) and by the keyword used for the declaration.

5.3.5 Local variables

If a variable can only be used within one POU, it is referred to as a local variable (the scope of its validity is "Local").

Local variables must be declared in the variables worksheet of the corresponding POU using one of the following keywords: VAR, VAR_INPUT or VAR_OUTPUT.

Since local variables cannot be connected to terminals (physical inputs and outputs), they are called symbolic variables. Symbolic variables are stored by SafeDESIGNER in free memory areas of the safety controller. The user does not know the addresses. Symbolic variables are permitted to have an optional initial value.

Function block instances are treated in the same way as local variables: their instances must be declared with VAR.

In debug mode, local variables can be overwritten.

5.3.6 Global variables

If a variable can be used in any POU in the project, it is called a global variable.

Variables with a global **scope** must be declared in the global variables worksheet.

In SafeDESIGNER, global variables are only permitted as I/O variables. This means that they must be connected to a terminal (physical input or output). In the IEC standard, these variables are referred to as "addressed" variables. The difference in SafeDESIGNER, however, is that a logical address does not have to be entered manually, but – for safety reasons – the terminal is connected to the global variable using drag-and-drop. In addition, the system automatically checks whether the sizes of the global variable and the terminal match. If necessary, the table editor adapts the data type of global variables to the data type of the connected terminal.

Like symbolic variables, I/O variables are also permitted to have an optional initial value (see "Initializing variables" on page 1028). In debug mode, I/O variables can be forced. Outputs are only permitted to be written to once in the program.

5.3.7 Initializing variables

Per IEC 61131-3, initial values can be assigned to a variable. This means that a variable used for the first time in the PLC program is called with its initial value. Initial values can be assigned for local (symbolic) and global output variables. Global variables assigned to a physical input cannot be initialized. The entered initial value must match the data type. For example, it is not possible to assign initial value "5" to a variable of data type BOOL. In this case, the system displays an error message when compiling. The initialization of variables is optional. If no initial value is used, the variable is initialized with the default initial value of the respective data type.

Information:

Initial values must be added in column "Initial value" of the variable table.

5.3.8 Keywords for variable declaration

Per IEC 61131-3, variables are declared using keywords.

In SafeDESIGNER, the keywords for variable declarations are selected in column "Usage" of the variable table.

The IEC-defined keywords relevant for the SafeDESIGNER software are described in the following table.

Keyword	Used for the declaration of
VAR	Local (symbolic) variables in the local variables worksheet of a POU. - Function block instances in the local variables worksheet of a POU.
VAR_INPUT	Variables that are input parameters of function block POUs. Input variables are used to pass a value to a function block, e.g. from another POU. An input variable can only be a local (symbolic) variable. The input variable can only be read by the function block; write access is not possible.
VAR_OUTPUT	Variables that are output parameters of function block POUs. Output variables are used by a function block to provide a value to another POU, for example. An output variable can only be a local (symbolic) variable. It can be written and read by the function block.
VAR_GLOBAL (Not visible in the safe programming system)	Although this keyword is defined in the IEC standard for declaring global variables, it does not have to be selected manually in the global variables worksheet. Since all variables declared in the global declaration table are automatically global, keyword VAR_GLOBAL is also assigned automatically to each variable declared there. As a result, column "Usage" is **not displayed** in the global worksheet. In the safe programming system, global variables are always I/O variables (assigned to a physical input or output).

5.3.8.1 Data types in SafeDESIGNER

Data types define the properties for the values of a variable. They define the initial value, range of possible values and number of bits.

The following elementary data types defined in IEC 61131-3 are available in SafeDESIGNER:

Standard (non-safe) data types

Data type	Description	Size	Range
BOOL	Boolean	1	0 or 1
INT	Integer	16	-32,768 to 32,767
DINT	Double integer	32	-2,147,483,648 to 2,147,483,647
TIME	Duration	32	0 to 2,147,483,647 s
BYTE	Bit string of length	8	0 to 255 (0x00 to 0xFF)
WORD	Bit string of length	16	0 to 65,535 (0x00 to 0xFFFF)
DWORD	Bit string of length	32	0 to 4,294,967,295 (0x00 to FFFFFFFF)

Safe data types

Data type	Description	Size	Range
SAFEBOOL	Boolean	1	0 or 1
SAFEINT	Integer	16	-32,768 to 32,767
SAFEDINT	Double integer	32	-2,147,483,648 to 2,147,483,647
SAFETIME	Duration	32	0 to 2,147,483.647 s
SAFEBYTE	Bit string of length	8	0 to 255 (0x00 to 0xFF)
SAFEWORD	Bit string of length	16	0 to 65,535 (0x00 to FFFF)
SAFEDWORD	Bit string of length	32	0 to 4,294,967,295 (0x00 to FFFFFFFF)

Corresponding conversion blocks are available to convert data types in the SAFE group to the IEC 61131 data types.

Danger!

Using a "non-safe-to-safe" converter in the wrong place can make an application unsafe. These types of conversions are usually only allowed when combined with a logical AND operator and a safe signal. In the course of an FMEA, ensure that the safety-related functionality of the application still meets the required SIL or PL level when using the conversion block.

5.4 Online communication

You can configure the online communication between SafeDESIGNER and the safety controller by selecting the "Online - TCPIP communication settings" menu option.

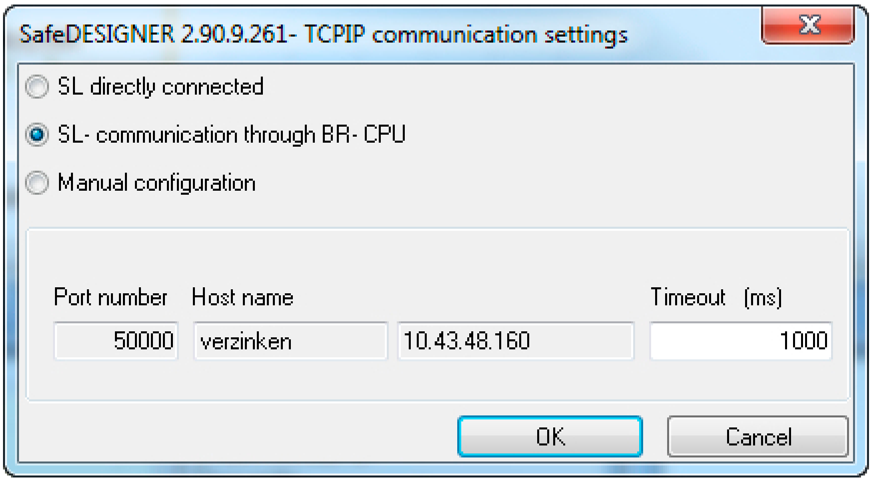


Figure 337: SafeDESIGNER online communication

The following list shows the options for online communication:

- Direct connection (point-to-point)
- Via the standard CPU
- Manual setting - No longer necessary

5.4.1 Direct connection

In this mode, the safety controller is connected directly with the PC and runs in standalone mode (no communication with I/O modules).

The safety controller has a static IP address: 192.168.100.xxx. The last position is determined by the node number switches. You must set the IP address of the PC from this network domain.

In order to make a direct connection, select the "Direct communication with the SL" option in SafeDESIGNER and specify the last position of the IP address accordingly.

5.4.2 Via the standard CPU

In this mode, the PC is connected to the CPU; the safety controller is located behind this CPU. An Ethernet connection is also established between the PC and CPU.

The CPU is responsible for ensuring that packets reach the safety controller and are returned to the PC. You must set a corresponding safety controller parameter in Automation Studio for this.

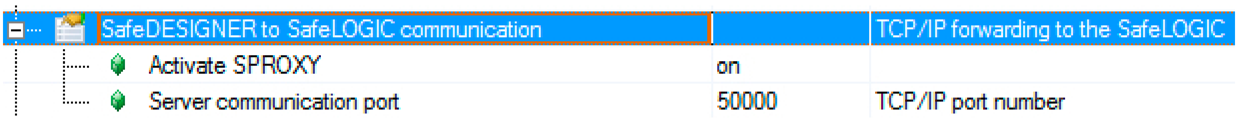


Figure 338: SPROXY settings

Information:

The port number for the safety controller must be unique in the project.

SafeDESIGNER uses the port number and current online configuration from Automation Studio as the basis for online communication.

For the connection, select the "SL communication via the CPU" option in SafeDESIGNER. As mentioned, SafeDESIGNER gets the port number, hostname and IP address from Automation Studio.

Communication when using ARwin

Additional points must be taken into account if the CPU is a Windows-based system (ARwin).

The first thing to do is set up communication between the PC and CPU. An additional setting must then be made for the SPROXY in the Internet connection sharing settings. To do so, use the port number configured for the safety controller from Automation Studio as well as the CPU's IP address.

Select the "SL communication via the CPU" option again in SafeDESIGNER.

6 Libraries

This section contains a description of the libraries available in SafeDESIGNER.

6.1 DATA_to_SafeDATA_SF

Library "DATA_to_SafeDATA_SF" allows the user to generate safe analog values from standard (non-safe) data sources.

The function blocks in this library receive analog values from 2 separate data sources in the form of standard X20 or X67 modules. The provided analog and time values are checked, further processed and then converted into safe analog values.

For additional information, see section "Technical information".

6.1.1 General information

This chapter lists the system requirements for using library "DATA_to_SafeDATA_SF". Major changes compared to the previous version are listed in the version history.

6.1.1.1 System requirements

Library "DATA_to_SafeDATA_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements must be met in order to use library "DATA_to_SafeDATA_SF":

- SafeDESIGNER: 4.3.3 or later
- Automation Studio: 4.3 or later
- SafeLOGIC: Safety Release 1.10 or later
- SafeLOGIC-X: Not supported
- Software license X20SF2106

6.1.1.2 Version history

Version	Date	Comment
1.11	May 2019	Chapter 6.1.2.1 "General notices": Added required hardware upgrades for modules.
1.10	January 2018	First version

Table 487: Version history

6.1.2 Technical information

The notes contained in this chapter must be observed when using the functions. Safety-related functionality is only provided under the basic conditions listed in this chapter.

In addition, the application diagrams listed in chapter "Applications" on page 1049 are binding and must be applied without modification. Deviating connections can cause failure of the safety function. Application-specific adaptations are only permitted in the appropriate blocks.

6.1.2.1 General notices

- Only modules of the following type are permitted to be used as a standard data source:

Model number	Short description	Support starting with hardware upgrade
X20DC1073	X20 digital counter module, 1x SinCos, 1 Vss, 400 kHz input frequency, encoder monitoring, NetTime function	V1.1.0.0
X20DC1176	X20 digital counter module, 1 ABR incremental encoder, 5 V, 600 kHz input frequency, 4x evaluation, encoder monitoring, NetTime function	V1.1.0.0
X20DC1178	X20 digital counter module, 1 SSI absolute encoder, 5 V, 1 Mbit/s, 32-bit, encoder monitoring, NetTime function	V1.1.0.0
X67AI2744	X67 analog input module, 2 full-bridge strain gauge inputs, 10 V, 24-bit converter resolution	V1.2.0.0

- The sensors must be used diversely so that signal values result in opposing analog values. This is achieved, for example, by the opposite installation of encoders to determine the safe speed. The opposing signal must be connected to channel 1 of the signal pair.
- Separate cables must be used for the sensor lines.
- Each analog value and associated NetTime must originate from a separate module. 2 standard analog values (including associated NetTime) and therefore 2 standard modules are needed for 1 safe analog value. Free channels on the modules are not permitted to be used for additional "DATA_to_SafeDATA" data sources.
- The circuit examples listed in chapter "Applications" must be applied without modification.
- The standard modules are not permitted to be operated in latch mode.
- The standard modules must provide the NetTime as a 32-bit value.
- All standard modules being used as a data source must be configured with a unique code ("SourceRef"). When using multiple "DATA_to_SafeDATA" instances or multiple SafeLOGIC controllers on the same POWERLINK network, these codes must be uniquely assigned on the POWERLINK network. The user must ensure this uniqueness since this is not checked by the system.
- Data from the standard modules (analog values, NetTime, "SourceRefIn", "CheckSum") must be made available to the function blocks in SafeDESIGNER via POWERLINK cross-traffic using "CPU-to-SafeLOGIC communication". Providing the data via a user task in the CPU is not permitted.

6.1.2.2 Information about the safety response time

The safety response time results as follows:

Max (response time of sensor 1, response time of sensor 2)
+
50 ms (worst-case scenario for the standard modules and POWERLINK network)
+
Max (SF_AnalogValueValidation_1.S_NetTimeFreezeDetectionTime, SF_AnalogValueValidation_2.S_NetTimeFreezeDetectionTime)
+
SF_1oo2Calculation.S_DiscrepancyTime (if only output parameters "S_DataOut" and "S_DataOutOK" of function block "SF_1oo2Calculation" are used for the evaluation or calculation)
+
SF_VelocityCalculation.S_TimeBase (if present; otherwise, 0 ms)
+
Response time for the signal transmission from the SafeLOGIC controller until the actuator drops out

Information:

For more information about "SF_AnalogValueValidation", "SF_1oo2Calculation" and "SF_VelocityCalculation", see the description of the function blocks.

6.1.2.3 Information about safety-related accuracy

With regard to safety-related accuracy, the following influences must be taken into account when performing the calculation of the residual error:

- Accuracy of the sensors being used
- Accuracy of the standard modules
- Monitoring tolerances of the "SF_1oo2Calculation" function blocks
- If applicable, the dx/dt function of function block "SF_VelocityCalculation"
- If applicable, further calculation steps of the safety application

6.1.2.4 Information about configuring the function blocks

The following parameters for the monitoring functions must be defined as small as possible in accordance with safety-related criteria.

The functions implemented in the function blocks are mainly used to diagnose errors in the B&R system, but they can also be used to diagnose the connected sensor.

The following notes must additionally be observed.

Function block / Parameter	Information in the B&R system	Information for sensors	Relevant for
SF_AnalogValueValidation			
S_MinAnalogValue, S_MaxAnalogValue	-	Can be used for sensor diagnostics.	-
S_MinChangeForFreezeDetection	≥1	Can be used for sensor diagnostics.	-
S_AnalogFreezeDetectionTime	>SafeLOGIC cycle time	Can be used for sensor diagnostics.	-
S_NetTimeFreezeDetectionTime	>SafeLOGIC cycle time	-	Response time
S_SourceRef	Unique in the entire POWERLINK network (also across multiple SafeLOGIC applications)	-	-
SF_VelocityCalculation			
S_TimeBase	-	If applicable, take the sensor data refresh time into account.	Response time
SF_1oo2Calculation			
S_MaxDifference	-	How "equal" are the signals returned by both sensors?	Accuracy
S_DiscrepancyTime	-	How "equal" are the signals returned by both sensors?	Accuracy and, if applicable, response time (see chapter "Applications" on page 1049)

6.1.2.5 Reliable safety function

The following safety functions are reliable from a safety point of view:

- Safe analog value: Strain gauge with X67AI2744 or position with X20DC1176
- Safe speed: Calculation from position with X20DC1176

6.1.2.6 Safety characteristics

The following safety characteristics per EN ISO 13849-1:2015 apply to the functions under the operating conditions listed in this chapter:

Characteristic value	Abbreviation	Value
Diagnostic coverage	DC	Medium
Category	Cat.	Cat. 3
Mean time to dangerous failure	MTTFD	100 years
Common cause failure	CCF	75 [1)]

1) Due to the 2-channel hardware structure, a score of 75 can be assumed.

6.1.3 Function blocks

6.1.3.1 General parameters

This section describes general parameters that are implemented in multiple function blocks in this library.

6.1.3.1.1 Activate

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable	FALSE	Enables the function block. TRUE: The function block is enabled.

Function description

This input parameter enables the function block.

If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.

If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.

In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.

You also have the option of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

6.1.3.1.2 Reset

I/O	Name	Data type	Connection	Initial value	Description
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment

Function description

Depending on the safety requirements, you must use data type SAFEBOOL or BOOL for the connection. With a SAFEBOOL connection, you avoid unexpected startups that may result from errors in the standard (non-safe) system.
You can reduce the risk of unexpected startup with further measures such as an additional function stop.

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.1.3.1.3 Ready

I/O	Name	Data type	Connection	Initial value	Description
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.

Function description

This output parameter indicates whether the function block is enabled or not.

6.1.3.1.4 Error

I/O	Name	Data type	Connection	Initial value	Description
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or the current value for numeric information and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
For numerical outputs, the current value is output.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.1.3.1.5 DiagCode

I/O	Name	Data type	Connection	Initial value	Description
OUT	DiagCode	WORD	Variable	0	Function block diagnostic message

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.1.3.2 Extended constants

Constants can be used in a safety application. These have the property that their values cannot be modified during runtime. For flexible solutions, however, it may be necessary to safely adjust parameters during runtime. "Safe commissioning options" are available for this purpose.
Extended constant means that both the conventional constants as well as the safe options are permitted to be connected on the input parameter.
Connecting an extended constant with a safe variable is technically not latched. However, all parameters specified as extended constants are not permitted to change during runtime; otherwise, the function block will issue an error message at runtime.

Possible applications of extended constants:

- Connecting with a SafeDESIGNER constant, e.g. SAFEINT#315
- Connecting with a SafeDESIGNER-global constant
- Connecting with a safe commissioning option

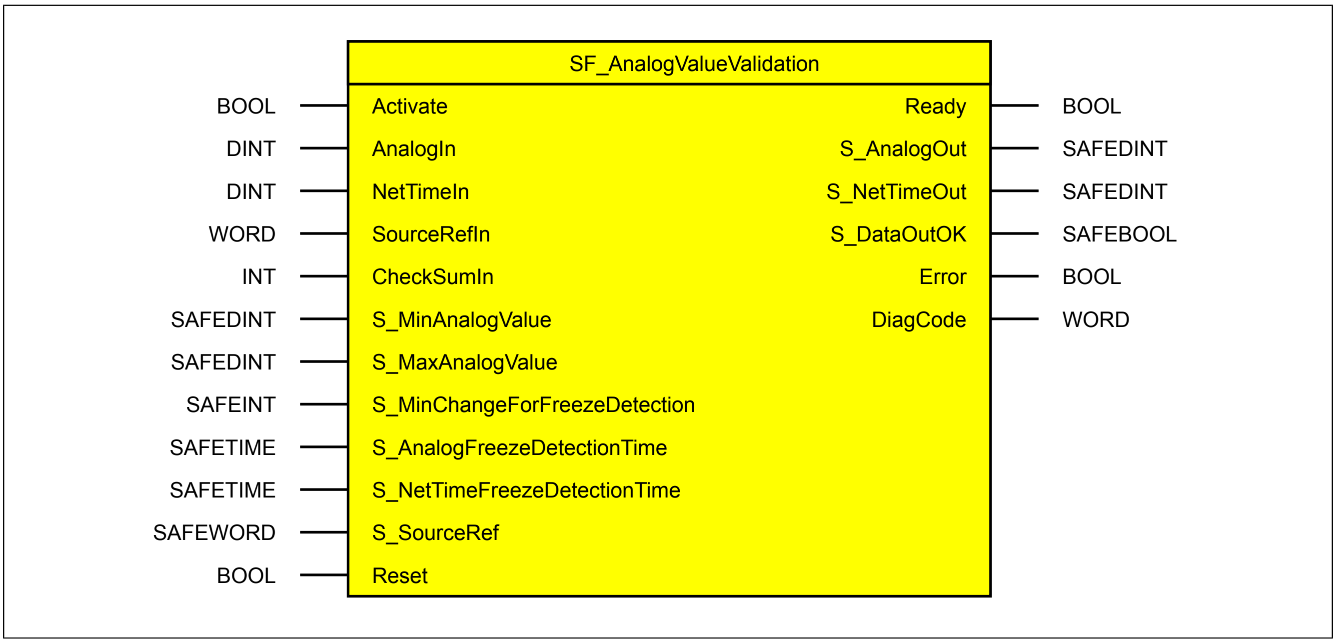
6.1.3.3 List of function blocks

Name	Description
SF_AnalogValueValidation	This function block validates the data from a standard analog value source.
SF_VelocityCalculation	This function block determines the velocity with the help of a position and time value.
SF_1oo2Calculation	This function block compares 2 analog signals and returns 1 safe analog signal as the result.

6.1.3.3.1 SF_AnalogValueValidation

This function block validates the data from a standard analog value source.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	AnalogIn	DINT	Variable	0	Analog value of the standard module
IN	NetTimeIn	DINT	Variable	0	NetTime of the standard module
IN	SourceRefIn	WORD	Variable	0	Unique code ("SourceRef") of the standard module
IN	CheckSumIn	INT	Variable	0	Checksum of the standard module
IN	S_MinAnalogValue	SAFEDINT	Extended constant	-2,147,483,648	Specification of the lower range limit of "AnalogIn"
IN	S_MaxAnalogValue	SAFEDINT	Extended constant	2,147,483,647	Specification of the upper range limit of "AnalogIn"
IN	S_MinChangeForFreezeDetection	SAFEINT	Extended constant	1	Specification of the min. change for the "Freeze detection" monitoring function of signal "AnalogIn". Range of values: 1 to max. INT
IN	S_AnalogFreezeDetectionTime	SAFETIME	Extended constant	20#ms	Specification of the monitoring time from "AnalogIn". Range of values: 10 ms to 10 s
IN	S_NetTimeFreezeDetectionTime	SAFETIME	Extended constant	5#ms	Specification of the monitoring time from "NetTimeIn". Range of values: 1 to 50 ms
IN	S_SourceRef	SAFEWORD	Extended constant	0	Specification for the unique code ("SourceRef") of the standard module
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_AnalogOut	SAFEDINT	Variable	0	Function block output signal
OUT	S_NetTimeOut	SAFEDINT	Variable	0	NetTime signal output of the function block
OUT	S_DataOutOK	SAFEBOOL	Variable	FALSE	Indicates that the output signal or NetTime signal is valid. TRUE: The signal is valid.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	0	Function block diagnostic message

6.1.3.3.1.1 Function description

The function block performs the following functions:

- Validation of the analog signal from the standard module:
  It is checked whether the signal on input parameter "AnalogIn" is within the configured limit values (input parameters "S_MinAnalogValue" and "S_MaxAnalogValue") and thus represents a valid value. The function block also checks whether the value present on input parameter "AnalogIn" changes within the time specified on "S_AnalogFreezeDetectionTime" by at least the value specified on "S_MinChangeForFreezeDetection". This detects freezing values.
- Validation of the NetTime signal from the standard module:
  The function block checks whether the value present on input parameter "NetTimeIn" changes within the time specified on "S_NetTimeFreezeDetectionTime". This detects freezing values. The function block can also perform an internal calculation that compares the time provided by the SafeLOGIC controller with the NetTime. A drifting apart of these two times likewise indicates a problem in the timing behavior of the standard module.
- Check of the "SourceRef" signal from the standard module:
  The function block checks whether the signal present on input parameter "SourceRefIn" corresponds to the value specified on "S_SourceRef". This reveals signal distortions.
- Check of the checksum signal from the standard module:
  The function block checks whether the signal present on input parameter "CheckSumIn" corresponds to the calculation rule for the checksum via input parameters "AnalogIn", "NetTimeIn" and "SourceRefIn". This reveals data corruption.

The configured values for "S_AnalogFreezeDetectionTime" and "S_NetTimeFreezeDetectionTime" must be greater than the cycle time of the safety controller; otherwise, deadlock situations may occur internally in the system. For this reason, smaller values result in an error message.

6.1.3.3.1.2 Status and error information

Additional details about output parameters that provide status and error information.

State of the output parameters in the event of error

In the event of error, the respective current values are output on output parameters "S_AnalogOut" and "S_NetTimeOut". Output parameter "S_DataOutOK" is set to FALSE. In addition, output parameter "Error" is set to TRUE. The error code is indicated by output parameter "DiagCode".

Status numbers

Errors

Code (hex)	Description	Reset required
C001	The configured value for "S_MinAnalogValue" is greater than "S_MaxAnalogValue".	Yes
C002	The configured value for "S_AnalogFreezeDetectionTime" is outside the permissible range of values.	Yes
C003	The input value for "AnalogIn" is less than "S_MinAnalogValue".	Yes
C004	The input value for "AnalogIn" is greater than "S_MaxAnalogValue".	Yes
C005	The value for "AnalogIn" did not change after expiration of the monitoring time specified on "S_AnalogFreezeDetectionTime" or changed by less than the value specified on "S_MinChangeForFreezeDetection".	Yes
C006	The configured value for "S_MinAnalogValue" / "S_MaxAnalogValue" was changed.	Yes
C007	The configured value for "S_AnalogFreezeDetectionTime" was changed.	Yes
C008	The value for "S_AnalogFreezeDetectionTime" is less than the cycle time of the safety controller.	Yes
C010	The function block is waiting for a rising edge on "Reset".	Yes
C011	The configured value for "S_NetTimeFreezeDetectionTime" is above the permissible range of values.	Yes
C012	The configured value for "S_NetTimeFreezeDetectionTime" is below the permissible range of values.	Yes
C013	The configured value for "S_NetTimeFreezeDetectionTime" was changed.	Yes
C014	The value for "NetTimeIn" did not change after expiration of the monitoring time specified on "S_NetTimeFreezeDetectionTime".	Yes
C015	The value for "S_NetTimeFreezeDetectionTime" is less than the cycle time of the safety controller.	Yes
C016	The delta value between the NetTime and SafeLOGIC cycle time is too large.	Yes
C017	The maximum number of function block instances (≤32) was exceeded.	-
C020	The value for "SourceRefIn" does not correspond to the value for "S_SourceRef".	Yes
C021	The value for "S_SourceRef" was changed.	Yes
C022	The value for "CheckSumIn" does not correspond to the calculation rule for the checksum via input parameters "AnalogIn", "NetTimeIn" and "SourceRefIn".	Yes
C023	The configured value for "S_MinChangeForFreezeDetection" is less than or equal to 0 (not permitted).	Yes
C024	The configured value for "S_MinChangeForFreezeDetection" changed.	Yes

Table 488: "SF_AnalogValueValidation": Error codes

Status information

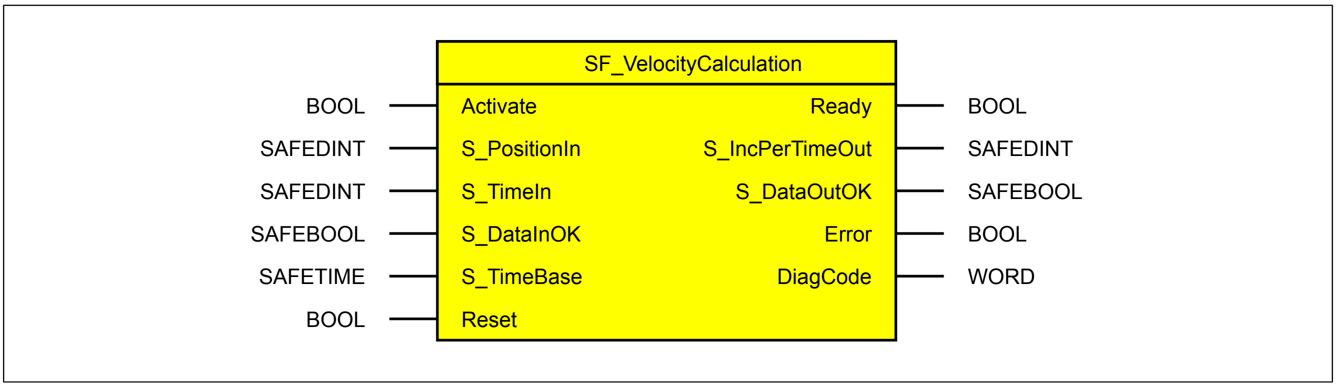
Code (hex)	Description
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.
8000	Normal operation
8001	Initialization of the function block after it has been enabled.
8010	The function block is waiting on new data from the standard module on input parameters "XXXIn".

Table 489: "SF_AnalogValueValidation": Diagnostic codes

6.1.3.3.2 SF_VelocityCalculation

This function block determines the velocity with the help of a position and time value.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_PositionIn	SAFEDINT	Variable	0	Input for the position value (must be connected to output parameter "S_AnalogOut" of function block "SF_AnalogValueValidation")
IN	S_TimeIn	SAFEDINT	Variable	0	Input for the time value (must be connected to output parameter "S_NetTimeOut" of function block "SF_AnalogValueValidation")
IN	S_DataInOK	SAFEBOOL	Variable	FALSE	Status of "S_PositionIn" and "S_TimeIn" (must be connected to output parameter "S_DataOutOK" of function block "SF_AnalogValueValidation"). TRUE: Values are valid.
IN	S_TimeBase	SAFETIME	Extended constant	10#ms	Specification of the time base for calculating the velocity. Range of values: 2 to 100 ms. Must be greater than or equal to 2 SafeLOGIC cycles.
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_IncPerTimeOut	SAFEDINT	Variable	0	Velocity value. Result from the difference calculation.
OUT	S_DataOutOK	SAFEBOOL	Variable	FALSE	Indicates that the output signal of "S_IncPerTimeOut" is valid. TRUE: The output signal is valid.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	0	Function block diagnostic message

6.1.3.3.2.1 Function description

Function block "SF_VelocityCalculation" generates a velocity value with the help of a difference calculation. The results of function block "SF_AnalogValueValidation" are used during this. When connecting the output parameters of function block "SF_AnalogValueValidation" to the corresponding input parameters of function block "SF_VelocityCalculation", it is important to ensure that both the analog value and the NetTime originate from the same module. If this is not the case, an invalid result occurs that the function block cannot detect.

The calculation for output parameter "S_IncPerTimeOut" is done with a valid value on "S_DataInOK".

S_TimeBase

The sampling time results from input parameter "S_TimeBase" and the SafeLOGIC cycle time. The following must be taken into account:

- The higher the sampling time, the more accurate the calculated values. More values are used for the calculation.
- The lower the sampling time, the faster the values are calculated. Fewer values are used for the calculation.

Example for "S_TimeBase" = 5 cycles:

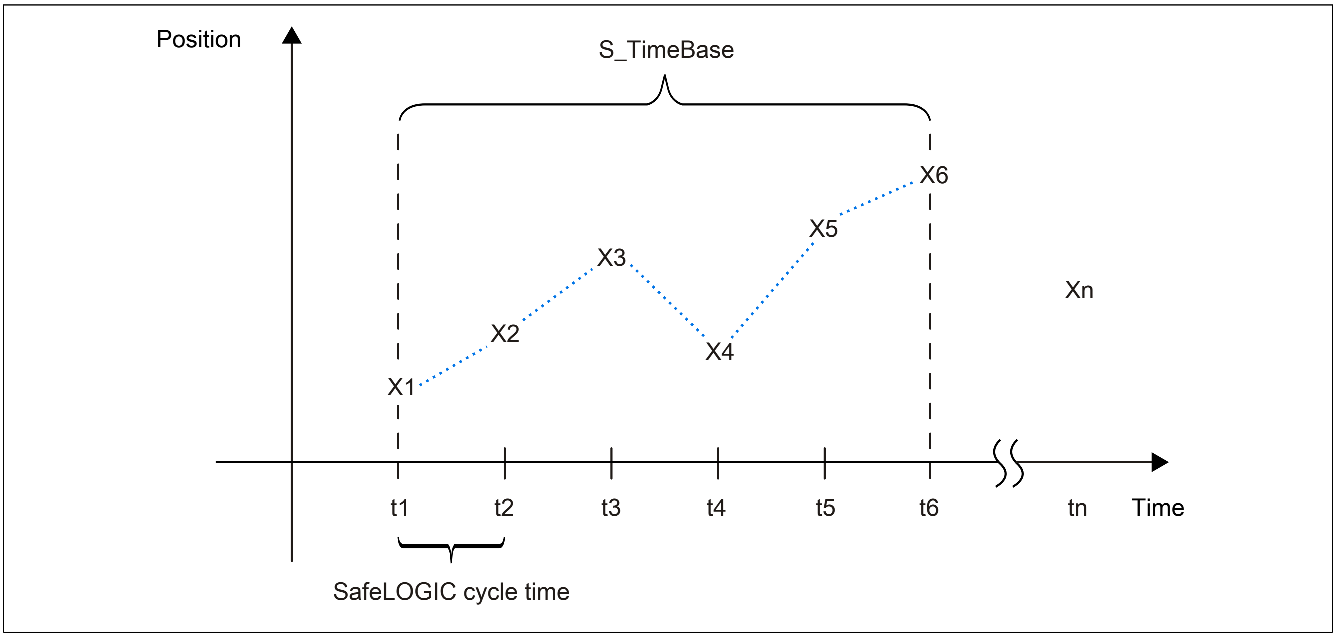


Figure 339: Representation of "S_TimeBase"

X Position value
t Time value

The velocity is calculated according to the following formula:

$$V1 = \frac{(X6 - X1)}{(t6 - t1)}$$

In the example, t6-t1 corresponds to the sampling time.
This results for the following cycle:

$$V2 = \frac{(X7 - X2)}{(t7 - t2)}$$

6.1.3.3.2.2 Status and error information

Additional details about output parameters that provide status and error information.

State of the output parameters in the event of error

In the event of error, the current value is output on output parameter "S_IncPerTimeOut". Output parameter "S_DataOutOK" is set to FALSE. In addition, output parameter "Error" is set to TRUE. The error code is indicated by output parameter "DiagCode".

Status numbers

Errors

Code (hex)	Description	Reset required
C001	The configured value for "S_TimeBase" is below the permissible range of values.	Yes
C002	The configured value for "S_TimeBase" is above the permissible range of values.	Yes
C003	The configured value for "S_TimeBase" is less than 2 times the configured cycle time of the safety controller.	Yes
C004	The value for "S_TimeBase" was changed.	Yes
C005	Input data "S_PositionIn" and/or "S_TimeIn" is invalid.	Yes
C006	The maximum number of function block instances (≤32) was exceeded.	-
C010	The function block is waiting for a rising edge on "Reset".	Yes

Table 490: "SF_VelocityCalculation": Error codes

Status information

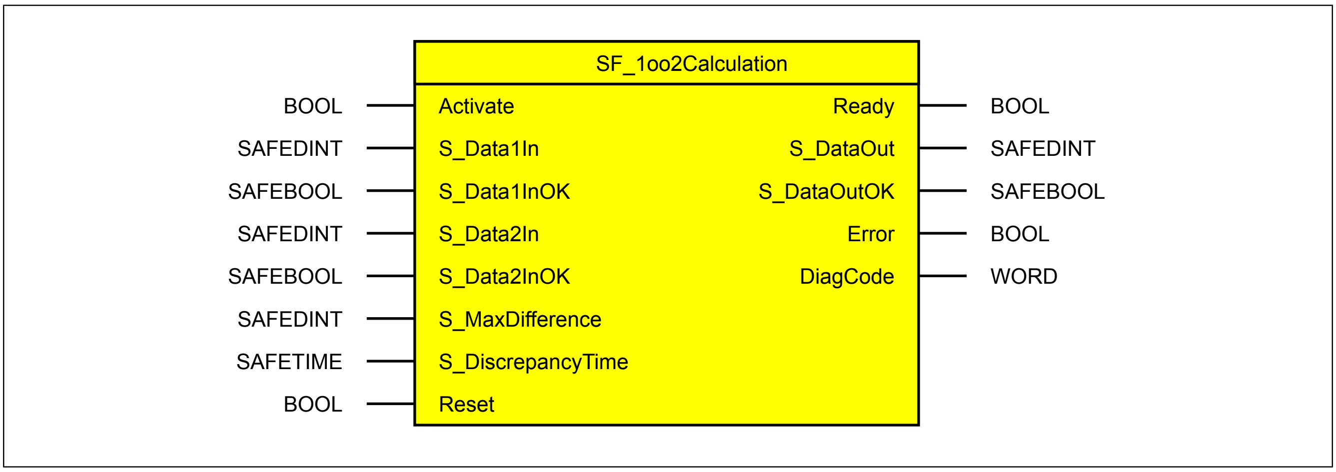
Code (hex)	Description
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.
8000	Cyclic execution of the velocity calculation.
8001	Initialization of the function block after it has been enabled.
8010	The function block is waiting for valid data on "S_DataInOK".

Table 491: "SF_VelocityCalculation": Diagnostic codes

6.1.3.3.3 SF_1oo2Calculation

This function block compares 2 analog signals and returns 1 safe analog signal as the result.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_Data1In	SAFEDINT	Variable	0	Input data from source 1
IN	S_Data1InOK	SAFEBOOL	Variable	FALSE	Indicates valid data from source 1. TRUE: The data from source 1 is valid.
IN	S_Data2In	SAFEDINT	Variable	0	Input data from source 2
IN	S_Data2InOK	SAFEBOOL	Variable	FALSE	Indicates valid data from source 2. TRUE: The data from source 2 is valid.
IN	S_MaxDifference	SAFEDINT	Extended constant	0	Specification of the maximum permissible difference between "S_Data1In" and "S_Data2In" Range of values: 0 to max. DINT
IN	S_DiscrepancyTime	SAFETIME	Extended constant	10#ms	Specification of the maximum permissible discrepancy time between "S_Data1In" and "S_Data2In" Range of values: 0 to 10 s
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_DataOut	SAFEDINT	Variable	0	Output data (corresponds to the mean value of "S_Data1In" and "S_Data2In")
OUT	S_DataOutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data on "S_DataOut". TRUE: Output data is valid.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	0	Function block diagnostic message

6.1.3.3.3.1 Function description

Function block "SF_1oo2Calculation" compares 2 analog signals and returns 1 safe analog output signal as the result. The two inputs are permitted to differ by the value specified on "S_MaxDifference". Overshooting or undershooting this limit is only permitted for a maximum duration of the time specified on "S_DiscrepancyTime" and otherwise results in an error.

6.1.3.3.3.2 Status and error information

Additional details about output parameters that provide status and error information.

State of the output parameters in the event of error

In the event of error, the current value is output on output parameter "S_DataOut". Output parameter "S_DataOutOK" is set to FALSE. In addition, output parameter "Error" is set to TRUE. The error code is indicated by output parameter "DiagCode".

Status numbers

Errors

Code (hex)	Description	Reset required
C001	The value for "S_MaxDifference" was changed.	Yes
C002	The value for "S_DiscrepancyTime" was changed.	Yes
C003	"S_Data1InOK" or "S_Data2InOK" is returning state FALSE.	Yes
C004	The value for "S_DiscrepancyTime" was overshot.	Yes
C005	The value for "S_MaxDifference" is invalid.	Yes
C006	The value for "S_DiscrepancyTime" is invalid.	Yes
C010	The function block is waiting for a rising edge on "Reset".	Yes

Table 492: "SF_1oo2Calculation": Error codes

Status information

Code (hex)	Description
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.
8000	Cyclic check of the valid input values.
8001	Initialization of the function block after it has been enabled.
8002	Determination of the exact error after "S_MaxDifference" is overshot.
8010	Check of the validity of the input parameters.

Table 493: "SF_1oo2Calculation": Diagnostic codes

6.1.4 Applications

The application diagrams listed are binding and must be applied without modification. Deviating connections can cause failure of the safety function. Application-specific adaptations are only permitted in the appropriate blocks (colored green).

6.1.4.1 Safe analog value

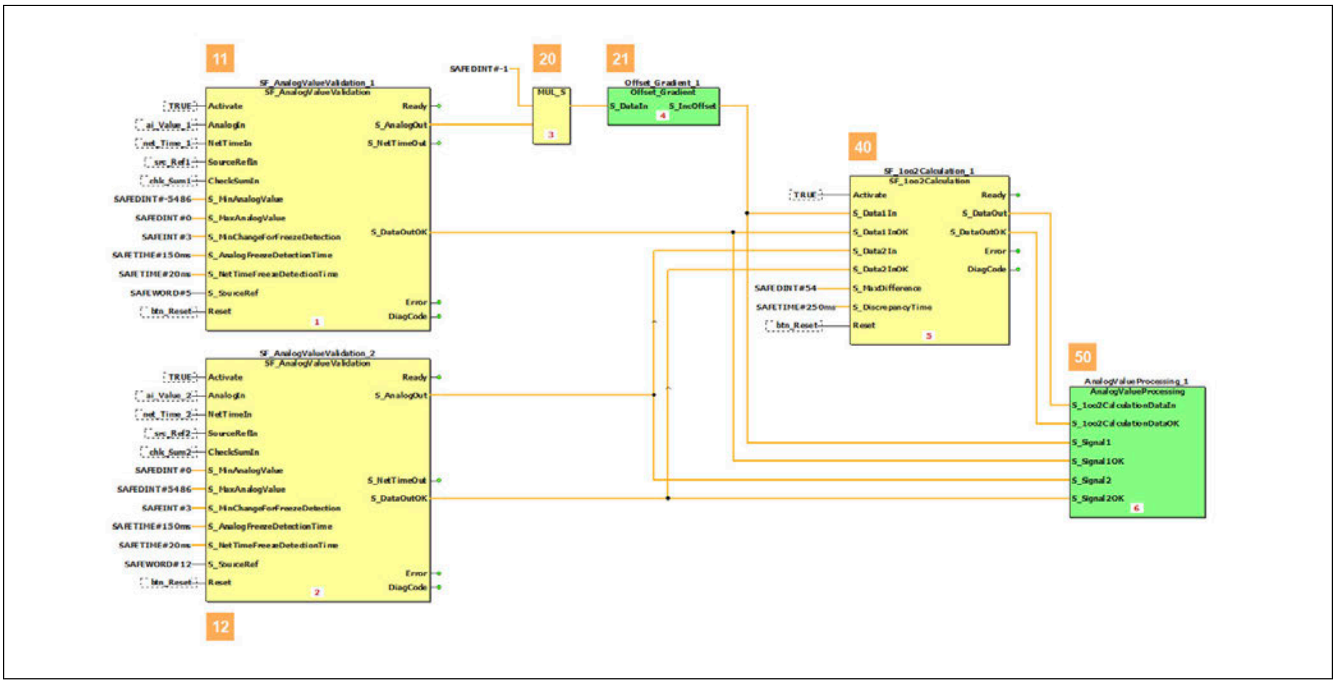


Figure 340: Safe analog value

Description of the application example

- **11/12:**
  This area is used for signal monitoring. Here, the system checks whether a value is frozen or a communication problem has occurred. This is done by checking the analog value and associated NetTime for continuous signal changes. In addition, the unique code of the signals and checksum of the data are checked. The two signals must come from the same source, i.e. the same module. For example, area 11 shows the monitoring of the two signals of module #1 and area 12 the signals of module #2.
- **20:**
  The inverse signal is corrected by multiplying it by -1.
- **21:**
  If further signal adjustments such as offset or slope corrections are necessary in addition to signal inversion (see #20), they are permitted to be made here in the application.
- **40:**
  Here, a 1oo2 calculation is made. The two signals are checked for their delta value. This is not permitted to exceed the value configured on the function block for "S_MaxDifference" longer than the configured discrepancy time ("S_DiscrepancyTime"). The result is provided via output parameter "S_DataOut".
- **50:**
  2 variants are available for evaluations or further calculations:
  - If signals "S_Signal1", "S_Signal1OK", "S_Signal2", "S_Signal2OK", "SF_1oo2Calculation.S_DataOut" and "SF_1oo2Calculation.S_DataOutOK" are evaluated, the value specified in parameter "SF_1oo2Calculation.S_DiscrepancyTime" is permitted to be ignored when considering the response time.
  - If only signals "SF_1oo2Calculation.S_DataOut" and "SF_1oo2Calculation.S_DataOutOK" are evaluated, the value specified in parameter "SF_1oo2Calculation.S_DiscrepancyTime" must be added when considering the response time.

6.1.4.2 Safe speed

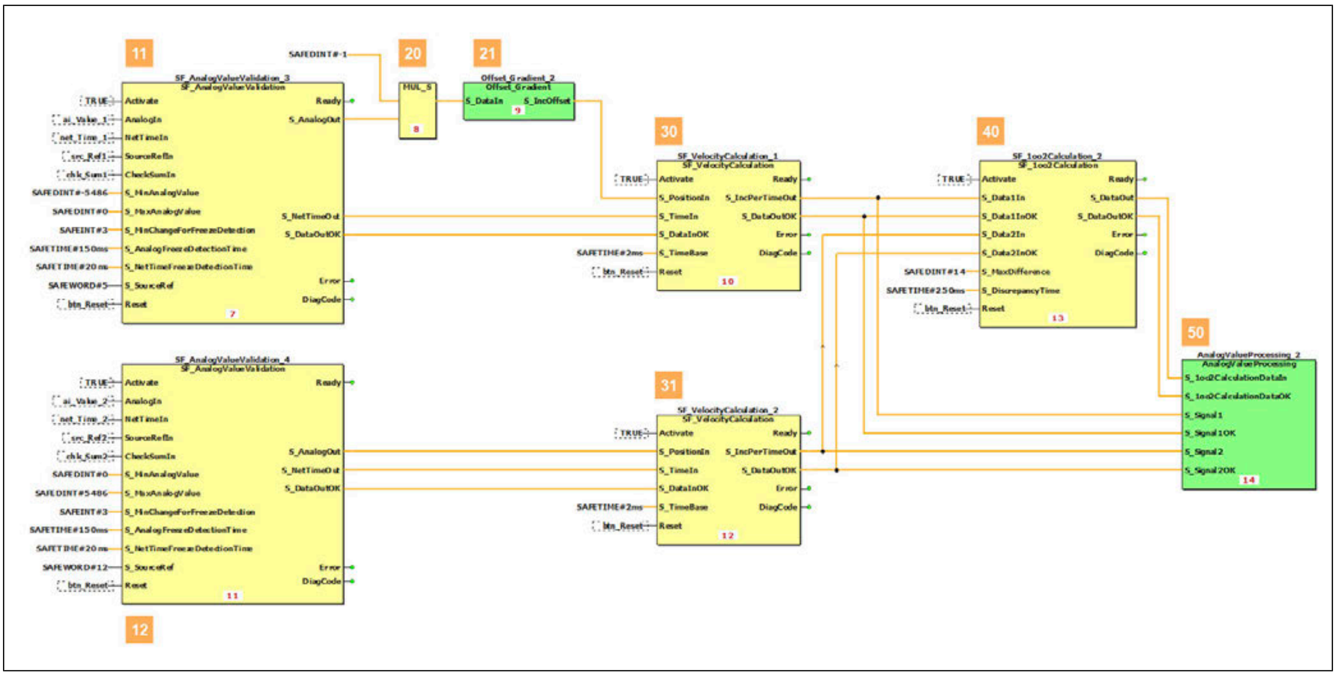


Figure 341: Safe speed

Description of the application example

- **11/12:**
  This area is used for signal monitoring. Here, the system checks whether a value is frozen or a communication problem has occurred. This is done by checking the analog value and associated NetTime for continuous signal changes. In addition, the unique code of the signals and checksum of the data are checked. The two signals must come from the same source, i.e. the same module. For example, area 11 shows the monitoring of the two signals of module #1 and area 12 the signals of module #2.
- **20:**
  The inverse signal is corrected by multiplying it by -1.
- **21:**
  If further signal adjustments such as offset or slope corrections are necessary in addition to signal inversion (see #20), they are permitted to be made here in the application.
- **30/31:**
  In this area, the speed signal is determined by means of the encoder signal (input parameter "S_PositionIn") and time (input parameter "S_TimeIn"). The result is provided on output parameter "S_IncPerTimeOut" of the function block. For example, area 30 shows the evaluation of module #1 and area 31 the evaluation of module #2. The derivation is carried out via the difference calculation depending on input parameter "S_TimeBase".
- **40:**
  Here, a 1oo2 calculation is made. The two signals are checked for their delta value. This is not permitted to exceed the value configured on the function block for "S_MaxDifference" longer than the configured discrepancy time ("S_DiscrepancyTime"). The result is provided via output parameter "S_DataOut".
- **50:**
  2 variants are available for evaluations or further calculations:
  - If signals "S_Signal1", "S_Signal1OK", "S_Signal2", "S_Signal2OK", "SF_1oo2Calculation.S_DataOut" and "SF_1oo2Calculation.S_DataOutOK" are evaluated, the value specified in parameter "SF_1oo2Calculation.S_DiscrepancyTime" is permitted to be ignored when considering the response time.
  - If only signals "SF_1oo2Calculation.S_DataOut" and "SF_1oo2Calculation.S_DataOutOK" are evaluated, the value specified in parameter "SF_1oo2Calculation.S_DiscrepancyTime" must be added when considering the response time.

6.1.4.3 Configuration in SISTEMA

In order to simplify the calculation of the safety characteristics, function "DATA_to_SafeDATA" is available in B&R library SISTEMA. The following example shows its use together with a linear encoder from Heidenhain.

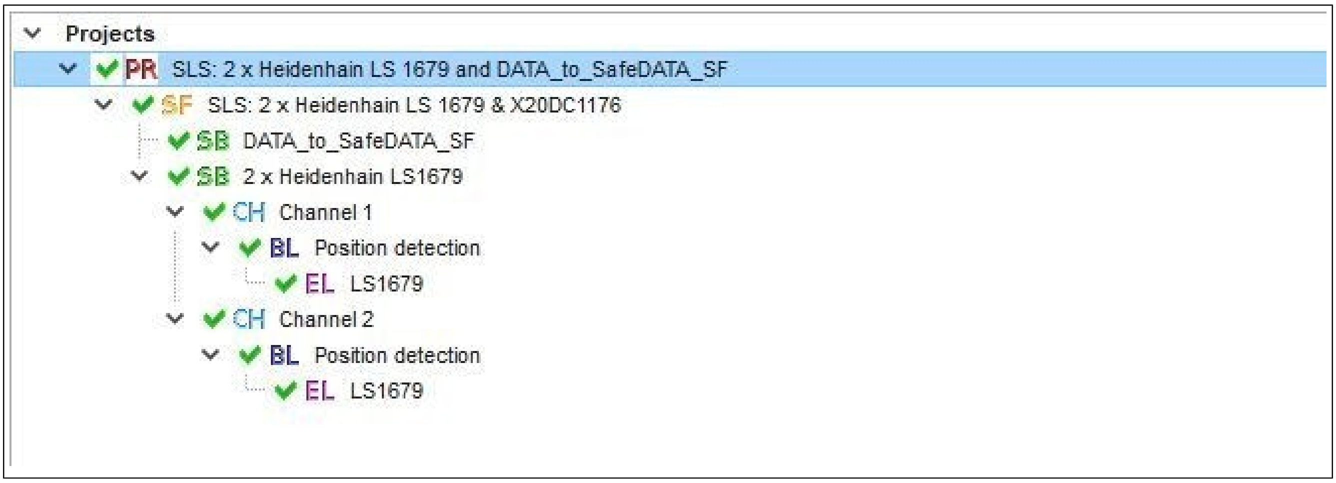


Figure 342: Configuration in SISTEMA

6.2 LightCurtain_SF

Using light curtains that support the openSAFETY Vision profile provides an extended range of functions compared to conventional hardwired devices. Single-beam evaluation is among the most important of these functions. In order to be able to use the full range of functions, function blocks have been developed that support the openSAFETY Vision profile. They have been grouped together in this library.

Users can take advantage of these to develop flexible solutions for implementing muting and blanking applications. They access the state of each individual beam and evaluate it accordingly. This opens up possibilities such as dividing a light curtain into areas that can be evaluated as needed. Object dimensions, for example, are defined and checked when using the blanking function. The provided state is available for further processing in the safety application.

For an overview of supported function blocks, see section "Function blocks".

6.2.1 General information

This chapter lists the system requirements for using library "LightCurtain_SF". Major changes compared to the previous version are listed in the version history.

6.2.1.1 System requirements

Library "LightCurtain_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements must be met in order to use library "LightCurtain_SF":

- SafeDESIGNER: 4.2.5 or later or 4.3.2 or later
- Automation Studio: 4.2 or later or 4.3 or later
- SafeLOGIC: Safety Release 1.10 or later
- SafeLOGIC-X: Not supported

6.2.1.2 Version history

Version	Date	Comment
1.22	May 2019	• Chapter 6.2.2.1.1 "Table data": Updated description.
1.21	March 2018	• Updated surveillance function. • Chapter "Function blocks": Added information. • SF_ReducedResolution: Chapter "Function description": Added danger warning.
1.20	February 2018	Editorial changes
1.10	November 2017	First version

Table 494: Version history

6.2.2 Technical information

A light curtain can be divided into several zones (muting) or objects (blanking). Configuration and parameterization takes place using tables. The following table formats are available to facilitate transparency and simple application:

- Table format C (partial muting) for implementing partial muting
- Table format D (blanking) for implementing blanking

A zone or object represents an area in the light curtain. The definition is made using single-beam evaluation of the light curtain. In this process, individual beams directly adjacent to one another are defined as a zone/object. Several zones/objects can be configured in a light curtain; they are not permitted to overlap, and a gap of at least one beam must be maintained between areas.

6.2.2.1 Tables

List of the table formats contained in this library and supported function blocks sorted in descending order starting with the version from which they are available.

Supported starting with SafeDESIGNER	Name	Short description	Supported by
4.3.3	Table format C (partial muting)	Masking and configuring muting areas of a light curtain	SF_BeamMask
4.3.3	Table format D (blanking)	Masking and configuring blanking areas of a light curtain	SF_Blanking

6.2.2.1.1 Table data

Table data can be edited via SafeDESIGNER. It uses the program for editing CSV files for this.

> **Information:**
>
> **To edit a table object, right-click on it in SafeDESIGNER and select "Edit".**

File structure

The file is divided into a header and data section. This structure must be maintained at all times; otherwise, the file can no longer be processed correctly. A CRC value, timestamp and the name of the Windows user who locked the file are entered in the file automatically.

The user specifies the row or column where the data begins and ends. Additional information (notes/comments) can also be included as long as it does not overlap with the data section (row/column).

The file size is calculated from the number of table entries and their data type.

Header section

- TableID:
  - "TableID" is updated by the system automatically when a table is edited or imported.
  - Permissible values: 1 to 99
  - Depends on the table object being used in SafeDESIGNER
- TableFormat: Specifies the type of table
- Data: Specifies the row or column where the data begins and ends (defined by the user)
- Additional parameters valid for the entire table (defined by the user)

Data section

The data section is determined by the table type or table format being used.

6.2.2.1.2 Table format C (partial muting)

Function description

This table format makes it possible to mask areas of a light curtain. A typical area of application would be the muting function.

Header section

There are no user-specific parameters for this table format.

Data section

- Zone [1 to 5]: Zone number
- Position [1 to 256]: Defines the starting point of the zone. This is the number of a beam in the light curtain.
- Dimension [1 to 255]: Defines the dimension of the zone. The starting point of the light curtain beam is part of the dimension and must therefore be taken into account.

In the following example, the data section (A5 up to and including C9) is highlighted in yellow. The gray cells (A4 to C4) contain the column headers. A total of 5 zones are defined, each with a dimension of 5 light curtain beams.

	A	B	C	D	E
1	TableID	1			
2	TableFormat	C			
3	Data	A	5	C	9
4	Zone	Position	Dimension		
5	1	2	2		
6	2	6	1		
7	3	13	5		
8	4	22	7		
9	5	31	8		
10					
11	3862637065				
12	MaxMuster	01.01.2017 11:22			

Figure 343: Table format C - Example

The following image shows part of the light curtain. The green areas highlight the valid, defined zones 1 and 2.

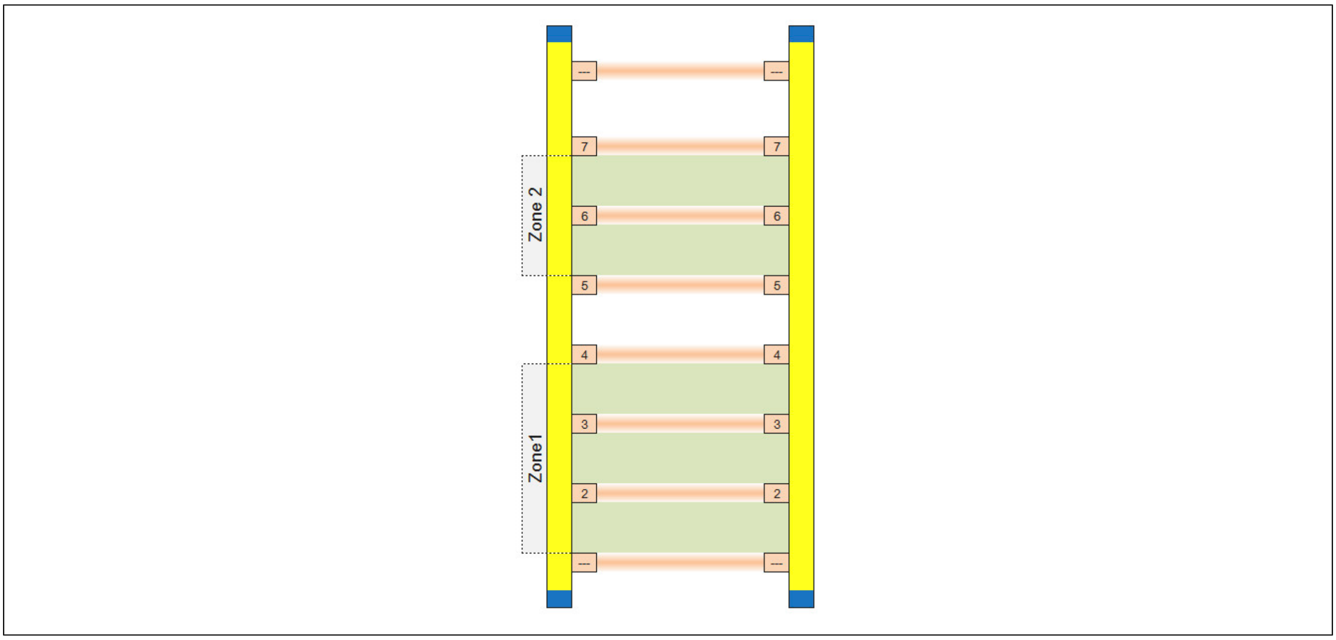


Figure 344: Zones 1 and 2 of the light curtain - Example

Information:

Zones must be unique and sorted in ascending order.

Overlaps are not permitted. A minimum gap of one light curtain beam between 2 zones must be taken into account.

6.2.2.1.3 Table format D (blanking)

Function description

This table format makes it possible to parameterize and configure blanking areas for a light curtain.

Header section

The blanking function monitors whether the defined objects are continuously present (light curtain is interrupted). Parameter "Surveillance" can be used to select between fixed blanking and partial monitoring.

Surveillance: Applies to all zones

- Partial monitoring [0]: The defined zones are not monitored. Interrupted or uninterrupted light curtain beams in the defined zones do not result in a safety request.
- Fixed blanking [1] (default): The defined zones are monitored. Uninterrupted light curtain beams in the defined zones result in a safety request.

Data section

"Surveillance" specifies whether the light curtain beams in the defined zones should be monitored for interruptions. This applies to the same extent to all objects and cannot be individually adjusted.

Parameters per object

- Object[1 to 5]: Object number
- MinimumBeam[1 to 256]: Specifies the light curtain beam with the lowest value that is permitted to be interrupted.
- MaximumBeam[1 to 256]: Specifies the light curtain beam with the highest value that is permitted to be interrupted.
- Dimension[1 to 255]: Defines the dimension of the blanking object, including the starting light curtain beam
- DimensionTolerance[0 to 255]: Tolerance of light curtain beams by which the dimension is permitted to vary

	A	B	C	D	E
1	TableID	2			
2	TableFormat	D			
3	Data	A	6	E	10
4	Surveillance	1			
5	Object	MinimumBeam	MaximumBeam	Dimension	DimensionTolerance
6	1	1	7	5	1
7	2	11	17	5	1
8	3	21	27	5	1
9	4	31	37	5	1
10	5	41	47	5	1
11					
12	1799223249				
13	maxmuster	19.10.2017 15:06			

Figure 345: Table format D - Example

The following image shows part of the light curtain. The green area highlights the valid range for the object in zone 3.

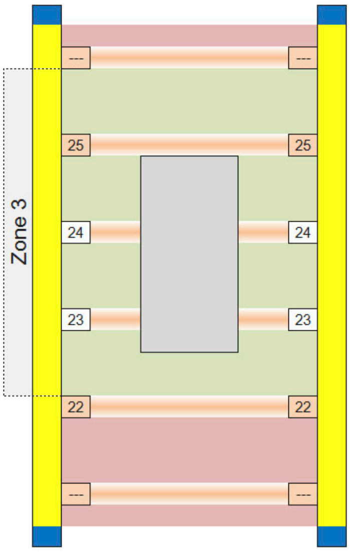


Figure 346: Zone 3 of the light curtain - Example

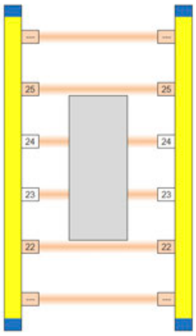
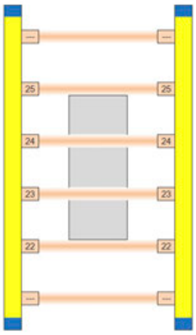
Information:

Zones must be unique and sorted in ascending order.

Overlaps are not permitted. A minimum gap of one light curtain beam between 2 zones must be taken into account.

Surveillance

- Checks whether light curtain beams are permanently interrupted

State of the light curtain beam	Surveillance	
	Partial monitoring (0)	Fixed blanking (1)
 Interrupted	No safety request	No safety request
 Uninterrupted	No safety request	Safety request

6.2.3 Function blocks

List of function blocks included in this library.

> **Information:**
>
> **A non-synchronized light curtain always results in a safety request for the function blocks.**
>
> **For more information, see the documentation for the respective light curtain.**

openSAFETY Vision profile

Name	Description
SF_LightCurtainBasic	This function block returns status information and transmits commands to a connected device that supports the openSAFETY Vision profile.

Muting

Name	Description
SF_BeamMask	This function block identifies the state of masked and unmasked beam areas.
SF_Muting_Type_L	This function block makes it possible to implement L-type muting.

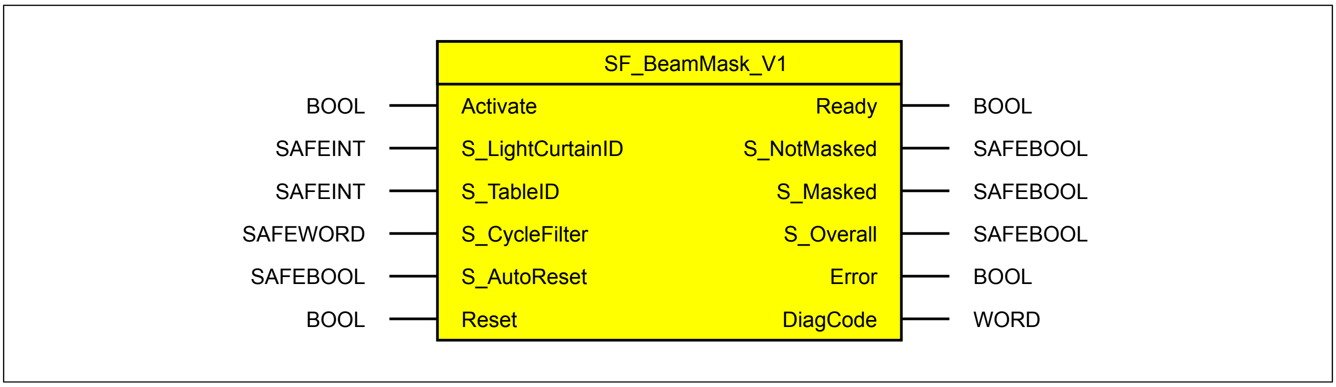
Blanking

Name	Description
SF_Blanking	This function block makes it possible to use blanking functions.
SF_ReducedResolution	This function block makes it possible to reduce the resolution for a device that supports the openSAFETY Vision profile.

6.2.3.1 SF_BeamMask

This function block identifies the state of masked and unmasked beam areas.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_LightCurtainID	SAFEINT	Constant	0	Reference to the object of the electro-sensitive protective equipment (light curtain)
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
IN	S_CycleFilter	SAFEWORD	Constant	0	Suppresses the safety requests of a light curtain for the specified number of light curtain cycles. Range of values: 0 to 5 light curtain cycles
IN	S_AutoReset	SAFEBOOL	Variable/ Constant	FALSE	Specifies the start interlock if proper signals are present on the input parameters. TRUE: The start interlock is not active.
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_NotMasked	SAFEBOOL	Variable	FALSE	State of the unmasked beams. TRUE: A safety request is not present for the beams in the unmasked area.
OUT	S_Masked	SAFEBOOL	Variable	FALSE	State of the masked beams. TRUE: A safety request is not present for the beams in the masked areas.
OUT	S_Overall	SAFEBOOL	Variable	FALSE	Function block release signal for the entire muting area. TRUE: A safety request is not present in any of the areas.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.2.3.1.1 Function description

Function block "SF_BeamMask" checks the state of the specified beam areas. They are configured using tables. Table format C (partial muting) is used in this case.

The function block returns the state of masked and unmasked light curtain areas as well as the state of the entire light curtain.

Masked zones are shown in green in the following image. Unmasked light curtain areas are shown in white.

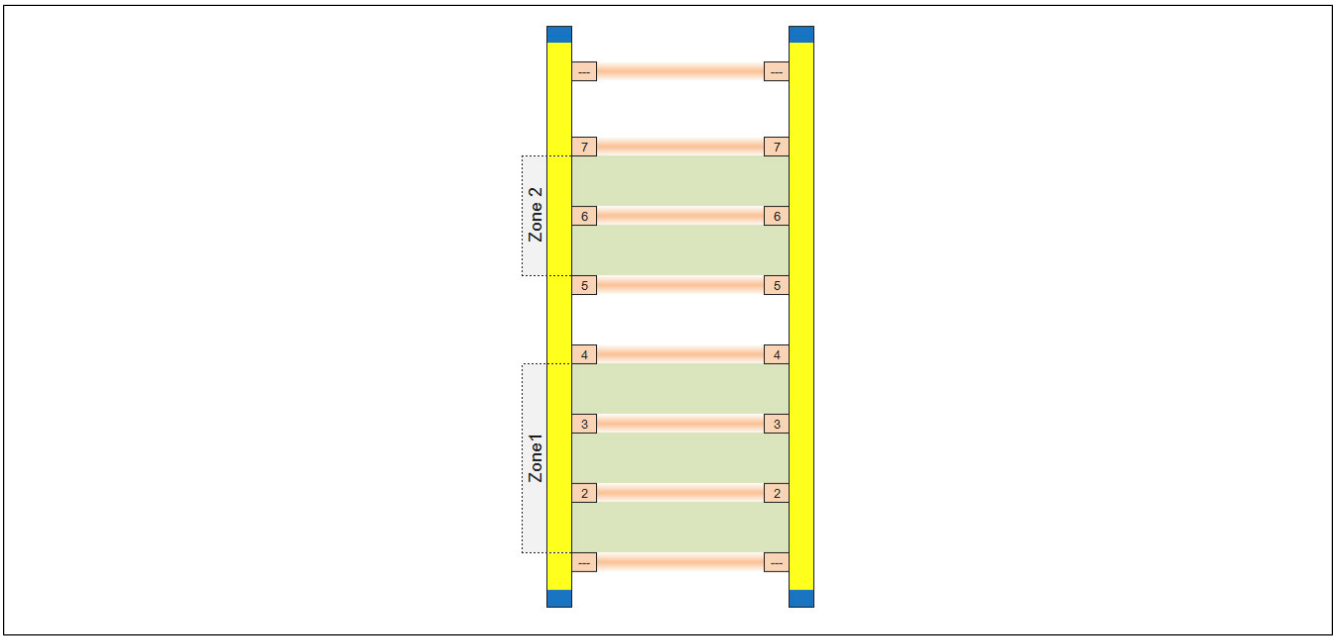


Figure 347: Example for 2 masked muting zones

The state of the unmasked areas could be connected to PLCopen function block "SF_ESPE", for example. The state of the entire light curtain can be linked to blanking functions, for example.

6.2.3.1.2 Input parameters

Description of the function block input parameters.

6.2.3.1.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- You also have the option of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.2.3.1.2.2 S_LightCurtainID

General function

- Reference to the object of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEINT

Connection

- Constant

Information:

Connect this input parameter to the signal of a safe device that is connected to the electro-sensitive protective equipment and supports the openSAFETY Vision profile. Input parameter "S_LightCurtainID" is then controlled using this signal.

Function description

The signal connected to input parameter "S_LightCurtainID" is processed by the function block.

The signal input processes the state of the electro-sensitive protective equipment.

The signal input is state-controlled. These states only lead to connection results if the function block is enabled ("Activate" = TRUE).

6.2.3.1.2.3 S_TableID

General function

- Assigns a table object to the function block

Data type

- TableID

Connection

- Constant

Function description

This input parameter assigns a table object to the function block.

Connect the corresponding table object to this input parameter in SafeDESIGNER using drag-and-drop.

> **Information:**
>
> **The same "TableID" can be used in multiple table function blocks of type C in the safety application.**

6.2.3.1.2.4 S_CycleFilter

General function

- Suppresses the safety requests of a light curtain for the specified number of light curtain cycles

Data type

- SAFEWORD

Connection

- Constant

Function description

The cycle-filter functionality suppresses the safety requests of a light curtain for a specified number of light curtain cycles. As a result, a light curtain state must be present and stable for the light curtain cycles set on input parameter "S_CycleFilter" before a response is given.

Note the following points when using "S_CycleFilter":

- This parameter is influenced by the light curtain cycle. This is not the cycle time of the SafeLOGIC controller!
- For information about the light curtain cycle, see the documentation for the respective light curtain.
- The parameter has an immediate effect on the response time. The function block output signal is delayed by "S_CycleFilter" * Light curtain cycle.
- The default value is 0 light curtain cycles.
- The function block provides the suppressed signal on the output.

Range of values: 0 to 5 light curtain cycles

The following figure shows that output signal "S_XXX" is shifted by one cycle if "S_CycleFilter" = 1.

The upper area of the figure shows a suppressed signal. Since the bit only switches to 0 for one cycle (n+1), the output remains at 1. In contrast, the signal in the lower area indicates 0 over 2 cycles (n+1 and n+2). In this case, the output is set to 0.

S_CycleFilter = 1

	n	n+1	n+2	n+3
Status	1	0	1	1
S_XXX	1	1	1	1

	n	n+1	n+2	n+3
Status	1	0	0	1
S_XXX	1	1	0	0

Figure 348: "S_CycleFilter" - Functionality

Legend:

n	Light curtain cycle
Status	Internally evaluated state of the function block (not visible to the user)
S_XXX	Function block enable output acted upon by "S_CycleFilter"

6.2.3.1.2.5 S_AutoReset

General function

- Specification of the start interlock if correct signals are present on the input parameters

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If a variable is used to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified using a constant, then you only have to validate the startup behavior for the specified value.**

Function description

This input parameter determines the operating behavior of the function block after correct signals have returned to the input parameters.

TRUE

After correct signals return to the input parameters, the function block does not support a start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination for this is valid.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After correct signals return to the input parameters, the function block supports a start interlock.

You must change "Reset" from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination for this is valid.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.2.3.1.2.6 Reset

General function

- Input parameter for resetting error messages if the error has been corrected
  or
- Input parameter for supporting a manual reset device if a start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset"

Data type

- BOOL

Depending on the safety requirements, data type SAFEBOOL or BOOL must be used for the connection. A SAFEBOOL connection avoids unexpected startups that may result from errors in the standard (non-safe) system. The risk of unexpected startup can be reduced with further measures such as an additional function stop.

Connection

- Variable

Function description

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.2.3.1.3 Output parameters

Description of the function block output parameters.

6.2.3.1.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.2.3.1.3.2 S_NotMasked

General function

- State of the unmasked beams

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter returns the state of the beams from the unmasked area.

TRUE

A safety request is not present for the beams in the unmasked area.

FALSE

A safety request is present for the beams in the unmasked area.

6.2.3.1.3.3 S_Masked

General function

- State of the masked beams

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter returns the state of the beams from the masked areas.

TRUE

A safety request is not present for the beams in the masked areas.

FALSE

A safety request is present for the beams in the masked areas.

6.2.3.1.3.4 S_Overall

General function

- Function block release signal for the entire muting area

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter returns the state of the entire muting area. This includes both the masked ("S_Masked") and unmasked ("S_NotMasked") areas.

TRUE

A safety request is not present in any of the areas.

FALSE

A safety request is present in at least one of the areas.

6.2.3.1.3.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.2.3.1.3.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.2.3.1.4 Status numbers

Errors

Code (hex)	Description	Corrective measures
C001	After the safety request is reset, the function block detected a static TRUE signal on "Reset".	Check the control device that controls input parameter "Reset" and the corresponding wiring.
C002	Data and/or "S_LightCurtainID" is invalid.	Check the validity of the value on "S_LightCurtainID".
C003	The state of the light curtain is invalid.	Check the state of the light curtain.
C004	The table data is invalid. The table data is not suitable for the configured light curtain (e.g. the table references beams that are not physically present).	Check the entries in the table and the selected light curtain.
C005	An input parameter is outside the defined limit values.	Configure valid values for "S_CycleFilter".

Table 495: "SF_BeamMask": Error codes

Status information

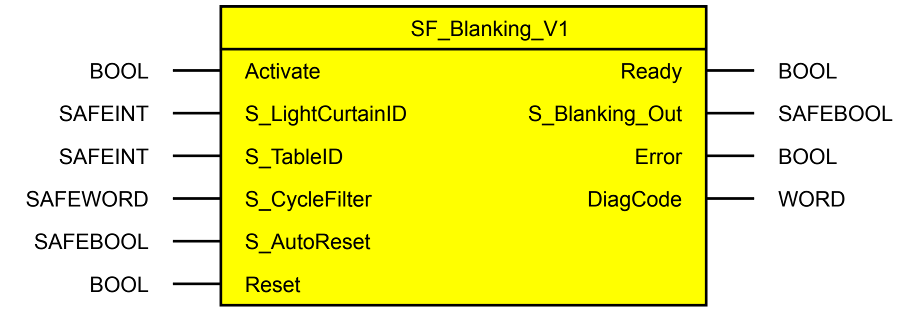
Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	A safety request is not present in any of the areas.	No corrective measures are required.
8Y03	Y = Error that must be reset. The safety request or error state was reset. The function block is waiting for a reset.	Acknowledge the state with a rising edge on "Reset".
8003	The function block detected error "C003".	Acknowledge the state with a rising edge on "Reset".
8203	The function block detected an error in status "8012".	Acknowledge the state with a rising edge on "Reset".
8303	The function block detected an error in status "8013".	Acknowledge the state with a rising edge on "Reset".
8011	A safety request is present in one or more masked areas.	Cancel the safety request.
8012	A safety request is present in the unmasked area.	Cancel the safety request.
8013	A safety request is present in both the masked and unmasked areas.	Cancel the safety request.

Table 496: "SF_BeamMask": Diagnostic codes

6.2.3.2 SF_Blanking

This function block makes it possible to use blanking functions.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_LightCurtainID	SAFEINT	Constant	0	Reference to the object of the electro-sensitive protective equipment (light curtain)
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
IN	S_CycleFilter	SAFEWORD	Constant	0	Suppresses the safety requests of a light curtain for the specified number of light curtain cycles. Range of values: 0 to 5 light curtain cycles
IN	S_AutoReset	SAFEBOOL	Variable/ Constant	FALSE	Specifies the start interlock if proper signals are present on the input parameters. TRUE: The start interlock is not active.
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_Blanking_Out	SAFEBOOL	Variable	FALSE	Function block release signal. TRUE: A safety request is not present for the specified blanking objects and protected light curtain areas.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.2.3.2.1 Function description

Function block "SF_Blanking" evaluates connected and configured blanking objects.
The information originates from a blanking table (see Table format D (blanking)). This is specified to the function block on input parameter "S_TableID". The configured objects are monitored during cyclic execution. The function block returns an error if there is a deviation.

Evaluation is performed according to the following rules:

Fixed blanking, see parameter "Surveillance" (1):

- The position/dimensions (± tolerance) of the defined object must match exactly with the object in the light curtain.
- All defined objects must be present at the light curtain.

Partial monitoring, see parameter "Surveillance" (0):

- The position/dimensions (± tolerance) of the object is not monitored.

The following points must be noted when using function block "SF_Blanking" and the associated table format:

- Areas of the light curtain that deviate from the specified object table are not permitted to be interrupted at any time.
- Entries in the table correspond to the order of objects at the light curtain.
- At least one unobstructed beam must be between objects in the light curtain; otherwise, the object dimensions will not be perceived correctly. Overlaps or changes in the position of objects result in an error.
- The function block itself checks whether the objects defined in the table are in the light curtain area (maximum position < number of beams).

6.2.3.2.2 Input parameters

Description of the function block input parameters.

6.2.3.2.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- You also have the option of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.2.3.2.2.2 S_LightCurtainID

General function

- Reference to the object of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEINT

Connection

- Constant

Information:

Connect this input parameter to the signal of a safe device that is connected to the electro-sensitive protective equipment and supports the openSAFETY Vision profile. Input parameter "S_LightCurtainID" is then controlled using this signal.

Function description

The signal connected to input parameter "S_LightCurtainID" is processed by the function block.

The signal input processes the state of the electro-sensitive protective equipment.

The signal input is state-controlled. These states only lead to connection results if the function block is enabled ("Activate" = TRUE).

6.2.3.2.2.3 S_TableID

General function

- Assigns a table object to the function block

Data type

- TableID

Connection

- Constant

Function description

This input parameter assigns a table object to the function block.

Connect the corresponding table object to this input parameter in SafeDESIGNER using drag-and-drop.

> ## Information:
> **The same "TableID" can be used in multiple table function blocks of type D in the safety application.**

6.2.3.2.2.4 S_CycleFilter

General function

- Suppresses the safety requests of a light curtain for the specified number of light curtain cycles

Data type

- SAFEWORD

Connection

- Constant

Function description

The cycle-filter functionality suppresses the safety requests of a light curtain for a specified number of light curtain cycles. As a result, a light curtain state must be present and stable for the light curtain cycles set on input parameter "S_CycleFilter" before a response is given.

Note the following points when using "S_CycleFilter":

- This parameter is influenced by the light curtain cycle. This is not the cycle time of the SafeLOGIC controller!
- For information about the light curtain cycle, see the documentation for the respective light curtain.
- The parameter has an immediate effect on the response time. The function block output signal is delayed by "S_CycleFilter" * Light curtain cycle.
- The default value is 0 light curtain cycles.
- The function block provides the suppressed signal on the output.

Range of values: 0 to 5 light curtain cycles

The following figure shows that output signal "S_XXX" is shifted by one cycle if "S_CycleFilter" = 1.

The upper area of the figure shows a suppressed signal. Since the bit only switches to 0 for one cycle (n+1), the output remains at 1. In contrast, the signal in the lower area indicates 0 over 2 cycles (n+1 and n+2). In this case, the output is set to 0.

S_CycleFilter = 1

	n	n+1	n+2	n+3
Status	1	0	1	1
S_XXX	1	1	1	1

	n	n+1	n+2	n+3
Status	1	0	0	1
S_XXX	1	1	0	0

Figure 349: "S_CycleFilter" - Functionality

Legend:

n	Light curtain cycle
Status	Internally evaluated state of the function block (not visible to the user)
S_XXX	Function block enable output acted upon by "S_CycleFilter"

6.2.3.2.2.5 S_AutoReset

General function

- Specification of the start interlock if correct signals are present on the input parameters

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If a variable is used to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified using a constant, then you only have to validate the startup behavior for the specified value.**

Function description

This input parameter determines the operating behavior of the function block after correct signals have returned to the input parameters.

TRUE

After correct signals return to the input parameters, the function block does not support a start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination for this is valid.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After correct signals return to the input parameters, the function block supports a start interlock.

You must change "Reset" from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination for this is valid.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.2.3.2.2.6 Reset

General function

- Input parameter for resetting error messages if the error has been corrected
  or
- Input parameter for supporting a manual reset device if a start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset"

Data type

- BOOL

Depending on the safety requirements, data type SAFEBOOL or BOOL must be used for the connection. A SAFEBOOL connection avoids unexpected startups that may result from errors in the standard (non-safe) system. The risk of unexpected startup can be reduced with further measures such as an additional function stop.

Connection

- Variable

Function description

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.2.3.2.3 Output parameters

Description of the function block output parameters.

6.2.3.2.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.2.3.2.3.2 S_Blanking_Out

General function

- Function block release signal

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter specifies whether a safety request is present for the specified blanking objects.

TRUE

A safety request is not present for the specified blanking objects and protected light curtain areas.

FALSE

A safety request is present for the specified blanking objects or protected light curtain areas. This could be a monitoring violation (check "Surveillance") or a blanking error (see Table format D (blanking)).

6.2.3.2.3.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.2.3.2.3.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.2.3.2.4 Status numbers

Errors

Code (hex)	Description	Corrective measures
C001	After the safety request is reset, the function block detected a static TRUE signal on "Reset".	Check the control device that controls input parameter "Reset" and the corresponding wiring.
C002	Data and/or "S_LightCurtainID" is invalid.	Check the validity of the value on "S_LightCurtainID".
C003	The state of the light curtain is invalid.	Check the state of the light curtain.
C004	The table data is invalid. The table data is not suitable for the configured light curtain (e.g. the table references beams that are not physically present).	Check the entries in the table and the selected light curtain.
C005	An input parameter is outside the defined limit values.	Configure valid values for "S_CycleFilter".
C006	Safety request. Beams in the unmasked area were interrupted or monitoring requests are not present (check "Surveillance").	Check the unmasked area and existence or dimensions of the objects if "Surveillance" is configured.

Table 497: "SF_Blanking": Error codes

Status information

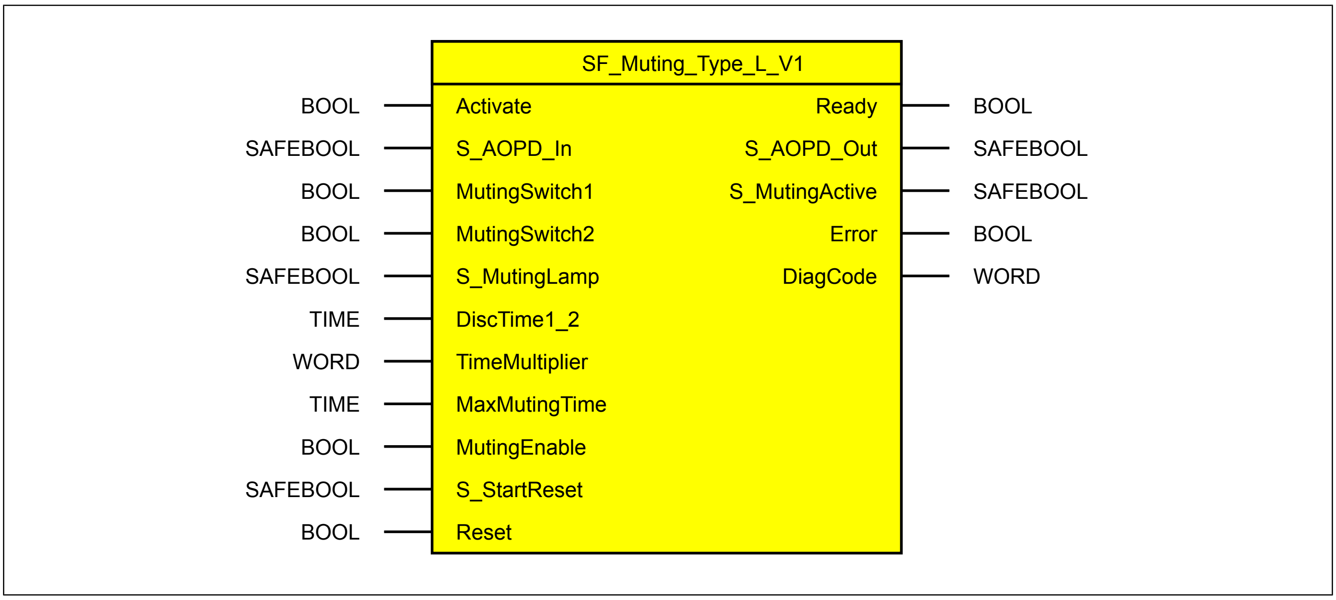
Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	Monitoring of the blanking function is active.	No corrective measures are required.
8Y03	Y = Error that must be reset. The safety request or error state was reset. The function block is waiting for a reset.	Acknowledge the state with a rising edge on "Reset".
8003	The function block detected error "C003".	Check the state of the light curtain.
8103	The function block detected error "C006".	Check the unmasked area and existence or dimensions of the objects if "Surveillance" is configured.

Table 498: "SF_Blanking": Diagnostic codes

6.2.3.3 SF_Muting_Type_L

This function block makes it possible to implement L-type muting.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_AOPD_In	SAFEBOOL	Variable	FALSE	Signal input of the protective equipment (light curtain). TRUE: The connected protective equipment is not damped.
IN	MutingSwitch1	BOOL	Variable	FALSE	Signal input of muting sensor 1. TRUE: The muting sensor is damped.
IN	MutingSwitch2	BOOL	Variable	FALSE	Signal input of muting sensor 2. TRUE: The muting sensor is damped.
IN	S_MutingLamp	SAFEBOOL	Variable/ Constant	FALSE	Feedback signal of the muting lamp. TRUE: The function of the muting lamp is not impaired.
IN	DiscTime1_2	TIME	Constant	0#ms	Specification of the maximum discrepancy time between muting sensor 1 and muting sensor 2. Range of values: 1 to 16 s
IN	TimeMultiplier	WORD	Constant	0	Multiplication factor for the value of the timer of the existing muting function. Range of values: 2 to 16
IN	MaxMutingTime	TIME	Constant	0#ms	Specification of the maximum time for the complete muting process. Range of values: 1 s to 10 min
IN	MutingEnable	BOOL	Variable/ Constant	FALSE	Specification of the start of the muting process. TRUE: Starting the muting function is possible.
IN	S_StartReset	SAFEBOOL	Variable/ Constant	FALSE	Specification of the start interlock after the function block is enabled and/or after a cold restart of the safety controller. TRUE: The start interlock is not active.
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_AOPD_Out	SAFEBOOL	Variable	FALSE	Function block release signal. TRUE: The process to be controlled is enabled. No stop function is active.
OUT	S_MutingActive	SAFEBOOL	Variable	FALSE	State of the muting process. TRUE: The muting process is enabled and being executed.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.2.3.3.1 Function description

Function block "SF_Muting_Type_L" provides support for L-type muting in an application.

L-type muting with 2 sensors is used to transport material out of the danger zone.

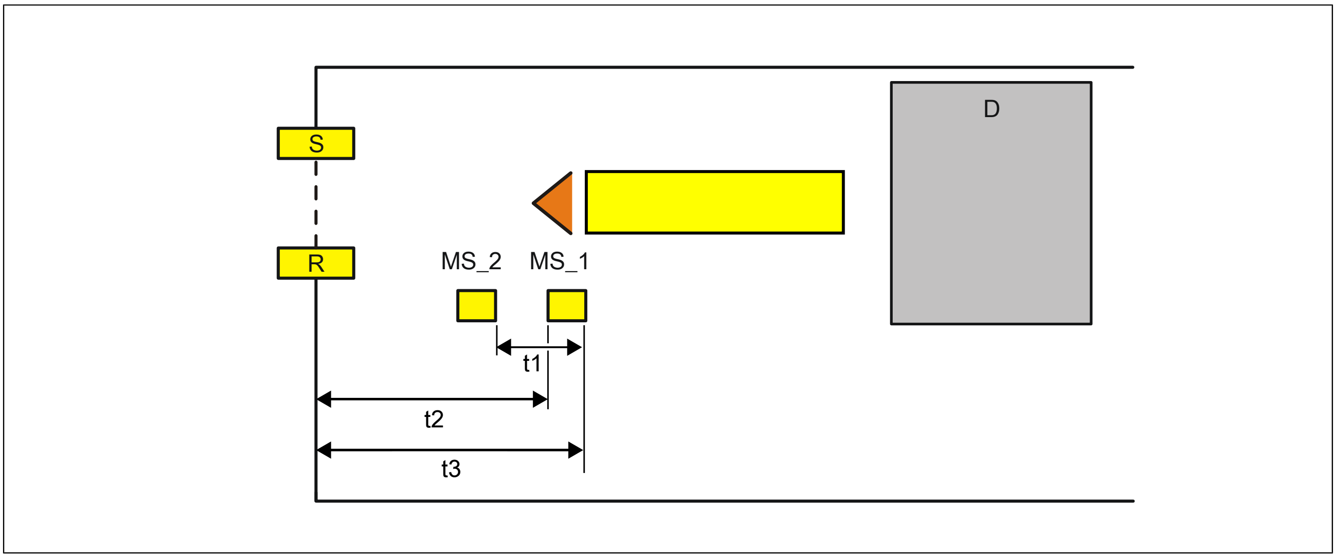


Figure 350: Schematic diagram of muting elements

Legend:

MS_1	Muting sensor 1
MS_2	Muting sensor 2
S, R	Transmitter (send) and receiver (receive) of the light curtain
D	Danger zone
t1	Sensor interval between a rising edge on "MutingSwitch1" and rising edge on "MutingSwitch2"
t2	"TimeMultiplier" * t1. Value "TimeMultiplier" must be specified by the user (permissible values: 2 to 16).
t3	Specification on "MaxMutingTime"

The device executes the muting function if the input parameters are switched in the following order:

- First, input parameter "MutingEnable" must be set to TRUE; "MutingSwitch1" must then be set to TRUE. This starts evaluation on "MaxMutingTime" and "DiscTime1_2".
- Active muting starts when input parameter "MutingSwitch2" is set to TRUE. The sensor interval between a rising edge on "MutingSwitch1" and rising edge on "MutingSwitch2" is measured as a time (t1).
- If "MutingSwitch2" is enabled before "MutingSwitch1", this does not correspond to a valid muting sequence and an error will be output.
- If "MutingSwitch1" is set to FALSE, the evaluation of "DiscTime1_2" is started. "MutingSwitch2" must be set to FALSE before this time expires.
- If the described muting sequence is not completed within the time specified on "MaxMutingTime", this will result in an error.

Input parameter "DiscTime1_2" specifies the maximum delay between "MutingSwitch1" and "MutingSwitch2" when enabling/disabling.
This value must be defined by the user (permissible values: 1 to 16 seconds).

The muting function ends after time "t2".

The difference with the other muting functions is the interval between the second muting sensor and the electro-sensitive protective equipment. This is variable and a multiple of time "t2".

6.2.3.3.2 Input parameters

Description of the function block input parameters.

6.2.3.3.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.2.3.3.2.2 S_AOPD_In

General function

- Signal input of the protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the protective equipment (e.g. light curtain) in the muting application over 1 or 2 channels. Input parameter "S_AOPD_In" is then controlled using this signal.**

Function description

The function block evaluates the state of the connected protective equipment using the signal connected on input parameter "S_AOPD_In".

Regardless of whether the protective equipment is connected to the safe device over 1 or 2 channels, "S_AOPD_In" is only connected to one signal.

If protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected protective equipment is not damped.

FALSE

The connected protective equipment is damped.

If muting is active, "S_AOPD_Out" does not initiate the safe state.
If muting is inactive, "S_AOPD_Out" initiates the safe state.

6.2.3.3.2.3 MutingSwitch1

General function

- Signal input of muting sensor 1

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_1" in the muting application. Input parameter "MutingSwitch1" is then controlled using this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch1". The result of the evaluation causes the muting process to be started or stopped. Whether the connected muting sensor is in an invalid state is also detected.

Note the following points if you are using a safe input device to evaluate the muting sensors.

Regardless of whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch1" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch1". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch1". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.2.3.3.2.4 MutingSwitch2

General function

- Signal input of muting sensor 2

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_2" in the muting application. Input parameter "MutingSwitch2" is then controlled using this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch2". The result of the evaluation causes the muting process to be started or stopped. Whether the connected muting sensor is in an invalid state is also detected.

Note the following points if you are using a safe input device to evaluate the muting sensors.

Regardless of whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch2" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch2". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch2". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.2.3.3.2.5 S_MutingLamp

General function

- Feedback signal of the muting lamp

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

Connect this input parameter to the signal of a safe device that is connected to the feedback signal of the muting lamp in the muting application over 1 or 2 channels. Input parameter "S_MutingLamp" is then controlled using this signal.

Information:

If your risk analysis determines that a muting lamp is not required in your muting application, note that you can specify constant TRUE on this input parameter.

Function description

The function block evaluates the state of the connected muting lamp (lamp operational / not operational) using the signal connected on input parameter "S_MutingLamp".

Note that the feedback signal of the muting lamp must permanently indicate state TRUE if its function is not impaired. If the lamp function is impaired, the feedback signal must permanently indicate state FALSE.

Regardless of whether the muting lamp is connected to the safe device over 1 or 2 channels, "S_MutingLamp" is only connected to one signal.

If the muting lamp is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The function of the muting lamp is not impaired.

FALSE

The function of the muting lamp is impaired.

6.2.3.3.2.6 DiscTime1_2

General function

- Specification of the maximum discrepancy time between muting sensor 1 and muting sensor 2

Data type

- TIME

Connection

- Constant

Function description

The function block starts the muting process if "MutingSwitch1" and "MutingSwitch2" switch from FALSE to TRUE within the specified time on "DiscTime1_2". If the second required signal change on "MutingSwitch1" and "MutingSwitch2" does not take place during this time frame, enable output "S_AOPD_Out" initiates the safe state (FALSE).

Likewise, when the signal on "MutingSwitch1" switches from TRUE to FALSE, whether "MutingSwitch2" is also disabled within time "DiscTime1_2" is checked.

You must define and validate the time value for input parameter "DiscTime1_2" based on your application and risk analysis.

Range of values: 1 to 16 seconds

6.2.3.3.2.7 TimeMultiplier

General function

- Multiplication factor for the value of the timer of the existing muting function

Data type

- WORD

Connection

- Constant

Function description

This input parameter specifies the multiplication factor for the value of the timer of the existing muting function.

Permissible values: 2 to 16

6.2.3.3.2.8 MaxMutingTime

General function

- Specification of the maximum time for the complete muting process

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the maximum time for the complete muting process. This time starts when the signal on "MutingSwitch1" changes from FALSE to TRUE.

You must define and validate the time value for input parameter "MaxMutingTime" based on your application and risk analysis.

Range of values: 1 second to 10 minutes

> **Information:**
>
> **Note that the minimum value can lock the muting process if it is too low.**
> **Make sure that this value is greater than the result of multiplying "t2" (see Fig. 350 "Schematic diagram of muting elements").**

6.2.3.3.2.9 MutingEnable

General function

- Specification of the start of the muting process

Data type

- BOOL

Connection

- Variable or constant

Information:

Control this input parameter using a signal from the standard application that enables the muting process. If your risk analysis determines that you do not need a release signal from the standard application, alternatively specify constant TRUE.

Function description

Input parameter "MutingEnable" receives the starting signal for enabling the muting process from the standard application. This is a measure for reducing the risk of an unintended muting process.

TRUE

Starting the muting function is possible.

FALSE

Starting the muting function is not possible.

6.2.3.3.2.10 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or after a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
> **If a variable is used to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified using a constant, then you only have to validate the startup behavior for the specified value.**

Function description

This input parameter defines the startup behavior of the function block after it is enabled and/or after a cold restart of the safety controller.

TRUE

After the function block is enabled, it does not support start interlocks.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination for this is valid.

An unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.

FALSE

After the function block is enabled, it supports a start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination for this is valid.

If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.

The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur when starting the safety controller.

6.2.3.3.2.11 Reset

General function

- Input parameter for resetting error messages if the error has been corrected
  or
- Input parameter for supporting a manual reset device if a start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset"

Data type

- BOOL

Depending on the safety requirements, data type SAFEBOOL or BOOL must be used for the connection. A SAFEBOOL connection avoids unexpected startups that may result from errors in the standard (non-safe) system. The risk of unexpected startup can be reduced with further measures such as an additional function stop.

Connection

- Variable

Function description

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.2.3.3.3 Output parameters

Description of the function block output parameters.

6.2.3.3.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.2.3.3.3.2 S_AOPD_Out

General function

- Function block release signal

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **Connect this output parameter to the safety application in such a way that the safety application takes on and maintains the safe state if signal FALSE is output.**

Function description

The release signal is the safe enable signal for the safeguarded area that is used to control an output on a safe device and therefore the process. This output parameter indicates the state of the protective device in the muting application. The release signal is controlled based on the state of the protective device and start interlock.

In addition, the release signal controls the request for the stop function. Control the stop function of the connected safety application by connecting "S_AOPD_Out" accordingly.

Since the release signal is present on output "S_AOPD_Out", this output is referred to as the "enable output".

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- And: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = TRUE (light curtain)
- Or: The muting process is active and the function block did not detect an invalid muting sequence.
- And: A start interlock is not active.
- And: The function block did not detect an error.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

Possible causes:

- The function block is not enabled ("Activate" = FALSE).
- And: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = FALSE (light curtain).
- Or: The muting process is active and the function block detected an invalid muting sequence.
- Or: A start interlock is active.
- Or: The function block detected an error.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_AOPD_Out" is only set to TRUE if input parameter "S_AOPD_In" indicates state TRUE and a reset has been carried out (no start interlock active).

The following table explains this behavior in detail.

Input parameter		Action	Start interlock	Reset	Enable output
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_AOPD_Out" to TRUE if the input signal combination is valid. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 499: "SF_Muting_Type_L": Input parameter "S_StartReset"

6.2.3.3.3.3 S_MutingActive

General function

- State of the muting process

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **If your risk analysis determines that you must use a muting lamp in the muting application, then connect this output parameter to a safe output device that is connected to the muting lamp.**
>
> **In addition, connect this output parameter to the safety application in such a way that the safety application is controlled according to the muting state.**

Function description

This output parameter indicates whether a muting process is enabled and being executed.

TRUE

The function block has been enabled ("Activate" = TRUE).

The muting process is enabled ("MutingEnable" = TRUE) and being executed. "S_AOPD_In" = FALSE does not result in the safe state on "S_AOPD_Out" (FALSE).

FALSE

The muting process is not enabled ("MutingEnable" = FALSE). "S_AOPD_In" = FALSE results in the safe state on "S_AOPD_Out" (FALSE).

6.2.3.3.3.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.2.3.3.3.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.2.3.3.4 Status numbers

Errors

Code (hex)	Description	Corrective measures
C001	Constant reset request in status "8001".	Reset the function block.
C002	Constant reset request in status "8003".	Acknowledge the state with a rising edge on "Reset".
C003	The feedback signal of the muting lamp is invalid.	Check the signal on "S_MutingLamp".
CYx4	Y = State in which the error occurred. x = State of "MutingSwitch1" and "MutingSwitch2" (LSB = "MutingSwitch1", LSB + 1 bit = "MutingSwitch2") An error in the muting sequence has occurred.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch1" and "MutingSwitch2" must indicate state FALSE. The 2 muting sensors connected to "MutingSwitch1" and "MutingSwitch2" must be undamped for this. • Reset the function block if "MutingSwitch1" and "MutingSwitch2" indicate state FALSE.
0xC0x4	An error in status "0x8000" occurred.	
0xC1x4	An error in status "0x8011" occurred.	
0xC2x4	An error in status "0x8012" occurred.	
0xC3x4	An error in status "0x8013" occurred.	
0xCFx4	"MutingEnable" was not requested.	
C005	An input parameter is outside the defined limit values.	Check the configured values.
C006	The muting process was not yet completed when time monitoring of the maximum muting time expired.	Check the value on "MaxMutingTime" and/or the muting process.

Table 500: "SF_Muting_Type_L": Error codes

Status information

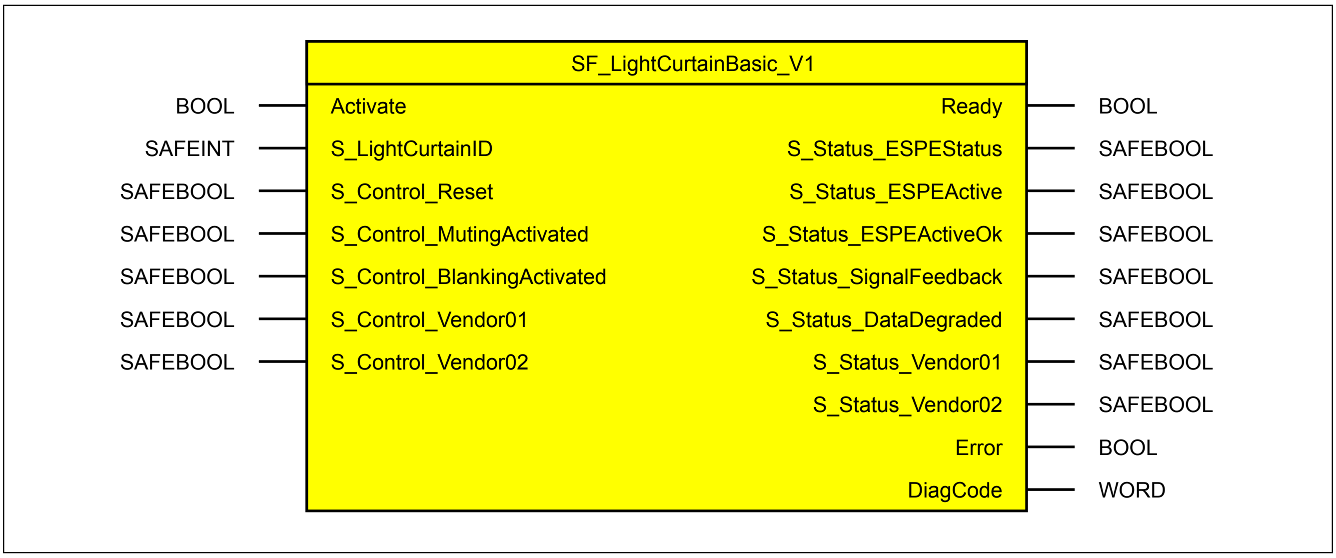
Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	A safety request is not present and muting is inactive.	No corrective measures are required.
8001	Function block initialization.	No corrective measures are required.
8002	A safety request is pending and muting is inactive.	Check the value on "S_AOPD_In".
8003	The safety request or error state was reset. The function block is waiting for a reset.	Acknowledge the state with a rising edge on "Reset".
8005	The safety function was activated.	No corrective measures are required.
8011	The muting process was requested on "MutingSwitch1". Waiting for "MutingSwitch2".	No corrective measures are required.
8012	The muting process is active. Waiting for "MutingSwitch1" and "MutingSwitch2" to be disabled again.	No corrective measures are required.
8013	The muting process is active. Waiting for the muting time to expire.	No corrective measures are required.

Table 501: "SF_Muting_Type_L": Diagnostic codes

6.2.3.4 SF_LightCurtainBasic

This function block returns status information and transmits commands to a connected device that supports the openSAFETY Vision profile.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_LightCurtainID	SAFEINT	Constant	0	Reference to the object of the electro-sensitive protective equipment (light curtain)
IN	S_Control_Reset	SAFEBOOL	Variable/ Constant	FALSE	Resets error messages and the electro-sensitive protective equipment (light curtain) if the cause of error is no longer present. RTRIG: Acknowledgment
IN	S_Control_MutingActivated	SAFEBOOL	Variable/ Constant	FALSE	Enables the muting process. TRUE: The muting process is enabled.
IN	S_Control_BlankingActivated	SAFEBOOL	Variable/ Constant	FALSE	Enables the blanking process. TRUE: The blanking process is enabled.
IN	S_Control_Vendor01	SAFEBOOL	Variable/ Constant	FALSE	Selects/Deselects vendor-specific safety function "Vendor01"
IN	S_Control_Vendor02	SAFEBOOL	Variable/ Constant	FALSE	Selects/Deselects vendor-specific safety function "Vendor02"
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_Status_ESPEStatus	SAFEBOOL	Variable	FALSE	State of the electro-sensitive protective equipment (light curtain) and indication of a non-resettable state. TRUE: An error did not occur.
OUT	S_Status_ESPEActive	SAFEBOOL	Variable	FALSE	Indication of the current state of the electro-sensitive protective equipment (light curtain). TRUE: The electro-sensitive protective equipment is in state "Operational" and no beams are interrupted.
OUT	S_Status_ESPEActiveOk	SAFEBOOL	Variable	FALSE	Indication of a resettable error state of the electro-sensitive protective equipment (light curtain). TRUE: An error state is not present.
OUT	S_Status_SignalFeedback	SAFEBOOL	Variable	FALSE	Indication of active muting/blanking. TRUE: Muting and/or blanking is being indicated.
OUT	S_Status_DataDegraded	SAFEBOOL	Variable	FALSE	Indication of required maintenance work on the electro-sensitive protective equipment (light curtain). TRUE: Maintenance work is required.
OUT	S_Status_Vendor01	SAFEBOOL	Variable	FALSE	Status information of vendor-specific safety function "Vendor01"
OUT	S_Status_Vendor02	SAFEBOOL	Variable	FALSE	Status information of vendor-specific safety function "Vendor02"
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.2.3.4.1 Function description

This function block returns status information and writes commands to a device that supports the openSAFETY Vision profile.

6.2.3.4.2 Input parameters

Description of the function block input parameters.

6.2.3.4.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- You also have the option of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.2.3.4.2.2 S_LightCurtainID

General function

- Reference to the object of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEINT

Connection

- Constant

> ## Information:
>
> **Connect this input parameter to the signal of a safe device that is connected to the electro-sensitive protective equipment and supports the openSAFETY Vision profile. Input parameter "S_LightCurtainID" is then controlled using this signal.**

Function description

The signal connected to input parameter "S_LightCurtainID" is processed by the function block.

The signal input processes the state of the electro-sensitive protective equipment.

The signal input is state-controlled. These states only lead to connection results if the function block is enabled ("Activate" = TRUE).

6.2.3.4.2.3 S_Control_Reset

General function

- Resets error messages and the electro-sensitive protective equipment (light curtain) if the cause of error is no longer present

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

This input parameter confirms an error or startup of the electro-sensitive protective equipment (light curtain).

The function block internally monitors edge transitions for this input parameter. The reset function is only executed on a rising edge. A continuing static TRUE signal following a rising edge does not trigger the function again. The function block detects a static TRUE signal on "S_Control_Reset" as an error in states where a rising edge on "S_Control_Reset" is required. Set "S_Control_Reset" to FALSE to exit this error state.

6.2.3.4.2.4 S_Control_MutingActivated

General function

- Enables the muting process

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

This input parameter enables the muting process.

TRUE

The muting process is enabled.
The requested safety functions are being executed by the function block.

FALSE

The muting process is not enabled.
The requested safety functions are not being executed by the function block.

6.2.3.4.2.5 S_Control_BlankingActivated

General function

- Enables the blanking process

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

This input parameter enables the blanking process.

TRUE

The blanking process is enabled.
The requested safety functions are being executed by the function block.

FALSE

The blanking process is not enabled.
The requested safety functions are not being executed by the function block.

6.2.3.4.2.6 S_Control_Vendor01

General function

- Selects/Deselects vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

"Vendor01" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective light curtain.**

6.2.3.4.2.7 S_Control_Vendor02

General function

- Selects/Deselects vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

"Vendor02" is a vendor-specific safety function.

> ## Information:
> **For information about this input parameter, see the documentation for the respective light curtain.**

6.2.3.4.3 Output parameters

Description of the function block output parameters.

6.2.3.4.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.2.3.4.3.2 S_Status_ESPEStatus

General function

- State of the electro-sensitive protective equipment (light curtain) and indication of a non-resettable state

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether a non-resettable error state (e.g. hardware defect) occurred in the electro-sensitive protective equipment (light curtain).

TRUE

An error did not occur.

FALSE

A non-resettable error occurred.

6.2.3.4.3.3 S_Status_ESPEActive

General function

- Indication of the current state of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates the current state of the electro-sensitive protective equipment (light curtain).

TRUE

The electro-sensitive protective equipment (light curtain) is in state "Operational" and no beams are interrupted.

FALSE

The electro-sensitive protective equipment (light curtain) is triggered.

6.2.3.4.3.4 S_Status_ESPEActiveOk

General function

- Indication of a resettable error state of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether an error state of the electro-sensitive protective equipment (light curtain) must be reset. Output parameter "S_Status_ESPEStatus" is not affected by this.

TRUE

An error state is not present.

FALSE

A reset is required.
For more information, see the documentation for the respective light curtain.

6.2.3.4.3.5 S_Status_SignalFeedback

General function

- Indication of active muting/blanking

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether active muting and/or blanking is being indicated by the electro-sensitive protective equipment (light curtain). This signal can be used to control a required muting lamp, for example.

TRUE

Muting and/or blanking is being indicated.

FALSE

Neither muting nor blanking is being indicated.

6.2.3.4.3.6 S_Status_DataDegraded

General function

- Indication of required maintenance work on the electro-sensitive protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates that the transferred data is valid but that maintenance on the electro-sensitive protective equipment (light curtain) is urgently recommended. This could be due to dirty lenses, for example.

TRUE

Maintenance work is required.

FALSE

Maintenance work is not required, or the function is not supported.

6.2.3.4.3.7 S_Status_Vendor01

General function

- Status information of vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates the state of "Vendor01".

> **Information:**
>
> **For information about this state, see the documentation for the respective light curtain.**

6.2.3.4.3.8 S_Status_Vendor02

General function

- Status information of vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates the state of "Vendor02".

> ## Information:
> **For information about this state, see the documentation for the respective light curtain.**

6.2.3.4.3.9 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.2.3.4.3.10 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.2.3.4.4 Status numbers

Errors

Code (hex)	Description	Corrective measures
C001	The function set for the control byte was not found.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).
C002	The function set for the status byte was not found.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).
C003	The function set ID that was read does not match.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).
C004	The data length of the function set that was read is invalid.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).
C005	The status byte could not be read.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).
C006	The control byte could not be written.	Check whether a current, functioning OSDD file has been imported for the electro-sensitive protective equipment (light curtain).

Table 502: "SF_LightCurtainBasic": Error codes

Status information

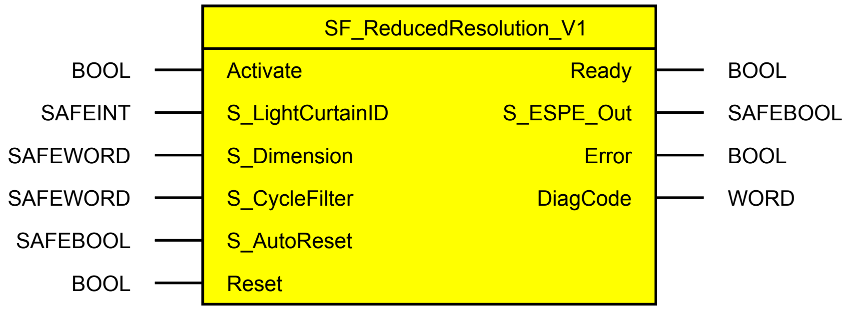
Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block was executed successfully.	No corrective measures are required.

Table 503: "SF_LightCurtainBasic": Diagnostic codes

6.2.3.5 SF_ReducedResolution

This function block makes it possible to reduce the resolution for a device that supports the openSAFETY Vision profile.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_LightCurtainID	SAFEINT	Constant	0	Reference to the object of the electro-sensitive protective equipment (light curtain)
IN	S_Dimension	SAFEWORD	Variable/ Constant	0	Specification of the maximum object dimensions (measured in beams) that are permitted to pass through the electro-sensitive protective equipment (light curtain). If the object is subject to vibrations, a tolerance must be added. Range of values: 0 to 256 beams
IN	S_CycleFilter	SAFEWORD	Constant	0	Suppresses the safety requests of a light curtain for the specified number of light curtain cycles. Range of values: 0 to 5 light curtain cycles
IN	S_AutoReset	SAFEBOOL	Variable/ Constant	FALSE	Specifies the start interlock if proper signals are present on the input parameters. TRUE: The start interlock is not active.
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists. RTRIG: Acknowledgment
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_ESPE_Out	SAFEBOOL	Variable	FALSE	Function block release signal. TRUE: A safety request is not present for the reduced resolution.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.2.3.5.1 Function description

This function is used for small objects (cables, particulate matter, wood shavings, etc.) that are permitted to cross the protected area.

With reduced resolution, objects that are smaller than specified on input parameter "S_Dimension" do not trigger a safety request when they pass through the electro-sensitive protective equipment. The number of objects that are permitted to interrupt the electro-sensitive protective equipment is unlimited.

The objects are permitted to move. At least one beam must be free between objects in the light curtain, however; otherwise, the object dimensions will not be perceived correctly. Overlaps or changes in the position of objects result in an error.

> **Danger!**
>
> **Recalculation of the safety distance**
>
> **A reduced resolution can increase the required minimum distance between the light curtain and danger zone.**
>
> **To calculate the correct safety distance, observe the instructions in EN ISO 13855.**

6.2.3.5.2 Input parameters

Description of the function block input parameters.

6.2.3.5.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- You also have the option of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.2.3.5.2.2 S_LightCurtainID

General function

- Reference to the object of the electro-sensitive protective equipment (light curtain)

Data type

- SAFEINT

Connection

- Constant

Information:

Connect this input parameter to the signal of a safe device that is connected to the electro-sensitive protective equipment and supports the openSAFETY Vision profile. Input parameter "S_LightCurtainID" is then controlled using this signal.

Function description

The signal connected to input parameter "S_LightCurtainID" is processed by the function block.

The signal input processes the state of the electro-sensitive protective equipment.

The signal input is state-controlled. These states only lead to connection results if the function block is enabled ("Activate" = TRUE).

6.2.3.5.2.3 S_Dimension

General function

- Specification of the maximum object dimensions (measured in beams) that are permitted to pass through the electro-sensitive protective equipment (light curtain)

Data type

- SAFEWORD

Connection

- Variable or constant

Function description

This input parameter specifies the maximum object dimensions that are permitted to pass through the electro-sensitive protective equipment (light curtain).

Since the function block does not have a dedicated tolerance input, this must be taken into account when configuring "S_Dimension".

Permissible values: 0 to 256 beams

6.2.3.5.2.4 S_CycleFilter

General function

- Suppresses the safety requests of a light curtain for the specified number of light curtain cycles

Data type

- SAFEWORD

Connection

- Constant

Function description

The cycle-filter functionality suppresses the safety requests of a light curtain for a specified number of light curtain cycles. As a result, a light curtain state must be present and stable for the light curtain cycles set on input parameter "S_CycleFilter" before a response is given.

Note the following points when using "S_CycleFilter":

- This parameter is influenced by the light curtain cycle. This is not the cycle time of the SafeLOGIC controller!
- For information about the light curtain cycle, see the documentation for the respective light curtain.
- The parameter has an immediate effect on the response time. The function block output signal is delayed by "S_CycleFilter" * Light curtain cycle.
- The default value is 0 light curtain cycles.
- The function block provides the suppressed signal on the output.

Range of values: 0 to 5 light curtain cycles

The following figure shows that output signal "S_XXX" is shifted by one cycle if "S_CycleFilter" = 1.

The upper area of the figure shows a suppressed signal. Since the bit only switches to 0 for one cycle (n+1), the output remains at 1. In contrast, the signal in the lower area indicates 0 over 2 cycles (n+1 and n+2). In this case, the output is set to 0.

S_CycleFilter = 1

	n	n+1	n+2	n+3
Status	1	0	1	1
S_XXX	1	1	1	1

	n	n+1	n+2	n+3
Status	1	0	0	1
S_XXX	1	1	0	0

Figure 351: "S_CycleFilter" - Functionality

Legend:

n	Light curtain cycle
Status	Internally evaluated state of the function block (not visible to the user)
S_XXX	Function block enable output acted upon by "S_CycleFilter"

6.2.3.5.2.5 S_AutoReset

General function

- Specification of the start interlock if correct signals are present on the input parameters

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

If a variable is used to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified using a constant, then you only have to validate the startup behavior for the specified value.

Function description

This input parameter determines the operating behavior of the function block after correct signals have returned to the input parameters.

TRUE

After correct signals return to the input parameters, the function block does not support a start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination for this is valid.

Danger!

Automatic startup

Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.

FALSE

After correct signals return to the input parameters, the function block supports a start interlock.

You must change "Reset" from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination for this is valid.

Danger!

If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.

The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.

6.2.3.5.2.6 Reset

General function

- Input parameter for resetting error messages if the error has been corrected
  or
- Input parameter for supporting a manual reset device if a start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset"

Data type

- BOOL

Depending on the safety requirements, data type SAFEBOOL or BOOL must be used for the connection. A SAFEBOOL connection avoids unexpected startups that may result from errors in the standard (non-safe) system. The risk of unexpected startup can be reduced with further measures such as an additional function stop.

Connection

- Variable

Function description

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.2.3.5.3 Output parameters

Description of the function block output parameters.

6.2.3.5.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.2.3.5.3.2 S_ESPE_Out

General function

- Function block release signal

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter specifies whether a safety request is present for the reduced resolution.

TRUE

A safety request is not present for the reduced resolution.

FALSE

A safety request is present for the reduced resolution.

6.2.3.5.3.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.2.3.5.3.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.2.3.5.4 Status numbers

Errors

Code (hex)	Description	Corrective measures
C001	The function block detected a static TRUE signal on "Reset".	Check the control device that controls input parameter "Reset" and the corresponding wiring.
C002	Data and/or "S_LightCurtainID" is invalid.	Check the validity of the value on "S_LightCurtainID".
C003	The state of the light curtain is invalid.	Check the state of the light curtain.
C005	An input parameter is outside the defined limit values.	Configure valid values for "S_Dimension" and "S_CycleFilter".
C006	Safety request. An area of beams larger than configured was detected.	Check the objects located in the area of the electro-sensitive protective equipment (light curtain).

Table 504: "SF_ReducedResolution": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The blanking process is active. A safety request is not present.	No corrective measures are required.
8Y03	Y = Error that must be reset. The safety request or error state was reset. The function block is waiting for a reset.	Acknowledge the state with a rising edge on "Reset".
8003	The function block detected error "C003".	Check the state of the light curtain.
8103	The function block detected error "C006".	Check the objects located in the area of the electro-sensitive protective equipment (light curtain).

Table 505: "SF_ReducedResolution": Diagnostic codes

6.3 Math_Utilities_SF

Library "Math_Utilities_SF" makes it possible to implement floating-point functions using SAFEDINT data types.

If a scaled value is needed for the calculation, the scaling is performed via input "S_Scale". This is provided as a SAFEINT data type and interpreted as 10 to the power of "S_Scale" (10^{S_Scale}). As a result, it is possible to represent large numbers (>SAFEDINT) as well as numbers between -1 and +1. If a function block has input "S_Scale", then the value shown on output "S_Out" corresponds to the scaled value.

6.3.1 System requirements

Library "Math_Utilities_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements apply in order to use library "Math_Utilities_SF":

- SafeDESIGNER: 4.3.1 or later
- Automation Studio: 4.3.3 or later
- Automation Runtime: 4.33 or later
- SafeLOGIC: Safety Release 1.10 or later
- SafeLOGIC-X: Safety Release 1.10 or later
- Possession of a software license for using library "Math_Utilities_SF"

6.3.2 Version history

Version	Date	Comment
1.11	May 2019	Added error number C007 for "Exp_S_BR".
1.10	May 2017	First edition

Table 506: Version history

6.3.3 Technical information

6.3.3.1 Error handling

Error handling is to be implemented such that the function blocks detect errors and inform the user via output "DiagCode". Examples of detected errors include value range violations and division by zero.

In the event of error, the safety controller does not change to state "FailSafe"; instead, it continues cyclic execution of the safety application. The error description is provided on output "DiagCode" of the affected function block.

The result is provided on the function block's "S_Out" output as a SAFEDINT value. Note that the user must ensure the validity of this value by checking the function block's "S_OutOK" output, which is of type SAFEBOOL. An invalid value can lead to undesirable results if processed further. This should be avoided in all cases. TRUE on output "S_OutOK" indicates a valid value. FALSE indicates that the returned value is invalid.

6.3.3.2 Scaling and rounding values

If the function block has input "S_Scale", then values will be scaled.
Scaling involves the application of a scaling factor in order to implement floating-point number calculations, for example. This factor is set via "S_Scale" and based on 10^{S_Scale}.

Scaling examples:

- To scale value 2.3 to value 23, set "S_Scale" = 1.
- To scale value 4375873210 to value 43758732, set "S_Scale" = -2.

For which value the scaling factor will be applied to, see the function description for the respective function block.

Information:

Note in the calculation that output "S_Out" returns the scaled value.

Information:

The calculated values for output "S_Out" are mathematically rounded to whole numbers.

This means:

- **Values from 0 up to and including 4 are rounded down.**
- **Values from 5 up to and including 9 are rounded up.**

Example: A calculated value of 0.4 will return value 0 on the output. In contrast, a calculated value of 0.5 will result in value 1 on the output.

6.3.4 Function blocks

List of the function blocks included in this library sorted in descending order according to the version from which they are available.

Supported starting with SafeDESIGNER	Name	Short description
4.3.1	ABS_S_BR	This function block returns the absolute value of the input value.
4.3.1	ADD_S_BR	This function block performs addition with 2 safe inputs.
4.3.1	DIV_S_BR	This function block performs division with 2 safe inputs.
4.3.1	EXP_S_BR	This function block makes it possible to calculate exponential functions with 2 safe inputs.
4.3.1	MAX_S_BR	This function block compares the 2 input values and returns the maximum value.
4.3.1	MIN_S_BR	This function block compares the 2 input values and returns the minimum value.
4.3.1	MOD_S_BR	This function block performs a modulo operation with 2 safe inputs.
4.3.1	MUL_S_BR	This function block performs multiplication with 2 safe inputs.
4.3.1	ROOT_S_BR	This function block performs a root calculation with 2 safe inputs.
4.3.1	SCALE_S_BR	This function block makes it possible to scale a safe value of data type SAFEDINT.
4.3.1	SUB_S_BR	This function block performs subtraction with 2 safe inputs.

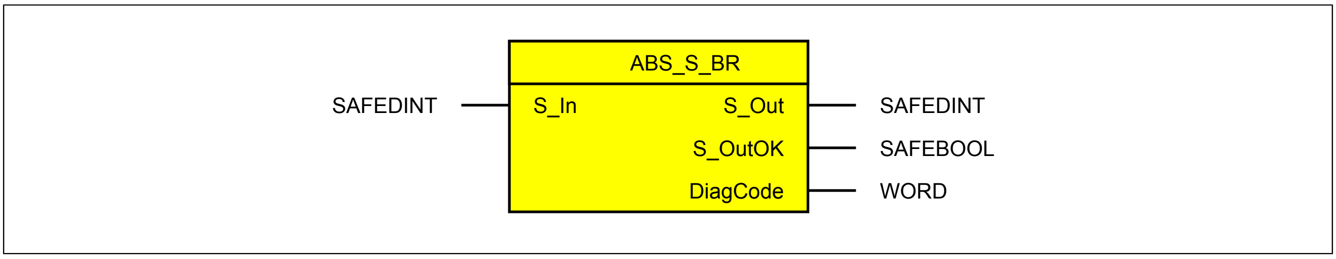
6.3.4.1 ABS_S_BR

This function block returns the absolute value of the input value.

The calculation is performed as follows:

$S\_Out = ABS(S\_In)$

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In	SAFEDINT	Variable	0	Value
OUT	S_Out	SAFEDINT	Variable	0	Absolute value
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.1.1 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

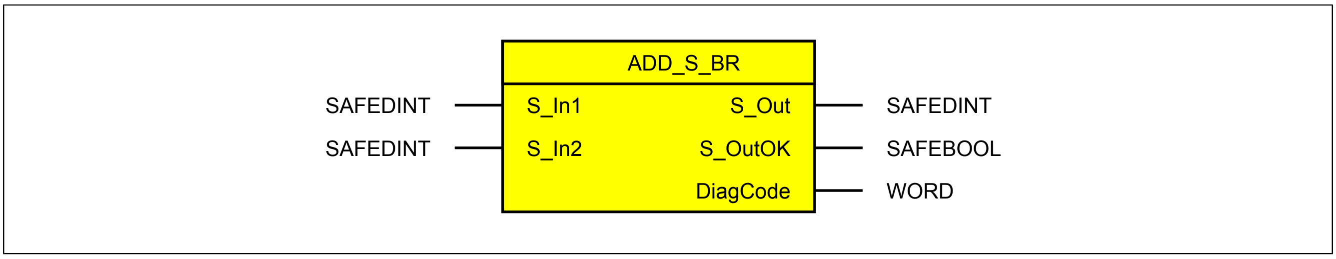
6.3.4.2 ADD_S_BR

This function block performs addition with 2 safe inputs.

The calculation is performed as follows:

$S\_Out = S\_In1 + S\_In2$

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	First summand
IN	S_In2	SAFEDINT	Variable	0	Second summand
OUT	S_Out	SAFEDINT	Variable	0	Sum
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.2.1 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.3 DIV_S_BR

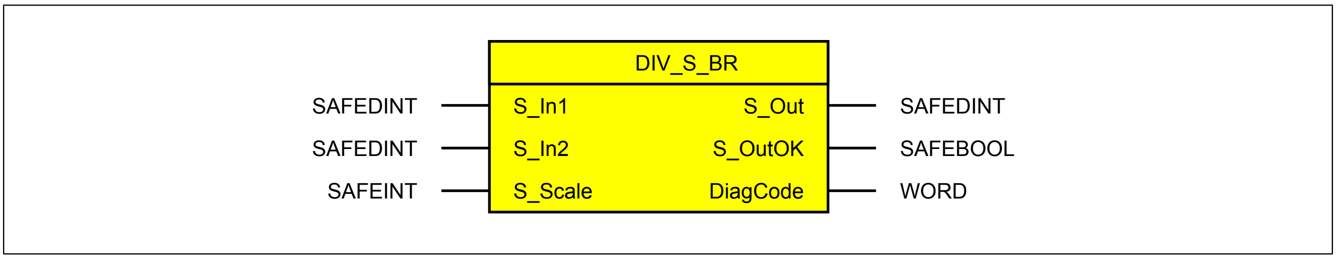
This function block performs division with 2 safe inputs. In order to perform calculations with floating-point numbers, for example, a scaling factor (input "S_Scale") is available. This is necessary for the function block to know which factor was applied to the inputs in order to perform the correct arithmetic operation. The result is represented as a scaled value on output "S_Out".

The calculation is performed as follows:

$$S\_Out = \frac{S\_In1}{S\_In2}$$

> **Information:**
>
> **Note in the calculation that output "S_Out" returns the scaled value.**

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Dividend
IN	S_In2	SAFEDINT	Variable	0	Divisor
IN	S_Scale	SAFEINT	Variable	0	Base 10 (10^{S_Scale}) scaling parameter
OUT	S_Out	SAFEDINT	Variable	0	Quotient (scaled value)
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.3.1 Function description

Scaling floating-point numbers results in rounded values on output "S_Out".
For additional information, see section "Scaling and rounding values" on page 1139.

The sections below provide examples of the following situations that must be taken into account when using the function block:

- Dividend greater than divisor
- Dividend less than divisor
- Floating-point numbers: Dividend greater than divisor
- Floating-point numbers: Dividend less than divisor

Dividend greater than divisor

$$\frac{600}{15} = 40$$

Inputs and outputs	Values
S_In1	600
S_In2	15
S_Scale	0
S_Out	40
S_OutOK	TRUE

Dividend less than divisor

In this example, it is important that an appropriate scaling factor (input "S_Scale") is used; otherwise, an underflow can occur. In this case, output "S_Out" returns value 0, and "DiagCode" returns error code C010.

$$\frac{4}{200} = 0.02$$

Inputs and outputs	Values
S_In1	40
S_In2	2000
S_Scale	1
S_Out	0
S_OutOK	FALSE
DiagCode	C010

To receive a value greater than 0, the selected scaling factor must be correspondingly high.

Inputs and outputs	Values
S_In1	400
S_In2	20000
S_Scale	2
S_Out	2
S_OutOK	TRUE

Floating-point numbers: Dividend greater than divisor

In this example, the scaling factor affects the precision of the result.

$$\frac{0.82}{0.5} = 1.64$$

Inputs and outputs	Values
S_In1	82
S_In2	50
S_Scale	2
S_Out	164
S_OutOK	TRUE

Floating-point numbers: Dividend less than divisor

In this example, it is important that an appropriate scaling factor (input "S_Scale") is used; otherwise, an underflow can occur. In this case, output "S_Out" returns value 0, and "DiagCode" returns error code C010.

$$\frac{1.9}{20} = 0.095$$

Inputs and outputs	Values
S_In1	19
S_In2	200
S_Scale	1
S_Out	1
S_OutOK	TRUE

To increase precision, a larger scaling factor must be selected.

Inputs and outputs	Values
S_In1	1900
S_In2	20000
S_Scale	3
S_Out	95
S_OutOK	TRUE

The next example shows division with 2 floating-point numbers where the dividend is less than the divisor.

$$\frac{0.19}{0.8} = 0.2375$$

Inputs and outputs	Values
S_In1	19
S_In2	80
S_Scale	2
S_Out	24
S_OutOK	TRUE

To increase precision, a larger scaling factor must be selected.

Inputs and outputs	Values
S_In1	1900
S_In2	8000
S_Scale	4
S_Out	2375
S_OutOK	TRUE

6.3.4.3.2 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.
C002	Division by zero has taken place.	Check the input values.
C010	Underflow: The result of the calculation is too small to be displayed.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.4 EXP_S_BR

This function block makes it possible to calculate exponential functions with 2 safe inputs of data type SAFEDINT. In order to perform calculations with a floating-point number (input "S_In"), for example, a scaling factor (input "S_Scale") is available. The result is represented as a scaled value.

The calculation is performed as follows:

$S\_Out = S\_In^{S\_Exp}$

Information:

Note in the calculation that the scaling factor (input "S_Scale") only affects input "S_In" and therefore also the calculated result.
Also note that exponents (input "S_Exp") must be whole numbers.

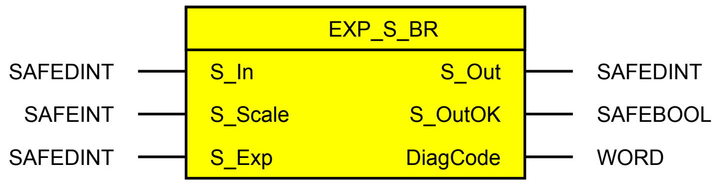
Information:

Note in the calculation that output "S_Out" returns the scaled value.

Information:

The root function is made possible by function block **"ROOT_S_BR" on page 1154****.**

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In	SAFEDINT	Variable	0	Base
IN	S_Scale	SAFEINT	Variable	0	Base 10 (10^{S_Scale}) scaling parameter
IN	S_Exp	SAFEDINT	Variable	0	Exponent. Permissible range: -30 to 30
OUT	S_Out	SAFEDINT	Variable	0	Power (scaled value)
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.4.1 Function description

Scaling floating-point numbers results in rounded values on output "S_Out".
For additional information, see section "Scaling and rounding values" on page 1139.

The sections below provide examples of the following situations that must be taken into account when using the function block:

- Negative exponent
- Floating-point number as a base

Negative exponent

In this example, it is important that an appropriate scaling factor (input "S_Scale") is used; otherwise, an underflow can occur. In this case, output "S_Out" returns value 0, and "DiagCode" returns error code C010.

$100^{-1} = 0.01$

Inputs and outputs	Values
S_In	100
S_Scale	0
S_Exp	-1
S_Out	0
S_OutOK	FALSE
DiagCode	C010

To receive a value greater than 0, the selected scaling factor must be correspondingly high.

Inputs and outputs	Values
S_In	10000
S_Scale	2
S_Exp	-1
S_Out	1
S_OutOK	TRUE

Floating-point number as a base

In this example, it is important that an appropriate scaling factor (input "S_Scale") is used; otherwise, an underflow can occur. In this case, output "S_Out" returns value 0, and "DiagCode" returns error code C010.

$0.2^{3} = 0.008$

Inputs and outputs	Values
S_In	2
S_Scale	1
S_Exp	3
S_Out	0
S_OutOK	FALSE
DiagCode	C010

To receive a value greater than 0, the selected scaling factor must be correspondingly high.

Inputs and outputs	Values
S_In	200
S_Scale	3
S_Exp	3
S_Out	8
S_OutOK	TRUE

6.3.4.4.2 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.
C002	The value on input "S_In" is 0, and there is a negative value on input "S_Exp".	Check the input values.
C006	The valid range of values of "S_Exp" was violated.	Check the input value.
C007	The value on input "S_In" is 0; "S_Exp" also indicates value 0.	Check the input values.
C010	Underflow: The result of the calculation is too small to be displayed.	Check the input values.

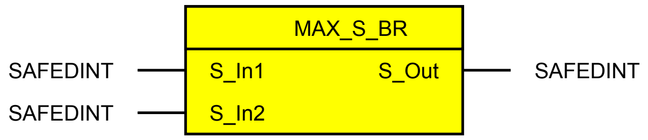
Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.5 MAX_S_BR

This function block compares the 2 input values and returns the maximum value.

Function block



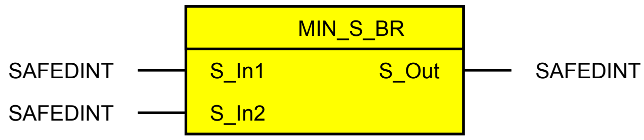
Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Value 1
IN	S_In2	SAFEDINT	Variable	0	Value 2
OUT	S_Out	SAFEDINT	Variable	0	Maximum value of the comparison values

6.3.4.6 MIN_S_BR

This function block compares the 2 input values and returns the minimum value.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Value 1
IN	S_In2	SAFEDINT	Variable	0	Value 2
OUT	S_Out	SAFEDINT	Variable	0	Minimum value of the comparison values

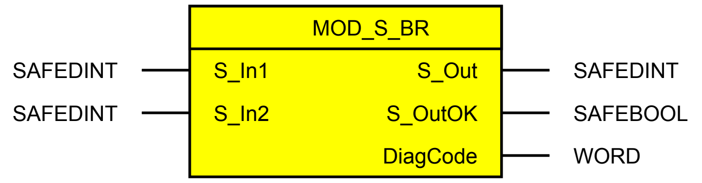
6.3.4.7 MOD_S_BR

This function block performs a modulo operation with 2 safe inputs.

The calculation is performed as follows:

$S\_Out = S\_In1 \ \mathrm{mod} \ S\_In2$

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Dividend
IN	S_In2	SAFEDINT	Variable	0	Divisor
OUT	S_Out	SAFEDINT	Variable	0	Modulo after division of dividend by divisor
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.7.1 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C002	Division by zero has taken place.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.8 MUL_S_BR

This function block performs multiplication with 2 safe inputs. In order to perform calculations with floating-point numbers, for example, a scaling factor (input "S_Scale") is available. This is necessary for the function block to know which factor was applied to the inputs in order to perform the correct arithmetic operation. The result is represented as a scaled value on output "S_Out".

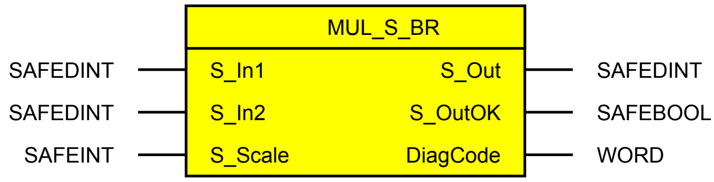
The calculation is performed as follows:

$S\_Out = S\_In1 \times S\_In2$

Information:

Note in the calculation that output "S_Out" returns the scaled value.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Factor
IN	S_In2	SAFEDINT	Variable	0	Factor
IN	S_Scale	SAFEINT	Variable	0	Base 10 (10^{S_Scale}) scaling parameter
OUT	S_Out	SAFEDINT	Variable	0	Product (scaled value)
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.8.1 Function description

Scaling floating-point numbers results in rounded values on output "S_Out".
For additional information, see section "Scaling and rounding values" on page 1139.

The sections below provide examples of the following situations that must be taken into account when using the function block:

- Multiplication of 2 whole numbers
- Multiplication of a whole number by a floating-point number
- Multiplication of 2 floating-point numbers

Multiplication of 2 whole numbers

$317 \times 211 = 66887$

Inputs and outputs	Values
S_In1	317
S_In2	211
S_Scale	0
S_Out	66887
S_OutOK	TRUE

To multiply 2 very large numbers that would exceed the range of data type SAFEDINT, the scaling factor (input "S_Scale") can be used to scale the numbers down.

$125321441 \times 4581181 = 574120204401821$

Inputs and outputs	Values
S_In1	125321
S_In2	4581
S_Scale	-3
S_Out	0
S_OutOK	FALSE
DiagCode	C001

Multiplication of a whole number by a floating-point number

When multiplying a whole number by a floating-point number, it is important that the specified values have the same base; otherwise, the calculation will be distorted.

$243 \times 1.23 = 298.89$

Inputs and outputs	Values
S_In1	24300
S_In2	123
S_Scale	2
S_Out	29889
S_OutOK	TRUE

In this case, forgetting to scale the whole number ("S_In1") will result in the function block not returning the desired result.

Inputs and outputs	Values
S_In1	243
S_In2	123
S_Scale	2
S_Out	299
S_OutOK	TRUE

Multiplication of 2 floating-point numbers

When multiplying 2 floating-point numbers, it is important that the specified values have the same base; otherwise, the calculation will be distorted.

$1.13 \times 2.2 = 2.486$

Inputs and outputs	Values
S_In1	113
S_In2	220
S_Scale	2
S_Out	249
S_OutOK	TRUE

If the product requires a precision of 3 decimal places in this case, this can be achieved using a higher scaling factor. When doing so, it is important to adjust the input values accordingly.

Inputs and outputs	Values
S_In1	1130
S_In2	2200
S_Scale	3
S_Out	2486
S_OutOK	TRUE

6.3.4.8.2 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.
C010	Underflow: The result of the calculation is too small to be displayed.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.9 ROOT_S_BR

This function block performs a root calculation with 2 safe inputs of data type SAFEDINT. In order to perform calculations with a floating-point number (input "S_In"), for example, a scaling factor (input "S_Scale") is available. The result is represented as a scaled value.

The calculation is performed as follows:

$$S\_Out = \sqrt[S\_Exp]{S\_In}$$

Information:

Note the following with regard to the calculation:

- **The scaling factor (input "S_Scale") only affects input "S_In" and therefore also the calculated result.**
- **Only positive whole numbers are used for nth root (input "S_Exp").**
- **Complex numbers are not supported. That means that if the radicand (input "S_In") is negative, only odd nth roots (input "S_Exp") will produce valid results. If an even nth root is used, the function block will detect an error (C004) and provide it on output "DiagCode".**

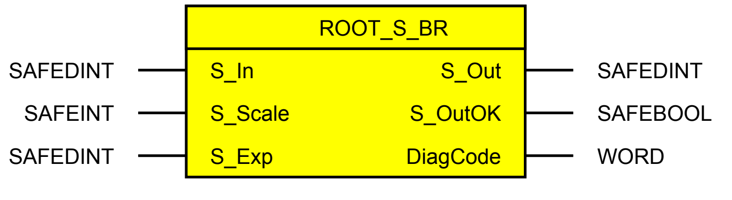
Information:

Note in the calculation that output "S_Out" returns the scaled value.

Information:

The calculation of exponential functions is made possible by function block "EXP_S_BR" on page 1146.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In	SAFEDINT	Variable	0	Radicand
IN	S_Scale	SAFEINT	Variable	0	Base 10 (10^{S_Scale}) scaling parameter
IN	S_Exp	SAFEDINT	Variable	1	Nth root (min. 1)
OUT	S_Out	SAFEDINT	Variable	0	Root (scaled value)
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.9.1 Function description

Scaling floating-point numbers results in rounded values on output "S_Out".
For additional information, see section "Scaling and rounding values" on page 1139.

The sections below provide examples of the following situations that must be taken into account when using the function block:

- Radicand is a whole number
- Radicand is a floating-point number
- Radicand is a negative number

Radicand is a whole number

$\sqrt[3]{80} = 4.3088694$

Inputs and outputs	Values
S_In	80
S_Scale	0
S_Exp	3
S_Out	4
S_OutOK	TRUE

To increase precision, a larger scaling factor must be selected.

Inputs and outputs	Values
S_In	800
S_Scale	1
S_Exp	3
S_Out	43
S_OutOK	TRUE

Radicand is a floating-point number

$\sqrt[3]{0.8} = 0.9283178$

Inputs and outputs	Values
S_In	8
S_Scale	1
S_Exp	3
S_Out	9
S_OutOK	TRUE

To increase precision, a larger scaling factor must be selected.

Inputs and outputs	Values
S_In	800
S_Scale	3
S_Exp	3
S_Out	928
S_OutOK	TRUE

Radicand is a negative number

$\sqrt[3]{-800} = -9.2831777$

Inputs and outputs	Values
S_In	-8000
S_Scale	1
S_Exp	3
S_Out	-93
S_OutOK	TRUE

To increase precision, a larger scaling factor must be selected.

Inputs and outputs	Values
S_In	-80000
S_Scale	2
S_Exp	3
S_Out	-928
S_OutOK	TRUE

$\sqrt[2]{-60}$

To use negative radicands (input "S_In"), odd nth roots must be used. If even nth roots are used, output "S_Out" returns value 0 and "DiagCode" returns error code C004.

Inputs and outputs	Values
S_In	-600
S_Scale	1
S_Exp	2
S_Out	0
S_OutOK	FALSE
DiagCode	C004

6.3.4.9.2 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.
C003	The exponent ("S_Exp") is a negative number.	Check the input values.
C004	The value on input "S_In" is negative, and there is an even number on input "S_Exp".	Check the input values. "S_Exp" must be an odd whole number. "S_Out" is not permitted to be a complex number.
C005	The exponent ("S_Exp") has value 0.	Check the input values.
C010	Underflow: The result of the calculation is too small to be displayed.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.3.4.10 SCALE_S_BR

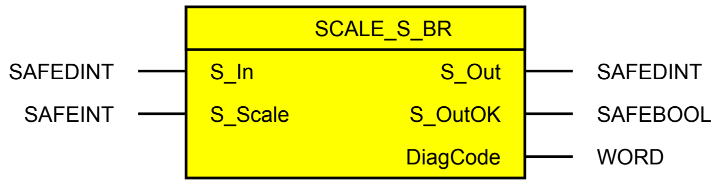
This function block makes it possible to scale a safe value of data type SAFEDINT. The base used for scaling is represented by 10^{S_Scale}, where whole number value "S_Scale" can be selected as either positive or negative. The result is represented as a scaled value.

The calculation is performed as follows:

$$S\_Out = SCALE(S\_In)$$

> **Information:**
>
> **Note in the calculation that output "S_Out" returns the scaled value.**

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In	SAFEDINT	Variable	0	Value
IN	S_Scale	SAFEINT	Variable	0	Base 10 (10^{S_Scale}) scaling parameter
OUT	S_Out	SAFEDINT	Variable	0	Value (scaled value)
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.10.1 Function description

Scaling floating-point numbers results in rounded values on output "S_Out".
For additional information, see section "Scaling and rounding values" on page 1139.

The sections below provide examples of the following situations that must be taken into account when using the function block:

- Positive scaling value on input "S_Scale"
- Negative scaling value on input "S_Scale"

Positive scaling value on input "S_Scale"

63 to 6300

Inputs and outputs	Values
S_In	63
S_Scale	2
S_Out	6300

Negative scaling value on input "S_Scale"

7350 to 7

Inputs and outputs	Values
S_In	7350
S_Scale	-3
S_Out	7

6.3.4.10.2 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.
C010	Underflow: The result of the calculation is too small to be displayed.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

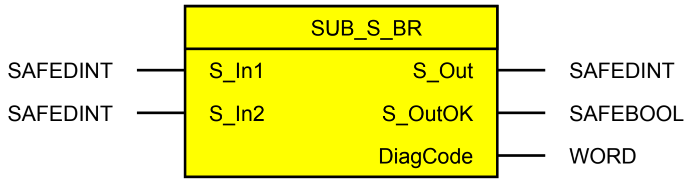
6.3.4.11 SUB_S_BR

This function block performs subtraction with 2 safe inputs.

The calculation is performed as follows:

$$S\_Out = S\_In1 - S\_In2$$

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_In1	SAFEDINT	Variable	0	Minuend
IN	S_In2	SAFEDINT	Variable	0	Subtrahend
OUT	S_Out	SAFEDINT	Variable	0	Difference
OUT	S_OutOK	SAFEBOOL	Variable	FALSE	Indicates valid output data. TRUE: Output data is valid. FALSE: Output data is invalid.
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.3.4.11.1 Status numbers

Information:

Each error message causes output "S_OutOK" to be set to FALSE since the values on "S_Out" are invalid.

Error

Code (hex)	Description	Corrective measures
C001	The valid range of values was violated.	Check the input values.

Status information

Code (hex)	Description	Corrective measures
8000	The output data is valid.	No corrective measures are required.

6.4 openSAFETY_Motion_SF

This library makes it possible to use openSAFETY Motion profiles.

Connection "S_AxisID" references the axis to be used. This axis ID is made available as a constant by SafeDESIGNER.

> ### Danger!
> **Make sure that the correct "S_AxisID" is always used on the input!**
> **Each assignment must be validated separately.**

All other connections correspond to the requirements or the status of the safety functions made available by the safe axis.

> ### Danger!
> **For information about integrated safety functions and the safe encoder connection, see the documentation for the safe drive.**

> ### Information:
> **Library "openSAFETY_Motion_SF" can only be used to control servo drives that support the openSAFETY Motion profile (e.g. ACOPOS P3, 3rd-party, etc.).**

> ### Information:
> **This library covers the entire openSAFETY Motion profile.**
> **For information about actual supported functions and their behavior, see the manufacturer documentation for the safe drive.**

> ### Information:
> **When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.**

> ### Information:
> **Only the inputs of the safety function that are actually used in the safety application are permitted to be linked.**
> **Linking an input of a safety function to TRUE or FALSE is not permitted since this identifies the function as being used but does not allow it to be tested!**

6.4.1 Overview

Overview of function blocks in library "openSAFETY_Motion_SF".

Function block	Description
SF_oS_MOTION_Basic	Basic set of the openSAFETY Motion profile
SF_oS_MOTION_Speed	Speed extension set of the openSAFETY Motion profile
SF_oS_MOTION_Advanced	Advanced extension set of the openSAFETY Motion profile
SF_oS_MOTION_EncoderBasic	Encoder basic set of the openSAFETY Motion profile
SF_oS_MOTION_Data_Acceleration	Links the safe acceleration of an axis and the associated status
SF_oS_MOTION_Data_Position	Links the safe position of an axis and the associated status
SF_oS_MOTION_Data_Speed	Links the safe speed of an axis and the associated status of the encoder error
SF_oS_MOTION_Data_Torque	Links the safe torque of an axis and the associated status

6.4.2 System requirements

Library "openSAFETY_Motion_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements apply in order to use library "openSAFETY_Motion_SF":

- SafeDESIGNER: 4.2.2 or later
- Automation Studio: 4.2.5 or later
- SafeLOGIC: Safety Release 1.10 or later
- SafeLOGIC-X: Safety Release 1.10 or later; X20(c)SLX910 supports max. 1 axis
- Axes with profile support (ACOPOSmulti SafeMOTION does not support profiles)
- When using a B&R drive: The safety functions being used must be unlocked using a Technology Guard.
- When using a 3rd-party drive: A software license on the SafeKEY is required to use library "openSAFETY_Motion_SF".

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.3 Term definitions

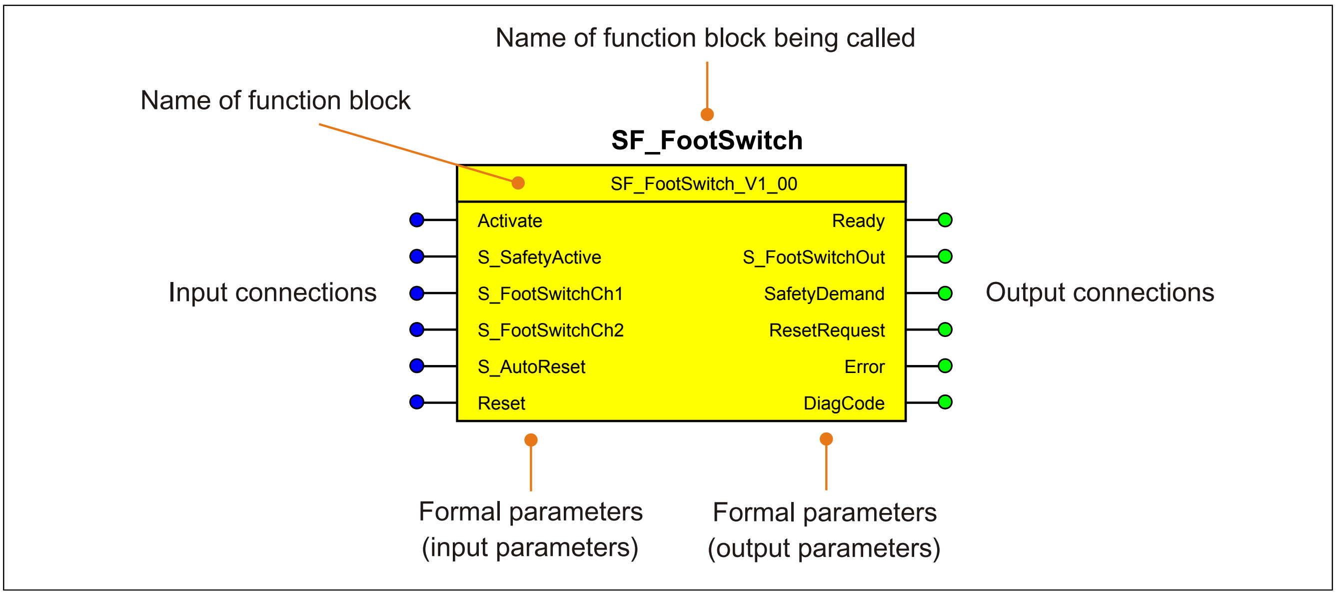


Figure 352: Components of a function block

When calling a function block, the inputs supply the input parameters with the current values of the variables or constants.
The output parameters supply the outputs with the associated values.

Inputs and outputs do not need to have the same name as the associated formal parameters, but they must be of the same data type. A difference in data type between formal parameters and inputs/outputs is reported as an error after compilation.

The name of a function block is composed of the function itself (e.g. "SF_FootSwitch", SF = safety function) and its version (Vx_yz). The format used to represent the version number in this document, Vx_yz, is a placeholder. For the actual version, see the function block being used.

6.4.4 SF_oS_MOTION_Basic

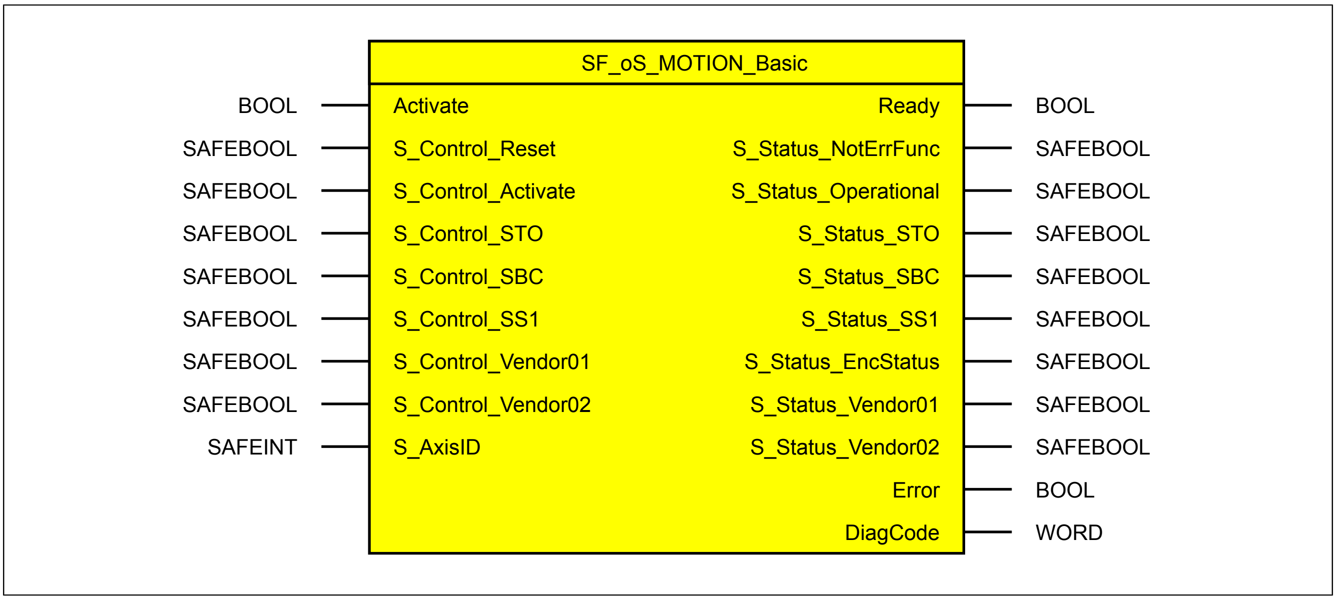


Figure 353: Function block "SF_oS_MOTION_Basic"

6.4.4.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Control_Reset	SAFEBOOL	Variable	Edge	FALSE	Resets error messages and the safe axis after the cause of the error has been removed
S_Control_Activate	SAFEBOOL	Variable/ Constant	Status	FALSE	Enables the safe state of the axis and safety functions. FALSE: Sets the state machine of the safe axis to state IDLE
S_Control_STO	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Torque Off" (STO). FALSE: The safety function is selected.
S_Control_SBC	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Brake Control" (SBC). FALSE: The safety function is selected.
S_Control_SS1	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Stop 1" (SS1). FALSE: The safety function is selected.
S_Control_Vendor01	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor01"
S_Control_Vendor02	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor02"
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 507: "SF_oS_MOTION_Basic": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Status_NotErrFunc	SAFEBOOL	Variable	Status	FALSE	Information about the error state of the safe axis. FALSE: State FUNCTIONAL FAIL SAFE
S_Status_Operational	SAFEBOOL	Variable	Status	FALSE	Status information for the safe axis. TRUE: The state machine is in state OPERATIONAL.
S_Status_STO	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Torque Off" (STO)
S_Status_SBC	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Brake Control" (SBC)
S_Status_SS1	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Stop 1" (SS1)
S_Status_EncStatus	SAFEBOOL	Variable	Status	FALSE	Indicates the validity of the encoder values
S_Status_Vendor01	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor01"
S_Status_Vendor02	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor02"
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 508: "SF_oS_MOTION_Basic": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

You have the option of linking a safe signal with a non-safe input parameter. To do so, use a function block for type conversion.

Danger!

You are responsible for any conversion of a non-safe input parameter to a safe signal.

6.4.4.2 Function

Function block "SF_oS_MOTION_Basic" provides functions for a safe axis.

These functions include the following:

- Reset
- Activate
- STO - Safe Torque Off
- SBC - Safe Brake Control
- SS1 - Safe Stop 1
- Vendor0x - Vendor-specific safety functions

Information:

The use of this function block is mandatory for each axis.

6.4.4.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.4.4.3.1 Exceeding monitored limits

The motion profile itself does not provide monitoring of the configured limits. This is solely the task of the axis. For information about how the axis monitors the configured limits, see the corresponding documentation for the drive.

The following points must be considered in order to prevent a monitored limit from being violated:

- The movement of the axis must be adapted to the requested safety function and initiated on time.
- The monitored limits must match the calculated limits and movement limitations. Make sure that the different configurations of the unit system match in the safety application and in the standard application.

6.4.4.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.4.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.4.4.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.4.4 Input parameters

Information:

For detailed information about individual safety functions, see the documentation for the safe drive.

6.4.4.4.1 General information about "S_Control" inputs

"S_Control" inputs are used to request the respective safety functions.

Information:

If a safety function is not used in the application, then the respective input must remain open.

Danger!

The safety functions that are used must be tested.
A function is considered to be used if the respective input variable is connected!

Information:

To enable the function block itself and assign the functions to a defined axis, inputs "Activate" and "S_AxisID" must be connected at a minimum.

Information:

It is mandatory for function block "SF_oS_MOTION_Basic" to be applied to each axis being used in the safety application.

In addition to inputs "Activate" and "S_AxisID", inputs "S_Control_Reset" and "S_Control_Activate" must also be used. Otherwise, the SafeDESIGNER project cannot be compiled.

6.4.4.4.2 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.4.4.3 S_Control_Reset

General function

- Resets error messages and the safe axis after the cause of the error has been removed

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This input parameter is used to confirm an error or the booting of the safe axis.

The edges of the input parameter are monitored internally by the function block. The reset function only takes place on a rising edge. An additional static TRUE signal following a rising edge does not trigger the function again. The function block detects a static TRUE signal on "S_Control_Reset" as an error in states where a rising edge on "S_Control_Reset" is required. Set "S_Control_Reset" to FALSE to exit this error state.

Relevant configuration parameters

> ## Information:
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.4.4.4 S_Control_Activate

General function

- Enables the safe state of the axis and safety functions

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the safe state of the axis and safety functions.

TRUE

The safe state of the axis is enabled.
The requested safety functions are executed by the function block.

FALSE

Sets the state machine of the safe axis to state IDLE.
The requested safety functions are not executed by the function block.

6.4.4.4.5 S_Control_STO

General function

- Selects/Deselects safety function "Safe Torque Off" (STO)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "STO".

TRUE

The safety function is deselected. Safe pulse disabling is not active.

FALSE

The safety function is selected. Safe pulse disabling is active. Torque/Power are switched off on the axis.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.4.4.6 S_Control_SBC

General function

- Selects/Deselects safety function "Safe Brake Control" (SBC)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SBC".

TRUE

The safety function is deselected. The motor holding brake output is enabled and can be used by the standard application.

FALSE

The safety function is selected. The motor holding brake output is switched to 0 V.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.4.4.7 S_Control_SS1

General function

- Selects/Deselects safety function "Safe Stop 1" (SS1)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SS1".

TRUE

The safety function is deselected. "SS1" is not executed.

FALSE

The safety function is selected. Safe pulse disabling is activated after the end of ramp monitoring.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.4.4.8 S_Control_Vendor01

General function

- Selects/Deselects vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor01" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is deselected.

FALSE

Safety function "Vendor01" is selected.

Not connected

Vendor-specific safety function "Vendor01" is deactivated.

6.4.4.4.9 S_Control_Vendor02

General function

- Selects/Deselects vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor02" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is deselected.

FALSE

Safety function "Vendor02" is selected.

Not connected

Vendor-specific safety function "Vendor02" is deactivated.

6.4.4.4.10 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

Information:

There can be only one combination of "S_AxisID" and function block "SF_oS_MOTION_Basic" in the safety application. Otherwise, it will not be possible to compile the safety application.

6.4.4.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.4.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.4.5.2 S_Status_NotErrFunc

General function

- Information about the error state of the safe axis

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter specifies the error state of the safe axis.

Danger!

The purpose of this signal is only to provide additional information. It only provides information in connection with the requested safety functions.
Output parameter "S_Status_NotErrFunc" does not represent the functional safe state of the safe axis!

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

No error was detected on the safe axis.

FALSE

FUNCTIONAL FAIL SAFE state
An error was detected on the safe axis (e.g. monitored limit exceeded), or the function block was not enabled.
If the error is a functional error, then it can be acknowledged by changing the signal on input "S_Control_Reset" from FALSE to TRUE (rising edge)!

6.4.4.5.3 S_Status_Operational

General function

- Status information for the safe axis

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter provides information about state OPERATIONAL of the safe axis.

TRUE

The state machine is in state OPERATIONAL.

FALSE

The state machine is not in state OPERATIONAL.

6.4.4.5.4 S_Status_STO

General function

- Status information for safety function "Safe Torque Off" (STO)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "STO".

TRUE

Safety function "STO" is active and currently in its safe state.

FALSE

Safety function "STO" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.4.5.5 S_Status_SBC

General function

- Status information for safety function "Safe Brake Control" (SBC)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SBC".

TRUE

Safety function "SBC" is active and currently in its safe state.

FALSE

Safety function "SBC" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.4.5.6 S_Status_SS1

General function

- Status information for safety function "Safe Stop 1" (SS1)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SS1".

TRUE

Safety function "SS1" is active and currently in its safe state.

FALSE

Safety function "SS1" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.4.5.7 S_Status_EncStatus

General function

- Indicates the validity of the encoder values

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the error state of the signal for a defined safe encoder.

> **Danger!**
>
> **The purpose of this signal is only to provide additional information. It only provides information in connection with the requested safety functions.**
> **"S_Status_EncStatus" does not represent the functional safe state of the safe axis!**

TRUE

An error was not detected on the encoder signal.

FALSE

The encoder signal from a defined safe axis is faulty, or the axis itself is in an error state. The safe drive is in an error state or the function block was not enabled.

6.4.4.5.8 S_Status_Vendor01

General function

- Status information for vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor01".

> **Information:**
>
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is active and currently in its safe state.

FALSE

Safety function "Vendor01" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.4.5.9 S_Status_Vendor02

General function

- Status information for vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor02".

> **Information:**
>
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is active and currently in its safe state.

FALSE

Safety function "Vendor02" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.4.5.10 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.4.5.11 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.4.5.12 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
C001	Function set for control byte not found.	Check whether the required safety function is supported by the connected axis.
C002	Function set for status byte not found.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read status byte.	Check whether the required safety function is supported by the connected axis.
C006	Could not write control byte.	Check whether the required safety function is supported by the connected axis.

Table 509: "SF_oS_MOTION_Basic": Diagnostic codes

6.4.4.6 Signal sequence diagram of the function block

A general signal sequence diagram of the function block cannot be specified since it depends on which safety functions are selected or deselected.

See the documentation for the safe drive for more information about these functions.

6.4.5 SF_oS_MOTION_Speed

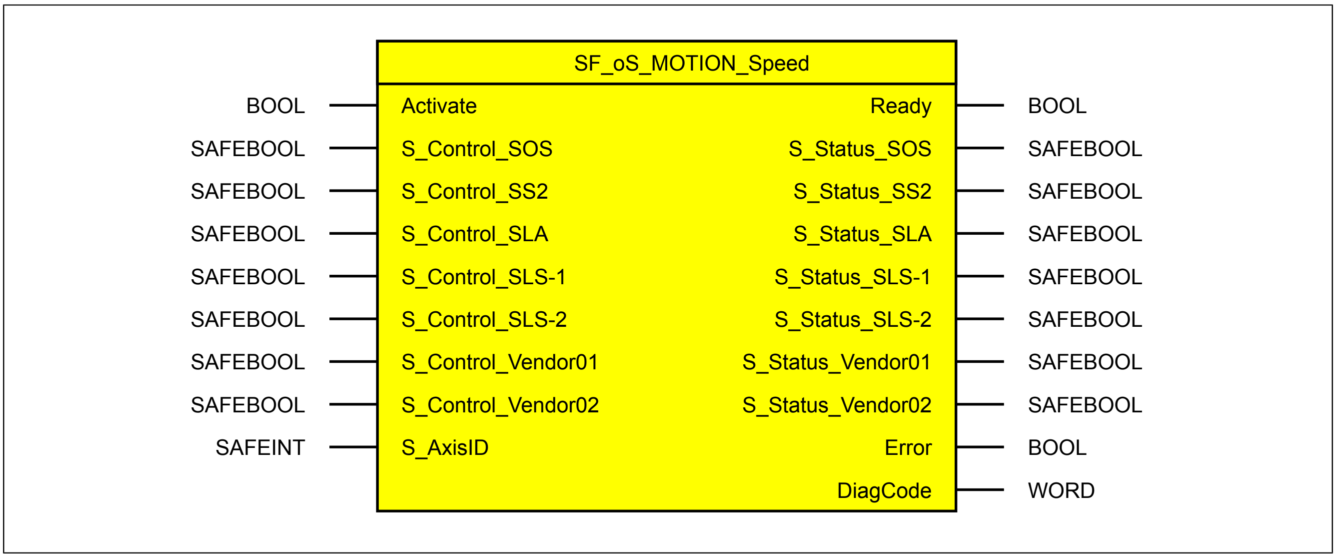


Figure 354: Function block "SF_oS_MOTION_Speed"

6.4.5.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Control_SOS	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Operating Stop" (SOS). FALSE: The safety function is selected.
S_Control_SS2	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Stop 2" (SS2). FALSE: The safety function is selected.
S_Control_SLA	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Acceleration" (SLA). FALSE: The safety function is selected.
S_Control_SLS-1	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Speed", speed limit 1 (SLS-1). FALSE: The safety function is selected.
S_Control_SLS-2	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Speed", speed limit 2 (SLS-2). FALSE: The safety function is selected.
S_Control_Vendor01	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor01"
S_Control_Vendor02	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor02"
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 510: "SF_oS_MOTION_Speed": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Status_SOS	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Operating Stop" (SOS)
S_Status_SS2	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Stop 2" (SS2)
S_Status_SLA	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Acceleration" (SLA)
S_Status_SLS-1	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Speed", speed limit 1 (SLS-1)
S_Status_SLS-2	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Speed", speed limit 2 (SLS-2)
S_Status_Vendor01	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor01"
S_Status_Vendor02	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor02"
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 511: "SF_oS_MOTION_Speed": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

You have the option of linking a safe signal with a non-safe input parameter. To do so, use a function block for type conversion.

Danger!

You are responsible for any conversion of a non-safe input parameter to a safe signal.

6.4.5.2 Function

Function block "SF_oS_MOTION_Speed" provides functions for a safe axis.

These functions include the following:

- SOS - Safe Operating Stop
- SS2 - Safe Stop 2
- SLA - Safely Limited Acceleration
- SLS-1 - Safely Limited Speed, speed limit 1
- SLS-2 - Safely Limited Speed, speed limit 2
- Vendor0x - Vendor-specific safety functions

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.5.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> ## Danger!
> **Always validate the entire safety function!**

6.4.5.3.1 Exceeding monitored limits

The motion profile itself does not provide monitoring of the configured limits. This is solely the task of the axis. For information about how the axis monitors the configured limits, see the corresponding documentation for the drive.

The following points must be considered in order to prevent a monitored limit from being violated:

- The movement of the axis must be adapted to the requested safety function and initiated on time.
- The monitored limits must match the calculated limits and movement limitations. Make sure that the different configurations of the unit system match in the safety application and in the standard application.

6.4.5.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> ## Danger!
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.5.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.4.5.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.5.4 Input parameters

> ## Information:
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.5.4.1 General information about "S_Control" inputs

"S_Control" inputs are used to request the respective safety functions.

> ## Information:
> **If a safety function is not used in the application, then the respective input must remain open.**

> ## Danger!
> **The safety functions that are used must be tested.**
> **A function is considered to be used if the respective input variable is connected!**

> ## Information:
> **To enable the function block itself and assign the functions to a defined axis, inputs "Activate" and "S_AxisID" must be connected at a minimum.**

> ## Information:
> **It is mandatory for function block "SF_oS_MOTION_Basic" to be applied to each axis being used in the safety application.**

6.4.5.4.2 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.5.4.3 S_Control_SOS

General function

- Selects/Deselects safety function "Safe Operating Stop" (SOS)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SOS".

TRUE

The safety function is deselected. Standstill tolerances are not being monitored.

FALSE

The safety function is selected. Standstill tolerances are being monitored.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> ## Information:
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.5.4.4 S_Control_SS2

General function

- Selects/Deselects safety function "Safe Stop 2" (SS2)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SS2".

TRUE

The safety function is deselected. "SS2" is not executed.

FALSE

The safety function is selected. Standstill monitoring is activated after the end of ramp monitoring.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.5.4.5 S_Control_SLA

General function

- Selects/Deselects safety function "Safely Limited Acceleration" (SLA)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLA".

TRUE

The safety function is deselected. "SLA" is not executed.

FALSE

The safety function is selected. A safe limit value for acceleration/deceleration is monitored with respect to the direction of movement.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.5.4.6 S_Control_SLS-1

General function

- Selects/Deselects safety function "Safely Limited Speed", speed limit 1 (SLS-1)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLS-1".

TRUE

The safety function is deselected. "SLS-1" is not executed.

FALSE

The safety function is selected. Speed limit 1 is monitored after the end of ramp monitoring.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.5.4.7 S_Control_SLS-2

General function

- Selects/Deselects safety function "Safely Limited Speed", speed limit 2 (SLS-2)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLS-2".

TRUE

The safety function is deselected. "SLS-2" is not executed.

FALSE

The safety function is selected. Speed limit 2 is monitored after the end of ramp monitoring.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> ## Information:
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.5.4.8 S_Control_Vendor01

General function

- Selects/Deselects vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor01" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is deselected.

FALSE

Safety function "Vendor01" is selected.

Not connected

Vendor-specific safety function "Vendor01" is deactivated.

6.4.5.4.9 S_Control_Vendor02

General function

- Selects/Deselects vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor02" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is deselected.

FALSE

Safety function "Vendor02" is selected.

Not connected

Vendor-specific safety function "Vendor02" is deactivated.

6.4.5.4.10 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

Information:

There can be only one combination of "S_AxisID" and function block "SF_oS_MOTION_Speed" in the safety application. Otherwise, it will not be possible to compile the safety application.

6.4.5.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.5.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.5.5.2 S_Status_SOS

General function

- Status information for safety function "Safe Operating Stop" (SOS)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SOS".

TRUE

Safety function "SOS" is active and currently in its safe state.

FALSE

Safety function "SOS" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.3 S_Status_SS2

General function

- Status information for safety function "Safe Stop 2" (SS2)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SS2".

TRUE

Safety function "SS2" is active and currently in its safe state.

FALSE

Safety function "SS2" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.4 S_Status_SLA

General function

- Status information for safety function "Safely Limited Acceleration" (SLA)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLA".

TRUE

Safety function "SLA" is active and currently in its safe state.

FALSE

Safety function "SLA" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.5 S_Status_SLS-1

General function

- Status information for safety function "Safely Limited Speed", speed limit 1 (SLS-1)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLS-1".

TRUE

Safety function "SLS-1" is active and currently in its safe state.

FALSE

Safety function "SLS-1" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.6 S_Status_SLS-2

General function

- Status information for safety function "Safely Limited Speed", speed limit 2 (SLS-2)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLS-2".

TRUE

Safety function "SLS-2" is active and currently in its safe state.

FALSE

Safety function "SLS-2" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.7 S_Status_Vendor01

General function

- Status information for vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor01".

> ## Information:
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is active and currently in its safe state.

FALSE

Safety function "Vendor01" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.8 S_Status_Vendor02

General function

- Status information for vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor02".

> **Information:**
>
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is active and currently in its safe state.

FALSE

Safety function "Vendor02" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.5.5.9 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.5.5.10 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.5.5.11 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
C001	Function set for control byte not found.	Check whether the required safety function is supported by the connected axis.
C002	Function set for status byte not found.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read status byte.	Check whether the required safety function is supported by the connected axis.
C006	Could not write control byte.	Check whether the required safety function is supported by the connected axis.

Table 512: "SF_oS_MOTION_Speed": Diagnostic codes

6.4.5.6 Signal sequence diagram of the function block

A general signal sequence diagram of the function block cannot be specified since it depends on which safety functions are selected or deselected.

See the documentation for the safe drive for more information about these functions.

6.4.6 SF_oS_MOTION_Advanced

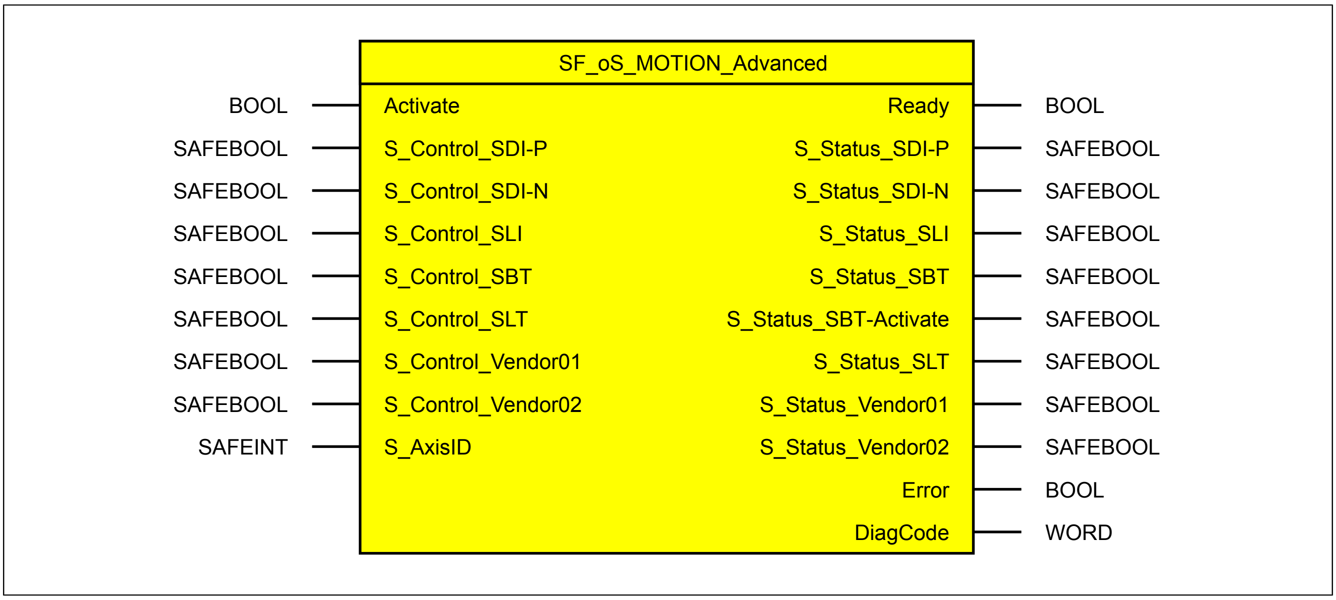


Figure 355: Function block "SF_oS_MOTION_Advanced"

6.4.6.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Control_SDI-P	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Direction" (SDI). Movement in the positive direction is allowed. FALSE: The safety function is selected.
S_Control_SDI-N	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safe Direction" (SDI). Movement in the negative direction is allowed. FALSE: The safety function is selected.
S_Control_SLI	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Increment" (SLI). FALSE: The safety function is selected.
S_Control_SBT	SAFEBOOL	Variable	Edge	FALSE	Selects/Deselects safety function "Safe Brake Test" (SBT). FALSE: The safety function is selected.
S_Control_SLT	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Torque" (SLT). FALSE: The safety function is selected.
S_Control_Vendor01	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor01"
S_Control_Vendor02	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor02"
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 513: "SF_oS_MOTION_Advanced": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Status_SDI-P	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Direction" (SDI). Movement in the positive direction is allowed.
S_Status_SDI-N	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Direction" (SDI). Movement in the negative direction is allowed.
S_Status_SLI	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Increment" (SLI)
S_Status_SBT	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Brake Test" (SBT)
S_Status_SBT-Activate	SAFEBOOL	Variable	Status	FALSE	Indicates an active safe brake test
S_Status_SLT	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Torque" (SLT)
S_Status_Vendor01	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor01"
S_Status_Vendor02	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor02"
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 514: "SF_oS_MOTION_Advanced": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

You have the option of linking a safe signal with a non-safe input parameter. To do so, use a function block for type conversion.

Danger!

You are responsible for any conversion of a non-safe input parameter to a safe signal.

6.4.6.2 Function

Function block "SF_oS_MOTION_Advanced" provides functions for a safe axis.

These functions include the following:

- SDI - Safe Direction for positive and negative direction
- SLI - Safely Limited Increment
- SBT - Safe Brake Test
- SLT - Safely Limited Torque
- Vendor0x - Vendor-specific safety functions

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.6.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

Danger!

Always validate the entire safety function!

6.4.6.3.1 Exceeding monitored limits

The motion profile itself does not provide monitoring of the configured limits. This is solely the task of the axis. For information about how the axis monitors the configured limits, see the corresponding documentation for the drive.

The following points must be considered in order to prevent a monitored limit from being violated:

- The movement of the axis must be adapted to the requested safety function and initiated on time.
- The monitored limits must match the calculated limits and movement limitations. Make sure that the different configurations of the unit system match in the safety application and in the standard application.

6.4.6.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

Danger!

You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!
Check the connections when validating the sub-application!

6.4.6.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.4.6.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.6.4 Input parameters

> ### Information:
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.6.4.1 General information about "S_Control" inputs

"S_Control" inputs are used to request the respective safety functions.

> ### Information:
> **If a safety function is not used in the application, then the respective input must remain open.**

> ### Danger!
> **The safety functions that are used must be tested.**
> **A function is considered to be used if the respective input variable is connected!**

> ### Information:
> **To enable the function block itself and assign the functions to a defined axis, inputs "Activate" and "S_AxisID" must be connected at a minimum.**

> ### Information:
> **It is mandatory for function block "SF_oS_MOTION_Basic" to be applied to each axis being used in the safety application.**

6.4.6.4.2 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.6.4.3 S_Control_SDI-P

General function

- Selects/Deselects safety function "Safe Direction" (SDI). Movement is allowed in the positive direction.

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SDI". Movement is allowed in the positive direction.

TRUE

The safety function is deselected. "SDI" is not executed.

FALSE

The direction of movement is monitored after the delay time has expired. Movement is allowed in the positive direction.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.6.4.4 S_Control_SDI-N

General function

- Selects/Deselects safety function "Safe Direction" (SDI). Movement is allowed in the negative direction.

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SDI". Movement is allowed in the negative direction.

TRUE

The safety function is deselected. "SDI" is not executed.

FALSE

The direction of movement is monitored after the delay time has expired. Movement is allowed in the negative direction.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.6.4.5 S_Control_SLI

General function

- Selects/Deselects safety function "Safely Limited Increment" (SLI)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLI".

TRUE

The safety function is deselected. "SLI" is not executed.

FALSE

The safety function is selected. A safe range of increments is monitored.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.6.4.6 S_Control_SBT

General function

- Selects/Deselects safety function "Safe Brake Test" (SBT)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This input parameter is used to select or deselect safety function "SBT".

Falling edge

A falling edge (state transition from TRUE to FALSE) on input parameter "S_Control_SBT" starts safety function "SBT".

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.6.4.7 S_Control_SLT

General function

- Selects/Deselects safety function "Safely Limited Torque" (SLT)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLT".

TRUE

The safety function is deselected. "SLT" is not executed.

FALSE

The safety function is selected.

Not connected

The safety function is deactivated.

Relevant configuration parameters

> **Information:**
>
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.6.4.8 S_Control_Vendor01

General function

- Selects/Deselects vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor01" is a vendor-specific safety function.

> ## Information:
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is deselected.

FALSE

Safety function "Vendor01" is selected.

Not connected

Vendor-specific safety function "Vendor01" is deactivated.

6.4.6.4.9 S_Control_Vendor02

General function

- Selects/Deselects vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor02" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is deselected.

FALSE

Safety function "Vendor02" is selected.

Not connected

Vendor-specific safety function "Vendor02" is deactivated.

6.4.6.4.10 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

Information:

There can be only one combination of "S_AxisID" and function block "SF_oS_MOTION_Advanced" in the safety application. Otherwise, it will not be possible to compile the safety application.

6.4.6.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.6.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.6.5.2 S_Status_SDI-P

General function

- Status information for safety function "Safe Direction" (SDI). Movement is allowed in the positive direction.

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SDI-P".

TRUE

Safety function "SDI-P" is active and currently in its safe state.

FALSE

Safety function "SDI-P" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.3 S_Status_SDI-N

General function

- Status information for safety function "Safe Direction" (SDI). Movement is allowed in the negative direction.

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SDI-N".

TRUE

Safety function "SDI-N" is active and currently in its safe state.

FALSE

Safety function "SDI-N" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.4 S_Status_SLI

General function

- Status information for safety function "Safely Limited Increment" (SLI)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLI".

TRUE

Safety function "SLI" is active and currently in its safe state.

FALSE

Safety function "SLI" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.5 S_Status_SBT

General function

- Status information for safety function "Safe Brake Test" (SBT)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter returns the result of the test requested by "S_Control_SBT".

TRUE

The safe brake test was successfully executed and completed.

FALSE

The safe brake test was not successfully executed or not yet completed.

6.4.6.5.6 S_Status_SBT-Activate

General function

- Indicates an active safe brake test

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter specifies an active safe brake test.

TRUE

A safe brake test is active.

FALSE

A safe brake test is not active.

6.4.6.5.7 S_Status_SLT

General function

- Status information for safety function "Safely Limited Torque" (SLT)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLT".

TRUE

Safety function "SLT" is active and currently in its safe state.

FALSE

Safety function "SLT" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.8 S_Status_Vendor01

General function

- Status information for vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor01".

> ## Information:
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is active and currently in its safe state.

FALSE

Safety function "Vendor01" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.9 S_Status_Vendor02

General function

- Status information for vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor02".

> ## Information:
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is active and currently in its safe state.

FALSE

Safety function "Vendor02" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.6.5.10 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.6.5.11 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.6.5.12 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
C001	Function set for control byte not found.	Check whether the required safety function is supported by the connected axis.
C002	Function set for status byte not found.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read status byte.	Check whether the required safety function is supported by the connected axis.
C006	Could not write control byte.	Check whether the required safety function is supported by the connected axis.

Table 515: "SF_oS_MOTION_Advanced": Diagnostic codes

6.4.6.6 Signal sequence diagram of the function block

A general signal sequence diagram of the function block cannot be specified since it depends on which safety functions are selected or deselected.

See the documentation for the safe drive for more information about these functions.

6.4.7 SF_oS_MOTION_EncoderBasic

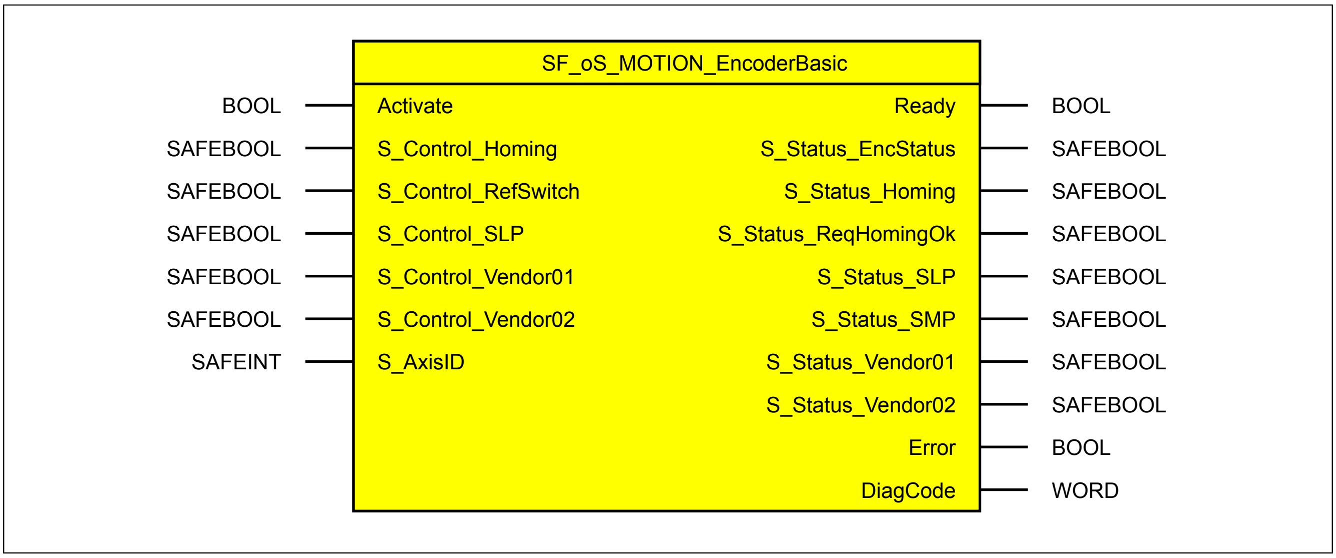


Figure 356: Function block "SF_oS_MOTION_EncoderBasic"

6.4.7.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Control_Homing	SAFEBOOL	Variable	Edge	FALSE	Request for safe homing. The request is made on a rising edge!
S_Control_RefSwitch	SAFEBOOL	Variable/ Constant	Status	FALSE	Safe input for a reference switch
S_Control_SLP	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects safety function "Safely Limited Position" (SLP). FALSE: The safety function is selected.
S_Control_Vendor01	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor01"
S_Control_Vendor02	SAFEBOOL	Variable/ Constant	Status	FALSE	Selects/Deselects vendor-specific safety function "Vendor02"
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 516: "SF_oS_MOTION_EncoderBasic": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Status_EncStatus	SAFEBOOL	Variable	Status	FALSE	Indicates the validity of the encoder values
S_Status_Homing	SAFEBOOL	Variable	Status	FALSE	Indicates the validity of the safe position
S_Status_ReqHomin-gOk	SAFEBOOL	Variable	Status	FALSE	Feedback for homing in SafeDESIGNER
S_Status_SLP	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safely Limited Position" (SLP)
S_Status_SMP	SAFEBOOL	Variable	Status	FALSE	Status information for safety function "Safe Maximum Position" (SMP)
S_Status_Vendor01	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor01"
S_Status_Vendor02	SAFEBOOL	Variable	Status	FALSE	Status information for vendor-specific safety function "Vendor02"
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 517: "SF_oS_MOTION_EncoderBasic": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

You have the option of linking a safe signal with a non-safe input parameter. To do so, use a function block for type conversion.

Danger!

You are responsible for any conversion of a non-safe input parameter to a safe signal.

6.4.7.2 Function

Function block "SF_oS_MOTION_EncoderBasic" provides functions for a safe axis.

These functions include the following:

- Safe homing
- SLP - Safely Limited Position
- Vendor0x - Vendor-specific safety functions

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.7.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.4.7.3.1 Exceeding monitored limits

The motion profile itself does not provide monitoring of the configured limits. This is solely the task of the axis. For information about how the axis monitors the configured limits, see the corresponding documentation for the drive.

The following points must be considered in order to prevent a monitored limit from being violated:

- The movement of the axis must be adapted to the requested safety function and initiated on time.
- The monitored limits must match the calculated limits and movement limitations. Make sure that the different configurations of the unit system match in the safety application and in the standard application.

6.4.7.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.7.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.4.7.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.7.4 Input parameters

> ## Information:
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.7.4.1 General information about "S_Control" inputs

"S_Control" inputs are used to request the respective safety functions.

> ## Information:
> **If a safety function is not used in the application, then the respective input must remain open.**

> ## Danger!
> **The safety functions that are used must be tested.**
> **A function is considered to be used if the respective input variable is connected!**

> ## Information:
> **To enable the function block itself and assign the functions to a defined axis, inputs "Activate" and "S_AxisID" must be connected at a minimum.**

> ## Information:
> **It is mandatory for function block "SF_oS_MOTION_Basic" to be applied to each axis being used in the safety application.**

6.4.7.4.2 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.7.4.3 S_Control_Homing

General function

- Request for safe homing

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This input parameter is used to start a safe homing procedure. A rising edge of the input starts the safety function.

Rising edge: Switch from FALSE to TRUE

Starts safe homing.

Falling edge: Switch from TRUE to FALSE

If still active, the homing procedure will be aborted by the falling edge. This state transition has no effect if the homing procedure has already been completed.

Not connected

Safe homing is disabled.

Relevant configuration parameters

> ## Information:
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.7.4.4 S_Control_RefSwitch

General function

- Safe input for a reference switch

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used as a reference switch input for safe homing.
The status of a safe reference switch that was read into the safety application via a safe input module (X20SIxxxx), for example, should be linked to the input.

Not connected

The reference switch is not being used.

Information:

For the homing variants in which evaluation takes place, see the documentation for the safe drive.

6.4.7.4.5 S_Control_SLP

General function

- Selects/Deselects safety function "Safely Limited Position" (SLP)

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

This input parameter is used to select or deselect safety function "SLP".

TRUE

The safety function is deselected. "SLP" is not executed.

FALSE

The configured position window will be safety-monitored after the configured delay time (see user's manual for the safe drive).

Not connected

The safety function is deactivated.

Relevant configuration parameters

> ## Information:
> **For corresponding configuration parameters, see the documentation for the respective drive.**

6.4.7.4.6 S_Control_Vendor01

General function

- Selects/Deselects vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor01" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is deselected.

FALSE

Safety function "Vendor01" is selected.

Not connected

Vendor-specific safety function "Vendor01" is deactivated.

6.4.7.4.7 S_Control_Vendor02

General function

- Selects/Deselects vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable or constant

Description of function

"Vendor02" is a vendor-specific safety function.

> **Information:**
>
> **For information about this input parameter, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is deselected.

FALSE

Safety function "Vendor02" is selected.

Not connected

Vendor-specific safety function "Vendor02" is deactivated.

6.4.7.4.8 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

> ## Information:
>
> **There can be only one combination of "S_AxisID" and function block "SF_oS_MOTION_EncoderBasic" in the safety application. Otherwise, it will not be possible to compile the safety application.**

6.4.7.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.7.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.7.5.2 S_Status_EncStatus

General function

- Indicates the validity of the encoder values

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the error state of the signal for a defined safe encoder.

> **Danger!**
>
> **The purpose of this signal is only to provide additional information. It only provides information in connection with the requested safety functions.**
> **"S_Status_EncStatus" does not represent the functional safe state of the safe axis!**

TRUE

An error was not detected on the encoder signal.

FALSE

The encoder signal from a defined safe axis is faulty, the axis itself is in an error state or the function block was not enabled.

6.4.7.5.3 S_Status_Homing

General function

- Indicates the validity of the safe position

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter specifies whether or not safe homing of the axis has been completed and whether or not the position signal is valid.

> **Danger!**
>
> **The purpose of this signal is only to provide additional information.**
> **"S_Status_Homing" does not represent the functional safe state of the safe axis!**

TRUE

The axis has been successfully homed, and the safe position is valid.

FALSE

The axis has not yet been successfully homed; the encoder signal of the axis is faulty. The safe axis is in an error state or the function block was not enabled.
The safe position is invalid.

6.4.7.5.4 S_Status_ReqHomingOk

General function

- Feedback for homing in SafeDESIGNER

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter is set to provide feedback in the event that homing is requested when already in a homed state.

TRUE

The input for homing is set and the safe position is valid.

FALSE

The input for homing is not set or the safe position is not valid. The safe axis is in an error state or the function block was not enabled.

6.4.7.5.5 S_Status_SLP

General function

- Status information for safety function "Safely Limited Position" (SLP)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SLP".

TRUE

Safety function "SLP" is active and currently in its safe state.

FALSE

Safety function "SLP" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.7.5.6 S_Status_SMP

General function

- Status information for safety function "Safe Maximum Position" (SMP)

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of safety function "SMP".

TRUE

Safety function "SMP" is active and currently in its safe state.

FALSE

Monitoring of the "SMP" position limits is not active.
Monitoring is not active because the safe axis has not yet been homed, the function or safe axis is in an error state or the function block was not enabled.

6.4.7.5.7 S_Status_Vendor01

General function

- Status information for vendor-specific safety function "Vendor01"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor01".

> **Information:**
>
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor01" is active and currently in its safe state.

FALSE

Safety function "Vendor01" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.7.5.8 S_Status_Vendor02

General function

- Status information for vendor-specific safety function "Vendor02"

Data type

- SAFEBOOL

Connection

- Variable

Description of function

This output parameter indicates the functional safe state of "Vendor02".

> ## Information:
> **For information about this status, see the documentation for the respective safe drive.**

TRUE

Safety function "Vendor02" is active and currently in its safe state.

FALSE

Safety function "Vendor02" is not requested or has not yet reached its safe state. The function or safe axis is in an error state or the function block was not enabled.

6.4.7.5.9 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.7.5.10 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.7.5.11 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
C001	Function set for control byte not found.	Check whether the required safety function is supported by the connected axis.
C002	Function set for status byte not found.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read status byte.	Check whether the required safety function is supported by the connected axis.
C006	Could not write control byte.	Check whether the required safety function is supported by the connected axis.

Table 518: "SF_oS_MOTION_EncoderBasic": Diagnostic codes

6.4.7.6 Signal sequence diagram of the function block

A general signal sequence diagram of the function block cannot be specified since it depends on which safety functions are selected or deselected.

See the documentation for the safe drive for more information about these functions.

6.4.8 SF_oS_MOTION_Data_Acceleration

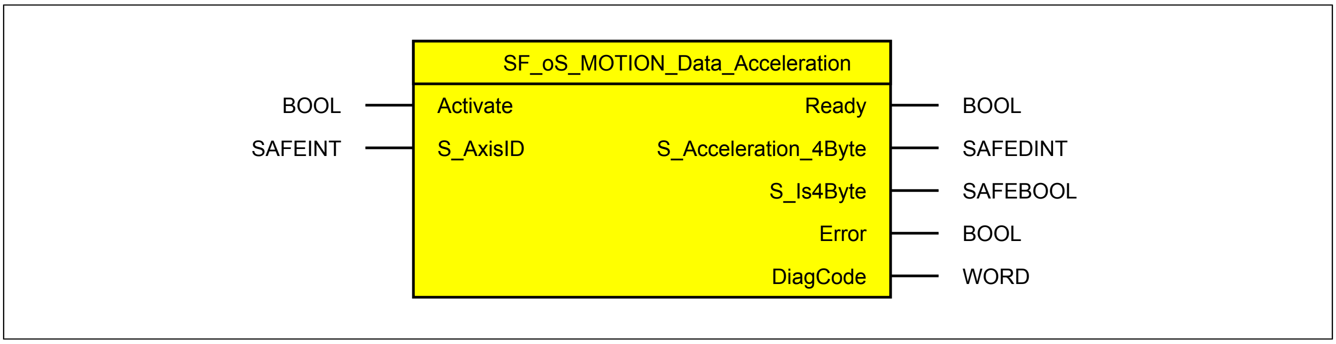


Figure 357: Function block "SF_oS_MOTION_Data_Acceleration"

6.4.8.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 519: "SF_oS_MOTION_Data_Acceleration": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Acceleration_4Byte	SAFEDINT	Variable	Value	DINT#0	Provides the acceleration value of the safe axis
S_Is4Byte	SAFEBOOL	Variable	Status	FALSE	Provides information about whether the value returned via the axis is a 2-byte or 4-byte value
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 520: "SF_oS_MOTION_Data_Acceleration": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.4.8.2 Function

Function block "SF_oS_MOTION_Data_Acceleration" provides the acceleration value of the axis. The value is always provided with a length of 4 bytes.
Output parameter "S_Is4Byte" indicates whether the returned value of the axis was implemented as a 2-byte or 4-byte value.
The safe axis is assigned using "S_AxisID".

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.8.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.4.8.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.8.3.2 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.8.4 Input parameters

> **Information:**
>
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.8.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.8.4.2 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

6.4.8.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.8.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.8.5.2 S_Acceleration_4Byte

General function

- Provides the acceleration value of the safe axis

Data type

- SAFEDINT

Connection

- Variable

Description of function

This output parameter provides the current acceleration value of the safe axis. The data length is 4 bytes. Status "S_Is4Byte" is used to evaluate whether the safe axis provides a 2-byte or 4-byte value.

6.4.8.5.3 S_Is4Byte

General function

- Provides information about whether the value returned via the axis is a 2-byte or 4-byte value

Data type

- SAFEBOOL

Connection

- Variable

Description of function

Output parameter "S_Acceleration_4Byte" is connected to a variable with a length of 4 bytes. The value is specified as either a 2-byte or 4-byte value depending on the safe axis being used. Evaluate this output parameter to determine the difference.

TRUE

The safe axis returns a 4-byte value.

FALSE

The safe axis returns a 2-byte value.

6.4.8.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.8.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.8.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	No function set for this InstanceID. 2 or 4 bytes not read.	Check whether the required safety function is supported by the connected axis.
C001	Could not read back acceleration value properly from axis.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read 2 or 4 bytes.	Check whether the required safety function is supported by the connected axis.

Table 521: "SF_oS_MOTION_Data_Acceleration": Diagnostic codes

6.4.8.6 Signal sequence diagram of the function block

A signal sequence diagram cannot be specified for this function block.

6.4.9 SF_oS_MOTION_Data_Position

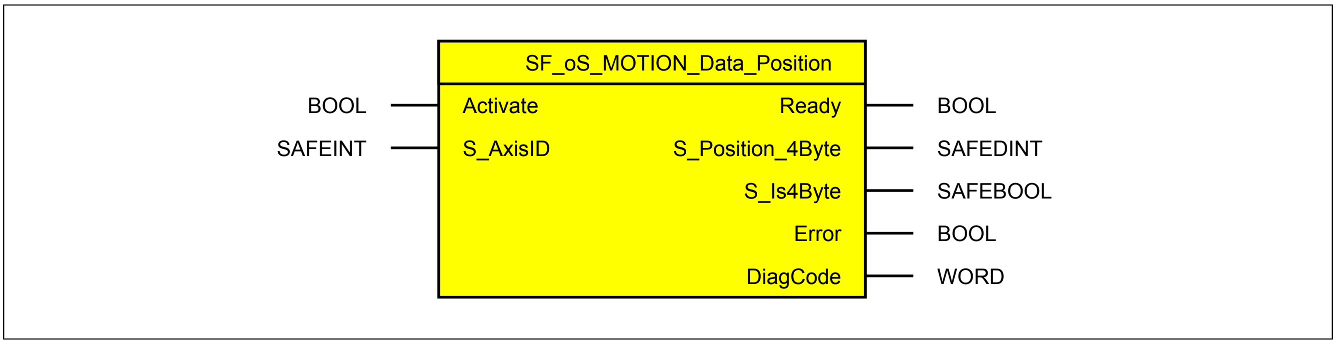


Figure 358: Function block "SF_oS_MOTION_Data_Position"

6.4.9.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 522: "SF_oS_MOTION_Data_Position": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Position_4Byte	SAFEDINT	Variable	Value	DINT#0	Provides the position value of the safe axis
S_Is4Byte	SAFEBOOL	Variable	Status	FALSE	Provides information about whether the value returned via the axis is a 2-byte or 4-byte value
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 523: "SF_oS_MOTION_Data_Position": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.4.9.2 Function

Function block "SF_oS_MOTION_Data_Position" provides the position value of the axis. The value is always provided with a length of 4 bytes.
Output parameter "S_Is4Byte" indicates whether the returned value of the axis was implemented as a 2-byte or 4-byte value.
The safe axis is assigned using "S_AxisID".

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.9.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

Danger!

Always validate the entire safety function!

6.4.9.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

Danger!

You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!
Check the connections when validating the sub-application!

6.4.9.3.2 Validating the position signal

For information about validating the position signal, see the documentation for the safe drive.

Danger!

If the position signal is not validated, then an invalid position could be used in the safety application. This can result in hazardous situations!

6.4.9.3.3 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

Danger!

You are responsible for the functional testing of protective equipment.
You must therefore validate the protective equipment!

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.9.4 Input parameters

> ## Information:
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.9.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.9.4.2 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

6.4.9.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.9.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.9.5.2 S_Position_4Byte

General function

- Provides the position value of the safe axis

Data type

- SAFEDINT

Connection

- Variable

Description of function

This output parameter provides the current position value of the safe axis. The data length is 4 bytes. Status "S_Is4Byte" is used to evaluate whether the safe axis provides a 2-byte or 4-byte value.

6.4.9.5.3 S_Is4Byte

General function

- Provides information about whether the value returned via the axis is a 2-byte or 4-byte value

Data type

- SAFEBOOL

Connection

- Variable

Description of function

Output parameter "S_Position_4Byte" is connected to a variable with a length of 4 bytes. The value is specified as either a 2-byte or 4-byte value depending on the safe axis being used. Evaluate this output parameter to determine the difference.

TRUE

The safe axis returns a 4-byte value.

FALSE

The safe axis returns a 2-byte value.

6.4.9.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.9.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.9.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	No function set for this InstanceID. 2 or 4 bytes not read.	Check whether the required safety function is supported by the connected axis.
C001	Could not read back position value properly from axis.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read 2 or 4 bytes.	Check whether the required safety function is supported by the connected axis.

Table 524: "SF_oS_MOTION_Data_Position": Diagnostic codes

6.4.9.6 Signal sequence diagram of the function block

A signal sequence diagram cannot be specified for this function block.

6.4.10 SF_oS_MOTION_Data_Speed

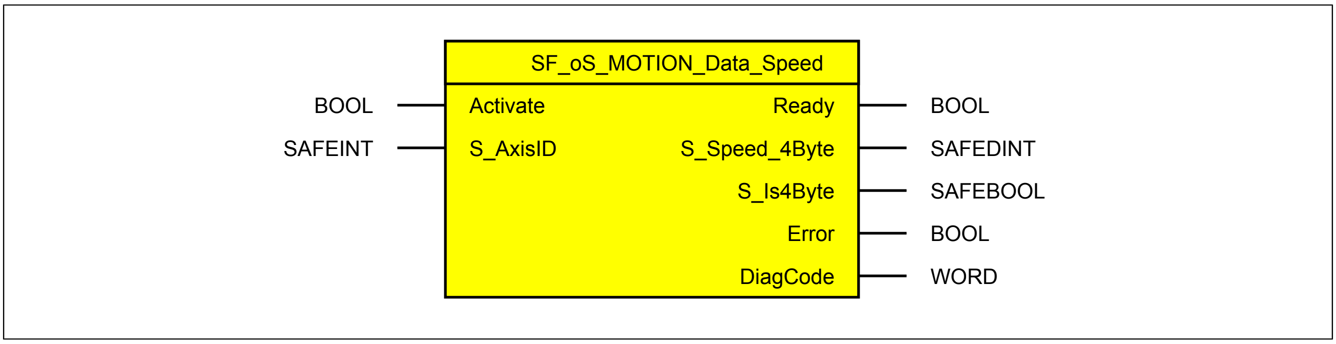


Figure 359: Function block "SF_oS_MOTION_Data_Speed"

6.4.10.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 525: "SF_oS_MOTION_Data_Speed": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Speed_4Byte	SAFEDINT	Variable	Value	DINT#0	Provides the speed value of the safe axis
S_Is4Byte	SAFEBOOL	Variable	Status	FALSE	Provides information about whether the value returned via the axis is a 2-byte or 4-byte value
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 526: "SF_oS_MOTION_Data_Speed": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.4.10.2 Function

Function block "SF_oS_MOTION_Data_Speed" provides the speed value of the axis. The value is always provided with a length of 4 bytes.
Output parameter "S_Is4Byte" indicates whether the returned value of the axis was implemented as a 2-byte or 4-byte value.
The safe axis is assigned using "S_AxisID".

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.10.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.4.10.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.10.3.2 Validating the speed signal

For information about validating the speed signal, see the documentation for the safe drive.

> **Danger!**
>
> **If the speed signal is not validated, then an invalid speed value could be used in the safety application. This can result in hazardous situations!**

6.4.10.3.3 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.10.4 Input parameters

Information:

For detailed information about individual safety functions, see the documentation for the safe drive.

6.4.10.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.10.4.2 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

6.4.10.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.10.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.10.5.2 S_Speed_4Byte

General function

- Provides the speed value of the safe axis

Data type

- SAFEDINT

Connection

- Variable

Description of function

This output parameter provides the current speed value of the safe axis. The data length is 4 bytes. Status "S_Is4Byte" is used to evaluate whether the safe axis provides a 2-byte or 4-byte value.

6.4.10.5.3 S_Is4Byte

General function

- Provides information about whether the value returned via the axis is a 2-byte or 4-byte value

Data type

- SAFEBOOL

Connection

- Variable

Description of function

Output parameter "S_Speed_4Byte" is connected to a variable with a length of 4 bytes. The value is specified as either a 2-byte or 4-byte value depending on the safe axis being used. Evaluate this output parameter to determine the difference.

TRUE

The safe axis returns a 4-byte value.

FALSE

The safe axis returns a 2-byte value.

6.4.10.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.10.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.10.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	No function set for this InstanceID. 2 or 4 bytes not read.	Check whether the required safety function is supported by the connected axis.
C001	Could not read back speed value properly from axis.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read 2 or 4 bytes.	Check whether the required safety function is supported by the connected axis.

Table 527: "SF_oS_MOTION_Data_Speed": Diagnostic codes

6.4.10.6 Signal sequence diagram of the function block

A signal sequence diagram cannot be specified for this function block.

6.4.11 SF_oS_MOTION_Data_Torque

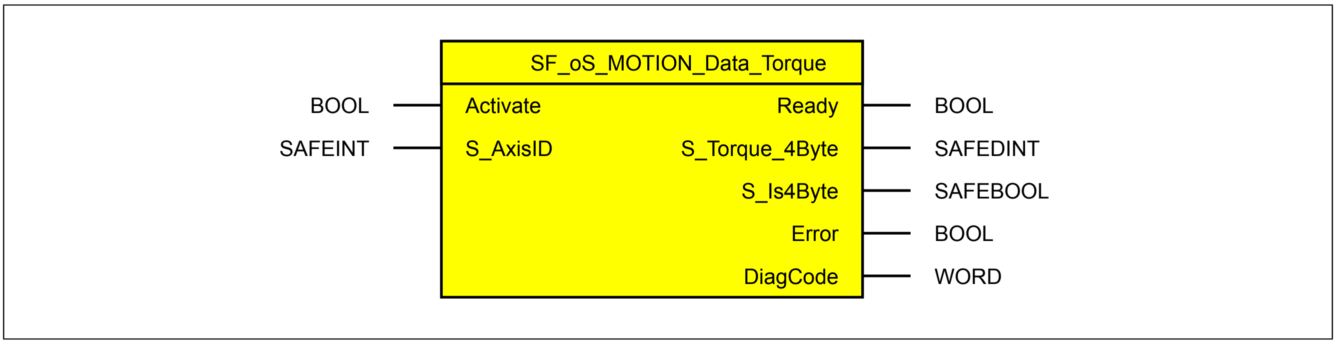


Figure 360: Function block "SF_oS_MOTION_Data_Torque"

6.4.11.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_AxisID	SAFEINT	Constant	Status	-1	Assigns an axis to the function block

Table 528: "SF_oS_MOTION_Data_Torque": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Torque_4Byte	SAFEDINT	Variable	Value	DINT#0	Provides the torque value of the safe axis
S_Is4Byte	SAFEBOOL	Variable	Status	FALSE	Provides information about whether the value returned via the axis is a 2-byte or 4-byte value
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 529: "SF_oS_MOTION_Data_Torque": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.4.11.2 Function

Function block "SF_oS_MOTION_Data_Torque" provides the torque value of the safe axis. The value is always provided with a length of 4 bytes.
Output parameter "S_Is4Byte" indicates whether the returned value of the axis was implemented as a 2-byte or 4-byte value.
The safe axis is assigned using "S_AxisID".

Information:

When using the function blocks in this library, keep in mind that function block "SF_oS_MOTION_Basic" must be instanced in order to use the functions included in all other function blocks in this library.

6.4.11.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.4.11.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.4.11.3.2 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.4.11.4 Input parameters

> ## Information:
> **For detailed information about individual safety functions, see the documentation for the safe drive.**

6.4.11.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

Data type

- BOOL

Connection

- Variable or constant

Description of function

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be connected to a variable that indicates the state (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the state of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.4.11.4.2 S_AxisID

General function

- Assigns an axis to the function block

Data type

- SAFEINT

Connection

- Constant

Description of function

This input parameter assigns a real axis to the function block.
The corresponding axis is connected to the input parameter using the drag-and-drop function in SafeDESIGNER.

6.4.11.5 Output parameters

Output parameters provide information about the state of the safe axis and individual safety functions.

6.4.11.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current state of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block output parameters set to FALSE.

6.4.11.5.2 S_Torque_4Byte

General function

- Provides the torque value of the safe axis

Data type

- SAFEDINT

Connection

- Variable

Description of function

This output parameter provides the current torque value of the safe axis. The data length is 4 bytes. Status "S_Is4Byte" is used to evaluate whether the safe axis provides a 2-byte or 4-byte value.

6.4.11.5.3 S_Is4Byte

General function

- Provides information about whether the value returned via the axis is a 2-byte or 4-byte value

Data type

- SAFEBOOL

Connection

- Variable

Description of function

Output parameter "S_Torque_4Byte" is connected to a variable with a length of 4 bytes. The value is specified as either a 2-byte or 4-byte value depending on the safe axis being used. Evaluate this output parameter to determine the difference.

TRUE

The safe axis returns a 4-byte value.

FALSE

The safe axis returns a 2-byte value.

6.4.11.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Description of function

This output parameter indicates a pending function block error message.

In order to exit an error state ("Error" = TRUE), the signal on input "S_Control_Reset" must change from FALSE to TRUE (rising edge).

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.
Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error.
Output parameter "DiagCode" indicates the state.

6.4.11.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Description of function

Specific function block diagnostic and status messages are output via this output parameter and can be provided automatically to the diagnostic tools being used.

These diagnostic tools cannot acknowledge diagnostic messages from the function block. This is done exclusively in the safety application.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.4.11.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	No function set for this InstanceID. 2 or 4 bytes not read.	Check whether the required safety function is supported by the connected axis.
C001	Could not read back torque value properly from axis.	Check whether the required safety function is supported by the connected axis.
C003	Read function set ID does not match.	Check whether the required safety function is supported by the connected axis.
C004	Data length of read function set is invalid.	Check whether the required safety function is supported by the connected axis.
C005	Could not read 2 or 4 bytes.	Check whether the required safety function is supported by the connected axis.

Table 530: "SF_oS_MOTION_Data_Torque": Diagnostic codes

6.4.11.6 Signal sequence diagram of the function block

A signal sequence diagram cannot be specified for this function block.

6.4.12 Version history

Version	Date	Comment
1.20	January 2018	• Chapter 6.4.2 "System requirements": Updated. • Function block "SF_oS_MOTION_Basic": Section 6.4.4.4.1 "General information about "S_Control" inputs": Updated information.
1.10	October 2016	First edition

Table 531: Version history

6.5 PLCopen_Press_SF

6.5.1 Overview of PLCopen function blocks for presses

The function blocks for safety-oriented applications standardized in the PLCopen package have revolutionized the development of safety applications. Because they are certified, they save time and reduce costs throughout all phases of a safety application's life cycle. From specification and implementation to testing and verification functions, the approach is more like virtual wiring than programming.

Unlike "real wiring", downloading the program to the safety controller guarantees that an identical copy will be stored. This completely eliminates wiring errors during series production. Naturally, all options for a safe programmable controller are available to handle even more complex challenges that can't be solved with "real wiring".

Function block	Description
"SF_CamMonitoring"	Monitors a cam switch unit
"SF_CamshaftMonitor"	Monitors the cam shaft for breakage
"SF_CycleControl"	Evaluates cyclic mode
"SF_DoubleValveMonitoring"	Controls a double valve with a feedback signal
"SF_FootSwitch"	Evaluates a footswitch
"SF_PressControl"	Sequential control of a press
"SF_SingleValveCycleMonitoring"	Controls a single valve with cycle monitoring
"SF_SingleValveMonitoring"	Controls a single valve with feedback signal
"SF_TwoHandControlTypeIIIC"	Evaluates a two-handed control device with monitoring of simultaneous actuation
"SF_TwoHandMultiOperator"	Evaluates two two-handed control devices
"SF_ValveGroupControl"	Controls a valve group with up to eight valves

6.5.2 System requirements

The library for presses is part of SafeDESIGNER and only permitted to be used therein.

The following prerequisites must be met before the library for presses can be used:

- SafeDESIGNER: 4.1.1 or higher
- Automation Studio: 4.1 or higher
- SafeLOGIC: Safety Release 1.7 or higher
- SafeLOGIC-X: Currently not supported
- Software license for using the library for presses

6.5.3 Definition of terms

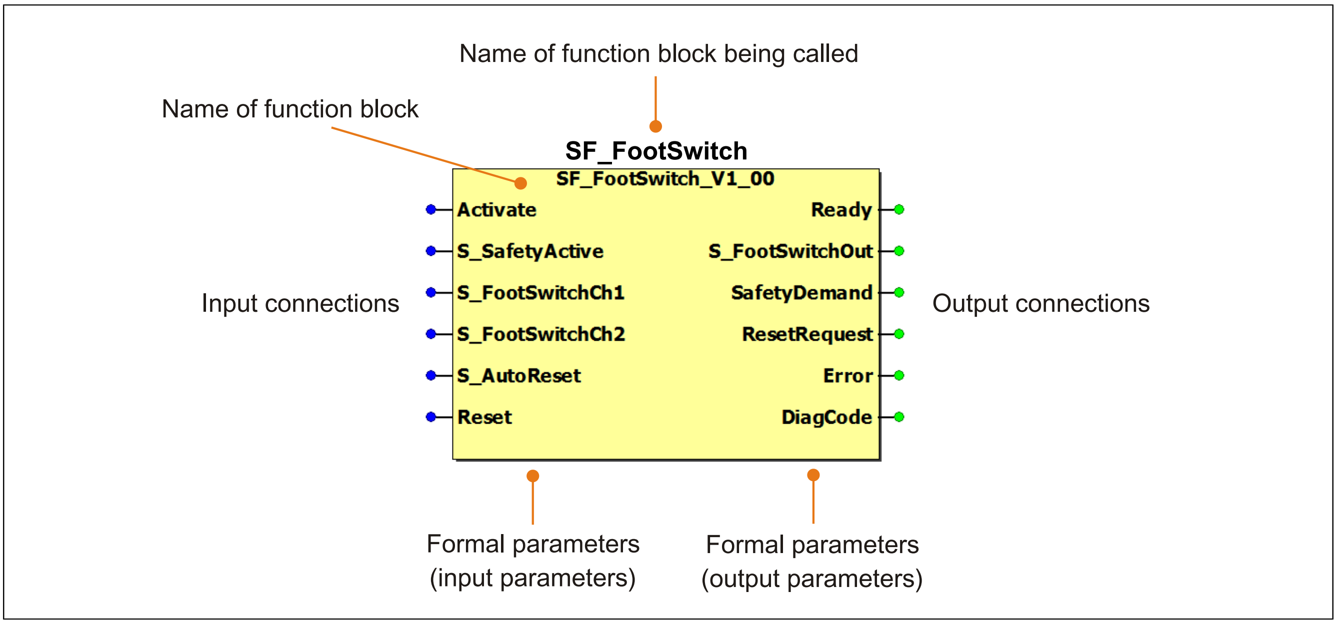


Figure 361: Function block label

When calling a function block, the inputs supply the input parameters with the current values of the variables or constants.
The output parameters supply the outputs with the associated values.

Inputs or outputs do not need to share the same name as the associated formal parameters, but they must be of the same data type. A difference in data type between formal parameters and inputs/outputs is reported as an error following compilation.

A function block's name is created from the function itself (e.g. "SF_FootSwitch", SF = safety function) and its version (Vx_yz). The representation for version Vx_yz used in this document is universal. The actual version can be determined from the function block in use.

6.5.4 SF_CamMonitoring

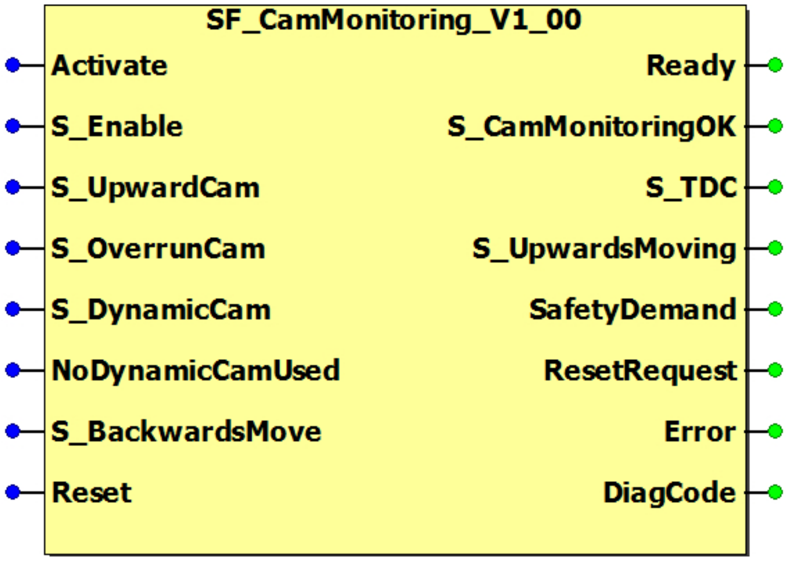


Figure 362: "SF_CamMonitoring" function block

6.5.4.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Enable	SAFEBOOL	Variable/ Constant	Status	FALSE	Control signal for enabling monitoring of a cam switch unit
S_UpwardCam	SAFEBOOL	Variable	Status	FALSE	Input for upward cam
S_OverrunCam	SAFEBOOL	Variable	Status	FALSE	Input for overrun cam
S_DynamicCam	SAFEBOOL	Variable	Status	FALSE	Input for dynamic cam
NoDynamicCamUsed	BOOL	Constant	Status	FALSE	Specifies the use of a dynamic cam
S_BackwardsMove	SAFEBOOL	Variable	Status	FALSE	Input for backward movement of the press
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 532: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_CamMonitoringOK	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_TDC	SAFEBOOL	Variable	Status	FALSE	Indicates top dead center (TDC)
S_UpwardsMoving	SAFEBOOL	Variable	Status	FALSE	Indicates the upward movement of the press
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 533: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.4.2 Function

The "SF_CamMonitoring" function block provides support for monitoring a mechanical cam switch unit.

> ## Information:
>
> **Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.**

A defined signal sequence from the cams must be adhered to. This function block supports operation with or without dynamic cams.

In order to enable cam switch unit monitoring, the "S_Enable" input parameter must be set to TRUE.

This function block provides information about whether the press is at top dead center (TDC) or performing an upward movement.

> ## Information:
>
> **The enabling signal "S_CamMonitoringOK" is set to TRUE as long as there are no errors in the sequence or with the signals.**
>
> **If errors in the sequence or with the signals occur, the function block transitions into an error state and "S_CamMonitoringOK" switches to FALSE.**

6.5.4.2.1 Operation without a dynamic cam

Setting the "NoDynamicCamUsed" input parameter to TRUE specifies that no dynamic cam is being used.

When operating without a dynamic cam, an upward cam (input parameter "S_UpwardCam") and overrun cam (input parameter "S_OverrunCam") are used.

The "S_DynamicCam" input parameter is not to be used in this mode.

Information:

If a TRUE signal is detected on the "S_DynamicCam" input parameter, then the function block transitions to an error state.

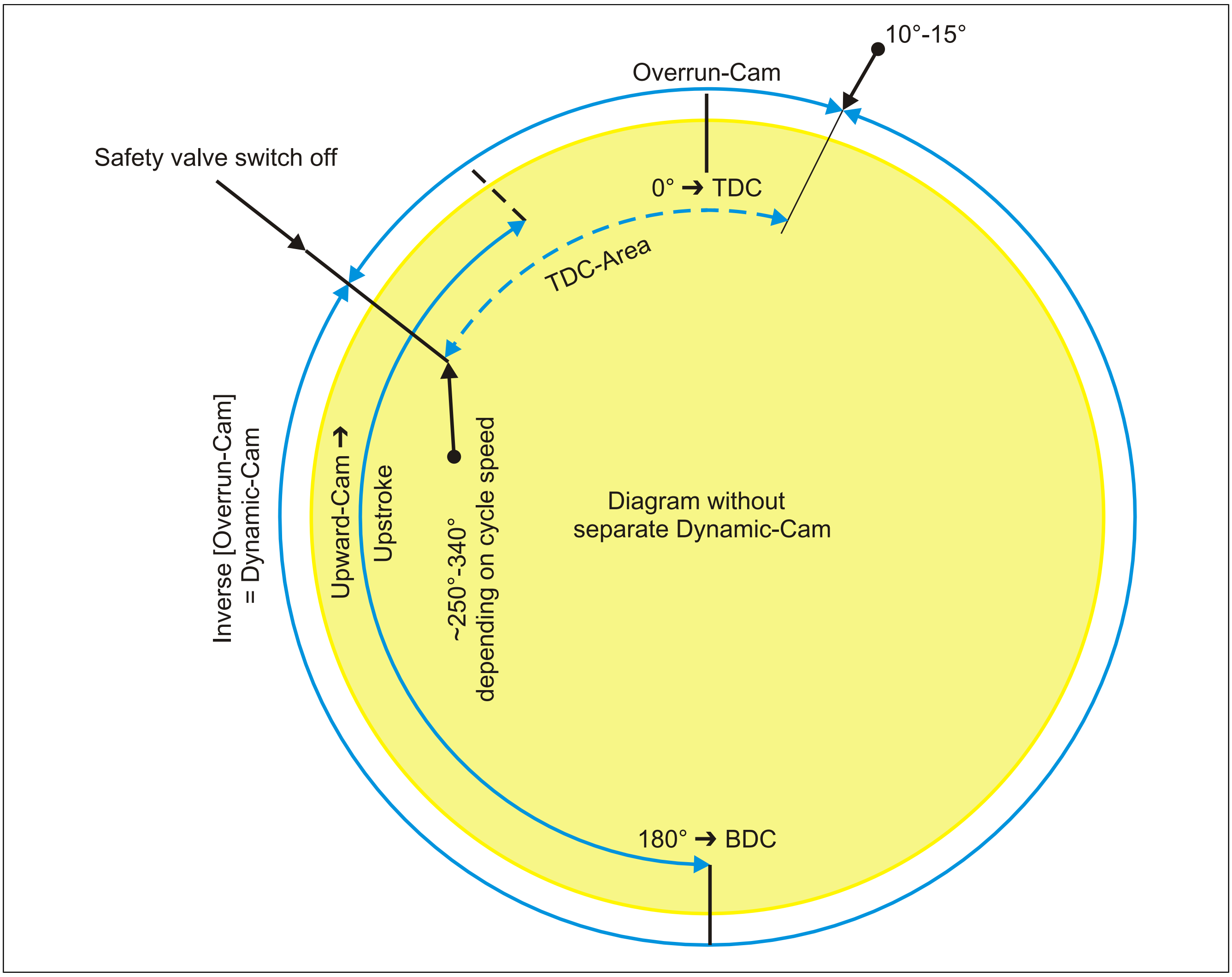


Figure 363: Operation without a dynamic cam

6.5.4.2.2 Operation with a dynamic cam

Setting the "NoDynamicCamUsed" input parameter to FALSE specifies that a dynamic cam is being used.

An upward cam (input parameter "S_UpwardCam"), overrun cam (input parameter "S_OverrunCam") and a dynamic cam (input parameter "S_DynamicCam") are used.

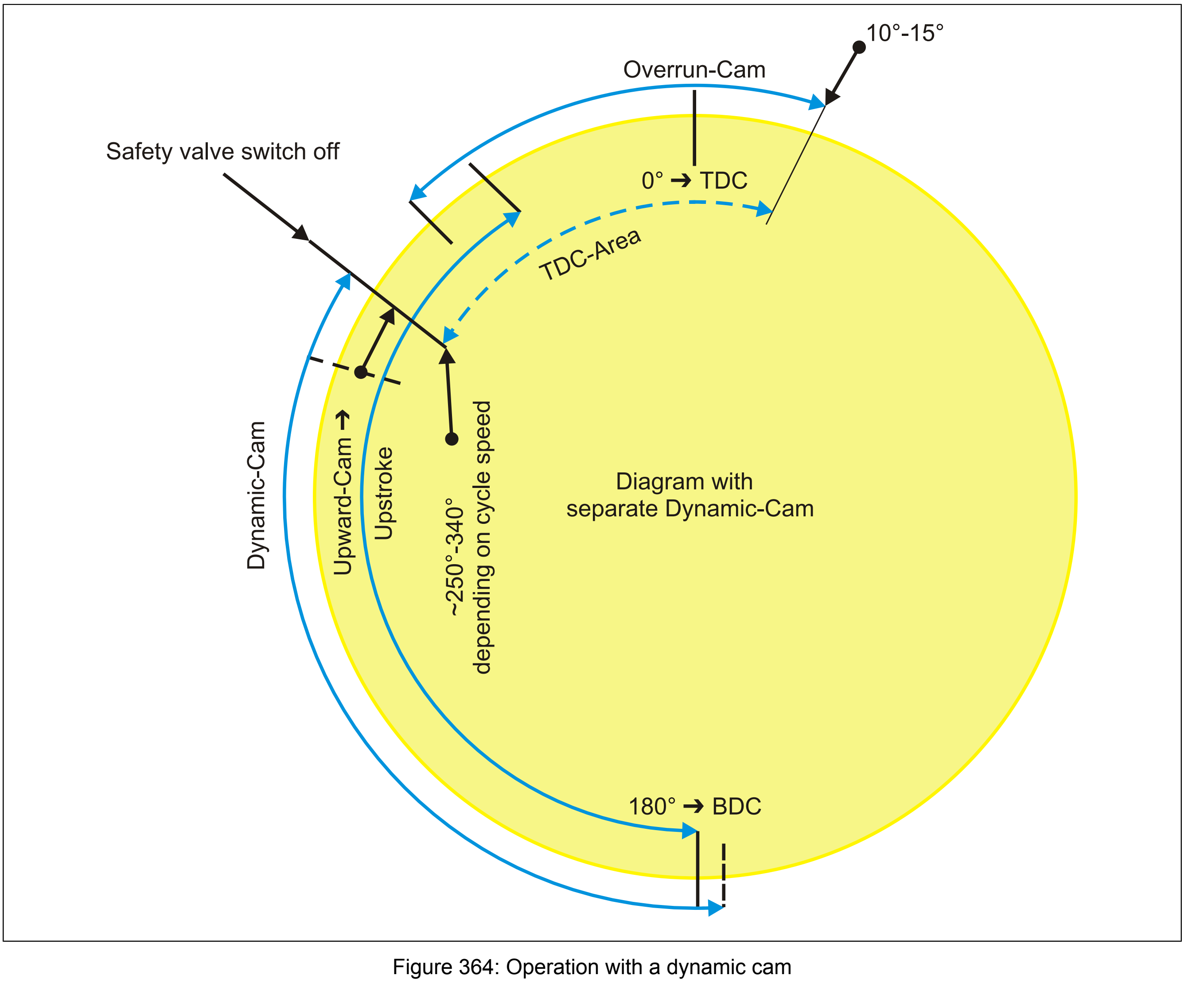


Figure 364: Operation with a dynamic cam

6.5.4.2.3 Monitoring the cam switch unit

In order to enable cam switch unit monitoring, the "S_Enable" input parameter must be set to TRUE.

A check is made at the beginning to determine whether the press is in the starting position – top dead center (TDC). If this is the case, this status is indicated by the "S_TDC" output parameter.

The cams must return the following signals for this:

- "S_UpwardCam" = FALSE
- "S_OverrunCam" = TRUE
- "S_DynamicCam" = FALSE

If the press cycle is started, then the press will move out of the TDC area.

The cams must return the following signals for this:

- "S_UpwardCam" = FALSE
- "S_OverrunCam" = FALSE
- "S_DynamicCam" = FALSE

After the downward movement has been performed, the press will be in the bottom dead center (BDC) area. At this point in time, the function block's "S_UpwardsMoving" output parameter indicates that the upward movement is active.

The cams must return the following signals for this:

- "S_UpwardCam" = TRUE
- "S_OverrunCam" = FALSE
- "S_DynamicCam" = TRUE (if a dynamic cam is being used)

During the upward movement, the press returns to the TDC area. If a dynamic cam is being used, a falling edge on it will indicate that this area has been reached via the "S_TDC" output parameter. If a dynamic cam is not being used, a rising edge of the "S_OverrunCam" input parameter is required.

The cams must return the following signals for this:

- "S_UpwardCam" = TRUE
- "S_OverrunCam" = TRUE
- "S_DynamicCam" = FALSE

The end of the upward movement is indicated by a falling edge of the "S_UpwardCam" input parameter and the state of the "S_UpwardsMoving" output parameter. At the end of the cycle, the press is again at TDC; for details, see the starting situation.

You can start a new cycle, which will repeat the sequence.

6.5.4.2.4 Backward movement

The possibility to perform a backward movement can be used to move the press backwards to TDC while monitoring of the cam switch unit is enabled. This could be the case when leaving TDC and then stopping the press shortly thereafter, for example.

To start, you first have to enable the backward movement with the "S_BackwardsMove" input parameter.

The cams must return the following signals for this:

- "S_UpwardCam" = FALSE
- "S_OverrunCam" = FALSE
- "S_DynamicCam" = FALSE

During the backward movement, the function block must detect a rising edge of the overrun cam ("S_OverrunCam" input parameter). This means that the press is located at TDC again ("S_TDC" output parameter = TRUE). The backward movement can be stopped.

6.5.4.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.4.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.4.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.4.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.4.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.4.4 Input parameters

6.5.4.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.4.4.2 S_Enable

General function

- Control signal for enabling monitoring of a cam switch unit

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

This input parameter serves as a control signal for enabling monitoring of a cam switch unit.

TRUE

Monitoring of cam switch unit is active.

FALSE

Monitoring of cam switch unit is not active.

6.5.4.4.3 S_UpwardCam

General function

- Input for upward cam

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_UpwardCam" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The upward cam is not active.

FALSE

The upward cam is active.

6.5.4.4.4 S_OverrunCam

General function

- Input for overrun cam

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_OverrunCam" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The overrun cam is not active.

FALSE

The overrun cam is active.

6.5.4.4.5 S_DynamicCam

General function

- Input for dynamic cam

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_DynamicCam" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The dynamic cam is not active.

FALSE

The dynamic cam is active.

6.5.4.4.6 NoDynamicCamUsed

General function

- Specifies the use of a dynamic cam

Data type

- BOOL

Connection

- Constant

Function description

This input parameter specifies the use of a dynamic cam for the cam switch unit.

TRUE

A dynamic cam is not being used for the cam switch unit. The "S_DynamicCam" input parameter is not to be used.

> ## Information:
>
> **If a TRUE signal is detected on the "S_DynamicCam" input parameter, then the function block transitions to an error state.**

FALSE

A dynamic cam is being used for the cam switch unit. The corresponding signal must be connected to the "S_DynamicCam" input parameter.

6.5.4.4.7 S_BackwardsMove

General function

- Input for backward movement of the press

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter serves as a control signal to indicate that a backward movement of the press is active.

This input parameter can be used to move the press backwards to TDC while monitoring of the cam switch unit is enabled. This could be the case when leaving TDC and then stopping the press shortly thereafter, for example.

> ## Information:
>
> **After completing the backward movement, set the "S_BackwardsMove" input parameter back to FALSE.**

TRUE

Moves the press backwards (counterclockwise)

FALSE

Moves the press forwards (clockwise)

6.5.4.4.8 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

> ## Danger!
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.4.5 Output parameters

6.5.4.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.4.5.2 S_CamMonitoringOK

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the status of cam switch unit monitoring.

If no monitoring errors exist or monitoring is not active, then the function block returns a TRUE signal.

The enable signal can be used for subsequent process control.

> **Danger!**
>
> **The enable signal is only permitted to control the process directly if it does not impair the safety function.**
>
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

Cam switch unit monitoring is not returning errors or is not active.

FALSE

Cam switch unit monitoring is returning an error, or there is a signal error.

6.5.4.5.3 S_TDC

General function

- Indicates top dead center (TDC)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether the press is at top dead center (TDC).

TRUE

The press is at TDC.

FALSE

The press is not at TDC.

6.5.4.5.4 S_UpwardsMoving

General function

- Indicates the upward movement of the press

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates the upward movement of the press – from bottom dead center (BDC) to top dead center (TDC).

TRUE

The press is performing an upward movement.

FALSE

The press is not performing an upward movement.

6.5.4.5.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.4.5.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.4.5.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.4.5.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.4.5.9 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The press is at TDC. • "S_CamMonitoringOK" = TRUE • "S_TDC" = TRUE • "S_UpwardsMoving" = FALSE	No corrective measures are necessary if the signal is intended. A falling edge on "S_OverrunCam" is required to leave the top dead center (TDC) area.
8001	Initialization of the function block after it has been enabled.	No corrective measures are required.
8010	The press is outside top dead center (TDC) or performing a downward movement. • "S_CamMonitoringOK" = TRUE • "S_TDC" = FALSE • "S_UpwardsMoving" = FALSE	No corrective measures are necessary if the signal is intended.
8020	The press is at bottom dead center (BDC) or performing an upward movement. • "S_CamMonitoringOK" = TRUE • "S_TDC" = FALSE • "S_UpwardsMoving" = TRUE	No corrective measures are necessary if the signal is intended.
8030	The press is in the top dead center (TDC) area. • "S_CamMonitoringOK" = TRUE • "S_TDC" = TRUE • "S_UpwardsMoving" = TRUE	No corrective measures are necessary if the signal is intended.
8040	The press is performing a backward movement. • "S_CamMonitoringOK" = TRUE • "S_TDC" = FALSE • "S_UpwardsMoving" = TRUE	No corrective measures are necessary if the signal is intended.
8050	The function block is enabled, but monitoring of the cam switch unit has not been enabled with the "S_Enable" input parameter. • "S_CamMonitoringOK" = TRUE	Enable the function by setting "S_Enable" to TRUE.
8802	The signal combination of connected cams is incorrect for starting. The following input parameter combination must be present: • "S_UpwardCam" = FALSE • "S_OverrunCam" = TRUE • "S_DynamicCam" = FALSE or "NoDynamicCamUsed" = TRUE	Bring the press to top dead center (starting position).
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C031	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C041	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C051	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C061	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C400	A TRUE signal was detected on the "S_DynamicCam" input parameter even though the dynamic cam is disabled ("NoDynamicCamUsed" = TRUE).	• Check whether a dynamic cam is being used. • If necessary, correct the error. • Reset the function block.

Table 534: Diagnostic codes

Code (hex)	Description	Corrective measures
C500	Incorrect signal combination at top dead center (TDC) • "S_BackwardsMove" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C510	Incorrect signal combination at top dead center (TDC) • "S_UpwardCam" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C520	Incorrect signal combination at top dead center (TDC) • "S_DynamicCam" = TRUE • "NoDynamicCamUsed" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C530	Incorrect signal combination for downward movement • "S_OverrunCam" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C540	Incorrect signal combination for downward movement • Falling edge on "S_DynamicCam" • "NoDynamicCamUsed" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C600	Incorrect signal combination for upward movement • "S_BackwardsMove" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C610	Incorrect signal combination for upward movement • "S_UpwardCam" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C620	Incorrect signal combination for upward movement • Falling edge on "S_DynamicCam" • "NoDynamicCamUsed" = FALSE • "S_OverrunCam" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C630	Incorrect signal combination for upward movement • Rising edge on "S_OverrunCam" • "S_DynamicCam" = TRUE • "NoDynamicCamUsed" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C640	Incorrect signal combination when entering top dead center (TDC) • "S_BackwardsMove" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C650	Incorrect signal combination when entering top dead center (TDC) • "S_OverrunCam" = FALSE • "NoDynamicCamUsed" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C660	Incorrect signal combination when entering top dead center (TDC) • "S_DynamicCam" = TRUE • "NoDynamicCamUsed" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C700	Incorrect signal combination for backward movement • "S_BackwardsMove" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C710	Incorrect signal combination for backward movement • "S_UpwardCam" = TRUE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C720	Incorrect signal combination for backward movement • "S_DynamicCam" = TRUE • "NoDynamicCamUsed" = FALSE	• Check the connected signals. • If necessary, correct the error. • Reset the function block.

Table 534: Diagnostic codes

6.5.4.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"NoDynamicCamUsed" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

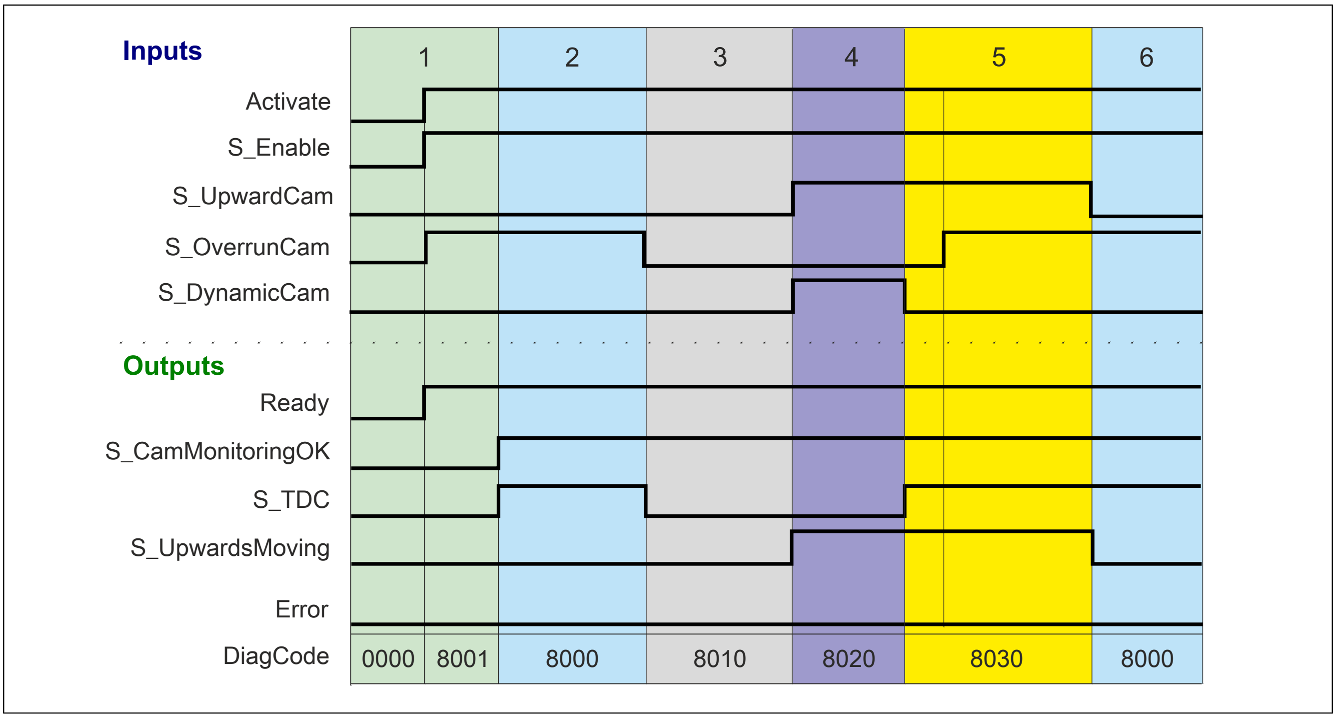


Figure 365: "SF_CamMonitoring": Signal sequence diagram 1

1 Initialization
2 Press at top dead center (TDC)
3 Press performing downward movement
4 Press performing upward movement
5 Press in top dead center (TDC) area
6 Press at top dead center (TDC)

Signal sequence diagram 2

"NoDynamicCamUsed" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

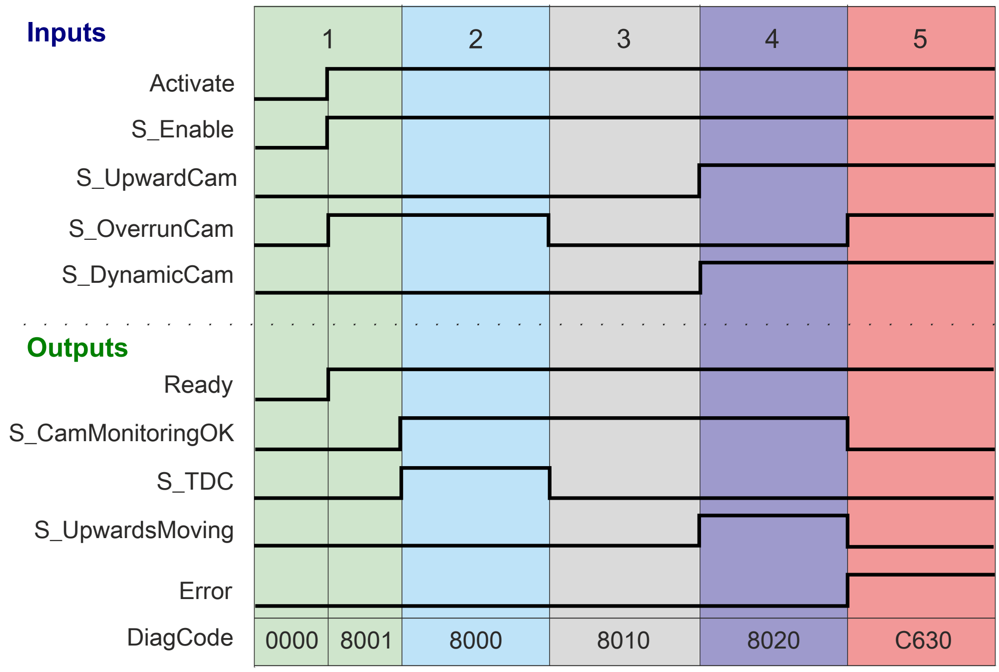


Figure 366: "SF_CamMonitoring": Signal sequence diagram 2

1 Initialization
2 Press at top dead center (TDC)
3 Press performing downward movement
4 Press performing upward movement
5 Error - Incorrect signal combination for upward movement

Signal sequence diagram 3

"NoDynamicCamUsed" = TRUE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

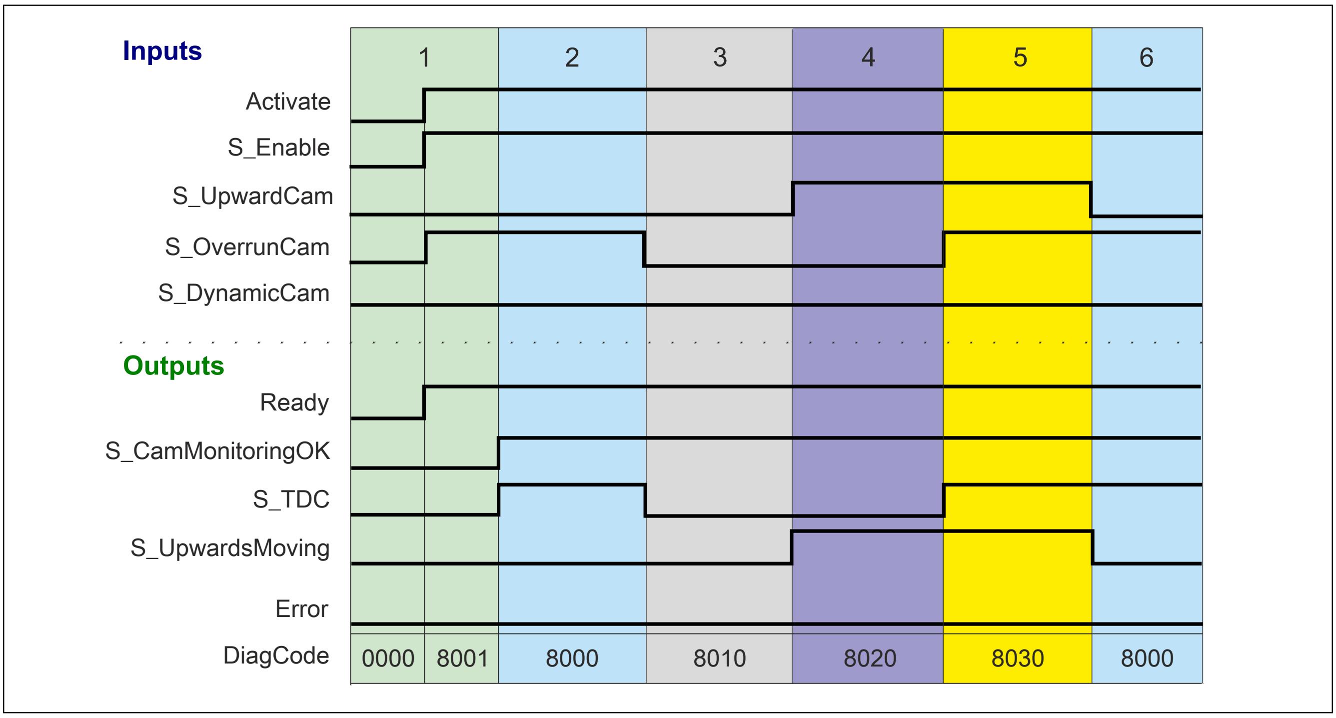


Figure 367: "SF_CamMonitoring": Signal sequence diagram 3

1 Initialization
2 Press at top dead center (TDC)
3 Press performing downward movement
4 Press performing upward movement
5 Press in top dead center (TDC) area
6 Press at top dead center (TDC)

6.5.4.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN 692	5.4.2.4 c 5.4.2.5 5.4.2.6 5.4.2.9 d 5.4.7.2 5.4.7.3
EN 693	5.4.5

Table 535: "SF_CamMonitoring": Implementing requirements from standards

6.5.5 SF_CamshaftMonitor

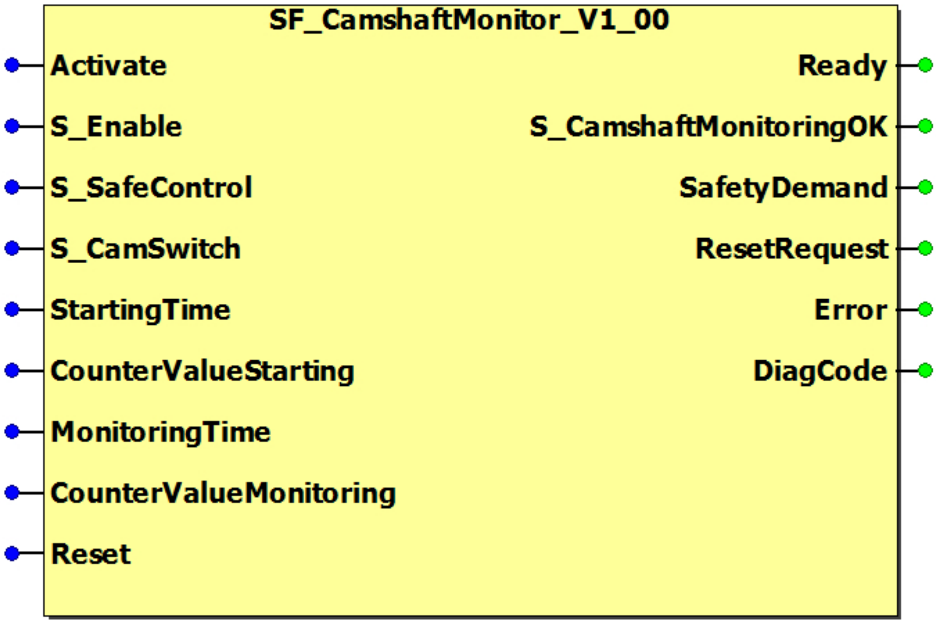


Figure 368: "SF_CamshaftMonitor" function block

6.5.5.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Enable	SAFEBOOL	Variable/ Constant	Status	FALSE	Control signal for enabling monitoring of the shaft
S_SafeControl	SAFEBOOL	Variable	Status	FALSE	Control signal for indicating the press is moving
S_CamSwitch	SAFEBOOL	Variable	Status	FALSE	Input for the cam switch of the shaft
StartingTime	TIME	Constant	Status	T#0ms	Specifies the startup time of the press
CounterValueStarting	INT	Constant	Status	INT#0	Specifies the number of signal changes to be reached during the startup time
MonitoringTime	TIME	Constant	Status	T#0ms	Specifies the monitoring time of the press
CounterValueMonitor-ing	INT	Constant	Status	INT#0	Specifies the number of signal changes to be reached during the monitoring time
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 536: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_CamshaftMonito-ringOK	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 537: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.5.2 Function

The "SF_CamshaftMonitor" function block provides support for monitoring the shaft.

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

The pulse from a cam (cam shaft monitoring) is used to check whether the shaft is defective.

The function block differentiates between the startup phase and the operating phase.

In order to enable shaft monitoring, the "S_Enable" input parameter must be set to TRUE.

Information:

If multiple cams should be used to monitor the shaft, then the function block must be instanced multiple times.

The function block supports up to 20 strokes (40 signal changes). During this, the rising and falling edges are counted on the "S_CamSwitch" input parameter.

Information:

The enabling signal "S_CamshaftMonitoringOK" is set to TRUE as long as there are no errors in the sequence or with the signals.

If errors in the sequence or with the signals occur, the function block transitions into an error state and the enable signal switches to FALSE.

6.5.5.2.1 Startup phase

The startup phase is defined by the startup time ("StartingTime" input parameter) and associated counter value ("CounterValueStarting" input parameter).

During the startup time, the function block must achieve at least the value defined with the "CounterValueStarting" input parameter via the signal change of the "S_CamSwitch" input parameter.

Information:

If the function block detects more than 50 signal changes during the startup time, it will switch to an error state.

6.5.5.2.2 Operating phase

The function block switches to the operating phase automatically after completing the startup phase. The operating phase is defined by the monitoring time ("MonitoringTime" input parameter) and associated counter value ("CounterValueMonitoring" input parameter).

During the monitoring time, the function block must achieve at least the value defined with the "CounterValueMonitoring" input parameter via the signal change of the "S_CamSwitch" input parameter.

Danger!

Make sure that the monitoring time is configured to be less than one press cycle.

Information:

If the function block detects more than 50 signal changes during the monitoring time, it will switch to an error state.

Information:

The monitoring time must be defined as less than the startup time. If the time is greater than this, the function block will transition to an error state.

6.5.5.2.3 Monitoring the shaft

In order to enable shaft monitoring, the "S_Enable" input parameter must be set to TRUE.

The "S_SafeControl" input parameter is used to notify the function block that a movement on the press is taking place. In addition, monitoring of the startup phase is started.

Information:

To control the "S_SafeControl" input parameter, you should use the control signal for the press safety valve or the feedback signal from the press safety valve.

During the startup time, the function block must achieve at least the value defined with the "CounterValueStarting" input parameter via the signal change of the "S_CamSwitch" input parameter. If this takes place, the function block transitions to the operating phase.

During the monitoring time, the function block must achieve at least the value defined with the "CounterValueMonitoring" input parameter via the signal change of the "S_CamSwitch" input parameter. If this takes place, the function block remains in the operating phase and monitors signal changes during the monitoring period.

6.5.5.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.5.3.1 "MonitoringTime" greater than one press cycle

The function block doesn't detect that "MonitoringTime" is greater than one press cycle.

> **Danger!**
>
> **Make sure that the monitoring time is configured to be less than one press cycle.**

6.5.5.3.2 Incorrect connection of "StartingTime" and "MonitoringTime"

The function block detects an error if the constant on the "MonitoringTime" input parameter is greater than the one connected to the "StartingTime" input parameter.

Possible cause:

- "MonitoringTime" constant greater than "StartingTime" constant

6.5.5.3.3 Incorrect connection of "CounterValueStarting" and "CounterValueMonitoring"

The function block detects an error if an illicit constant is connected to the "CounterValueStarting" or "CounterValueMonitoring" input parameter.

Possible cause:

- Constant connected to "CounterValueStarting" ≤ 0
- Constant connected to "CounterValueMonitoring" ≤ 0

6.5.5.3.4 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.5.3.5 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.5.3.6 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.5.3.7 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.5.4 Input parameters

6.5.5.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.5.4.2 S_Enable

General function

- Control signal for enabling monitoring of the shaft

Data type

- SAFEBOOL

Connection

- Variable or constant

Function description

This input parameter serves as a control signal for enabling monitoring of a shaft.

TRUE

Monitoring of the shaft is active.

FALSE

Monitoring of the shaft is not active.

6.5.5.4.3 S_SafeControl

General function

- Control signal for indicating the press is moving

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter serves as the control signal to indicate an active press movement and to start monitoring.

> **Information:**
>
> **To control the "S_SafeControl" input parameter, you should use the control signal for the press safety valve or the feedback signal from the press safety valve.**

TRUE

A movement of the press is taking place and monitoring is active.

FALSE

A movement of the press is not taking place and monitoring is inactive.

6.5.5.4.4 S_CamSwitch

General function

- Input for the cam switch of the shaft

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_CamSwitch" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The cam switch unit is active.

FALSE

The cam switch unit is not active.

6.5.5.4.5 StartingTime

General function

- Specifies the startup time of the press

Data type

- TIME

Connection

- Constant

Function description

This input parameter is used to define the startup time of the press.

> ## Information:
>
> **The value for the "StartingTime" input parameter must be defined and validated based on your application.**

6.5.5.4.6 CounterValueStarting

General function

- Specifies the number of signal changes to be reached during the startup time

Data type

- INT

Connection

- Constant

Function description

This input parameter is used to define the number of signal changes to be achieved during the startup time.

> **Information:**
>
> **The value for the "CounterValueStarting" input parameter must be defined and validated based on your application.**

> **Information:**
>
> **If at least the defined number of signal changes is not detected during the startup time, then the function block transitions to an error state.**

6.5.5.4.7 MonitoringTime

General function

- Specifies the monitoring time of the press

Data type

- TIME

Connection

- Constant

Function description

This input parameter is used to define the monitoring time during the operating phase.

> ## Information:
> **The value for the "MonitoringTime" input parameter must be defined and validated based on your application.**

> ## Information:
> **The monitoring time must be defined as less than the startup time. If the time is greater than this, the function block will transition to an error state.**

6.5.5.4.8 CounterValueMonitoring

General function

- Specifies the number of signal changes to be reached during the monitoring time

Data type

- INT

Connection

- Constant

Function description

This input parameter is used to define the number of signal changes to be achieved during the monitoring time.

> ## Information:
> **The value for the "CounterValueMonitoring" input parameter must be defined and validated based on your application.**

> ## Information:
> **If at least the defined number of signal changes is not detected during the monitoring time, then the function block transitions to an error state.**

6.5.5.4.9 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.5.5 Output parameters

6.5.5.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.5.5.2 S_CamshaftMonitoringOK

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the status of shaft monitoring.

If no monitoring errors exist or monitoring is not active, then the function block return a TRUE signal.

The enable signal can be used for subsequent process control.

> **Danger!**
>
> **The enable signal is only permitted to control the process directly if it does not impair the safety function.**
>
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

Shaft monitoring is not returning errors or is not active.

FALSE

Shaft monitoring is returning an error, or there is a signal error.

6.5.5.5.3 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.5.5.4 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.5.5.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.5.5.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.5.5.7 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	No request to start monitoring is present on the "S_SafeControl" input parameter. • "S_CamshaftMonitoringOK" = TRUE	In order to start monitoring, "S_SafeControl" must be set to TRUE.
8001	Initialization of the function block after it has been enabled.	Enable the function by setting "S_Enable" to TRUE.
8100	Monitoring of the startup phase is active. • "S_CamshaftMonitoringOK" = TRUE	No corrective measures are necessary if the signal is intended.
8200	Monitoring of the operating phase is active. • "S_CamshaftMonitoringOK" = TRUE	No corrective measures are necessary if the signal is intended.
8300	The function block is enabled, but monitoring of the shaft has not been enabled with the "S_Enable" input parameter. • "S_CamshaftMonitoringOK" = TRUE	Enable the function by setting "S_Enable" to TRUE.
C000	The monitoring time ("MonitoringTime") is greater than the startup time ("StartingTime").	Configure the monitoring time ("MonitoringTime") to be less than the startup time ("StartingTime").
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	The value for the signal change in the startup phase ("CounterValueStarting") or for monitoring the operating phase ("CounterValueMonitoring") is invalid.	The values for the signal change ("CounterValueStarting" and "CounterValueMonitoring") must be greater than 0.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C400	The number of detected signal changes during the startup phase was less than the defined value ("CounterValueStarting") or greater than 50.	• Check the connected signals. • If necessary, correct the error. • Reset the function block.
C410	The number of detected signal changes during cyclic monitoring was less than the defined value ("CounterValueMonitoring") or greater than 50.	• Check the connected signals. • If necessary, correct the error. • Reset the function block.

Table 538: Diagnostic codes

6.5.5.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"CounterValueStarting" = 9
"CounterValueMonitoring" = 15

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

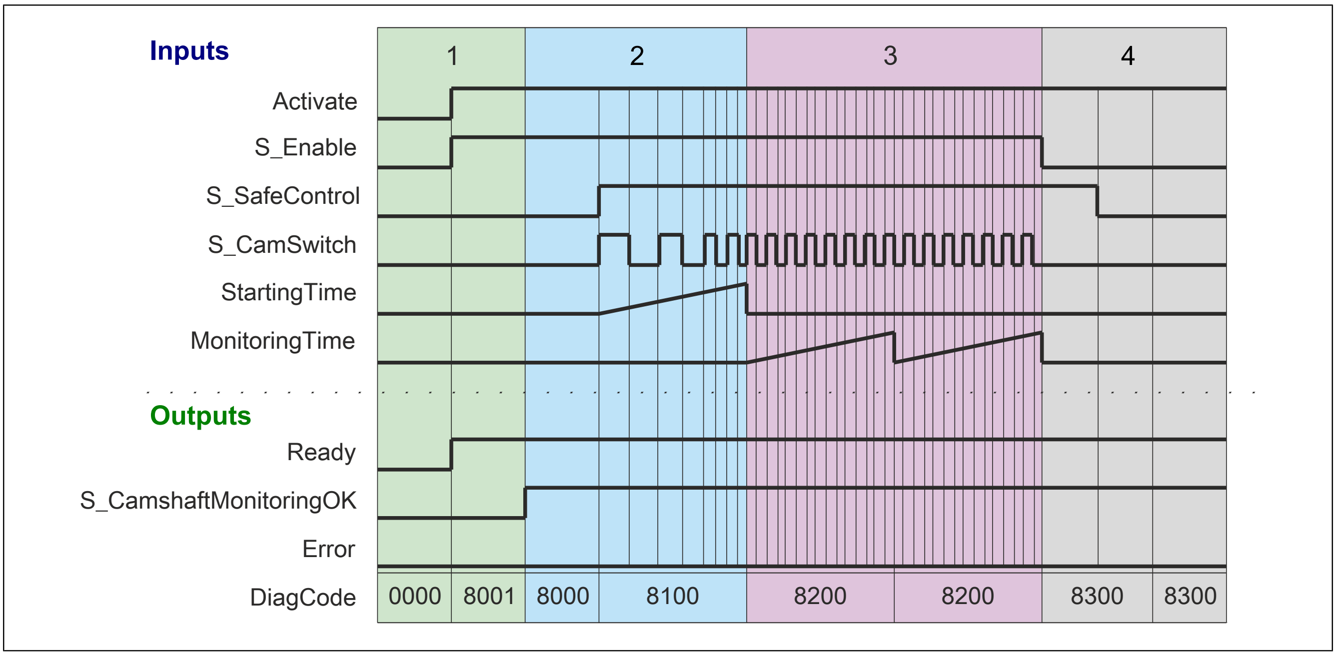


Figure 369: "SF_CamshaftMonitor": Signal sequence diagram 1

1 Initialization
2 Monitoring the startup phase
3 Monitoring the operating phase
4 Monitoring disabled

Signal sequence diagram 2

"CounterValueStarting" = 14
"CounterValueMonitoring" = 28

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

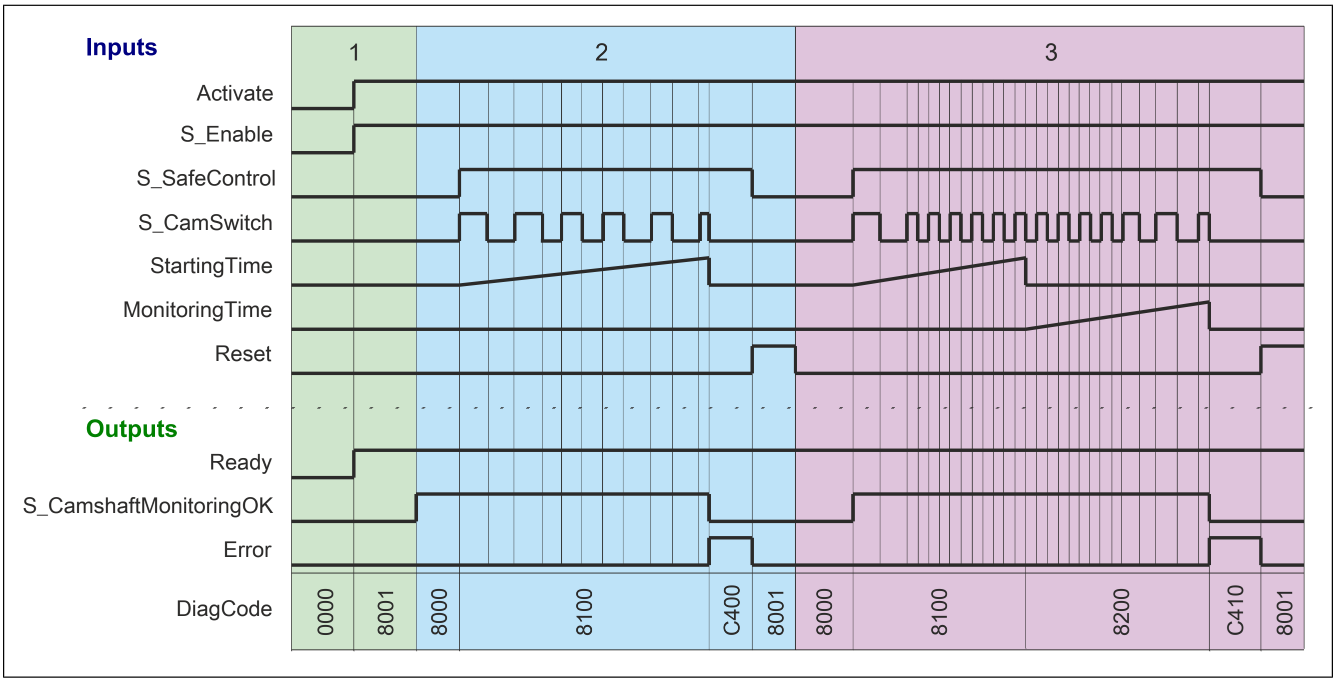


Figure 370: "SF_CamshaftMonitor": Signal sequence diagram 2

1 Initialization
2 Error - Insufficient signal change during startup phase
3 Error - Insufficient signal change during operating phase

6.5.5.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN 692	5.4.2.4 c

Table 539: "SF_CamshaftMonitor": Implementing requirements from standards

6.5.6 SF_CycleControl

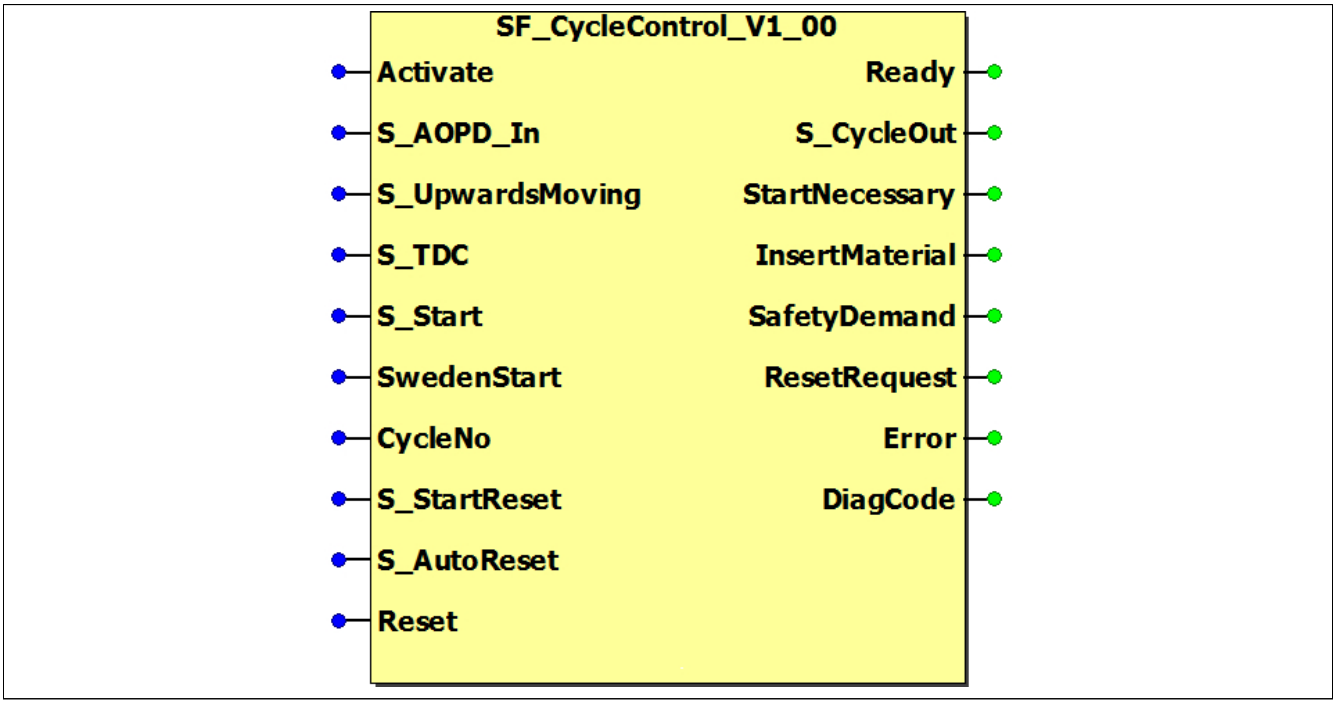


Figure 371: "SF_CycleControl" function block

6.5.6.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_AOPD_In	SAFEBOOL	Variable	Status	FALSE	Input of an active opto-electronic protective device (AOPD)
S_UpwardsMoving	SAFEBOOL	Variable/ Constant	Status	FALSE	Input for active upward movement of the press
S_TDC	SAFEBOOL	Variable	Status	FALSE	Input for position at top dead center (TDC)
S_Start	SAFEBOOL	Variable	Status	FALSE	Control signal for starting cyclic operation
SwedenStart	BOOL	Constant	Status	FALSE	Defines the start behavior
CycleNo	INT	Constant	Status	INT#0	Defines the number of cycles for operation
S_StartReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock if proper signals are present on the input parameters
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 540: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_CycleOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
StartNecessary	BOOL	Variable	Status	FALSE	Indicates that cyclic operation should be started
InsertMaterial	BOOL	Variable	Status	FALSE	Indicates that material should be inserted in order to start cyclic operation
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 541: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.6.2 Function

The "SF_CycleControl" function block supports cyclic operation of presses when using an active opto-electronic protective device (AOPD).

> **Information:**
>
> **Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.**

Two initialization modes are supported: standard startup and Sweden startup.

The function block returns information regarding whether a starting sequence is necessary or whether materials must be inserted into the press.

6.5.6.2.1 Initialization with a standard startup

The standard startup is defined by setting the "SwedenStart" input parameter to FALSE.

The first press cycle must be started manually. A check is made to determine whether the press is in the starting position – top dead center (TDC).

At the beginning, the protective device on the "S_AOPD_In" input parameter must be interrupted as many times as there are cycles on the "CycleNo" input parameter.

> **Information:**
>
> **A cycle is defined as the signal transition from FALSE to TRUE on the "S_AOPD_In" input parameter. The interruption must be longer than 100 ms.**

> **Information:**
>
> **The time between two interruptions / two cycles is not permitted to be longer than 30 seconds.**

If the corresponding number of cycles is reached, then a rising edge of the "S_Start" input parameter is required in order to set the "S_CycleOut" enable output to TRUE. This executes the first cycle.

> **Information:**
>
> **The starting signal is not permitted to be enabled later than 30 seconds after the last interrupted cycle.**

6.5.6.2.2 Initialization with a Sweden startup

The Sweden startup is defined by setting the "SwedenStart" input parameter to TRUE.

The first press cycle must be started manually. A check is made to determine whether the press is in the starting position – top dead center (TDC).

At the beginning, a rising edge of the "S_Start" input parameter is required.

The protective device on the "S_AOPD_In" input parameter must then be interrupted as many times as there are cycles on the "CycleNo" input parameter.

> **Information:**
>
> **A cycle is defined as the signal transition from FALSE to TRUE on the "S_AOPD_In" input parameter. The interruption must be longer than 100 ms.**

> **Information:**
>
> **The first cyclic interruption is not permitted to take place later than 30 seconds after the starting signal.**
>
> **The time between two interruptions / two cycles is not permitted to be longer than 30 seconds.**

If the corresponding number of cycles is reached, then the "S_CycleOut" enable output is set to TRUE and the first cycle can be executed.

6.5.6.2.3 Cyclic operation

After the first cycle has been executed – leaving and returning to top dead center (TDC) – the system switches to actual cyclic operation.

Information:

The start of a new cycle is not permitted to take place later than 30 seconds after the end of the previous cycle. If this is not observed, initialization will have to take place again.

If the press is at top dead center (TDC) (input parameter "S_TDC" = TRUE) or performing an upward movement (input parameter "S_UpwardsMoving"), then the next interaction for starting the next cycle can already be initiated.

If the press is at top dead center (TDC), the "S_CycleOut" enable signal is set to FALSE.

The protective device on the "S_AOPD_In" input parameter must be interrupted as many times as there are cycles on the "CycleNo" input parameter.

Information:

The time between two interruptions / two cycles is not permitted to be longer than 30 seconds.

If the corresponding number of cycles is reached, then the "S_CycleOut" enable output is set to TRUE and the next cycle can be executed.

The same sequence then starts again from the beginning.

Information:

The function block detects an interruption of the protective device during a downward movement of the press. In this case, the system transitions to an error state.

Danger!

If the upward movement is dangerous or no intervention is permitted, you are not permitted to use the "S_UpwardsMoving" input parameter, or it must be connected with a constant (FALSE).

Information:

As long as there are no errors in the sequence or for the signals, then the "S_CycleOut" enable signal is set to TRUE.

If errors in the sequence or with the signals occur, the function block transitions into an error state, the enable signal switches to FALSE and you will have to perform another initialization.

6.5.6.2.4 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters.

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state. The start interlock only becomes inactive after manual intervention (rising edge) on the "Reset" input parameter if the input signal combination is permitted for this.

Danger!

Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.

6.5.6.2.5 Start interlock after cold restart (optional)

Support for a start interlock must be defined accordingly on the input parameter after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the enable signal as needed.

6.5.6.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.6.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.6.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.6.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal during a cold restart of the safety controller causes an error message on the function block if the start interlock is defined for after the function block is enabled.

If this start interlock is not defined for a cold restart of the safety controller, then the status of "Reset" is irrelevant.

6.5.6.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.6.3.5 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.6.4 Input parameters

6.5.6.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.6.4.2 S_AOPD_In

General function

- Input of an active opto-electronic protective device (AOPD)

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_AOPD_In" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The protective device is not interrupted.

FALSE

The protective device is interrupted.

6.5.6.4.3 S_UpwardsMoving

General function

- Input for active upward movement of the press

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

The output signal "S_UpwardsMoving" from the "SF_CamMonitoring" function block must be used for the connection.

Danger!

If the upward movement is dangerous or no intervention is permitted, you are not permitted to use the "S_UpwardsMoving" input parameter, or it must be connected with a constant (FALSE).

Function description

The signal connected to the "S_UpwardsMoving" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The press is performing an upward movement.

FALSE

The press is not performing an upward movement.

6.5.6.4.4 S_TDC

General function

- Input for position at top dead center (TDC)

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **The output signal "S_TDC" from the "SF_CamMonitoring" function block must be used for the connection.**

Function description

The signal connected to the "S_TDC" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The press is at the starting position – at TDC.

FALSE

The press is not at the starting position – not at TDC.

6.5.6.4.5 S_Start

General function

- Control signal for starting cyclic operation

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter serves as the control signal for starting cyclic operation on a rising edge.

6.5.6.4.6 SwedenStart

General function

- Defines the start behavior

Data type

- BOOL

Connection

- Constant

Function description

This input parameter is used to define the initialization mode during startup.

TRUE

Initiates a Sweden startup

FALSE

Initiates a standard startup

6.5.6.4.7 CycleNo

General function

- Defines the number of cycles for operation

Data type

- INT

Connection

- Constant

Function description

This input parameter is used to define the number of cycles (interruptions) needed to set the enable output to TRUE.

> ## Information:
> **A cycle is defined as the signal transition from FALSE to TRUE on the "S_AOPD_In" input parameter. The interruption must be longer than 100 ms.**

6.5.6.4.8 S_StartReset

General function

- Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the startup behavior of the function block following its activation and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.5.6.4.9 S_AutoReset

General function

- Defines the start interlock if proper signals are present on the input parameters

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the operating behavior of the function block after proper signals have returned to the input parameters.

TRUE

After proper signals return to the input parameters, the function block does not support start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After proper signals return to the input parameters, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.5.6.4.10 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.6.5 Output parameters

6.5.6.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.6.5.2 S_CycleOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the status of cyclic operation.

The enable signal can be used for subsequent process control.

> **Danger!**
>
> **The enable signal is only permitted to control the process directly if it does not impair the safety function.**
>
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

Cyclic operation is not returning any errors.

FALSE

Cyclic operation is returning an error or some form of interaction is necessary (see output parameter "StartNecessary" or "InsertMaterial").

6.5.6.5.3 StartNecessary

General function

- Indicates that cyclic operation should be started

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether a rising edge of the "S_Start" input parameter is required to start cyclic operation.

> ## Information:
> **During a standard startup, startup must take place via a rising edge of the "S_Start" input parameter within 30 seconds of the request to start.**

TRUE

A starting signal with a rising edge of the "S_Start" input parameter is required.

FALSE

No starting signal is required.

6.5.6.5.4 InsertMaterial

General function

- Indicates that material should be inserted in order to start cyclic operation

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether material must be inserted to start cyclic operation.

The protective device on the "S_AOPD_In" input parameter must be interrupted as many times as there are cycles on the "CycleNo" input parameter.

> **Information:**
>
> **During a Sweden startup, the first cycle (interruption) must take place within 30 seconds of the request to insert material.**

> **Information:**
>
> **A cycle is defined as the signal transition from FALSE to TRUE on the "S_AOPD_In" input parameter. The interruption must be longer than 100 ms.**

TRUE

Material must be inserted to start cyclic operation.

FALSE

No request has been made for material to be inserted.

6.5.6.5.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.6.5.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.6.5.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.6.5.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.6.5.9 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_CycleOut" = TRUE	No corrective measures are necessary if the signal is intended.
8014	The press is at the starting position and cyclic operation can be started. To start cyclic operation, the signal on the "S_AOPD_In" input parameter must be interrupted as many times as there are cycles on the "CycleNo" input parameter. • "InsertMaterial" = TRUE	Carry out as many FALSE to TRUE signal changes on "S_AOPD_In" as there are cycles on "CycleNo".
8034	The press is at the starting position and cyclic operation can be started. Sweden startup is not enabled ("SwedenStart" = FALSE). To start cyclic operation, the signal on the "S_AOPD_In" input parameter must be interrupted as many times as there are cycles on the "CycleNo" input parameter. • "InsertMaterial" = TRUE	Carry out as many FALSE to TRUE signal changes on "S_AOPD_In" as there are cycles on "CycleNo".
8044	The press is at the starting position and cyclic operation can be started. Sweden startup is not enabled ("SwedenStart" = FALSE). To start cyclic operation, a rising edge of the "S_Start" input parameter is required. • "S_Start" = TRUE	Start cyclic operation with a rising edge on "S_Start".
8054	The press is at the starting position and cyclic operation can be started. Sweden startup is enabled ("SwedenStart" = TRUE). To start cyclic operation, a rising edge of the "S_Start" input parameter is required. • "S_Start" = TRUE	Start cyclic operation with a rising edge on "S_Start".
8401	The start interlock to be carried out after enabling the function block is defined ("S_StartReset" = FALSE), or the function block's start interlock is active ("S_AutoReset" = FALSE).	Reset the function block.
8402	The signal is TRUE again after an interruption on the "S_AOPD_In" input parameter.	Reset the function block.
8802	An irregular interruption on the "S_AOPD_In" input parameter was detected.	• Correct the interruption on "S_AOPD_In". • Check the wiring of the control device or the safe device connected to it.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	The signal for top dead center (TDC) on the "S_TDC" input parameter switched to FALSE.	Correct the problem - "S_TDC" must be TRUE.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C012	FALSE has been detected by the function block again after a static TRUE signal on "S_Start".	No corrective measures are necessary since "S_AutoReset" = TRUE.
C020	The function block detected a static TRUE signal on "S_Start".	• Correct the problem - "S_Start" must be FALSE. • Check the wiring of the control device or the safe device connected to it.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C030	The value for the number of cycles ("CycleNo") is invalid.	The value for the number of cycles must be greater than 0.
C412	FALSE has been detected by the function block again after a static TRUE signal on "S_Start".	Reset the function block.

Table 542: Diagnostic codes

6.5.6.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_StartReset" = FALSE
"S_AutoReset" = FALSE
"SwedenStart" = FALSE
"CycleNo" = INT#2

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

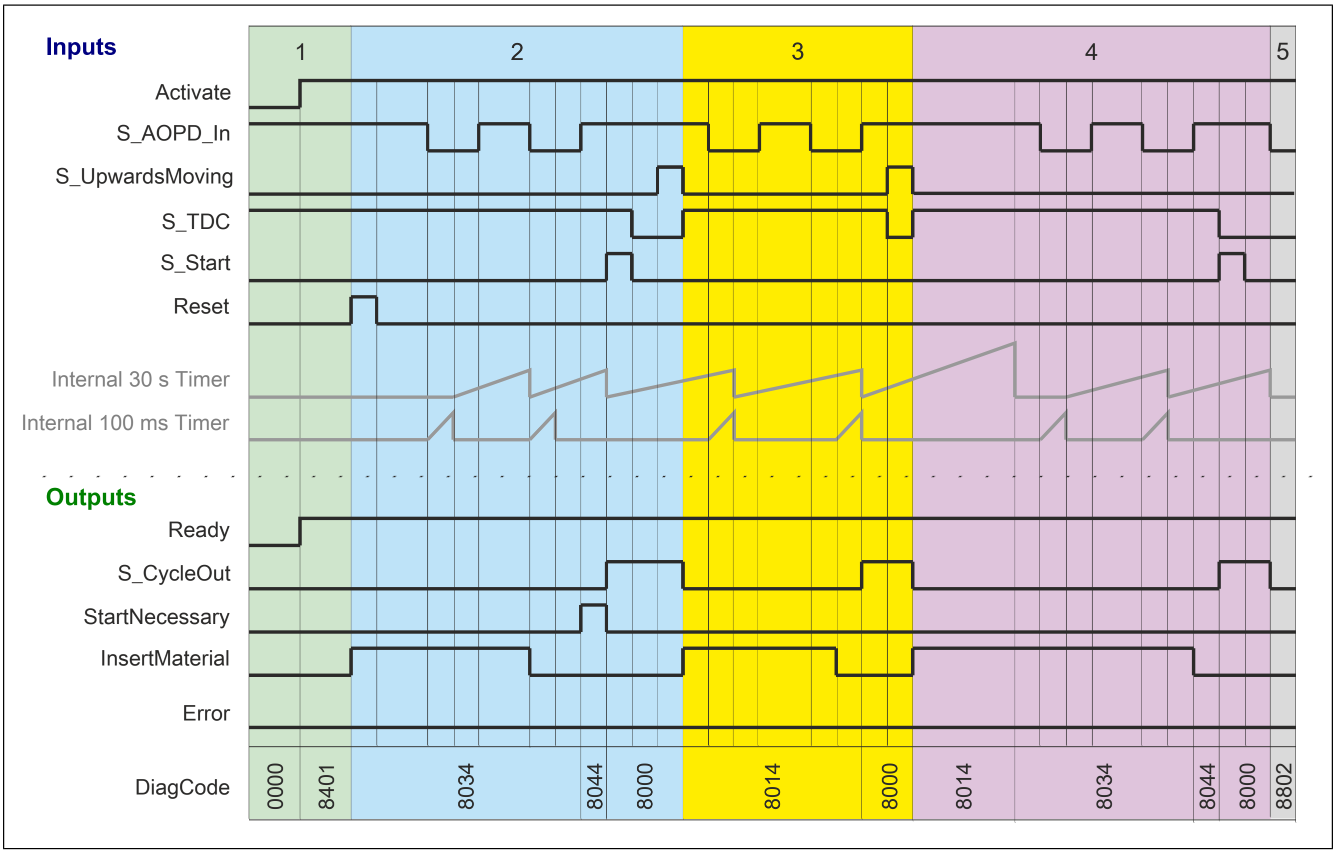


Figure 372: "SF_CycleControl": Signal sequence diagram 1

1 Initialization
2 Starts the first cycle with a standard startup
3 Cyclic operation
4 Time monitoring (30 s) expired
5 Interruption on "S_AOPD_In" input parameter

Signal sequence diagram 2

"S_StartReset" = FALSE
"S_AutoReset" = FALSE
"SwedenStart" = TRUE
"CycleNo" = INT#2

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

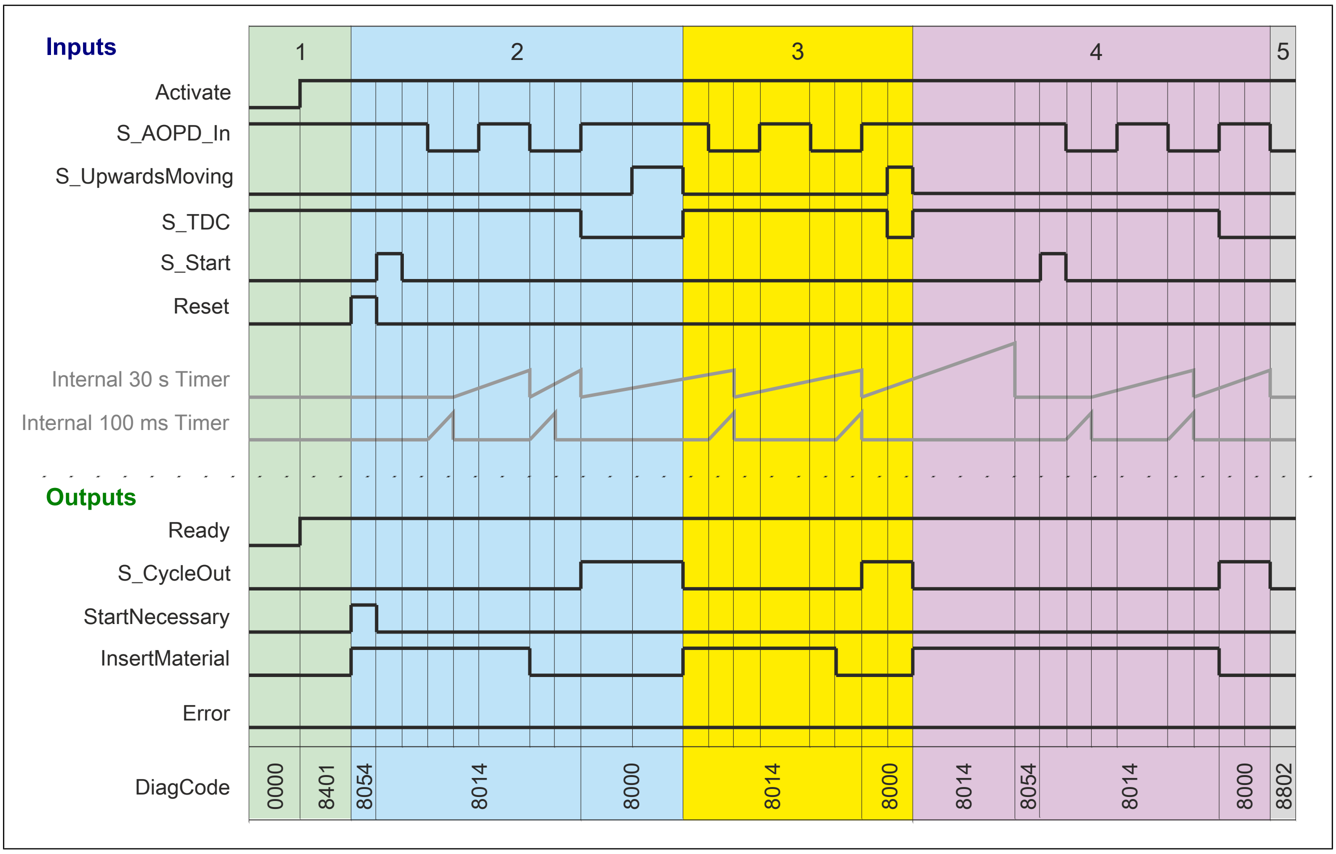


Figure 373: "SF_CycleControl": Signal sequence diagram 2

1 Initialization
2 Starts the first cycle with a Sweden startup
3 Cyclic operation
4 Time monitoring (30 s) expired
5 Interruption on "S_AOPD_In" input parameter

6.5.6.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN IEC 62046	4.7.4
EN ISO 61496-1	Appendix A A.1

Table 543: "SF_CycleControl": Implementing requirements from standards

6.5.7 SF_DoubleValveMonitoring

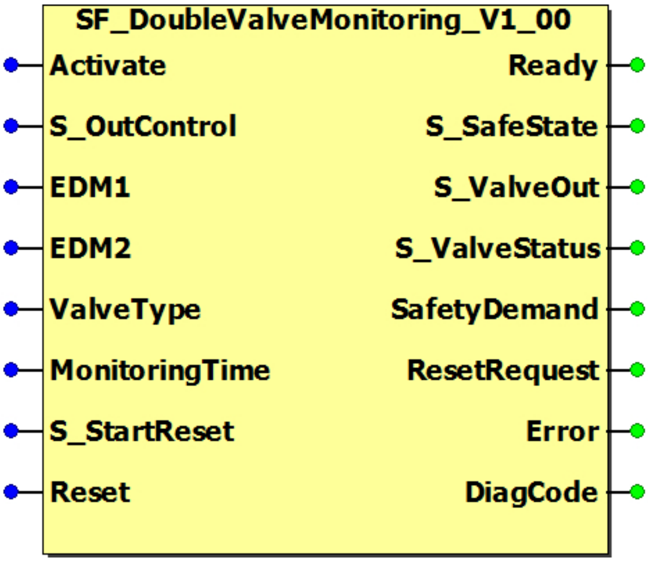


Figure 374: "SF_DoubleValveMonitoring" function block

6.5.7.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_OutControl	SAFEBOOL	Variable	Status	FALSE	Control signal for controlling valves
EDM1	BOOL	Variable	Status	FALSE	Input for feedback signal from valve 1
EDM2	BOOL	Variable	Status	FALSE	Input for feedback signal from valve 2
ValveType	BOOL	Constant	Status	FALSE	Specifies the polarity of the feedback signals
MonitoringTime	TIME	Constant	Status	T#0s	Specifies the monitoring time of the switching operation
S_StartReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 544: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_SafeState	SAFEBOOL	Variable	Status	FALSE	Indicates the basic state
S_ValveOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_ValveStatus	SAFEWORD	Variable	Status	16#0000	Status output for use with the "SF_ValveGroupControl" function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 545: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.7.2 Function

The "SF_DoubleValveMonitoring" function block checks the defined basic state and dynamic switching state of double valves (e.g. press safety valves) that are controlled by safe output devices.

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

The outputs of safe devices that are used to control the double valves to be monitored are energized to do so by the enable signal from this function block. The states of the connected double valves are fed back to the function block via safe input devices.

In the basic state, the fed back signals from the double valves being monitored must have the status defined by the function block ("ValveType" input parameter).

In the switching state, the fed back signals from the double valves being monitored must have the inverted status after the defined time window expires ("MonitoringTime" input parameter).

Information:

The "S_ValveStatus" output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group.

This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

Information:

The polarity of the feedback signals can be defined using the "ValveType" input parameter. This specifies whether the basic state is confirmed by a TRUE or FALSE signal.

6.5.7.2.1 Monitoring the basic state

If the function block detects the basic state ("ValveType" input parameter) of the double valves via the feedback signals present on the "EDM1" and "EDM2" input parameters during a switch-on request, then the function block will set its enable signal to TRUE and switch on the double valves.

If the basic state is not detected by the function block during a switch-on request, then the function block will set its enable signal to FALSE and will not switch on the double valves.

Information:

If the basic state of the valve is detected, then the "S_SafeState" output parameter is set to TRUE.

Leaving the basic state will switch "S_SafeState" to FALSE.

6.5.7.2.2 Monitoring the switching state

After energizing the double valves, the function block checks the reaction of the double valves within the specified time window ("MonitoringTime" input parameter).

If the function block does not detect the states of the feedback signals defined for the switching state (depends on the "ValveType" input parameter), then the function block will set its enable signal to FALSE and cut the double valves back off again.

6.5.7.2.3 Start interlock after cold restart (optional)

Support for a start interlock must be defined accordingly on the input parameter after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the enable signal as needed.

6.5.7.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.7.3.1 Connecting "ValveType"

> **Danger!**
>
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

6.5.7.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.7.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.7.3.4 Invalid static signals when cold restarting the safety controller

A static TRUE signal during a cold restart of the safety controller causes an error message on the function block if the start interlock is defined for after the function block is enabled.

If this start interlock is not defined for a cold restart of the safety controller, then the status of "Reset" is irrelevant.

6.5.7.3.5 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.7.3.6 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.7.4 Input parameters

6.5.7.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.7.4.2 S_OutControl

General function

- Control signal for controlling valves

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Note that no hardware input is connected on the "S_OutControl" input parameter.**

Function description

This input parameter serves as a control signal for controlling the valves.

> ## Information:
> **The "S_OutControl" input parameter should be connected to an enable signal from a safe function block.**
>
> **The status of the "S_OutControl" input parameter corresponds to the status of the upstream function block / safety function.**

The status of the "S_OutControl" input parameter controls the "S_ValveOut" output parameter under consideration of the states of the connected feedback signals.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream safety function was not triggered.

> ## Information:
> **The "S_ValveOut" output parameter is set to TRUE if the remaining input signal combination is valid for this.**

FALSE

The safety function has been initiated or the safe device connected for this safety function is switched off or faulty.

> ## Information:
> **The "S_ValveOut" output parameter is set to FALSE.**

6.5.7.4.3 EDM1

General function

- Input for feedback signal from valve 1

Data type

- BOOL

Connection

- Variable

Function description

The signal connected to the "EDM1" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

"EDM1" = TRUE and "ValveType" = FALSE

The valve is energized and not in the basic state.

"EDM1" = TRUE and "ValveType" = TRUE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM2" input parameter also returns a TRUE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM1" = FALSE and "ValveType" = FALSE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM2" input parameter also returns a FALSE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM1" = FALSE and "ValveType" = TRUE

The valve is energized and not in the basic state.

6.5.7.4.4 EDM2

General function

- Input for feedback signal from valve 2

Data type

- BOOL

Connection

- Variable

Function description

The signal connected to the "EDM2" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

"EDM2" = TRUE and "ValveType" = FALSE

The valve is energized and not in the basic state.

"EDM2" = TRUE and "ValveType" = TRUE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM1" input parameter also returns a TRUE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM2" = FALSE and "ValveType" = FALSE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM1" input parameter also returns a FALSE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM2" = FALSE and "ValveType" = TRUE

The valve is energized and not in the basic state.

6.5.7.4.5 ValveType

General function

- Specifies the polarity of the feedback signals

Data type

- BOOL

Connection

- Constant

Function description

This input parameter is used to define the polarity of the feedback signals.

> **Danger!**
>
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

TRUE

The feedback signal returns status TRUE in the basic state.

FALSE

The feedback signal returns status FALSE in the basic state.

6.5.7.4.6 MonitoringTime

General function

- Specifies the monitoring time of the switching operation

Data type

- TIME

Connection

- Constant

Function description

This input parameter is used to define the monitoring time. Switching operations on safe inputs must take place within this monitoring time in order to be recognized as valid.

Information:

The limits for the "MonitoringTime" input parameter must be defined and validated based on your application.

6.5.7.4.7 S_StartReset

General function

- Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the startup behavior of the function block following its activation and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.5.7.4.8 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.7.5 Output parameters

6.5.7.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.7.5.2 S_SafeState

General function

- Indicates the basic state

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether the valves are in the basic state.

TRUE

The valves are in the basic state.

FALSE

The valves are not in the basic state.

6.5.7.5.3 S_ValveOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is the safe signal used to control an output on a safe device, therefore also controlling the process. The enable signal is controlled depending on the status of the connected valve or valves as well as the start interlocks of the upstream safety functions.

TRUE

The output of a safe device is set to TRUE.

FALSE

The output of a safe device is set to FALSE.

6.5.7.5.4 S_ValveStatus

General function

- Status output for use with the "SF_ValveGroupControl" function block

Data type

- SAFEWORD

Connection

- Variable

Function description

This output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group. This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

6.5.7.5.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.7.5.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.7.5.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.7.5.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.7.5.9 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8010	The feedback signals on the "EDM1" and "EDM2" input parameters have changed their signal state when the request was enabled on the "S_OutControl" input parameter. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
8020	Waiting ("MonitoringTime" input parameter) for change to feedback signals on the "EDM1" and "EDM2" input parameters after the request was made via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
83FF	State for registering on the "SF_ValveGroupControl" function block.	No corrective measures are required.
8401	The start interlock to be carried out after enabling the function block is defined ("S_StartReset" = FALSE), or the function block's start interlock is active ("S_AutoReset" = FALSE).	Reset the function block.
8802	Waiting ("MonitoringTime" input parameter) on the feedback signals on the "EDM1" and "EDM2" input parameters if no request was made via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = FALSE	No corrective measures are necessary if the signal is intended.
8804	No request is present on the "S_OutControl" input parameter for setting the safe output to TRUE. • "S_SafeState" = TRUE • "S_ValveOut" = FALSE	"S_OutControl" must have status TRUE in order to set the safe output to TRUE.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	When the request is enabled using the "S_OutControl" input parameter, the state of the feedback signal on the "EDM1" input parameter (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C020	When the request is enabled using the "S_OutControl" input parameter, the state of the feedback signal on the "EDM2" input parameter (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C030	When the request is enabled using the "S_OutControl" input parameter, the state of the feedback signals on the "EDM1" and "EDM2" input parameters (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C031	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	After the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM1" (depending on the configuration with the "ValveType" input parameter) is not correct. After a valid state of the feedback signals has been detected, the state of "EDM1" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.

Table 546: Diagnostic codes

Code (hex)	Description	Corrective measures
C041	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C050	After the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct. After a valid state of the feedback signals has been detected, the state of "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C051	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C060	After the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM1" and "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct. After a valid state of the feedback signals has been detected, the state of "EDM1" and "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C061	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C070	After the request has been enabled and the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM1" (depending on the configuration with the "ValveType" input parameter) is not correct or the request is no longer present ("S_OutControl" = FALSE). After the request has been enabled and a valid state of the feedback signals has been detected, the state of "EDM1" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C071	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C080	After the request has been enabled and the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct or the request is no longer present ("S_OutControl" = FALSE). After the request has been enabled and a valid state of the feedback signals has been detected, the state of "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C081	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C090	After the request has been enabled and the monitoring time ("MonitoringTime" input parameter) for the feedback signals on the "EDM1" and "EDM2" input parameters has expired, the state of "EDM1" and "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct or the request is no longer present ("S_OutControl" = FALSE). After the request has been enabled and a valid state of the feedback signals has been detected, the state of "EDM1" and "EDM2" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM1" and "EDM2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C091	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C100	A rising edge was detected on the "S_OutControl" and "Reset" input parameters at the same time. There is a danger of a programming error.	• The same signals are not permitted to be used to control the "S_OutControl" and "Reset" input parameters. • Check the wiring of the control device or the safe device connected to it.
C410	After the request has been enabled via "S_OutControl", an incorrect signal was detected on the "EDM1" input parameter. The signal is now correct again, and a request is not being enabled via "S_OutControl".	Reset the function block.
C420	After the request has been enabled via "S_OutControl", an incorrect signal was detected on the "EDM2" input parameter. The feedback signals are now correct again, and a request is not being enabled via "S_OutControl".	Reset the function block.
C430	After the request has been enabled via "S_OutControl", an incorrect signal was detected on the "EDM1" and "EDM2" input parameters. The feedback signals are now correct again, and a request is not being enabled via "S_OutControl".	Reset the function block.

Table 546: Diagnostic codes

Code (hex)	Description	Corrective measures
C440	After the monitoring time has expired and an incorrect signal was detected on the "EDM1" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After an incorrect signal was detected on the "EDM1" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C450	After the monitoring time has expired and an incorrect signal was detected on the "EDM2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After an incorrect signal was detected on the "EDM2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C460	After the monitoring time has expired and an incorrect signal was detected on the "EDM1" and "EDM2" input parameters, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After an incorrect signal was detected on the "EDM1" and "EDM2" input parameters, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C470	After enabling the request, the monitoring time has expired and an incorrect signal was detected on the "EDM1" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After enabling the request and an incorrect signal was detected on the "EDM1" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C480	After enabling the request, the monitoring time has expired and an incorrect signal was detected on the "EDM2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After enabling the request and an incorrect signal was detected on the "EDM2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C490	After enabling the request, the monitoring time has expired and an incorrect signal was detected on the "EDM1" and "EDM2" input parameters, the feedback signals are correct again and a request is not being enabled via "S_OutControl". After enabling the request and an incorrect signal was detected on the "EDM1" and "EDM2" input parameters, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.

Table 546: Diagnostic codes

6.5.7.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram

"ValveType" = TRUE
"S_StartReset" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

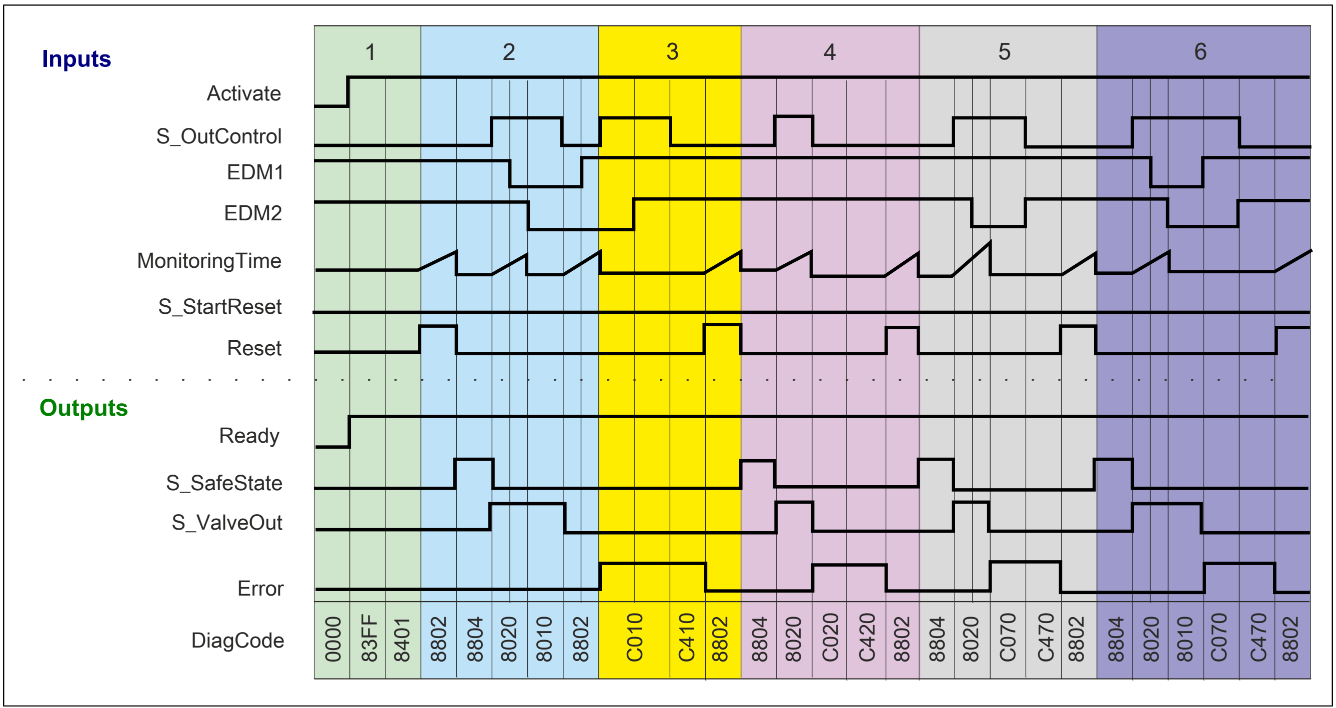


Figure 375: "SF_DoubleValveMonitoring": Signal sequence diagram

1 Initialization
2 Normal operation
3 Error - Request enabled but feedback signal on "EDM2" input parameter not correct
4 Error - Request revoked before feedback signals changed their state
5 Error - Feedback signal on "EDM1" input parameter didn't change its state within the defined time window
6 Error - Request enabled but feedback signal on "EDM1" input parameter not correct

6.5.7.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN ISO 13849-1	5.2.1 6.2
EN ISO 12100	6.2.11.4 6.2.11.6
EN ISO 4413	5.4.7.1 5.4.8.1
EN ISO 4414	5.4.6.1
EN 692	5.4.1.4 5.4.1.6 5.4.2.3

Table 547: "SF_DoubleValveMonitoring": Implementing requirements from standards

6.5.8 SF_FootSwitch

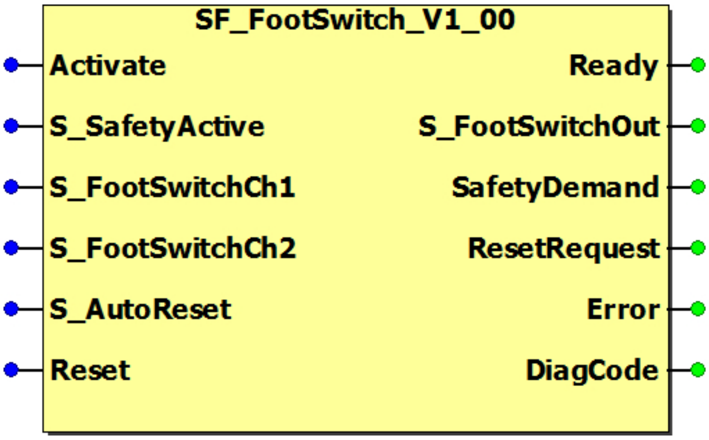


Figure 376: "SF_FootSwitch" function block

6.5.8.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_SafetyActive	SAFEBOOL	Variable/ Constant	Status	FALSE	Control signal indicating that the safe operating mode is active
S_FootSwitchCh1	SAFEBOOL	Variable	Status	FALSE	Input for resulting signal of contacts E1 and E2 of the connected footswitch
S_FootSwitchCh2	SAFEBOOL	Variable	Status	FALSE	Input for resulting signal of contacts E3 and E4 of the connected footswitch
S_AutoReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock if proper signals are present on the input parameters
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 548: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_FootSwitchOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 549: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.8.2 Function

The "SF_FootSwitch" function block supports the lifting of technical protective measures (in accordance with EN 60204) using a footswitch if the corresponding operating mode (e.g. limitation of the speed or power of motion, or the limitation of the range of motion) is selected and active.

> **Information:**
>
> **Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.**

> **Information:**
>
> **The corresponding operating mode must be selected outside of the function block.**

6.5.8.2.1 Footswitch requirements

The footswitch being used must support the following signal level for its switching stage:

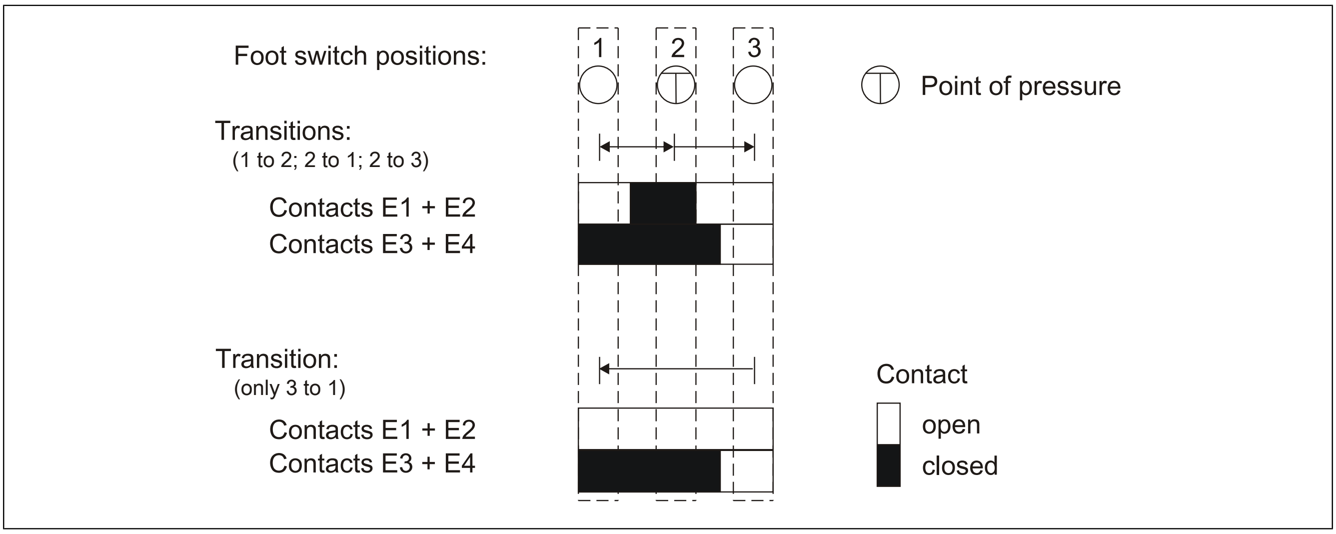


Figure 377: Footswitch: Switching elements and contact travel diagrams

Your footswitch must correspond to the requirements set forth in EN 60204 and the category/SIL required by your application.

6.5.8.2.2 Linking conditions

The resulting signal from normally open contacts E1 and E2 must be connected to the "S_FootSwitchCh1" input parameter. The resulting signal from positively driven normally closed contacts E3 and E4 must be connected to the "S_FootSwitchCh2" input parameter.

If the function block detects a selected safe operating mode on the "S_SafetyActive" input parameter, then the "S_FootSwitchCh1" and "S_FootSwitchCh2" input parameters must indicate the signals for switching stage 1 (footswitch not actuated).

The function block can detect the switching state of the footswitch and its switching direction using the defined order of signals from the contacts (switching stage 1 → switching stage 2 / switching stage 3 → switching stage 2).

> **Information:**
>
> **The function block's enable signal is only set to TRUE after switching from switching stage 1 to switching stage 2.**
>
> **For other switching directions or switching stages, the enable signal is set to FALSE.**

6.5.8.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters.

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state. The start interlock only becomes inactive after manual intervention (rising edge) on the "Reset" input parameter if the input signal combination is permitted for this.

Danger!

Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.

6.5.8.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

Danger!

Always validate the entire safety function!

6.5.8.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

Danger!

You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!
Check the connections when validating the sub-application!

6.5.8.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.8.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.8.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.8.4 Input parameters

6.5.8.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.8.4.2 S_SafetyActive

General function

- Control signal indicating that the safe operating mode is active

Data type

- SAFEBOOL

Connection

- Variable or constant

You will define requirements and/or corrective measures based on the risk analysis you have performed.

> ## Information:
> **Connect this input parameter with a signal that reports the selected safe operating mode.**
>
> **The "S_SafetyActive" output signal of the "SF_SafetyRequest" function block and/or the signals of connected safe peripheral devices are suitable for connection depending on requirements.**

> ## Information:
> **If the safe operating mode is safely implemented without confirmation in your application in accordance with your risk analysis, connect a static TRUE signal on the "S_SafetyActive" input parameter.**

Function description

This input parameter serves as a control signal to indicate that the safe operating mode is active.

> ## Information:
> **The corresponding operating mode must be selected outside of the function block.**

TRUE

The selected safe operating mode is active. The function block sets the "S_FootSwitchOut" output parameter to TRUE if the signal combination for switching stage 2 is present on the "S_FootSwitchCh1" and "S_FootSwitchCh2" input parameters.

FALSE

A safe operating mode is not active. The function block sets the "S_FootSwitchOut" output parameter to the safe state (FALSE) and keeps it in this state.

6.5.8.4.3 S_FootSwitchCh1

General function

- Input for resulting signal of contacts E1 and E2 of the connected footswitch

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_FootSwitchCh1" input parameter is processed by the function block for switching stages 1, 2 and 3.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected contacts are closed.

FALSE

The connected contacts are open.

6.5.8.4.4 S_FootSwitchCh2

General function

- Input for resulting signal of contacts E3 and E4 of the connected footswitch

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_FootSwitchCh2" input parameter is processed by the function block for switching stages 1, 2 and 3.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected contacts are closed.

FALSE

The connected contacts are open.

6.5.8.4.5 S_AutoReset

General function

- Defines the start interlock if proper signals are present on the input parameters

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the operating behavior of the function block after proper signals have returned to the input parameters.

TRUE

After proper signals return to the input parameters, the function block does not support start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After proper signals return to the input parameters, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.5.8.4.6 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.8.5 Output parameters

6.5.8.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.8.5.2 S_FootSwitchOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the switching stage.

The enable signal can be used for subsequent process control.

TRUE

The process to be controlled is enabled.

The following conditions must be met for this:

- The function block has been enabled ("Activate" = TRUE).
- The footswitch is actuated ("S_FootSwitchCh1" and "S_FootSwitchCh2" indicate the signal level for switching stage 2).
- No start interlock is active.
- The function block did not detect any errors.

Danger!

A TRUE signal on the "S_FootSwitchOut" output parameter is not permitted to allow or initiate a dangerous state on its own. For this, a conscious startup command independent of the footswitch is required.

FALSE

The process to be controlled is not enabled.

This may be due to one of the following reasons:

- The function block has not been enabled ("Activate" = FALSE).
- A non-actuated footswitch has been detected by the function block ("S_FootSwitchCh1" and "S_FootSwitchCh2" indicate the signal level for switching stage 1).
- The signal levels of switching stage 3 of the footswitch were detected on "S_FootSwitchCh1" and "S_FootSwitchCh2" by the function block.
- A FALSE signal is present on "S_SafetyActive".
- A start interlock is active.
- The function block detected an error.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safe application and an operational stop from the standard application.

The "S_FootSwitchOut" enable output is set to TRUE after an error only if the "S_SafetyActive" input has the state TRUE and a reset has been carried out (start interlock not active).

6.5.8.5.3 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.8.5.4 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.8.5.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.8.5.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.8.5.7 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_FootSwitchOut" = TRUE	No corrective measures are necessary if the signal is intended.
8004	No safe operating mode is active - Input parameter "S_SafetyActive" = FALSE. • "S_FootSwitchOut" = FALSE	Control the "S_SafetyActive" input parameter with a signal that reports back the safe state (e.g. safe reduced speed) of the monitored range with a TRUE signal. No corrective measures are necessary if the signal is intended.
8005	The safe state (e.g. safe reduced speed) is reported using the signal connected to the "S_SafetyActive" input parameter. • "S_FootSwitchOut" = FALSE	No corrective measures are required.
8006	The connected footswitch is in switching stage 1. • "S_FootSwitchCh1" = FALSE • "S_FootSwitchCh2" = TRUE • "S_FootSwitchOut" = FALSE	No corrective measures are necessary if the signal is intended.
8810	The connected footswitch is in switching stage 3. • "S_FootSwitchCh1" = FALSE • "S_FootSwitchCh2" = FALSE • "S_FootSwitchOut" = FALSE	No corrective measures are necessary if the signal is intended.
C120	The function block detected that the connected footswitch is not in switching stage 1.	• Set "S_SafetyActive" to FALSE. • Put the footswitch in switching stage 1. • Check the wiring of the control device or the safe device connected to it.
C140	The function block detected that the connected footswitch is in switching stage 1.	• Set "S_SafetyActive" to FALSE. • Reset the function block. • Check the wiring of the control device or the safe device connected to it.
C141	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C160	The function block detected that the connected footswitch was in switching stage 3 earlier and is now in switching stage 2.	• Set "S_SafetyActive" to FALSE. • Take the footswitch out of switching stage 2. • Check the wiring of the control device or the safe device connected to it.
C180	The function block detected that the connected footswitch is not in switching stage 2.	• Set "S_SafetyActive" to FALSE. • Reset the function block. • Check the wiring of the control device or the safe device connected to it.
C181	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 550: Diagnostic codes

6.5.8.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_AutoReset" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

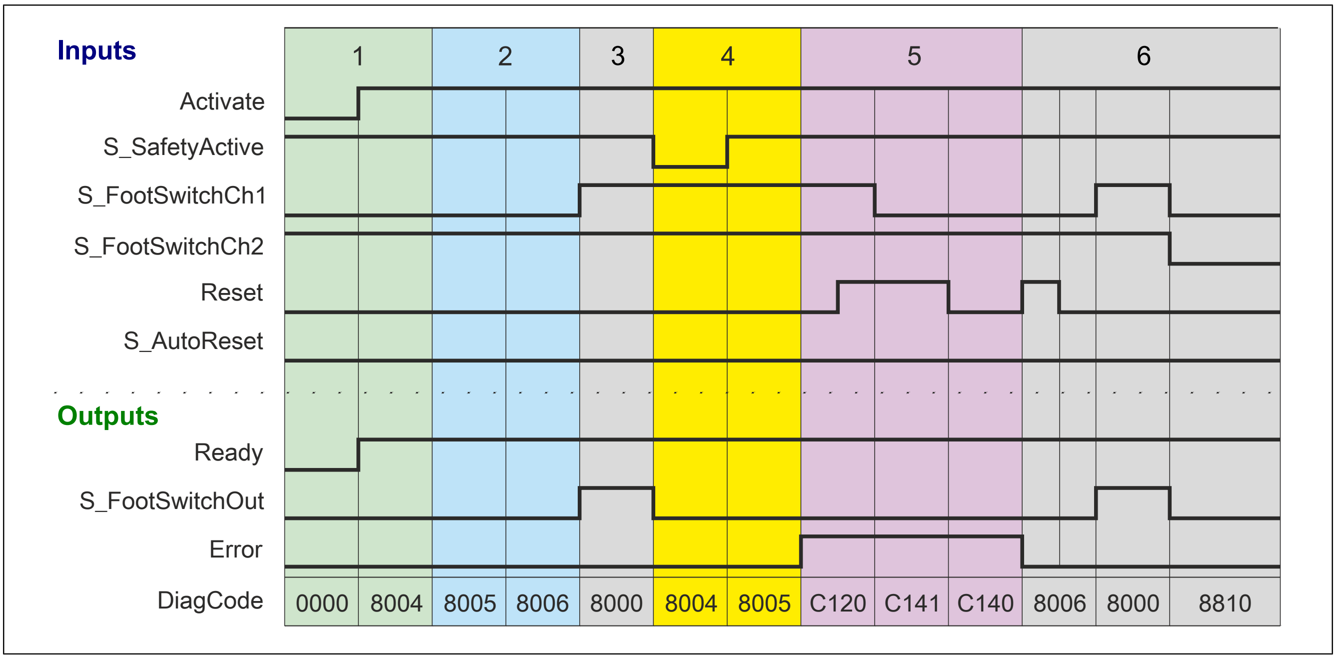


Figure 378: "SF_FootSwitch": Signal sequence diagram 1

1 Initialization
2 Footswitch in position 1
3 Footswitch in position 2
4 Control signal for safe operating mode not present on "S_SafetyActive" input parameter
5 Error - Footswitch not in position 1
6 Normal operation

Signal sequence diagram 2

"S_AutoReset" = TRUE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

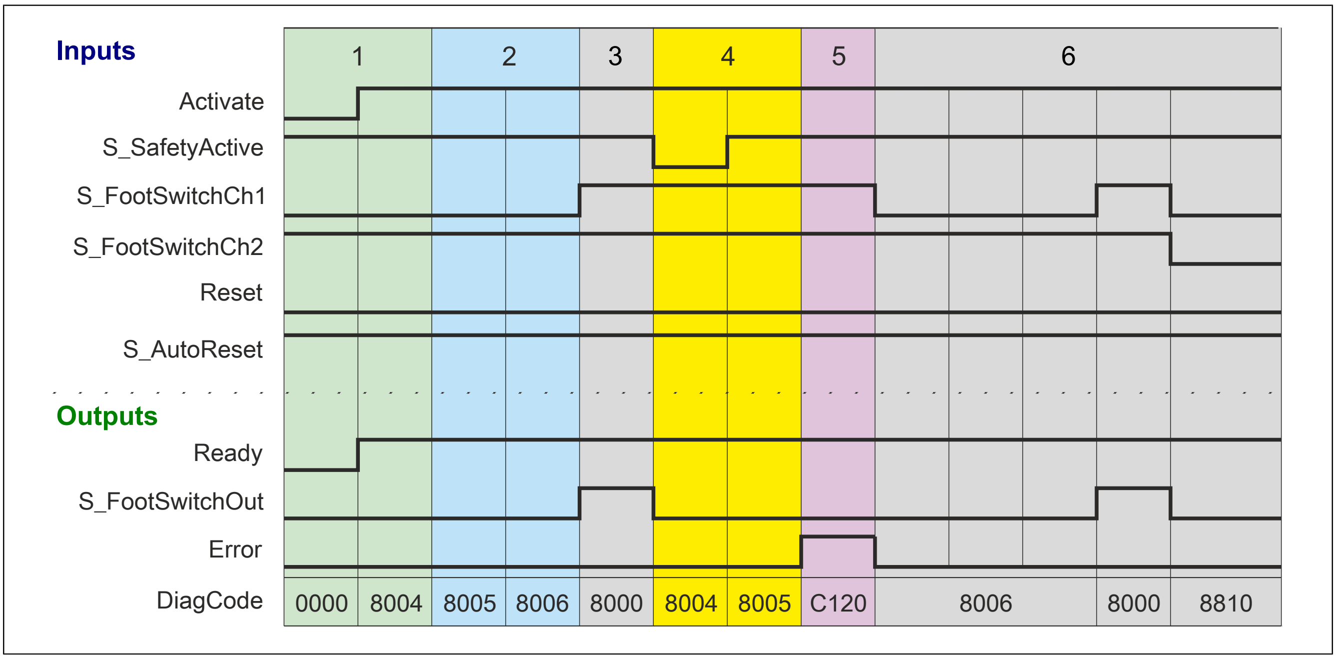


Figure 379: "SF_FootSwitch": Signal sequence diagram 2

1 Initialization
2 Footswitch in position 1
3 Footswitch in position 2
4 Control signal for safe operating mode not present on "S_SafetyActive" input parameter
5 Error - Footswitch not in position 1
6 Normal operation

6.5.8.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN IEC 60204-1	9.2.6.3 10.6 10.9
EN 692	5.4.8.1

Table 551: "SF_FootSwitch": Implementing requirements from standards

6.5.9 SF_PressControl

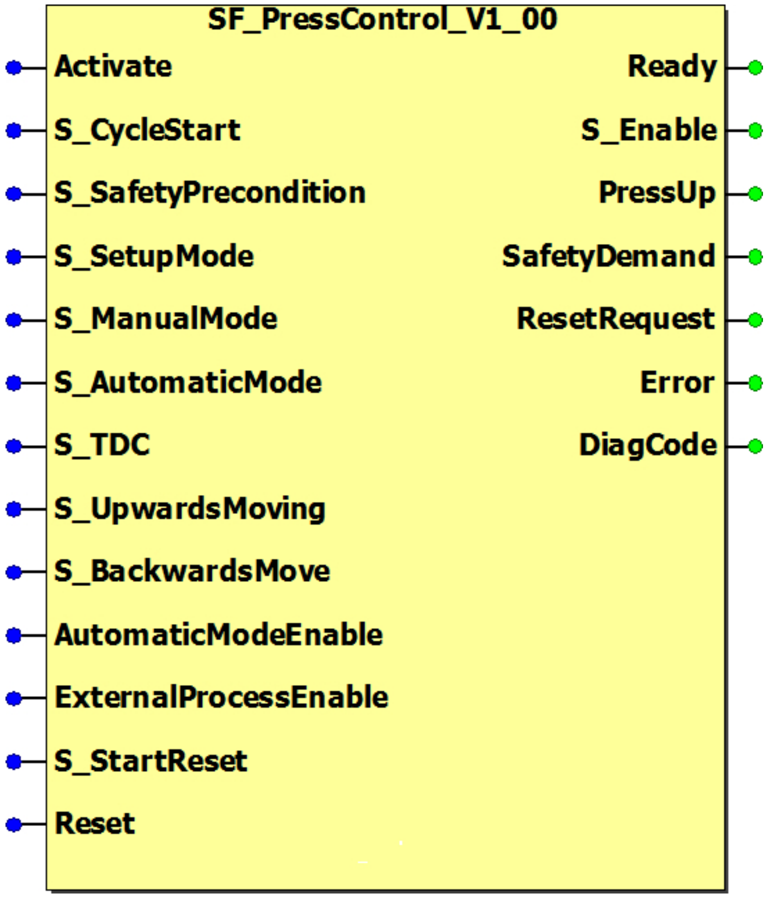


Figure 380: "SF_PressControl" function block

6.5.9.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_CycleStart	SAFEBOOL	Variable	Status	FALSE	Control signal for starting the press
S_SafetyPrecondition	SAFEBOOL	Variable	Status	FALSE	Input for enabling upstream safety functions
S_SetupMode	SAFEBOOL	Variable	Status	FALSE	Input for "Setup" mode
S_ManualMode	SAFEBOOL	Variable	Status	FALSE	Input for "Manual" mode
S_AutomaticMode	SAFEBOOL	Variable	Status	FALSE	Input for "Automatic" mode
S_TDC	SAFEBOOL	Variable	Status	FALSE	Input for position at top dead center (TDC)
S_UpwardsMoving	SAFEBOOL	Variable	Status	FALSE	Input for active upward movement of the press
S_BackwardsMove	SAFEBOOL	Variable	Status	FALSE	Input for backward movement of the press
AutomaticModeEnable	BOOL	Variable	Status	FALSE	Control signal from standard controller for "Automatic" mode
ExternalProcessEnable	BOOL	Variable/ Constant	Status	FALSE	Control signal from standard controller for enabling the process
S_StartReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 552: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_Enable	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
PressUp	BOOL	Variable	Status	FALSE	Indicates the upward movement of the press
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 553: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.9.2 Function

The "SF_PressControl" function block supports the control of the press with three operating modes. In doing so, status information from other function blocks is also processed.

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

Danger!

In all three operating modes, if the "S_SafetyPrecondition" input parameter is set to FALSE, the enable signal is immediately set to FALSE and the cycle is stopped.

If the signal on the "S_SafetyPrecondition" input parameter is then set to TRUE, acknowledgment on "Reset" is not required to start a new cycle.

If acknowledgment or restart interlock is required, then this must be accomplished via the upstream safety functions in order to form the signal for the "S_SafetyPrecondition" input parameter.

6.5.9.2.1 "Manual" mode

To start, the press must be at top dead center ("S_TDC" input parameter). Both the upstream safety functions ("S_SafetyPrecondition" input parameter) as well as the enable from the standard controller ("ExternalProcessEnable" input parameter) must return a TRUE signal. The cycle starts on a rising edge of the "S_CycleStart" input parameter. The enable signal ("S_Enable" output parameter) is set to TRUE.

Information:

The single-stroke function is active. In order to start a new cycle, a new starting signal is required on the "S_CycleStart" input parameter.

The signal on the "S_CycleStart" input parameter must be set to TRUE until a rising edge of the "S_UpwardsMoving" input parameter takes place. If this is not the case, the enable signal ("S_Enable" output parameter) is set to FALSE.

Information:

If the signal of the "S_CycleStart" input parameter switches to FALSE during the downward movement, then the downward movement can be immediately resumed with another rising edge on the "S_CycleStart" input parameter.

If the cycle is completed and top dead center ("S_TDC" input parameter) reached once more, then the enable signal is set to FALSE.

Information:

Cyclic overrun monitoring ensures that it is only possible to restart the cycle from top dead center after a rising edge of the "S_TDC" input parameter. If the "S_TDC" input parameter switches to FALSE, this error can only be acknowledged in "Setup" mode.

6.5.9.2.2 "Setup" mode (clockwise)

The press can be started from any position. Both the upstream safety functions ("S_SafetyPrecondition" input parameter) as well as the enable from the standard controller ("ExternalProcessEnable" input parameter) must return a TRUE signal. The cycle starts if a TRUE signal is present on the "S_CycleStart" input parameter. The enable signal ("S_Enable" output parameter) is set to TRUE.

> **Information:**
>
> **The "S_CycleStart" input parameter must be set to TRUE for the entire cycle.**
>
> **If a rising edge of the "S_TDC" input parameter is detected, the enable signal is set to FALSE.**
>
> **An error in overrun monitoring is detected and indicated by a corresponding state of the function block (see "Diagnostic codes"). Acknowledgment is not necessary.**

6.5.9.2.3 "Setup" mode (counterclockwise)

To start, you first have to enable the backward movement with the "S_BackwardsMove" input parameter.

The press can be started from any position. Both the upstream safety functions ("S_SafetyPrecondition" input parameter) as well as the enable from the standard controller ("ExternalProcessEnable" input parameter) must return a TRUE signal.

The cycle starts if a TRUE signal is present on the "S_CycleStart" input parameter. The enable signal ("S_Enable" output parameter) is set to TRUE.

> **Information:**
>
> **If a rising edge of the "S_TDC" input parameter is detected, the enable signal is set to FALSE.**

6.5.9.2.4 "Automatic" mode

The press can be started from any position. Both the upstream safety functions ("S_SafetyPrecondition" input parameter) as well as the enable from the standard controller ("ExternalProcessEnable" input parameter) must return a TRUE signal. The cycle starts on a rising edge of the "S_CycleStart" input parameter. The enable signal ("S_Enable" output parameter) is set to TRUE.

> **Information:**
>
> **The single-stroke function is not active. In order to start a new cycle, a new starting signal is not required on the "S_CycleStart" input parameter.**

"Automatic" mode can be stopped with a FALSE signal either on the "S_SafetyPrecondition" input parameter or the "AutomaticModeEnable" input parameter.

> **Information:**
>
> **If "Automatic" mode is stopped by a FALSE signal on the "AutomaticModeEnable" input parameter, then the enable signal is set to TRUE until top dead center (TDC) is reached.**

6.5.9.2.5 Start interlock after cold restart (optional)

Support for a start interlock must be defined accordingly on the input parameter after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the enable signal as needed.

6.5.9.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.9.3.1 Restart interlock combined with "S_SafetyPrecondition"

The function block does not support restart interlock for the signal on the "S_SafetyPrecondition" input parameter.

> **Danger!**
>
> **If acknowledgment or restart interlock is required, then this must be accomplished via the upstream safety function in order to form the signal for the "S_SafetyPrecondition" input parameter.**

6.5.9.3.2 More than one active operating mode

The enabled function block detected more than one active operating mode from the input parameters for operating modes.

Possible cause:

- The same or incorrect variables are connected to input parameters in the application (programming error, user error).

6.5.9.3.3 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.9.3.4 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.9.3.5 Invalid static signals when cold restarting the safety controller

A static TRUE signal during a cold restart of the safety controller causes an error message on the function block if the start interlock is defined for after the function block is enabled.

If this start interlock is not defined for a cold restart of the safety controller, then the status of "Reset" is irrelevant.

6.5.9.3.6 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.9.3.7 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.9.4 Input parameters

6.5.9.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.9.4.2 S_CycleStart

General function

- Control signal for starting the press

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter serves as the control signal for starting the cycle depending on the operating mode.

"Manual" mode

The "S_CycleStart" input parameter must return a TRUE signal until the upward movement ("S_UpwardsMoving") is initiated in order to set the enable signal to TRUE. A FALSE signal during the downward movement stops the cycle.

"Setup" mode

The "S_CycleStart" input parameter must return a TRUE signal over the entire cycle in order to set the enable signal to TRUE. A FALSE signal stops the cycle.

"Automatic" mode

A rising edge of the "S_CycleStart" input parameter starts the cycle and sets the enable signal to TRUE. A FALSE signal does not stop the cycle.

> **Information:**
>
> **"Automatic" mode is stopped by a FALSE signal on the "AutomaticModeEnable" input parameter.**

6.5.9.4.3 S_SafetyPrecondition

General function

- Input for enabling upstream safety functions

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **This signal is used to notify the function block of the state of the upstream safety functions (e.g. "SF_EmgergencyStop", "SF_GuardMonitoring", etc.).**

Function description

The signal connected to the "S_SafetyPrecondition" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The safety-related preconditions are satisfied.

FALSE

The safety-related preconditions are not satisfied.

6.5.9.4.4 S_SetupMode

General function

- Input for "Setup" mode

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **You should connect the selection of the "Setup" mode from the "SF_ModeSelector" function block here.**

Function description

The signal connected to the "S_SetupMode" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **Only one of the three operating modes is permitted to be active at a time. If this is not the case, the function block transitions to an error state.**

TRUE

"Setup" mode is active.

FALSE

"Setup" mode is not active.

6.5.9.4.5 S_ManualMode

General function

- Input for "Manual" mode

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **You should connect the selection of the "Manual" mode from the "SF_ModeSelector" function block here.**

Function description

The signal connected to the "S_ManualMode" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **Only one of the three operating modes is permitted to be active at a time. If this is not the case, the function block transitions to an error state.**

TRUE

"Manual" mode is active.

FALSE

"Manual" mode is not active.

6.5.9.4.6 S_AutomaticMode

General function

- Input for "Automatic" mode

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **You should connect the selection of the "Automatic" mode from the "SF_ModeSelector" function block here.**

Function description

The signal connected to the "S_AutomaticMode" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> ## Information:
> **Only one of the three operating modes is permitted to be active at a time. If this is not the case, the function block transitions to an error state.**

TRUE

"Automatic" mode is active.

FALSE

"Automatic" mode is not active.

6.5.9.4.7 S_TDC

General function

- Input for position at top dead center (TDC)

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **The output signal "S_TDC" from the "SF_CamMonitoring" function block must be used for the connection.**

Function description

The signal connected to the "S_TDC" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The press is at the starting position – at TDC.

FALSE

The press is not at the starting position – not at TDC.

6.5.9.4.8 S_UpwardsMoving

General function

- Input for active upward movement of the press

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **The output signal "S_UpwardsMoving" from the "SF_CamMonitoring" function block must be used for the connection.**

Function description

The signal connected to the "S_UpwardsMoving" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The press is performing an upward movement.

FALSE

The press is not performing an upward movement.

6.5.9.4.9 S_BackwardsMove

General function

- Input for backward movement of the press

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_BackwardsMove" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **The function for the backward movement can only be used in "Setup" mode.**

TRUE

Moves the press backwards (counterclockwise)

FALSE

Moves the press forwards (clockwise)

6.5.9.4.10 AutomaticModeEnable

General function

- Control signal from standard controller for "Automatic" mode

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to a control signal from the standard controller.

Function description

This input parameter serves as a control signal from the standard controller for enabling "Automatic" mode or for halting the press the next time it reaches top dead center (TDC).

TRUE

"Automatic" mode is active.

FALSE

"Automatic" mode is not active or the press is stopped at top dead center (TDC).

6.5.9.4.11 ExternalProcessEnable

General function

- Control signal from standard controller for enabling the process

Data type

- BOOL

Connection

- Variable or constant

> **Information:**
>
> **Connect this input parameter to a control signal from the standard controller.**

> **Information:**
>
> **If not using a control signal from the standard controller, connect a static TRUE signal to the "ExternalProcessEnable" input parameter.**

Function description

This input parameter serves as the control signal from the standard controller for enabling the process.

> **Information:**
>
> **If the control signal switches to the FALSE state, then the cycle can only be started by a new rising edge of the "S_CycleStart" input parameter.**

TRUE

An enable is present from the standard controller.

FALSE

An enable is not present from the standard controller.

6.5.9.4.12 S_StartReset

General function

- Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the startup behavior of the function block following its activation and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> **Danger!**
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> **Danger!**
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.5.9.4.13 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.9.5 Output parameters

6.5.9.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.9.5.2 S_Enable

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the cycle and operating mode.

The enable signal can be used for subsequent process control.

> ## Information:
> **Use this output parameter to control the valves.**

TRUE

Enable exists to move the press.

FALSE

Enable does not exist to move the press.

6.5.9.5.3 PressUp

General function

- Indicates the upward movement of the press

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates the upward movement of the press.

TRUE

The press is performing an upward movement.

FALSE

The press is not performing an upward movement.

6.5.9.5.4 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.9.5.5 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.9.5.6 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> ## Danger!
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.9.5.7 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.9.5.8 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8002	The operating mode enabled using the "S_SetupMode", "S_ManualMode" and "S_AutomaticMode" input parameters is being checked.	No corrective measures are required.
8100	The press is in "Manual" mode ("S_ManualMode" = TRUE). The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_Enable" = TRUE • "PressUp" changes from FALSE to TRUE on a rising edge of the "S_UpwardsMoving" input parameter.	No corrective measures are necessary if the signal is intended.
8110	The press is in "Setup" mode ("S_SetupMode" = TRUE). The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_Enable" = TRUE • "PressUp" changes from FALSE to TRUE on a rising edge of the "S_UpwardsMoving" input parameter.	No corrective measures are necessary if the signal is intended.
8120	The press is in "Automatic" mode ("S_AutomaticMode" = TRUE). The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_Enable" = TRUE • "PressUp" changes from FALSE to TRUE on a rising edge of the "S_UpwardsMoving" input parameter.	No corrective measures are necessary if the signal is intended.
8130	The press is in "Setup" mode ("S_SetupMode" = TRUE) performing a backward movement ("S_BackwardMove" = TRUE). The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_Enable" = TRUE • "PressUp" changes from FALSE to TRUE on a rising edge of the "S_UpwardsMoving" input parameter.	No corrective measures are necessary if the signal is intended.
8401	The start interlock to be carried out after enabling the function block is defined ("S_StartReset" = FALSE), or the function block's start interlock is active ("S_AutoReset" = FALSE).	Reset the function block.
8802	The press is in "Manual" mode ("S_ManualMode" = TRUE) and waiting for the start of the cycle via the "S_CycleStart" input parameter. • "S_Enable" = FALSE • "PressUp" = FALSE	"S_CycleStart" must have a rising edge in order to set the safe output to TRUE.
8804	The press is in "Setup" mode ("S_SetupMode" = TRUE) and waiting for the start of the cycle via the "S_CycleStart" input parameter. • "S_Enable" = FALSE • "PressUp" = FALSE	"S_CycleStart" must be set to TRUE in order to set the safe output to TRUE.
8806	The press is in "Automatic" mode ("S_AutomaticMode" = TRUE) and waiting for the start of the cycle via the "S_CycleStart" input parameter. • "S_Enable" = FALSE • "PressUp" = FALSE	"S_CycleStart" must have a rising edge in order to set the safe output to TRUE.
8808	A loss of signal on the "S_SafetyPrecondition" input parameter was detected. • "S_Enable" = FALSE • "PressUp" = FALSE	• Correct the loss of signal on "S_SafetyPrecondition". • Check the wiring of the control device or the safe device connected to it.
8812	The press is in "Manual" mode ("S_ManualMode" = TRUE). The signal on the "S_CycleStart" input parameter was lost before it was detected that the upward movement is enabled with the "S_UpwardsMoving" input parameter. • "S_Enable" = FALSE • "PressUp" = FALSE	"S_CycleStart" must have a rising edge in order to set the safe output to TRUE.
8814	The press is in "Setup" mode ("S_SetupMode" = TRUE) but not at top dead center ("S_TDC" input parameter = FALSE). • "S_Enable" = FALSE • "PressUp" = FALSE	"S_CycleStart" must be set to TRUE in order to set the safe output to TRUE.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	More than one operating mode is enabled via the input parameters.	• Correct the problem by enabling only one operating mode via the input parameters. • Check the wiring of the control device or the safe device connected to it.

Table 554: Diagnostic codes

Code (hex)	Description	Corrective measures
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C020	An invalid signal for top dead center ("S_TDC" input parameter = FALSE) was detected.	Correct the problem by switching to "Setup" mode and performing a reset on the function block.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C410	After several operating modes were active, now only one operating mode is active.	Reset the function block.
C420	"Setup" mode was entered after an invalid signal on the "S_TDC" input parameter was detected.	Reset the function block.

Table 554: Diagnostic codes

6.5.9.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_SetupMode" = TRUE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

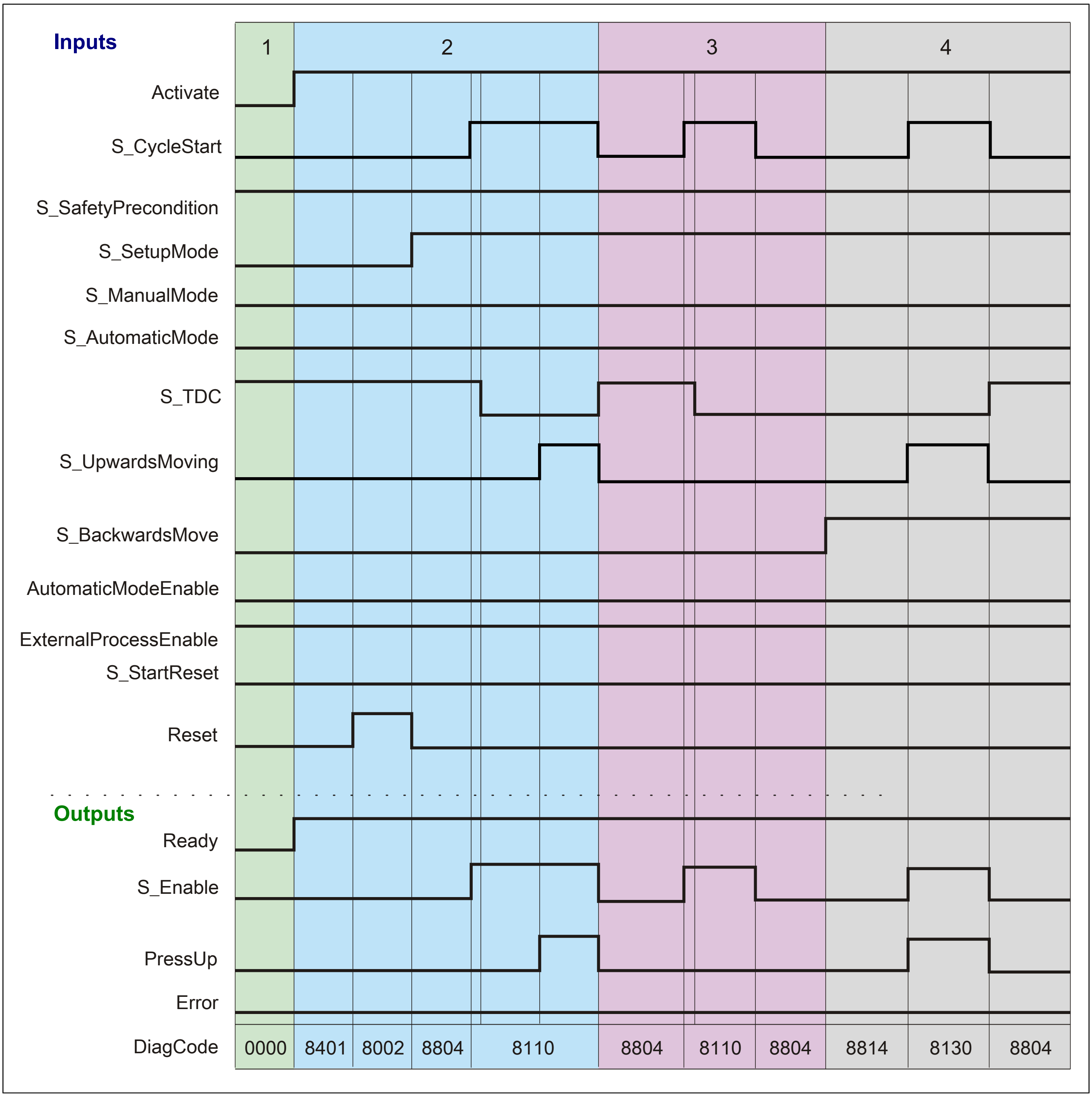


Figure 381: "SF_PressControl": Signal sequence diagram 1

1 Initialization
2 First single stroke
3 Second single stroke
4 Backward movement

Signal sequence diagram 2

"S_ManualMode" = TRUE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

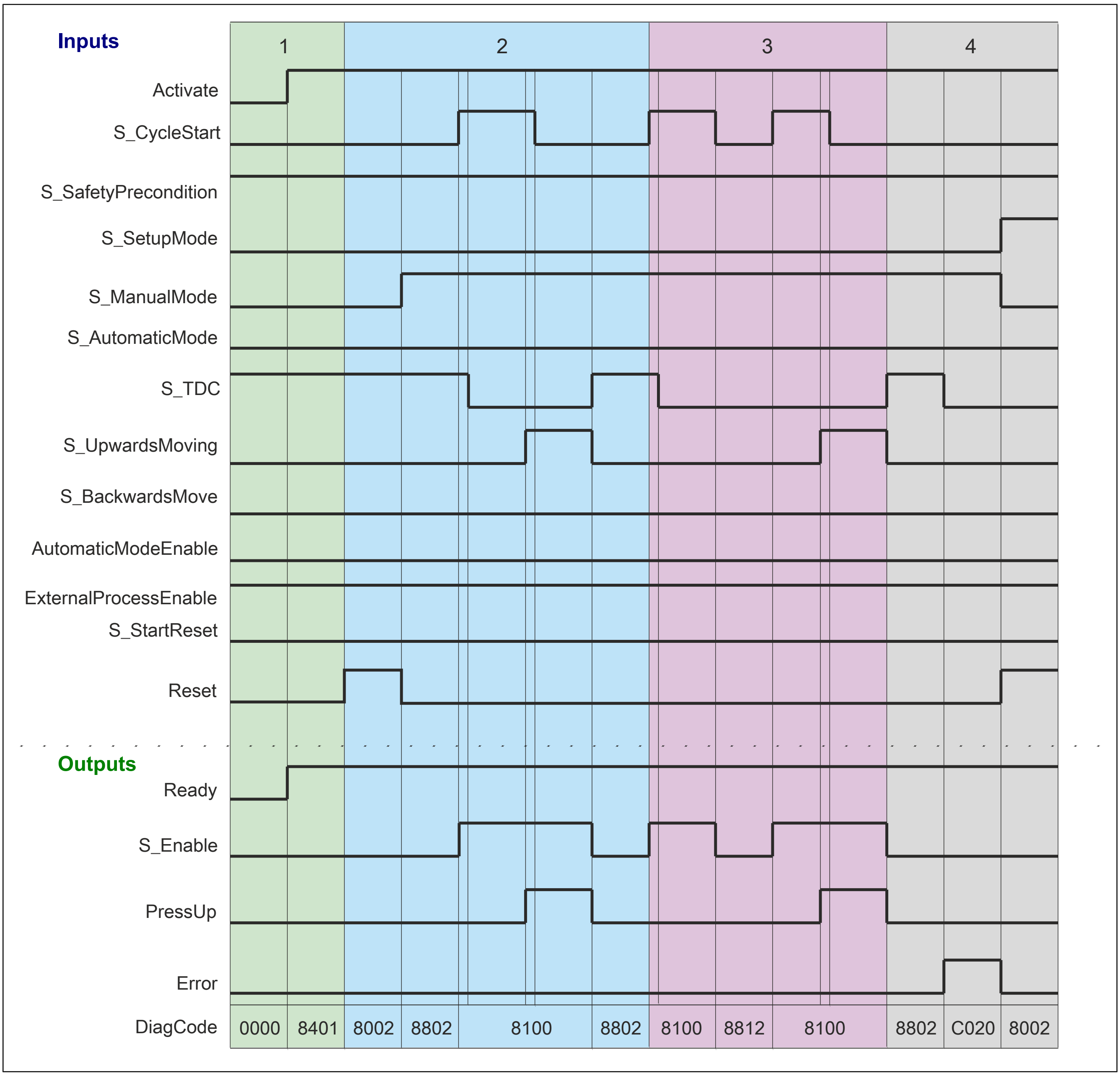


Figure 382: "SF_PressControl": Signal sequence diagram 2

1 Initialization
2 Single stroke
3 Single stroke with stop between top dead center (TDC) and bottom dead center (BDC) and restart from the downward movement
4 Overrun error after last stroke

Signal sequence diagram 3

"S_AutomaticMode" = TRUE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

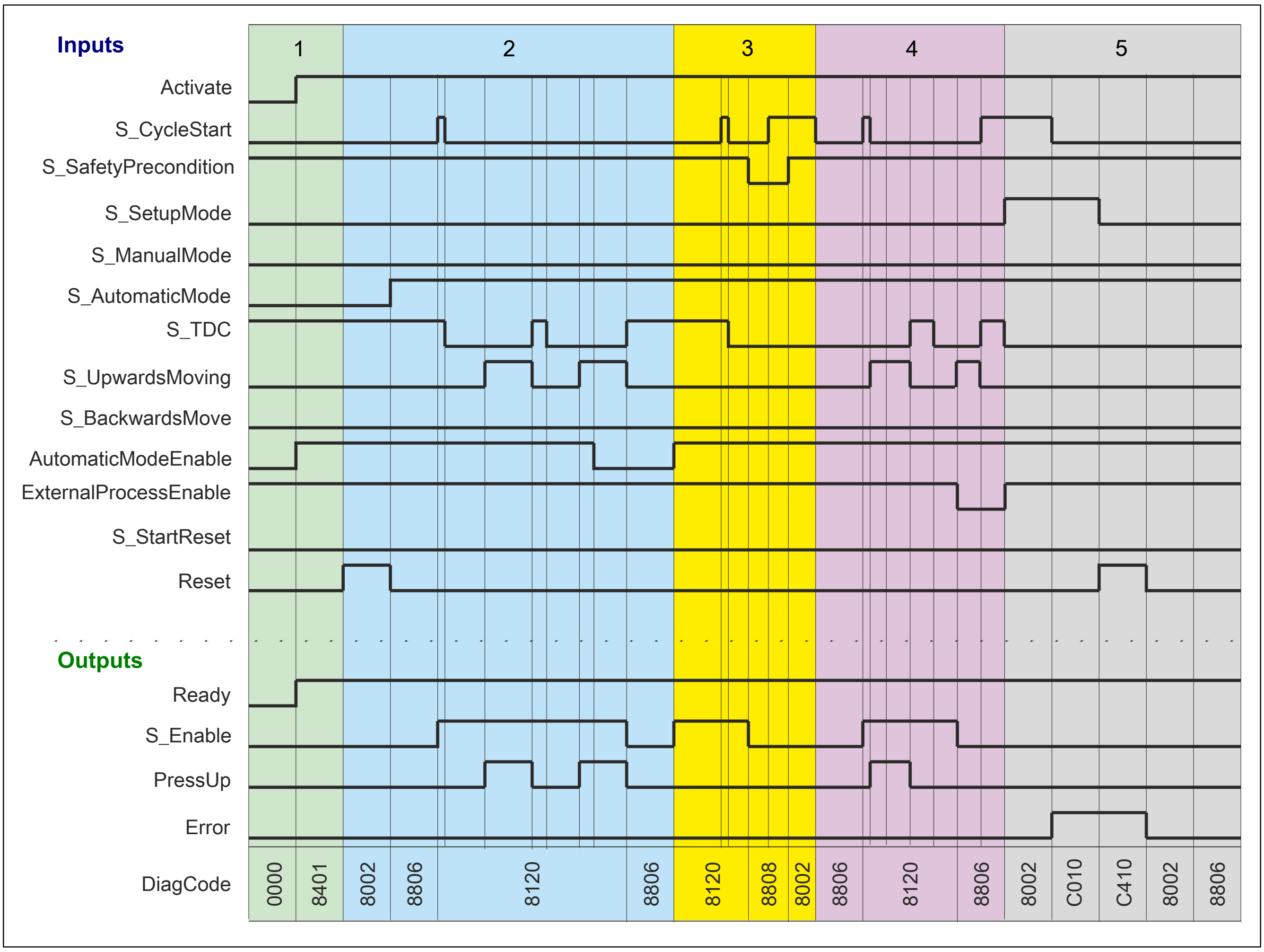


Figure 383: "SF_PressControl": Signal sequence diagram 3

1 Initialization
2 Stops the press
3 Switches the "S_SafetyPrecondition" input parameter to FALSE during stroke
4 Switches the "ExternalProcessEnable" input parameter to FALSE during stroke
5 More than one mode selected

6.5.9.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN 692	3.1.6 3.1.8 3.1.25 5.3.16 5.3.17 5.4.1.2 5.4.2.4
EN 693	3.4 3.19

Table 555: "SF_PressControl": Implementing requirements from standards

6.5.10 SF_SingleValveCycleMonitoring

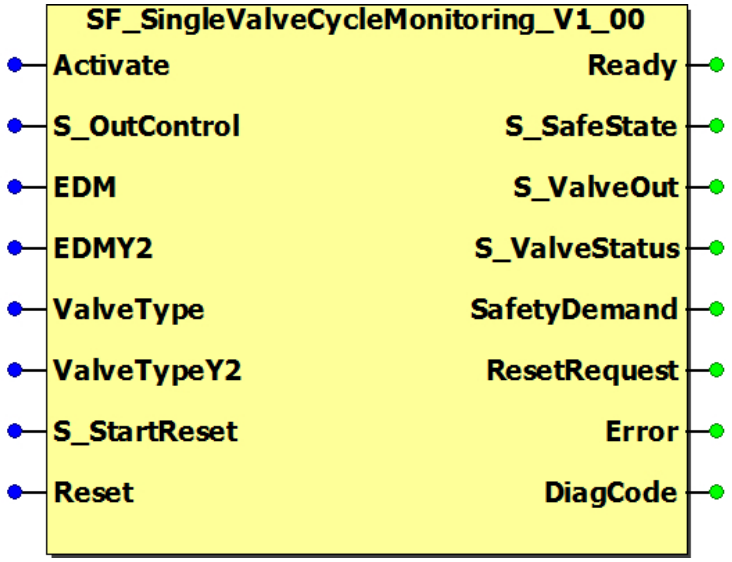


Figure 384: "SF_SingleValveCycleMonitoring" function block

6.5.10.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_OutControl	SAFEBOOL	Variable	Status	FALSE	Control signal for controlling valves
EDM	BOOL	Variable	Status	FALSE	Input for feedback signal from cartridge valve
EDMY2	BOOL	Variable	Status	FALSE	Input for feedback signal from valve Y2
ValveType	BOOL	Constant	Status	FALSE	Specifies the polarity of the feedback signal from the cartridge valve
ValveTypeY2	BOOL	Constant	Status	FALSE	Specifies the polarity of the feedback signal from valve Y2
S_StartReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock after the function block is en-abled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 556: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_SafeState	SAFEBOOL	Variable	Status	FALSE	Indicates the basic state
S_ValveOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_ValveStatus	SAFEWORD	Variable	Status	16#0000	Status output for use with the "SF_ValveGroupControl" function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 557: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.10.2 Function

In each press cycle, the "SF_SingleValveCycleMonitoring" function block checks the defined basic state and dynamic switching state of flow control valves (cartridge valves) that are controlled by safe output devices. If no signal change takes place within a press cycle, then the next cycle is prevented.

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

On a press, these types of valves are used to control the downward stroke (downward movement). Before the downward stroke can be started, the valve must be in the basic state ("EDM" input parameter). Likewise, the Y2 valve for the upward stroke (upward movement) must report being in the basic state ("EDMY2" input parameter). The basic state is defined with the "ValveType" and "ValveTypeY2" input parameters.

The outputs of safe devices that are used to control the cartridge valves to be monitored are energized to do so by the enable signal from this function block. The states of the connected cartridge valves are fed back to the function block via safe input devices.

In the switching state, the function block must detect a signal change in the signals fed back from the valves being monitored. In addition, the "EDMY2" input parameter is monitored and checked for dynamic changes.

Information:

The "S_ValveStatus" output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group.

This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

Information:

The polarity of the feedback signals can be defined using the "ValveType" and "ValveTypeY2" input parameters. This specifies whether the basic state is confirmed by a TRUE or FALSE signal.

6.5.10.2.1 Monitoring the basic state

If the function block detects the basic state ("ValveType" input parameter) of the valves via the feedback signals present on the "EDM" and "EDMY2" input parameters during a switch-on request, then the function block will set its enable signal to TRUE and switch on the valves.

If the basic state is not detected by the function block during a switch-on request, then the function block will set its enable signal to FALSE and will not switch on the valves.

Information:

If the basic state of the valve is detected, then the "S_SafeState" output parameter is set to TRUE.

Leaving the basic state will switch "S_SafeState" to FALSE.

6.5.10.2.2 Monitoring per press cycle

After controlling the valves, the function block checks the reaction of the valves each press cycle using the "EDM" and "EDMY2" input parameters. At the beginning, the state on the "EDM" input parameter has to change. After that, the state of the "EDMY2" input parameter also has to change after "S_OutControl" is no longer being controlled.

If the function block does not detect the states of the feedback signals defined for the switching state (depends on the "ValveType" and "ValveTypeY2" input parameters), then the function block will set its enable signal to FALSE and cut the valves back off again.

6.5.10.2.3 Start interlock after cold restart (optional)

Support for a start interlock must be defined accordingly on the input parameter after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the enable signal as needed.

6.5.10.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.10.3.1 Connecting "ValveType" and "ValveTypeY2"

> **Danger!**
>
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

6.5.10.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.10.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.10.3.4 Invalid static signals when cold restarting the safety controller

A static TRUE signal during a cold restart of the safety controller causes an error message on the function block if the start interlock is defined for after the function block is enabled.

If this start interlock is not defined for a cold restart of the safety controller, then the status of "Reset" is irrelevant.

6.5.10.3.5 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.10.3.6 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.10.4 Input parameters

6.5.10.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.10.4.2 S_OutControl

General function

- Control signal for controlling valves

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **Note that no hardware input is connected on the "S_OutControl" input parameter.**

Function description

This input parameter serves as a control signal for controlling the valves.

> **Information:**
>
> **The "S_OutControl" input parameter should be connected to an enable signal from a safe function block.**
>
> **The status of the "S_OutControl" input parameter corresponds to the status of the upstream function block / safety function.**

The status of the "S_OutControl" input parameter controls the "S_ValveOut" output parameter under consideration of the states of the connected feedback signals.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream safety function was not triggered.

> **Information:**
>
> **The "S_ValveOut" output parameter is set to TRUE if the remaining input signal combination is valid for this.**

FALSE

The safety function has been initiated or the safe device connected for this safety function is switched off or faulty.

> **Information:**
>
> **The "S_ValveOut" output parameter is set to FALSE.**

6.5.10.4.3 EDM

General function

- Input for feedback signal from cartridge valve

Data type

- BOOL

Connection

- Variable

Function description

The signal connected to the "EDM" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

"EDM" = TRUE and "ValveType" = FALSE

The valve is energized and not in the basic state.

"EDM" = TRUE and "ValveType" = TRUE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM" input parameter returns a TRUE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM" = FALSE and "ValveType" = FALSE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM" input parameter returns a FALSE signal, then the "S_SafeState" output parameter is set to FALSE.**

"EDM" = FALSE and "ValveType" = TRUE

The valve is energized and not in the basic state.

6.5.10.4.4 EDMY2

General function

- Input for feedback signal from valve Y2

Data type

- BOOL

Connection

- Variable

Function description

The signal connected to the "EDMY2" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

"EDMY2" = TRUE and "ValveTypeY2" = FALSE

The valve is energized and not in the basic state.

"EDMY2" = TRUE and "ValveTypeY2" = TRUE

The valve is not energized but is in the basic state.

"EDMY2" = FALSE and "ValveTypeY2" = FALSE

The valve is not energized but is in the basic state.

"EDMY2" = FALSE and "ValveTypeY2" = TRUE

The valve is energized and not in the basic state.

6.5.10.4.5 ValveType

General function

- Specifies the polarity of the feedback signal from the cartridge valve

Data type

- BOOL

Connection

- Constant

Function description

This input parameter is used to define the polarity of the feedback signal from the cartridge valve.

> **Danger!**
>
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

TRUE

The feedback signal returns status TRUE in the basic state.

FALSE

The feedback signal returns status FALSE in the basic state.

6.5.10.4.6 ValveTypeY2

General function

- Specifies the polarity of the feedback signal from valve Y2

Data type

- BOOL

Connection

- Constant

Function description

This input parameter is used to define the polarity of the feedback signal from the Y2 valve.

> ## Danger!
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

TRUE

The feedback signal returns status TRUE in the basic state.

FALSE

The feedback signal returns status FALSE in the basic state.

6.5.10.4.7 S_StartReset

General function

- Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the startup behavior of the function block following its activation and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.5.10.4.8 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

> ## Danger!
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.10.5 Output parameters

6.5.10.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.10.5.2 S_SafeState

General function

- Indicates the basic state

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether the valves are in the basic state.

TRUE

The valves are in the basic state.

FALSE

The valves are not in the basic state.

6.5.10.5.3 S_ValveOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is the safe signal used to control an output on a safe device, therefore also controlling the process. The enable signal is controlled depending on the status of the connected valve or valves as well as the start interlocks of the upstream safety functions.

TRUE

The output of a safe device is set to TRUE.

FALSE

The output of a safe device is set to FALSE.

6.5.10.5.4 S_ValveStatus

General function

- Status output for use with the "SF_ValveGroupControl" function block

Data type

- SAFEWORD

Connection

- Variable

Function description

This output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group. This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

6.5.10.5.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.10.5.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.10.5.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.10.5.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.10.5.9 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8010	The feedback signal on the "EDM" input parameter changed its signal state during an enabled request via the "S_OutControl" input parameter. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
8020	A request is present on the "S_OutControl" input parameter for setting the safe output to TRUE. The feedback signal on the "EDMY2" input parameter has a valid signal state while the request is enabled. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
83FF	State for registering on the "SF_ValveGroupControl" function block.	No corrective measures are required.
8401	The start interlock to be carried out after enabling the function block is defined ("S_StartReset" = FALSE), or the function block's start interlock is active ("S_AutoReset" = FALSE).	Reset the function block.
8802	Waiting for feedback signal on the "EDM" input parameter if no request is present via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = FALSE	No corrective measures are necessary if the signal is intended.
8804	No request is present on the "S_OutControl" input parameter for setting the safe output to TRUE. • "S_SafeState" = TRUE • "S_ValveOut" = FALSE	"S_OutControl" must have status TRUE in order to set the safe output to TRUE.
8806	Waiting for a rising edge on the feedback signal of the "EDMY2" input parameter if no request is present via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = FALSE	No corrective measures are necessary if the signal is intended.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	When the request is enabled using the "S_OutControl" input parameter, the state of the feedback signal on the "EDM" input parameter (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" and "EDMY2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	After the request has been enabled and a valid state of the feedback signals has been detected, the state of "EDMY2" (depending on the configuration with the "ValveTypeY2" input parameter) is not correct or the request is no longer present ("S_OutControl" = FALSE).	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" and "EDMY2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C041	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C0C0	After the request has been enabled and a valid state of the feedback signals has been detected, the state of "EDMY2" (depending on the configuration with the "ValveTypeY2" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" and "EDMY2" feedback signals (depending on the configuration) return the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C0C1	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C0D0	After the request was revoked, enabling of the request was detected.	You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" and "EDMY2" feedback signals (depending on the configuration) return the correct value in a de-energized state.

Table 558: Diagnostic codes

Code (hex)	Description	Corrective measures
C0D1	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C100	A rising edge was detected on the "S_OutControl" and "Reset" input parameters at the same time. There is a danger of a programming error.	• The same signals are not permitted to be used to control the "S_OutControl" and "Reset" input parameters. • Check the wiring of the control device or the safe device connected to it.
C410	After enabling the request and an incorrect signal was detected on the "EDM" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C440	After enabling the request and an incorrect signal was detected on the "EDMY2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C4C0	After enabling the request and an incorrect signal was detected on the "EDMY2" input parameter, the feedback signals are correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C4D0	After the request is re-enabled, the feedback signals are correct and a request is not being enabled via "S_OutControl".	Reset the function block.

Table 558: Diagnostic codes

6.5.10.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram

"ValveType" = TRUE
"S_StartReset" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

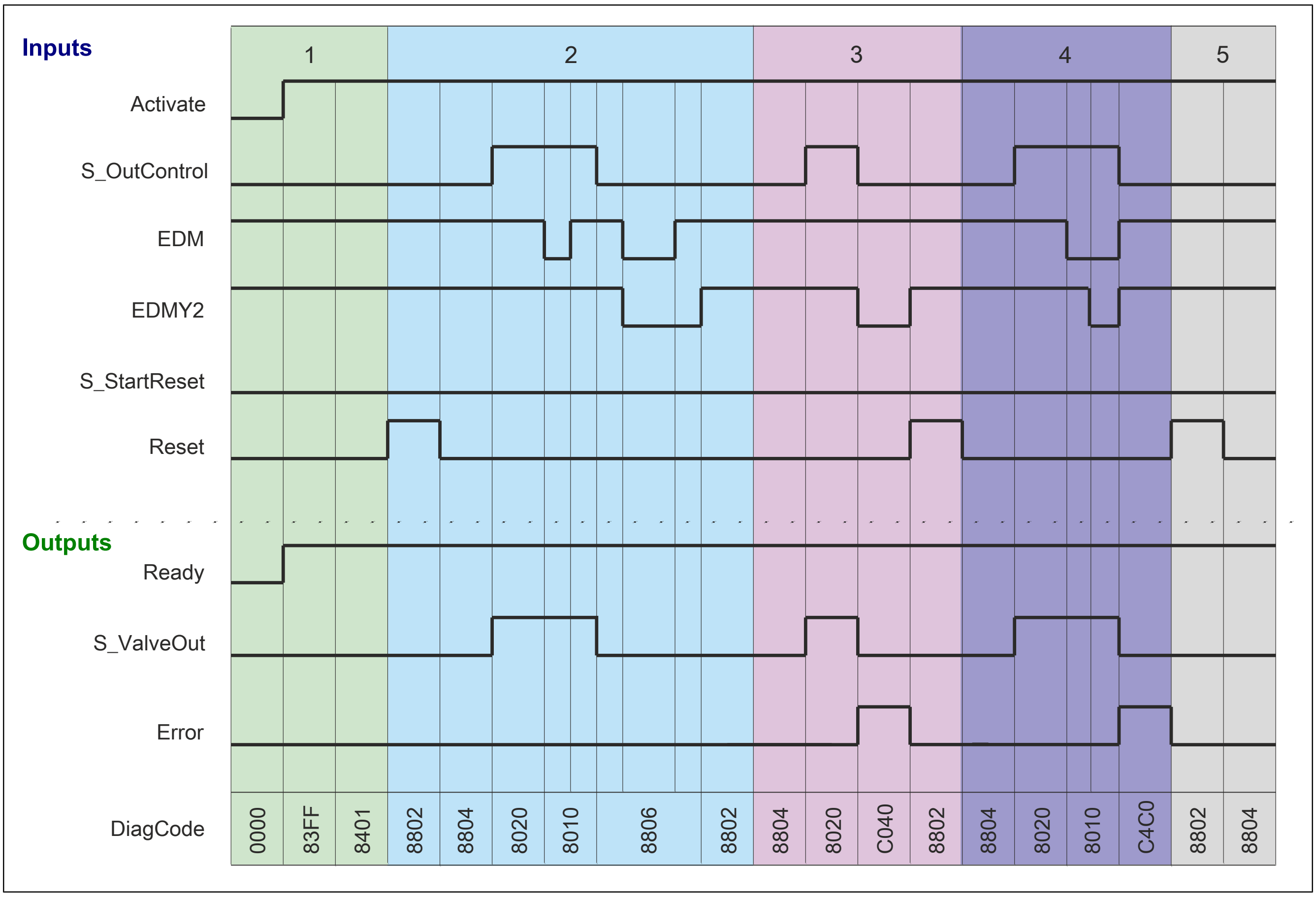


Figure 385: "SF_SingleValveCycleMonitoring": Signal sequence diagram

1 Initialization
2 Normal operation
3 Error - Request revoked before the state of the feedback signal on the "EDM" input parameter changed
4 Error - Request enabled but feedback signal on "EDMY2" input parameter not correct
5 Valve in basic state

6.5.10.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN ISO 13849-1	5.2.1 6.2
EN ISO 12100	6.2.11.4 6.2.11.6
EN ISO 4413	5.4.7.1 5.4.8.1
EN ISO 4414	5.4.6.1
EN 692	5.4.1.4 5.4.1.6 5.4.2.3

Table 559: "SF_SingleValveCycleMonitoring": Implementing requirements from standards

6.5.11 SF_SingleValveMonitoring

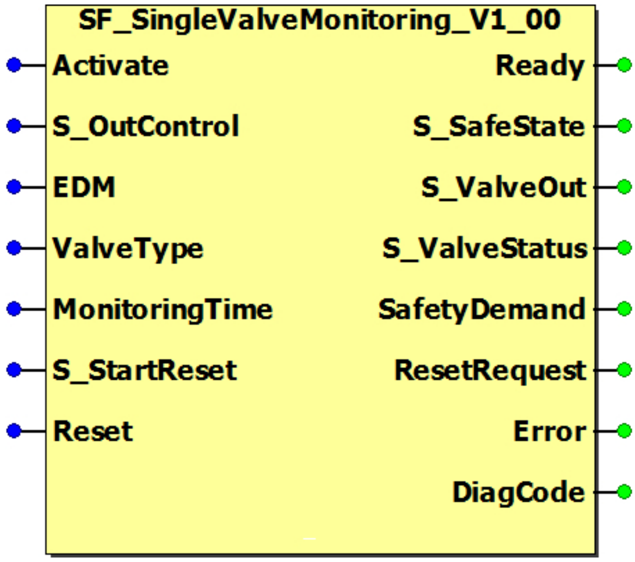


Figure 386: "SF_SingleValveMonitoring" function block

6.5.11.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_OutControl	SAFEBOOL	Variable	Status	FALSE	Control signal for controlling the valve
EDM	BOOL	Variable	Status	FALSE	Input for feedback signal from valve
ValveType	BOOL	Constant	Status	FALSE	Specifies the polarity of the feedback signal
MonitoringTime	TIME	Constant	Status	T#0s	Specifies the monitoring time of the switching operation
S_StartReset	SAFEBOOL	Variable/ Constant	Status	FALSE	Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 560: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_SafeState	SAFEBOOL	Variable	Status	FALSE	Indicates the basic state
S_ValveOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_ValveStatus	SAFEWORD	Variable	Status	16#0000	Status output for use with the "SF_ValveGroupControl" function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 561: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.11.2 Function

The "SF_SingleValveMonitoring" function block checks the defined basic state and dynamic switching state of valves that are controlled by safe output devices.

> **Information:**
>
> **Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.**

The outputs of safe devices that are used to control the valves to be monitored are energized to do so by the enable signal from this function block. The states of the connected valves are fed back to the function block via safe input devices.

In the basic state, the fed back signals from the valves being monitored must have the status defined by the function block ("ValveType" input parameter).

In the switching state, the fed back signals from the valves being monitored must have the inverted status after the defined time window expires ("MonitoringTime" input parameter).

> **Information:**
>
> **The "S_ValveStatus" output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group.**
>
> **This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).**

> **Information:**
>
> **The polarity of the feedback signals can be defined using the "ValveType" input parameter. This specifies whether the basic state is confirmed by a TRUE or FALSE signal.**

6.5.11.2.1 Monitoring the basic state

If the function block detects the basic state ("ValveType" input parameter) of the valves via the feedback signals present on the "EDM" input parameter during a switch-on request, then the function block will set its enable signal to TRUE and switch on the valves.

If the basic state is not detected by the function block during a switch-on request, then the function block will set its enable signal to FALSE and will not switch on the valves.

> **Information:**
>
> **If the basic state of the valve is detected, then the "S_SafeState" output parameter is set to TRUE.**
>
> **Leaving the basic state will switch "S_SafeState" to FALSE.**

6.5.11.2.2 Monitoring the switching state

After controlling the valve, the function block checks the reaction of the valves within the specified time window ("MonitoringTime" input parameter).

If the function block does not detect the states of the feedback signals defined for the switching state (depends on the "ValveType" input parameter), then the function block will set its enable signal to FALSE and cut the valves back off again.

6.5.11.2.3 Start interlock after cold restart (optional)

Support for a start interlock must be defined accordingly on the input parameter after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the enable signal as needed.

6.5.11.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.11.3.1 Connecting "ValveType"

> **Danger!**
>
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

6.5.11.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.11.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.11.3.4 Invalid static signals when cold restarting the safety controller

A static TRUE signal during a cold restart of the safety controller causes an error message on the function block if the start interlock is defined for after the function block is enabled.

If this start interlock is not defined for a cold restart of the safety controller, then the status of "Reset" is irrelevant.

6.5.11.3.5 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.11.3.6 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.11.4 Input parameters

6.5.11.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.11.4.2 S_OutControl

General function

- Control signal for controlling the valve

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Note that no hardware input is connected on the "S_OutControl" input parameter.**

Function description

This input parameter serves as a control signal for controlling the valve.

> ## Information:
> **The "S_OutControl" input parameter should be connected to an enable signal from a safe function block.**
>
> **The status of the "S_OutControl" input parameter corresponds to the status of the upstream function block / safety function.**

The status of the "S_OutControl" input parameter controls the "S_ValveOut" output parameter under consideration of the states of the connected feedback signals.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream safety function was not triggered.

> ## Information:
> **The "S_ValveOut" output parameter is set to TRUE if the remaining input signal combination is valid for this.**

FALSE

The safety function has been initiated or the safe device connected for this safety function is switched off or faulty.

> ## Information:
> **The "S_ValveOut" output parameter is set to FALSE.**

6.5.11.4.3 EDM

General function

- Input for feedback signal from valve

Data type

- BOOL

Connection

- Variable

Function description

The signal connected to the "EDM" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

"EDM" = TRUE and "ValveType" = FALSE

The valve is energized and not in the basic state.

"EDM" = TRUE and "ValveType" = TRUE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM" input parameter returns a TRUE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM" = FALSE and "ValveType" = FALSE

The valve is not energized but is in the basic state.

> **Information:**
>
> **If the "EDM" input parameter returns a FALSE signal, then the "S_SafeState" output parameter is set to TRUE.**

"EDM" = FALSE and "ValveType" = TRUE

The valve is energized and not in the basic state.

6.5.11.4.4 ValveType

General function

- Specifies the polarity of the feedback signal

Data type

- BOOL

Connection

- Constant

Function description

This input parameter is used to define the polarity of the feedback signal.

> ## Danger!
> **Note that a TRUE signal confirms the safe state with respect to the safety principle. You should therefore use valves that confirm the basic state with a TRUE signal.**

TRUE

The feedback signal returns status TRUE in the basic state.

FALSE

The feedback signal returns status FALSE in the basic state.

6.5.11.4.5 MonitoringTime

General function

- Specifies the monitoring time of the switching operation

Data type

- TIME

Connection

- Constant

Function description

This input parameter is used to define the monitoring time. Switching operations on safe inputs must take place within this monitoring time in order to be recognized as valid.

> ## Information:
>
> **The limits for the "MonitoringTime" input parameter must be defined and validated based on your application.**

6.5.11.4.6 S_StartReset

General function

- Defines the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the defined value.

Function description

This input parameter specifies the startup behavior of the function block following its activation and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.5.11.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.11.5 Output parameters

6.5.11.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.11.5.2 S_SafeState

General function

- Indicates the basic state

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates whether the valve is in the basic state.

TRUE

Valve in basic state

FALSE

Valve not in basic state

6.5.11.5.3 S_ValveOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is the safe signal used to control an output on a safe device, therefore also controlling the process. The enable signal is controlled depending on the status of the connected valve or valves as well as the start interlocks of the upstream safety functions.

TRUE

The output of a safe device is set to TRUE.

FALSE

The output of a safe device is set to FALSE.

6.5.11.5.4 S_ValveStatus

General function

- Status output for use with the "SF_ValveGroupControl" function block

Data type

- SAFEWORD

Connection

- Variable

Function description

This output parameter can be used together with the "SF_ValveGroupControl" function block to control several valves as a valve group. This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

6.5.11.5.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.11.5.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.11.5.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.11.5.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.11.5.9 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8010	The feedback signal on the "EDM" input parameter changed its signal state during an enabled request via the "S_OutControl" input parameter. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
8020	Waiting ("MonitoringTime" input parameter) for change to feedback signal on the "EDM" input parameter after the request was made via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = TRUE	No corrective measures are necessary if the signal is intended.
83FF	State for registering on the "SF_ValveGroupControl" function block.	No corrective measures are required.
8401	The start interlock to be carried out after enabling the function block is defined ("S_StartReset" = FALSE), or the function block's start interlock is active ("S_AutoReset" = FALSE).	Reset the function block.
8802	Waiting ("MonitoringTime" input parameter) for feedback signal on the "EDM" input parameter if no request is present via the "S_OutControl" input parameter. • "S_SafeState" = FALSE • "S_ValveOut" = FALSE	No corrective measures are necessary if the signal is intended.
8804	No request is present on the "S_OutControl" input parameter for setting the safe output to TRUE. • "S_SafeState" = TRUE • "S_ValveOut" = FALSE	"S_OutControl" must have status TRUE in order to set the safe output to TRUE.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	When the request is enabled using the "S_OutControl" input parameter, the state of the feedback signal on the "EDM" input parameter (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" feedback signal (depending on the configuration) returns the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	After the monitoring time ("MonitoringTime" input parameter) for the feedback signal on the "EDM" input parameter has expired, the state of "EDM" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" feedback signal (depending on the configuration) returns the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C041	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C070	After the request has been enabled and the monitoring time ("MonitoringTime" input parameter) for the feedback signal on the "EDM" input parameter has expired, the state of "EDM" (depending on the configuration with the "ValveType" input parameter) is not correct or the request is no longer present ("S_OutControl" = FALSE). After the request has been enabled and a valid state of the feedback signal has been detected, the state of "EDM" (depending on the configuration with the "ValveType" input parameter) is not correct.	• You can correct this problem by revoking the request with "S_OutControl" and making sure that the "EDM" feedback signal (depending on the configuration) returns the correct value in a de-energized state. • Check the wiring of the control device or the safe device connected to it.
C071	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 562: Diagnostic codes

Code (hex)	Description	Corrective measures
C100	A rising edge was detected on the "S_OutControl" and "Reset" input parameters at the same time. There is a danger of a programming error.	• The same signals are not permitted to be used to control the "S_OutControl" and "Reset" input parameters. • Check the wiring of the control device or the safe device connected to it.
C410	After enabling the request via "S_OutControl" and an incorrect signal was detected on the "EDM" input parameter, the signal is correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C440	After the monitoring time has expired and an incorrect signal was detected on the "EDM" input parameter, the feedback signal is correct again and a request is not being enabled via "S_OutControl".	Reset the function block.
C470	After enabling the request, the monitoring time has expired and an incorrect signal was detected on the "EDM" input parameter, the feedback signal is correct again and a request is not being enabled via "S_OutControl". After enabling the request and an incorrect signal was detected on the "EDM" input parameter, the feedback signal is correct again and a request is not being enabled via "S_OutControl".	Reset the function block.

Table 562: Diagnostic codes

6.5.11.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram

"ValveType" = TRUE
"S_StartReset" = FALSE

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

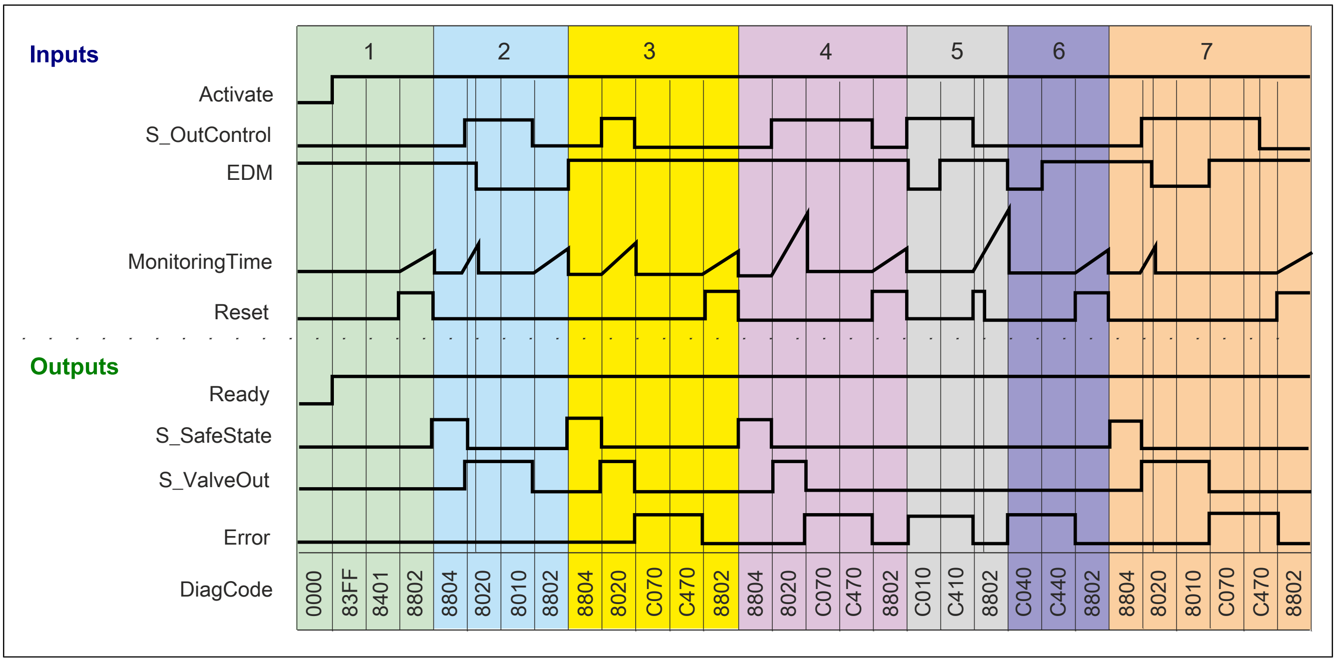


Figure 387: "SF_SingleValveMonitoring": Signal sequence diagram

1 Initialization
2 Normal operation
3 Error - Request revoked before the state of the feedback signal changed
4 Error - Feedback signal on "EDM" input parameter didn't change its state within the defined time window
5 Error - Request enabled but feedback signal on "EDM" input parameter not correct
6 Error - Feedback signal on "EDM" input parameter not correct after the specified time window has expired
7 Error - Request enabled but feedback signal on "EDM" input parameter not correct

6.5.11.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN ISO 13849-1	5.2.1 6.2
EN ISO 12100	6.2.11.4 6.2.11.6
EN ISO 4413	5.4.7.1 5.4.8.1
EN ISO 4414	5.4.6.1
EN 692	5.4.1.4 5.4.1.6 5.4.2.3

Table 563: "SF_SingleValveMonitoring": Implementing requirements from standards

6.5.12 SF_TwoHandControlTypeIIIC

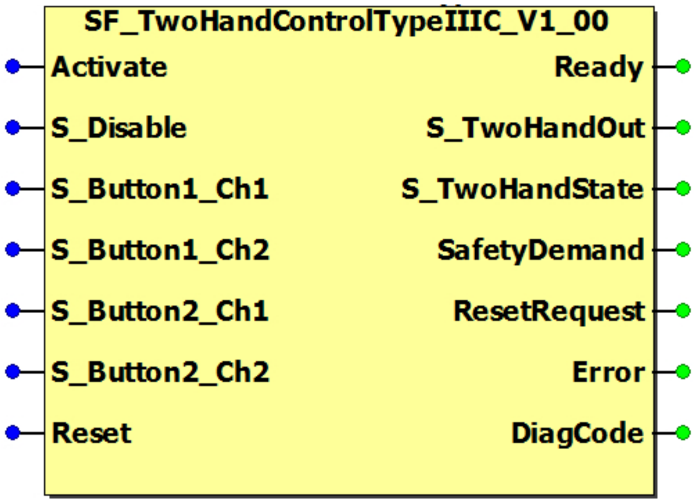


Figure 388: "SF_TwoHandControlTypeIIIC" function block

6.5.12.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_Disable	SAFEBOOL	Variable	Status	FALSE	Control signal for disabling and disconnecting the two-hand control device
S_Button1_Ch1	SAFEBOOL	Variable	Status	FALSE	Input for channel 1 of button 1 on the two-hand control device
S_Button1_Ch2	SAFEBOOL	Variable	Status	FALSE	Input for channel 2 of button 1 on the two-hand control device
S_Button2_Ch1	SAFEBOOL	Variable	Status	FALSE	Input for channel 1 of button 2 on the two-hand control device
S_Button2_Ch2	SAFEBOOL	Variable	Status	FALSE	Input for channel 2 of button 2 on the two-hand control device
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 564: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_TwoHandOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_TwoHandState	SAFEWORD	Variable	Status	16#0000	Status output for use with the "SF_TwoHandMultiOperator" function block
SafetyDemand	BOOL	Variable	Status	FALSE	Indicates that the safety function is being requested
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 565: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.12.2 Function

The "SF_TwoHandControlTypeIIIC" function block supports the use of plug-in two-hand control devices (Type III C) in accordance with EN 574.

> ## Information:
>
> **Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.**

The function block uses an internal timer to monitor whether a signal change on the "S_Button1_Ch1", "S_Button1_Ch2", "S_Button2_Ch1" or "S_Button2_Ch2" input parameter results in the state of all other signals being changed within 500 ms. This applies both when the buttons are pressed as well as when it is released.

The function block can be used to ensure the release of both buttons before the "S_TwoHandOut" output parameter is set back to TRUE. It is also possible to disable and disconnect the two-hand control device.

> ## Information:
>
> **You must use two two-channel buttons. The "S_Button1_Ch1" and "S_Button1_Ch2" represent button 1, while the "S_Button2_Ch1" and "S_Button2_Ch2" input parameters represent button 2.**
>
> **If an error is detected for the two-hand control device, acknowledgment via "Reset" is not necessary. It is sufficient that the two-hand control device is not in an actuated state.**

> ## Information:
>
> **The "S_TwoHandState" output parameter can be used together with the "SF_TwoHandMultiOperator" function block to monitor two control devices at the same time.**

6.5.12.2.1 Disabling and disconnecting two-hand control devices

Follow the steps below to disconnect the two-hand control device without generating an error message:

- The two-hand control device is connected and not being actuated.
- Set the "S_Disable" input parameter to TRUE.
- Disconnect the two-hand control device.

6.5.12.2.2 Reconnect the two-hand control device.

Follow the steps below to reconnect the two-hand control device without generating an error message:

- The two-hand control device is not connected and the "S_Disable" input parameter is set to TRUE.
- Connect the two-hand control device.
- Set the "S_Disable" input parameter to FALSE.
- The two-hand control device is connected and not being actuated.

6.5.12.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.12.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.12.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.12.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.12.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.12.4 Input parameters

6.5.12.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.12.4.2 S_Disable

General function

- Control signal for disabling and disconnecting the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter serves as the control signal for disabling and disconnecting the two-hand control device.

Information:

Follow the steps below to disconnect the two-hand control device without generating an error message:

- **The two-hand control device is connected and not being actuated.**
- **Set the "S_Disable" input parameter to TRUE.**
- **Disconnect the two-hand control device.**

Information:

Follow the steps below to reconnect the two-hand control device without generating an error message:

- **The two-hand control device is not connected and the "S_Disable" input parameter is set to TRUE.**
- **Connect the two-hand control device.**
- **Set the "S_Disable" input parameter to FALSE.**
- **The two-hand control device is connected and not being actuated.**

TRUE

The two-hand control device is disabled and can be disconnected.

FALSE

The two-hand control device is enabled and must be connected.

6.5.12.4.3 S_Button1_Ch1

General function

- Input for channel 1 of button 1 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_Button1_Ch1" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **A normally open contact is involved here.**

TRUE

The normally open contact is closed – the button is being actuated.

FALSE

The normally open contact is open – the button is not being actuated or the safe input device connected to this two-hand control device is shut off or defective.

6.5.12.4.4 S_Button1_Ch2

General function

- Input for channel 2 of button 1 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_Button1_Ch2" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **A normally closed contact is involved here.**

TRUE

The normally closed contact is closed – the button is not being actuated.

FALSE

The normally closed contact is open – the button is being actuated or the safe input device connected to this two-hand control device is shut off or defective.

6.5.12.4.5 S_Button2_Ch1

General function

- Input for channel 1 of button 2 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_Button2_Ch1" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **A normally open contact is involved here.**

TRUE

The normally open contact is closed – the button is being actuated.

FALSE

The normally open contact is open – the button is not being actuated or the safe input device connected to this two-hand control device is shut off or defective.

6.5.12.4.6 S_Button2_Ch2

General function

- Input for channel 2 of button 2 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Function description

The signal connected to the "S_Button2_Ch2" input parameter is processed by the function block.

The signal input is state-controlled. These states only lead to the following link results if the function block is enabled ("Activate" = TRUE).

> **Information:**
>
> **A normally closed contact is involved here.**

TRUE

The normally closed contact is closed – the button is not being actuated.

FALSE

The normally closed contact is open – the button is being actuated or the safe input device connected to this two-hand control device is shut off or defective.

6.5.12.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.12.5 Output parameters

6.5.12.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.12.5.2 S_TwoHandOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the status of the two-hand control device.

The enable signal can be used for subsequent process control.

> ## Danger!
>
> **The enable signal is only permitted to control the process directly if it does not impair the safety function.**
>
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The two-hand control device is actuated.

FALSE

The two-hand control device is not actuated.

6.5.12.5.3 S_TwoHandState

General function

- Status output for use with the "SF_TwoHandMultiOperator" function block

Data type

- SAFEWORD

Connection

- Variable

Function description

This output parameter can be used together with the "SF_TwoHandMultiOperator" function block to monitor two control devices at the same time.

6.5.12.5.4 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending request for the safety function. An appropriate action must be taken (see the "Diagnostic codes" table) to leave this state.

TRUE

The enabled function block has detected a request for the safety function. There are no errors, and the associated enable signal is set to FALSE.

FALSE

The function block is not enabled, or the enabled function block has not detected that the safety function has been requested.

6.5.12.5.5 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.12.5.6 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.12.5.7 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.12.5.8 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_TwoHandOut" = TRUE	No corrective measures are necessary if the signal is intended.
8001	Initialization of the function block after it has been enabled.	No corrective measures are required.
83FE	State for registering on the "SF_TwoHandMultiOperator" function block.	No corrective measures are required.
8802	The two-hand control device is connected but disabled via the "S_Disable" input parameter. • "S_TwoHandOut" = FALSE	No corrective measures are necessary if the signal is intended.
8804	The two-hand control device is disabled ("S_Disable" = TRUE) and also not connected. • "S_TwoHandOut" = FALSE	Reconnect the two-hand control device. No corrective measures are necessary if the signal is intended.
8806	The two-hand control device is disabled ("S_Disable" = TRUE), was disconnected and has now been reconnected. • "S_TwoHandOut" = FALSE	Enable the two-hand control device by setting "S_Disable" to FALSE. No corrective measures are necessary if the signal is intended.
8812	The two-hand control device is connected but not being actuated.	To set the safe output to TRUE, "S_Button1_Ch1", "S_Button1_Ch2", "S_Button2_Ch1" and "S_Button2_Ch2" must change their signal state within 500 ms.
8822	The two-hand control device was actuated.	No corrective measures are necessary if the signal is intended.
8832	The two-hand control device is no longer being actuated.	No corrective measures are necessary if the signal is intended.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C031	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C041	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C400	The two-hand control device is disconnected and also not disabled ("S_Disable" = FALSE).	Reset the function block.
C410	An unachievable disabling ("S_Disable" = TRUE) was detected by the function block when the buttons were pressed.	Reset the function block.
C420	An unachievable disabling ("S_Disable" = TRUE) was detected by the function block when the buttons were no longer pressed.	Reset the function block.
C430	An unachievable disabling ("S_Disable" = TRUE) was detected by the function block when the buttons are pressed.	Reset the function block.
C800	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C810	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C820	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C830	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C840	The function block detected an invalid signal combination. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C850	The function block detected an invalid signal combination. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.

Table 566: Diagnostic codes

Code (hex)	Description	Corrective measures
C860	The function block detected an invalid signal combination. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C870	The function block detected an invalid signal combination. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C880	The function block detected an invalid signal combination. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C890	The function block detected an invalid signal combination. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C8A0	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C8B0	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C8C0	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C8D0	The function block detected an invalid signal combination. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C8E0	The function block detected an invalid signal combination. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	No corrective measures are required.
C8F0	The function block detected an invalid signal combination. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C900	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C910	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C920	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C930	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C940	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C950	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C960	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C970	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C980	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.

Table 566: Diagnostic codes

Code (hex)	Description	Corrective measures
C990	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C9A0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C9B0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C9C0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C9D0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
C9E0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	No corrective measures are required.
C9F0	The function block detected a timeout (>500 ms) when actuating the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA00	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA10	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA20	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA30	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = FALSE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA40	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA50	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA60	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA70	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA80	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CA90	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.

Table 566: Diagnostic codes

Code (hex)	Description	Corrective measures
CAA0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = FALSE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CAB0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CAC0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CAD0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Error (channel 1 = TRUE / channel 2 = TRUE) • Button 2 - Error (channel 1 = TRUE / channel 2 = TRUE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.
CAE0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Not actuated (channel 1 = FALSE / channel 2 = TRUE) • Button 2 - Not actuated (channel 1 = FALSE / channel 2 = TRUE)	No corrective measures are required.
CAF0	The function block detected a timeout (>500 ms) when releasing the control device. • Button 1 - Actuated (channel 1 = TRUE / channel 2 = FALSE) • Button 2 - Actuated (channel 1 = TRUE / channel 2 = FALSE)	• Make sure that the control device is not being actuated. • Check the wiring of the control device or the safe device connected to it.

Table 566: Diagnostic codes

6.5.12.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

> **Information:**
>
> **The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.**

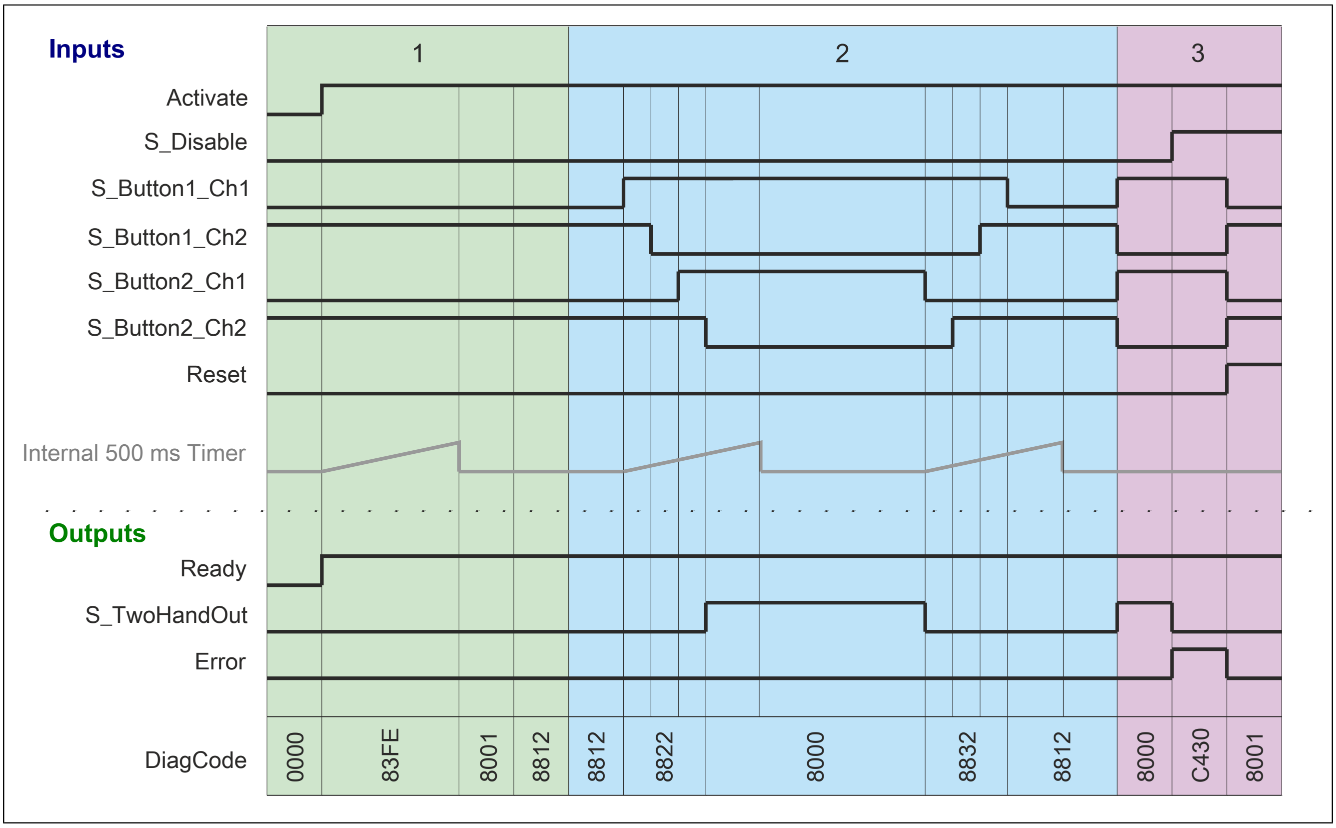


Figure 389: "SF_TwoHandControlTypeIIIC": Signal sequence diagram 1

1 Initialization
2 Normal operation
3 Error – Disabling not allowed

Signal sequence diagram 2

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

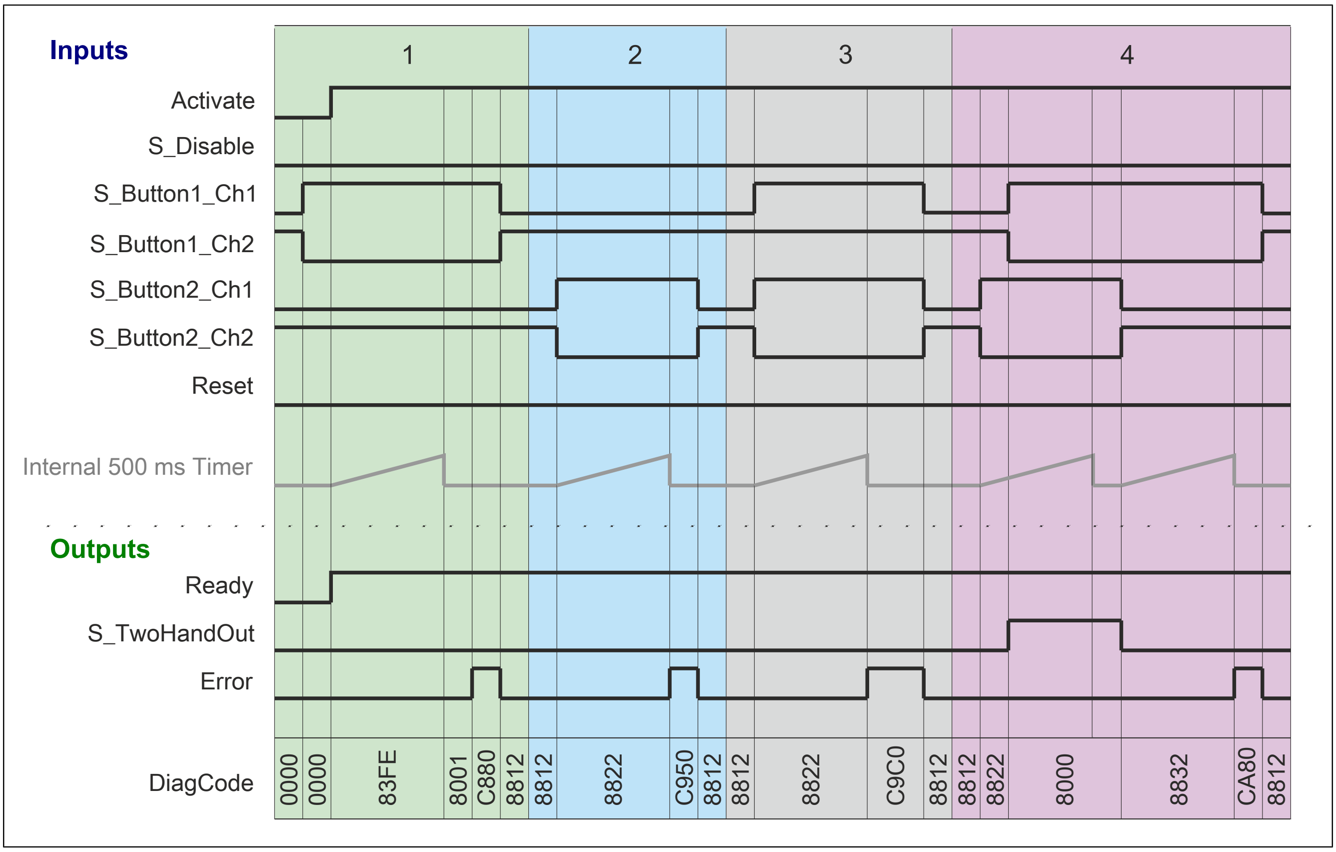


Figure 390: "SF_TwoHandControlTypeIIIC": Signal sequence diagram 2

1 Error - Button 1 is pressed during initialization
2 Error - Button 1 pressed
3 Error - After 500 ms the state of button 1 is not correct during actuation
4 Error - After 500 ms the state of button 1 is not correct during release

6.5.12.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN 574	5.1 5.2 5.3 5.6 5.7 6.3 6.4
ISO 12100-2	4.11.4

Table 567: "SF_TwoHandControlTypeIIIC": Implementing requirements from standards

6.5.13 SF_TwoHandMultiOperator

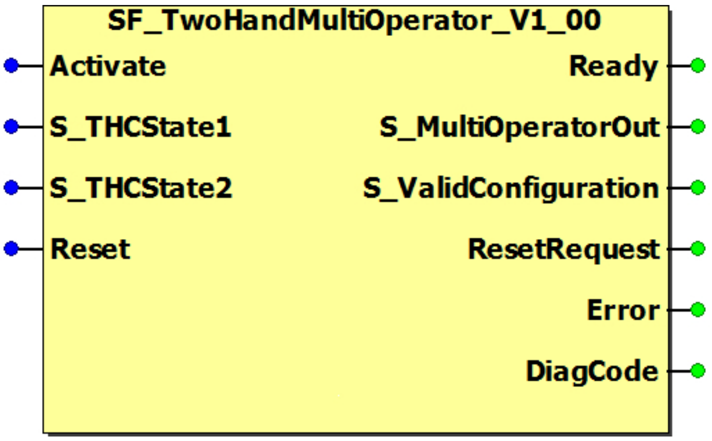


Figure 391: "SF_TwoHandMultiOperator" function block

6.5.13.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_THCState1	SAFEWORD	Variable	Status	16#0000	Input for evaluating the first two-hand control device
S_THCState2	SAFEWORD	Variable	Status	16#0000	Input for evaluating the second two-hand control device
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error is no longer pending

Table 568: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_MultiOperatorOut	SAFEBOOL	Variable	Status	FALSE	Enable signal of the function block
S_ValidConfiguration	SAFEBOOL	Variable	Status	FALSE	Indicates a valid configuration for both two-hand control devices
ResetRequest	BOOL	Variable	Status	FALSE	Indicates that a reset is required on the function block
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 569: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.13.2 Function

The "SF_TwoHandMultiOperator" function block supports the use of two control stations, each with one two-hand control device for controlling a press.

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

Information:

This function block is always used together with the "SF_TwoHandControlTypeIIIC" function block.

Information:

The two control stations must always be located on the same side of the press (front or back).

If there are more than two control stations, the function block must be instanced multiple times.

The function block determines a valid configuration of both two-hand control devices and controls the enable output depending on the status of the upstream "SF_TwoHandControlTypeIIIC" function blocks. For this, the "S_THCState1" and "S_THCState2" input parameters are connected to the "S_TwoHandState" output parameter of the respective instance.

The "S_MultiOperatorOut" is set to TRUE if a valid configuration has been detected and at least one of the two (or both) two-hand control devices is actuated.

The "S_ValidConfiguration" output parameter is set to TRUE if a valid configuration is detected.

Valid configurations include:

- Both two-hand control devices are connected.
- One two-hand control device is connected, the other is disabled and disconnected.
- Both two-hand control devices are disabled and disconnected.

Information:

If both two-hand control devices are disabled and disconnected, "S_ValidConfiguration" does switch to TRUE, but the "S_MultiOperatorOut" output parameter will remain set to FALSE.

Information:

The function block does not include any sort of time monitoring.

A registration algorithm is monitored to ensure that only a function block of type "SF_TwoHandControlTypeIIIC" can be used with this function block.

6.5.13.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.13.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.13.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.13.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.13.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.13.4 Input parameters

6.5.13.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.13.4.2 S_THCState1

General function

- Input for evaluating the first two-hand control device

Data type

- SAFEWORD

Connection

- Variable

> ## Information:
> **Connect the "S_TwoHandState" output signal of the preceding safety-oriented "SF_TwoHandControlTypeIIIC" function block here in order to evaluate the first two-hand control device.**

Function description

This signal input is used to evaluate the state of the first two-hand control device.

6.5.13.4.3 S_THCState2

General function

- Input for evaluating the second two-hand control device

Data type

- SAFEWORD

Connection

- Variable

Information:

Connect the "S_TwoHandState" output signal of the preceding safety-oriented "SF_TwoHandControlTypeIIIC" function block here in order to analyze the second two-hand control device.

Function description

This signal input is used to evaluate the state of the second two-hand control device.

6.5.13.4.4 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with the "S_StartReset" and/or "S_AutoReset" input parameter.

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of the input parameter are monitored internally by the function block. The function is only executed on a rising edge of the "Reset" input parameter. An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on the "Reset" input parameter as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.5.13.5 Output parameters

6.5.13.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.13.5.2 S_MultiOperatorOut

General function

- Enable signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is controlled depending on the status of both two-hand control devices.

The enable signal can be used for subsequent process control.

> ## Danger!
>
> **The enable signal is only permitted to control the process directly if it does not impair the safety function.**
>
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

> ## Information:
>
> **The "S_ValidConfiguration" output parameter can be used to determine whether the configuration of both two-hand control devices is valid.**

TRUE

The first two-hand control device, the second two-hand control device or both two-hand control devices are actuated.

FALSE

The two-hand control devices are not actuated.

6.5.13.5.3 S_ValidConfiguration

General function

- Indicates a valid configuration for both two-hand control devices

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates a valid configuration for both two-hand control devices.

TRUE

The configuration of both two-hand control devices is valid.

> ## Information:
> **If both two-hand control devices are disabled and disconnected, "S_ValidConfiguration" does switch to TRUE, but the "S_MultiOperatorOut" output parameter will remain set to FALSE.**

FALSE

The configuration of both two-hand control devices is invalid.

6.5.13.5.4 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates that the "Reset" input parameter must be acknowledged (see the "Diagnostic codes" table).

TRUE

The enabled function block has detected an error, and acknowledgment is necessary (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected any errors.

The "DiagCode" output parameter indicates the status.

6.5.13.5.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.13.5.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.13.5.7 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8010	Waiting for registration of function blocks of type "SF_TwoHandControlTypeIIIC" on the "S_THCState1" and "S_THCState2" input parameters.	No corrective measures are required.
8020	Waiting for information about the connected function blocks.	No corrective measures are required.
8030	The information from the control device to "S_THCState1" is valid, and the control device is connected. The information from the control device to "S_THCState2" is valid, but the control device is disconnected. • "S_MultiOperatorOut" = FALSE • "S_ValidConfiguration" = TRUE	Actuate the control device that is connected to "S_THCState1".
8040	The information from the control device to "S_THCState1" is valid, but the control device is disconnected. The information from the control device to "S_THCState2" is valid, and the control device is connected. • "S_MultiOperatorOut" = FALSE • "S_ValidConfiguration" = TRUE	Actuate the control device that is connected to "S_THCState2".
8050	The information from the control device to "S_THCState1" and "S_THCState2" is valid, and both control devices are connected. • "S_MultiOperatorOut" = FALSE • "S_ValidConfiguration" = TRUE	Actuate the control devices that are connected to "S_THCState1" and "S_THCState2".
8060	The information from the control device to "S_THCState1" and "S_THCState2" is valid, but both control devices are disconnected. • "S_MultiOperatorOut" = FALSE • "S_ValidConfiguration" = TRUE	No corrective measures are necessary if the signal is intended.
8100	The control device connected to "S_THCState1" is actuated. The control device connected to "S_THCState2" is disconnected. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_MultiOperatorOut" = TRUE • "S_ValidConfiguration" = TRUE	No corrective measures are necessary if the signal is intended.
8200	The control device connected to "S_THCState1" is disconnected. The control device connected to "S_THCState2" is actuated. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_MultiOperatorOut" = TRUE • "S_ValidConfiguration" = TRUE	No corrective measures are necessary if the signal is intended.
8300	The control device connected to "S_THCState1" and "S_THCState2" is actuated. The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_MultiOperatorOut" = TRUE • "S_ValidConfiguration" = TRUE	No corrective measures are necessary if the signal is intended.
8410	The function block detected an invalid signal on "S_THCState1".	Reset the function block.
8420	The function block detected an invalid signal on "S_THCState2".	Reset the function block.
8430	The function block detected an invalid signal on "S_THCState1" and "S_THCState2".	Reset the function block.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C011	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C021	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C031	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	The registration sequence on the "S_THCState1" input parameter is invalid.	Check the connection of the "S_THCState1" input parameter.
C050	The registration sequence on the "S_THCState2" input parameter is invalid.	Check the connection of the "S_THCState2" input parameter.

Table 570: Diagnostic codes

6.5.13.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

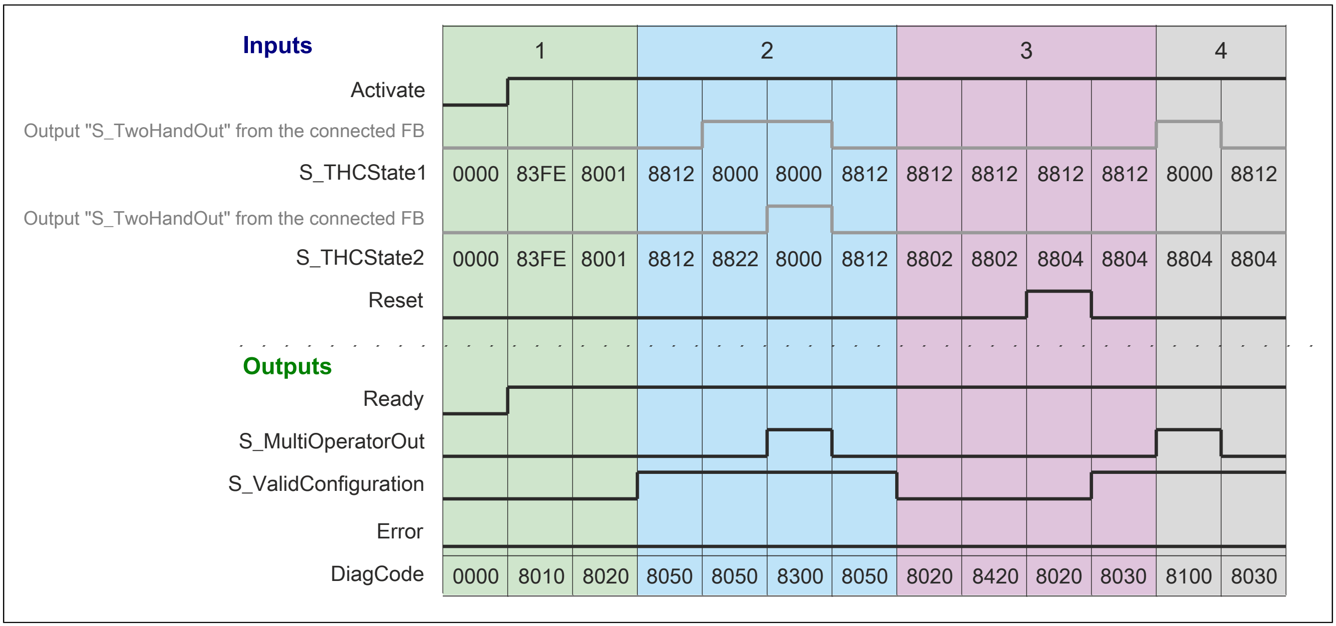


Figure 392: "SF_TwoHandMultiOperator": Signal sequence diagram

1 Initialization
2 Normal operation with two control stations
3 Control station disabled and disconnected
4 Normal operation with one control station

6.5.13.7 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter
EN 692	5.3.14 5.4.5.4

Table 571: "SF_TwoHandMultiOperator": Implementing requirements from standards

6.5.14 SF_ValveGroupControl

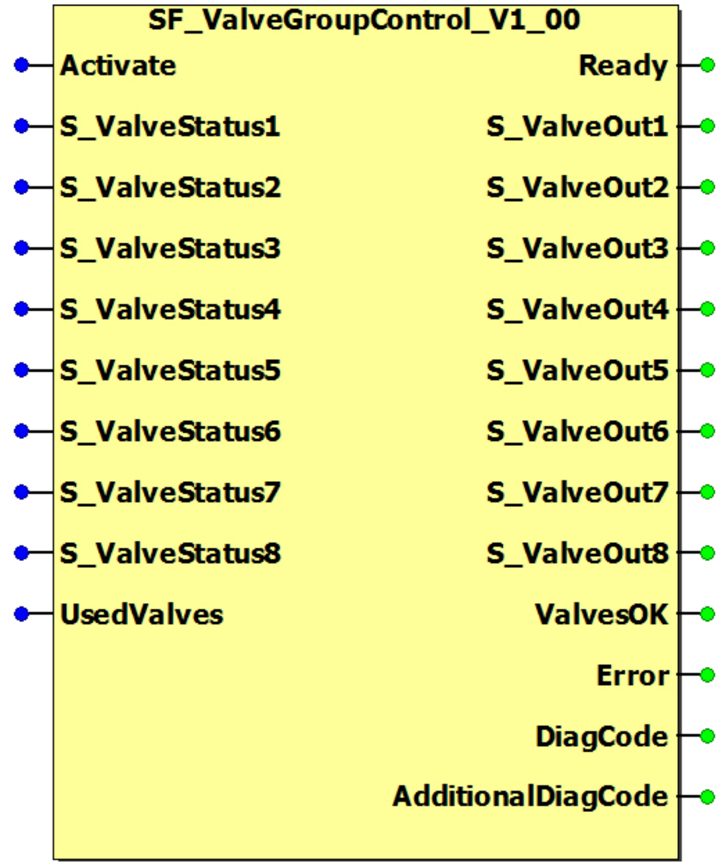


Figure 393: "SF_ValveGroupControl" function block

6.5.14.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	Status	FALSE	Enables the function block ("Activate" = TRUE)
S_ValveStatus1	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus2	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus3	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus4	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus5	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus6	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus7	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
S_ValveStatus8	SAFEWORD	Variable	Status	16#0000	Input for analyzing the valve
UsedValves	INT	Constant	Status	INT#0	Defines the number of valves

Table 572: Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	Status	FALSE	Indicates that the function block is enabled
S_ValveOut1	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut2	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut3	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut4	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut5	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut6	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut7	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
S_ValveOut8	SAFEBOOL	Variable	Status	FALSE	Enable signal from the function block for the valve
ValvesOK	BOOL	Variable	Status	FALSE	Indicates the state of all valves
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message
AdditionalDiagCode	BYTE	Variable	Status	16#00	Additional diagnostic message from the function block to indicate which valve is enabled or causing an error

Table 573: Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.5.14.2 Function

The "SF_ValveGroupControl" function block can control up to eight valves as a single valve group. This makes it possible to switch other valves to the safe state if one of the valves in the group fails (e.g. if the double valve doesn't switch or the feedback signals do not correspond to the expected signals).

Information:

Refer to the diagrams in the "Signal sequence diagrams of function block" section as the basis for determining whether the function block is being used properly. You must also observe the information specified in the "Error prevention" section.

Information:

This function block is always used together with up to eight valve blocks.

If several valve groups should be formed, then the function block needs to be instanced multiple times.

The function block controls the enable outputs depending on the state of the upstream valve blocks. For this, the "S_ValveStatus1" to "S_ValveStatus8" input parameters are connected to the "S_ValveStatus" output parameter of the respective valve block. The number of valves is defined with the "UsedValves" input parameter.

Information:

A registration algorithm is monitored to ensure that only valve blocks can be used with this function block.

The number of registered valves is compared with the "UsedValves" input parameter; gaps are not permitted between the input parameters being used.

If a valve reports an error state, then all other enable signals are also set to FALSE.

Information:

The additional diagnostic message from the "AdditionalDiagCode" output parameter indicates which of the eight valves are energized and which are returning an error.

Each valve is represented by one bit.

6.5.14.3 Error prevention

The following note regarding validation applies to all errors listed in this section.

> **Danger!**
>
> **Always validate the entire safety function!**

6.5.14.3.1 Incorrect connection of "UsedValves"

The function block detects an error if an illicit constant is connected to the "UsedValves" input parameter.

Possible cause:

- Connected constant < 0
- Connected constant > 8

6.5.14.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.5.14.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.5.14.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.5.14.3.5 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

Danger!

You are responsible for the functional testing of protective equipment.
You must therefore validate the protective equipment!

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.5.14.4 Input parameters

6.5.14.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When enabling or disabling safe devices, "Activate" must be linked to a variable that indicates the status (enabled or disabled) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is disabled. This linkage allows the function block to optionally support a start interlock (as long as the "S_StartReset" input parameter is present) after enabling the device if the states of the safe devices involved in the safety function have been switched to "Activate".
- If "Activate" is set to TRUE when a cold restart of the safety controller is performed, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, "Activate" must be set to FALSE so that defective hardware/peripherals on the function block do not result in a faulty diagnostic message.
- "Activate" can also be connected to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After being enabled, the function block supports an optional start interlock. The start interlock is reset with a rising edge of "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is disabled.

All binary output parameters are set to FALSE.
The "DiagCode" output parameter is set to WORD#16#0000.

In order to control function block diagnostics as needed in the diagnostic concept when error messages from safe devices and/or disabled safe devices occur, "Activate" must be connected to a signal that indicates the status of the safe devices utilizing the safety functionality supported by the function block. This signal can only be created for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of disabled safe devices.

6.5.14.4.2 S_ValveStatusX [1 to 8]

General function

- Input for analyzing the valve

Data type

- SAFEWORD

Connection

- Variable

Information:

Connect the "S_ValveState" output signal of the preceding safety-oriented function block (e.g. "SF_SingleValveMonitoring") in order to analyze the respective valve.

Function description

This signal input is used to evaluate the state of the respective valve.

6.5.14.4.3 UsedValves

General function

- Defines the number of valves

Data type

- INT

Connection

- Constant

Function description

This input parameter is used to define the number of valves for the valve group.

6.5.14.5 Output parameters

6.5.14.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether or not the function block is enabled.

TRUE

The function block is enabled ("Activate" = TRUE), with the output parameters indicating the current status of the safety function.

FALSE

The function block is disabled ("Activate" = FALSE), with the function block outputs set to FALSE.

6.5.14.5.2 S_ValveOutX [1 to 8]

General function

- Enable signal from the function block for the valve

Data type

- SAFEBOOL

Connection

- Variable

Function description

The enable signal is the safe signal used to control an output on a safe device, therefore also controlling the process. The enable signal is controlled depending on the status of the connected valve or valves.

TRUE

The output of a safe device is set to TRUE.

The following conditions must be met for this:

- The valve is energized by the associated function block (e.g. "SF_SingleValveMonitoring").
- The function block did not detect any errors.

FALSE

The output of a safe device is set to FALSE.

This may be due to one of the following reasons:

- The valve is not energized by the associated function block (e.g. "SF_SingleValveMonitoring").
- The function block detected an error.

6.5.14.5.3 ValvesOK

General function

- Indicates the state of all valves

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates the state of all valves in the group.

TRUE

No error in the connected valves

FALSE

One or more valves indicating an error

6.5.14.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message causes the safe output parameters to be set to and remain at FALSE.

To leave an error state ("Error" = TRUE), the "Reset" input parameter must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you have to set the "Reset" input parameter from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

All safe output parameters are set to FALSE. The "DiagCode" output parameter indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. The "DiagCode" output parameter indicates the status.

6.5.14.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via the "Error" output parameter.

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

For status messages, the "Error" output parameter is set to FALSE by the function block.

All other messages are error messages, i.e. "Error" = TRUE.

6.5.14.5.6 Diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is disabled. If "Activate" is connected to a variable that represents the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected an error in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that represents the state of a connected safe device (active, inactive or peripheral error detected), or correct the error in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. • "S_ValveOutX" = TRUE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" = TRUE • "AdditionalDiagCode" depends on the "UsedValves" input parameter and which valves are enabled.	No corrective measures are necessary if the signal is intended.
8001	Initialization of the function block after it has been enabled.	No corrective measures are required.
8002	Waiting for registration of valve function blocks on the "S_ValveStatusX" input parameters (X = 1 to the value of the "UsedValves" input parameter). • "AdditionalDiagCode" depends on the "UsedValves" input parameter and which valves are not yet registered.	No corrective measures are required.
8006	The valves specified with the "UsedValves" input parameter successfully registered on the function block. • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x00	Set the valves on "S_ValveStatusX" (x = 1 to "UsedValves") again.
8014	Valve 1 on the "S_ValveStatus1" input parameter has been deregistered ("S_ValveStatus1" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x01	Perform the registration sequence on the "S_ValveStatus1" input parameter once more.
8024	Valve 2 on the "S_ValveStatus2" input parameter has been deregistered ("S_ValveStatus2" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x02	Perform the registration sequence on the "S_ValveStatus2" input parameter once more.
8034	Valve 3 on the "S_ValveStatus3" input parameter has been deregistered ("S_ValveStatus3" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x04	Perform the registration sequence on the "S_ValveStatus3" input parameter once more.
8044	Valve 4 on the "S_ValveStatus4" input parameter has been deregistered ("S_ValveStatus4" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x08	Perform the registration sequence on the "S_ValveStatus4" input parameter once more.
8054	Valve 5 on the "S_ValveStatus5" input parameter has been deregistered ("S_ValveStatus5" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x10	Perform the registration sequence on the "S_ValveStatus5" input parameter once more.
8064	Valve 6 on the "S_ValveStatus6" input parameter has been deregistered ("S_ValveStatus6" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x20	Perform the registration sequence on the "S_ValveStatus6" input parameter once more.

Table 574: Diagnostic codes

Code (hex)	Description	Corrective measures
8074	Valve 7 on the "S_ValveStatus7" input parameter has been deregistered ("S_ValveStatus7" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x40	Perform the registration sequence on the "S_ValveStatus7" input parameter once more.
8084	Valve 8 on the "S_ValveStatus8" input parameter has been deregistered ("S_ValveStatus8" = 0x0000). • "S_ValveOutX" = FALSE (X = 1 to the value of the "UsedValves" input parameter) • "S_ValvesOK" depends on the state of the connected function blocks. • "AdditionalDiagCode" = 0x80	Perform the registration sequence on the "S_ValveStatus8" input parameter once more.
C000	The value on the "UsedValves" input parameter is invalid.	The value must be greater than 0 and less than 9.
C410	Valve 1 on the "S_ValveStatus1" input parameter has reported an error. • "AdditionalDiagCode" = 0x01	Correct the error on the function block connected via "S_ValveStatus1".
C420	Valve 2 on the "S_ValveStatus2" input parameter has reported an error. • "AdditionalDiagCode" = 0x02	Correct the error on the function block connected via "S_ValveStatus2".
C430	Valve 3 on the "S_ValveStatus3" input parameter has reported an error. • "AdditionalDiagCode" = 0x04	Correct the error on the function block connected via "S_ValveStatus3".
C440	Valve 4 on the "S_ValveStatus4" input parameter has reported an error. • "AdditionalDiagCode" = 0x08	Correct the error on the function block connected via "S_ValveStatus4".
C450	Valve 5 on the "S_ValveStatus5" input parameter has reported an error. • "AdditionalDiagCode" = 0x10	Correct the error on the function block connected via "S_ValveStatus5".
C460	Valve 6 on the "S_ValveStatus6" input parameter has reported an error. • "AdditionalDiagCode" = 0x20	Correct the error on the function block connected via "S_ValveStatus6".
C470	Valve 7 on the "S_ValveStatus7" input parameter has reported an error. • "AdditionalDiagCode" = 0x40	Correct the error on the function block connected via "S_ValveStatus7".
C480	Valve 8 on the "S_ValveStatus8" input parameter has reported an error. • "AdditionalDiagCode" = 0x80	Correct the error on the function block connected via "S_ValveStatus8".
C510	Valve 1 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x01	Check the signal connected to the "S_ValveStatus1" input parameter.
C520	Valve 2 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x02	Check the signal connected to the "S_ValveStatus2" input parameter.
C530	Valve 3 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x04	Check the signal connected to the "S_ValveStatus3" input parameter.
C540	Valve 4 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x08	Check the signal connected to the "S_ValveStatus4" input parameter.
C550	Valve 5 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x10	Check the signal connected to the "S_ValveStatus5" input parameter.
C560	Valve 6 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x20	Check the signal connected to the "S_ValveStatus6" input parameter.
C570	Valve 7 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x40	Check the signal connected to the "S_ValveStatus7" input parameter.
C580	Valve 8 is being used but has not been successfully registered on the function block. • "AdditionalDiagCode" = 0x80	Check the signal connected to the "S_ValveStatus8" input parameter.

Table 574: Diagnostic codes

6.5.14.5.7 AdditionalDiagCode

General function

- Additional diagnostic message from the function block to indicate which valve is enabled or causing an error

Data type

- BYTE

Connection

- Variable

Function description

This output parameter is used to output additional specific diagnostic messages from the function block. It indicates which valves are enabled and which are showing an error.

> ## Information:
> **Each valve is represented by one bit.**

6.5.14.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

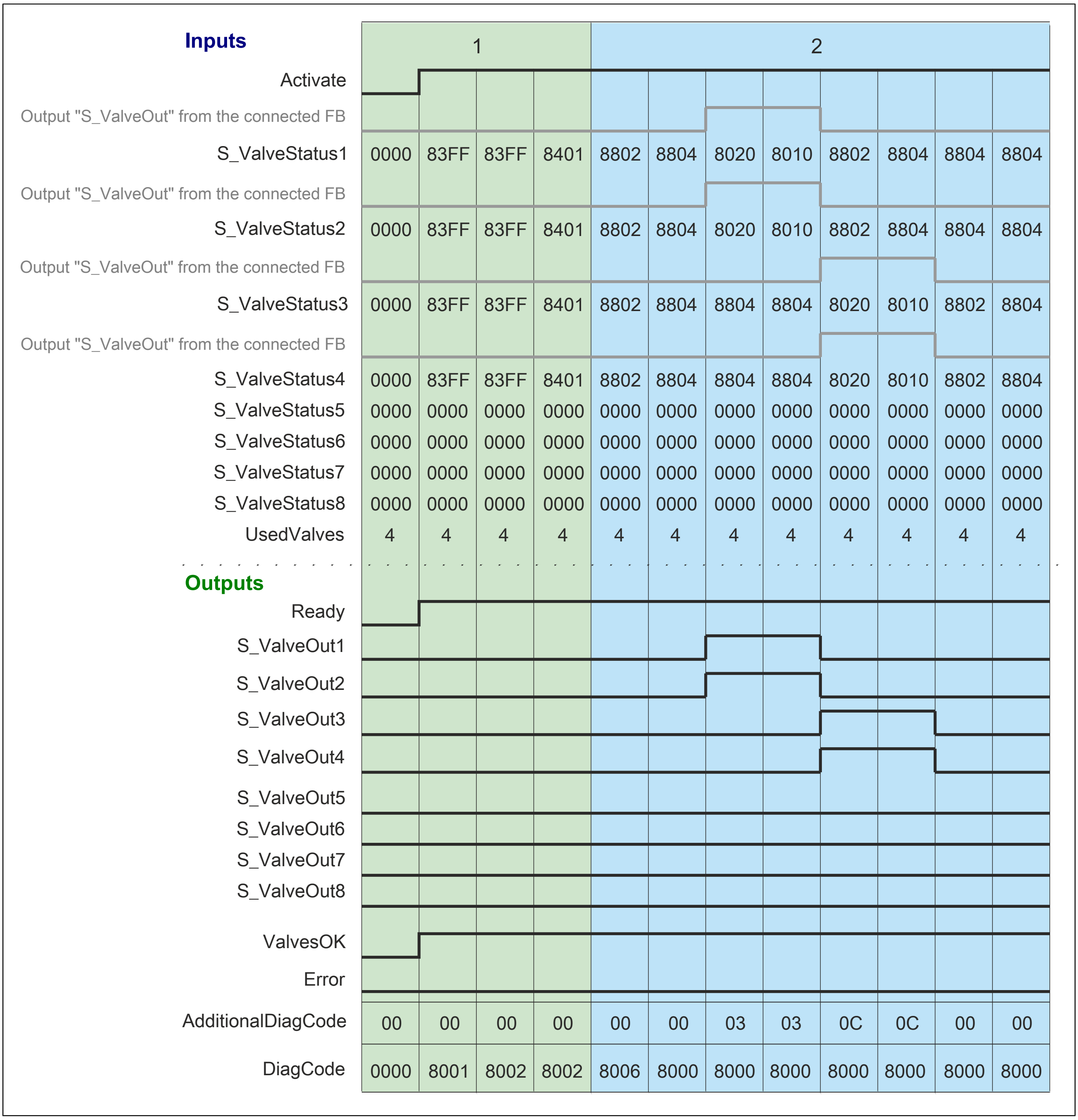


Figure 394: "SF_ValveGroupControl": Signal sequence diagram 1

1 Initialization
2 Normal operation with four valves

Signal sequence diagram 2

Information:

The "SafetyDemand" and "ResetRequest" outputs are not listed in the signal sequence diagrams.

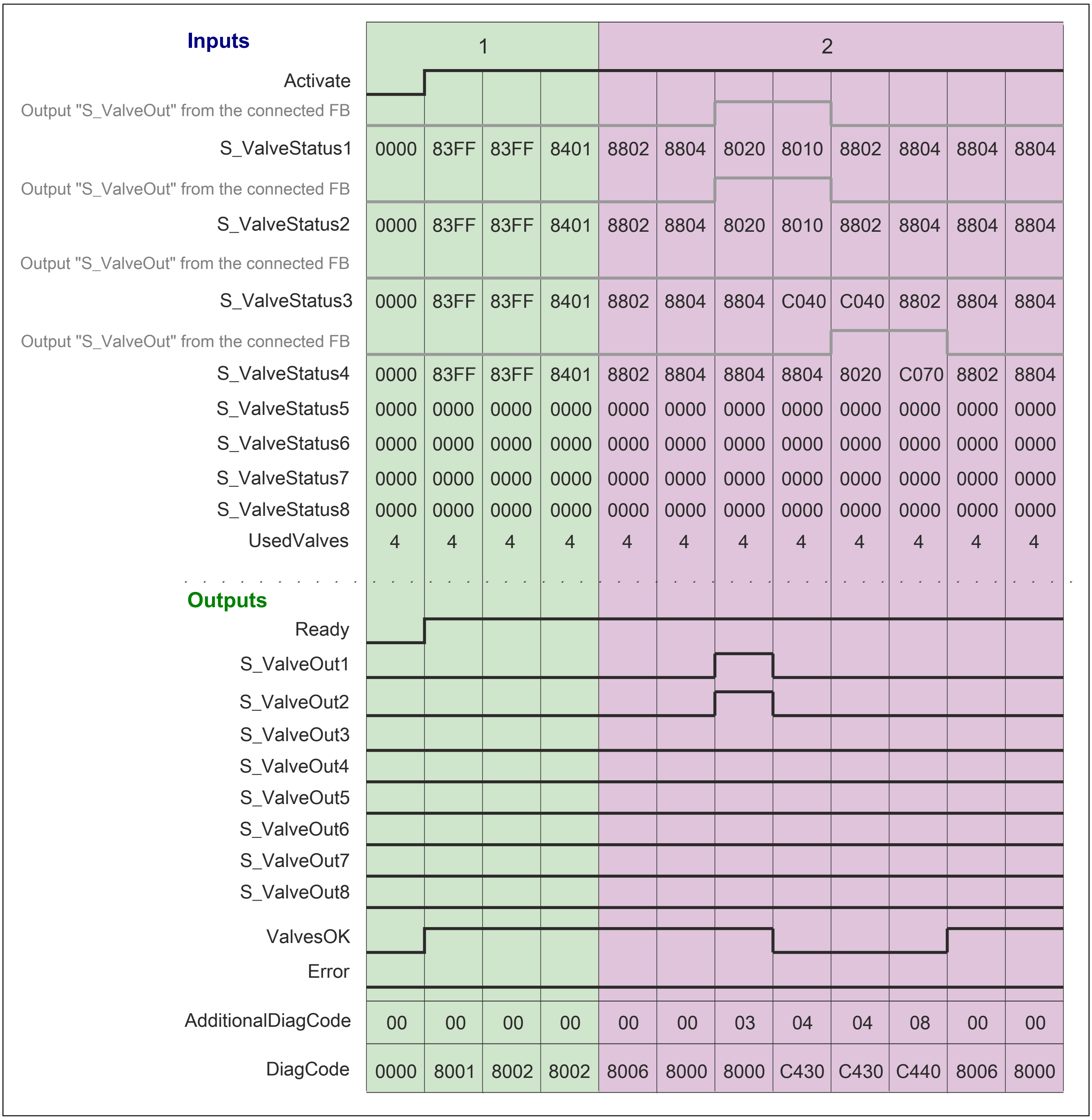


Figure 395: "SF_ValveGroupControl": Signal sequence diagram 2

1 Initialization
2 Error - 2 valves are reporting an error

6.5.15 Version history

Version	Date	Comment
1.00	February 2016	First edition

Table 575: Version history

6.6 PLCopen_SF

6.6.1 Overview of PLCopen function blocks

The function blocks for safety-oriented applications standardized in the PLCopen package have revolutionized the development of safety applications. Because they are certified, they save time and reduce costs throughout all phases of a safety application's lifecycle. From specification and implementation to testing and verification functions, the approach is more like virtual wiring than programming.

Unlike "real wiring", downloading the program to the safety controller guarantees that an identical copy will be stored. This completely eliminates wiring errors during series production. To be sure, all options for a safety controller are available to handle even more complex challenges that cannot be solved with "real wiring".

Actuator connections

Function block	Description
SF_OutControl	Controls an actuator with restart interlock
SF_EDM	Controls an actuator with evaluation of feedback signals
SF_SafetyRequest	General safety request with state monitoring
SF_GuardLocking	Controls a safety door with guard locking device

Sensor connections

Function block	Description
SF_Equivalent	1-out-of-2 evaluation of 2 equivalent contacts (N.C. / N.C. or N.O. / N.O.) with discrepancy time monitoring
SF_Antivalent	1-out-of-2 evaluation of 2 antivalent contacts (N.C. / N.O.) with discrepancy time monitoring
SF_ModeSelector	Mode selector switch (1-out-of-8 (max.) evaluation) with discrepancy time monitoring
SF_EmergencyStop	Emergency switch-off evaluation with restart interlock
SF_ESPE	Evaluates an ESPE signal with restart interlock
SF_TwoHandControlTypeII	Evaluates a two-hand control device without monitoring of simultaneous actuation
SF_TwoHandControlTypeIII	Evaluates a two-hand control device with monitoring of simultaneous actuation
SF_GuardMonitoring	Safety door monitoring with discrepancy time monitoring and restart interlock
SF_TestableSafetySensor	Tests ESPE with restart interlock
SF_EnableSwitch	Evaluates an enable switch with restart interlock

Muting

Function block	Description
SF_MutingSeq	Muting with sensors arranged sequentially
SF_MutingPar	Muting with sensors arranged in parallel
SF_MutingPar_2Sensor	Muting with safety sensors arranged in parallel
SF_Override	Implementation of the override function. This makes it possible to remove blockage that occurs in the safety zone during the muting process.

6.6.2 System requirements

The PLCopen library is part of SafeDESIGNER and only permitted to be used therein.

There are no specific requirements for using the PLCopen library.

6.6.3 Term definitions

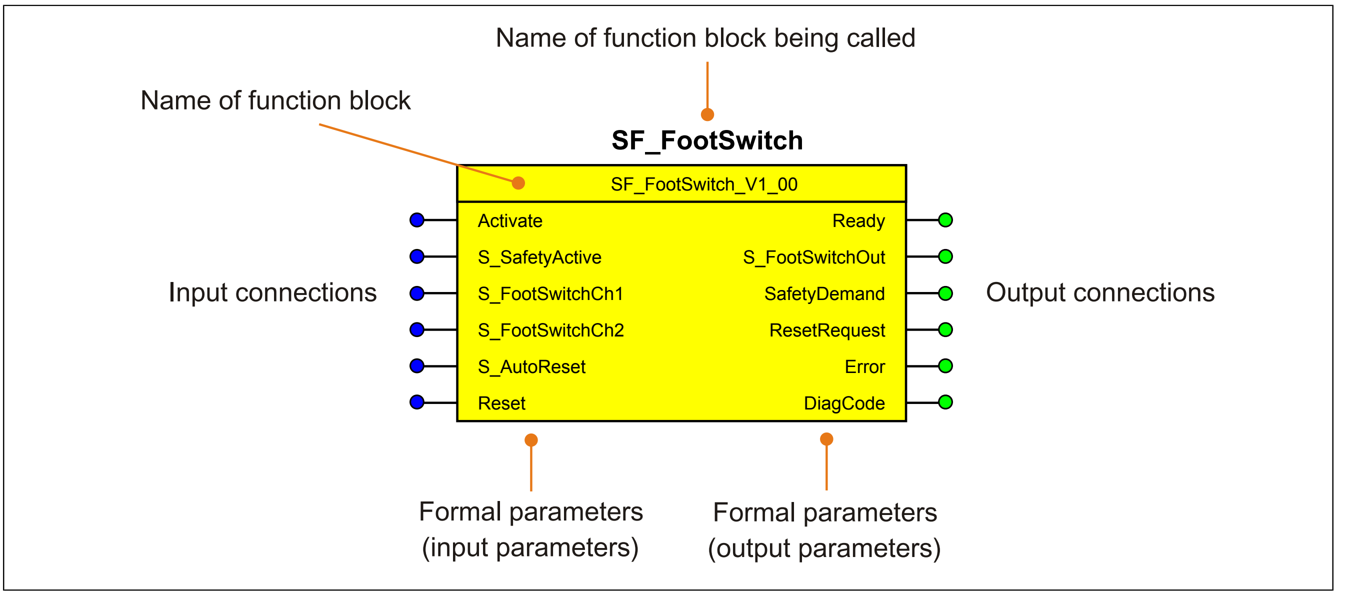


Figure 396: Components of a function block

When calling a function block, the inputs supply the input parameters with the current values of the variables or constants.
The output parameters supply the outputs with the associated values.

Inputs and outputs do not need to have the same name as the associated formal parameters, but they must be of the same data type. A difference in data type between formal parameters and inputs/outputs is reported as an error after compilation.

The name of a function block is composed of the function itself (e.g. "SF_FootSwitch", SF = safety function) and its version (Vx_yz). The format used to represent the version number in this document, Vx_yz, is a placeholder. For the actual version, see the function block being used.

6.6.4 SF_Antivalent

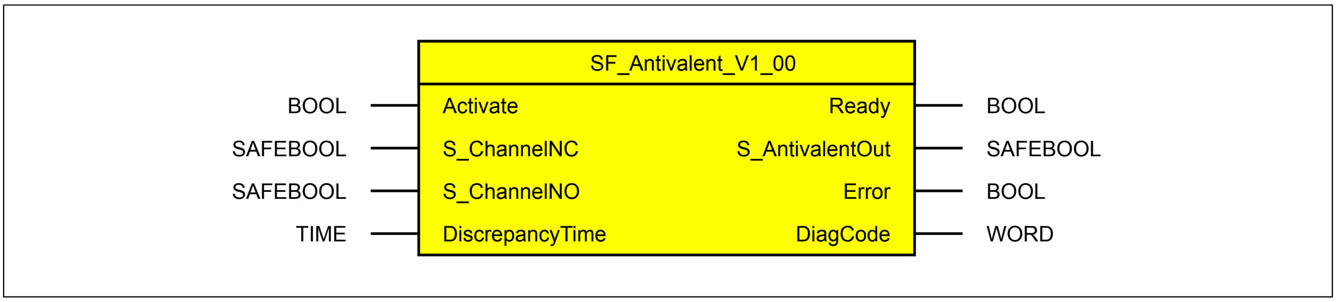


Figure 397: Function block "SF_Antivalent"

6.6.4.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_ChannelNC	SAFEBOOL	Variable	State	FALSE	NC input (normally closed)
S_ChannelNO	SAFEBOOL	Variable	State	FALSE	NO input (normally open)
DiscrepancyTime	TIME	Constant	State	#0ms	Specification for the monitoring time of the switching operation on "S_ChannelNC" and "S_ChannelNO"

Table 576: "SF_Antivalent": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_AntivalentOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 577: "SF_Antivalent": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 578: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.4.2 Function

Function block "SF_Antivalent" supports a monitoring function with regard to the antivalence of 2 safe signal inputs.

This function block compares 2 safe signal inputs of a sensor connected over 2 channels for antivalent signal states. Discrepancy time monitoring starts after the state of one of the signal inputs has changed (input parameter "DiscrepancyTime").

During this time, the function block monitors whether a switching operation has taken place on the other signal input (switching function) and whether both signal inputs are switching symmetrically (switching symmetry).

If the signal switches for both signal inputs within the discrepancy time, then discrepancy time monitoring ends. An error message is reported if the discrepancy time is exceeded.

The two signal inputs are interdependent. For this reason, either wire the NC signal input with a normally closed contact and the NO signal input with a normally open contact, or vice versa.

The result of the comparison is indicated on function block enable output "S_AntivalentOut". If the NC signal input switches from TRUE to FALSE and/or the NO signal input switches from FALSE to TRUE, then the enable output immediately switches to FALSE.

6.6.4.2.1 Discrepancy time monitoring

During the discrepancy time, both signal inputs are permitted to have the same state. The function block does not detect this as an error.

Discrepancy time monitoring begins when the state of one of the signal inputs changes. The function block detects an error if both signal inputs have the same state after the discrepancy time elapses.

6.6.4.2.2 Switching states

Monitoring watches both signal inputs for a change from FALSE to TRUE as well as a change from TRUE to FALSE. For this reason, the signal inputs must be switched symmetrically.

6.6.4.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.4.3.1 Asymmetrical switching signals on signal inputs NC and NO

If the signals on signal inputs NC and NO are switched asymmetrically (due to faulty protective equipment, faulty switch, faulty cables, etc.), then the safe state (enable output "S_AntivalentOut" = FALSE) will be maintained once the protective equipment has been closed.

6.6.4.3.2 Same states on signal inputs NC and NO

The same states on signal inputs NC and NO (due to faulty protective equipment, faulty switch, faulty cables, etc.) after the specified discrepancy time has expired (input parameter "DiscrepancyTime") is reported as a fault by the function block.

6.6.4.3.3 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.4.3.4 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.4.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.4.4 Input parameters

6.6.4.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.4.4.2 S_ChannelNC

General function

- NC input (normally closed)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal (NC channel) from a safety sensor connected over 2 channels. Input parameter "S_ChannelNC" is then controlled via this signal.

The following combinations are allowed with interconnected signal inputs:

- **"S_ChannelNC": Normally closed contact and "S_ChannelNO": Normally open contact, or**
- **"S_ChannelNC": Normally open contact and "S_ChannelNO": Normally closed contact**

Information:

Connect a normally closed contact if you want to generate a TRUE signal on "S_AntivalentOut" when the contacts are not actuated ("idle current" principle). If you want to generate a FALSE signal on "S_AntivalentOut", connect a normally open contact.

Function description

The signal connected to input parameter "S_ChannelNC" is processed by the function block.

Signal input NC processes the state of a signal (NC channel) from a safety sensor connected over 2 channels. This signal must indicate state TRUE in order to set enable output "S_AntivalentOut" to TRUE under consideration of the state on "S_ChannelNO".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

If a normally closed contact is connected, it is not actuated.
If a normally open contact is connected, it is actuated.

FALSE

If a normally closed contact is connected, it is actuated.
If a normally open contact is connected, it is not actuated.

6.6.4.4.3 S_ChannelNO

General function

- NO input (normally open)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal (NO channel) from a safety sensor connected over 2 channels. Input parameter "S_ChannelNO" is then controlled via this signal.

The following combinations are allowed with interconnected signal inputs:

- **"S_ChannelNC": Normally closed contact and "S_ChannelNO": Normally open contact, or**
- **"S_ChannelNC": Normally open contact and "S_ChannelNO": Normally closed contact**

Information:

Connect a normally open contact if you want to generate a TRUE signal on "S_AntivalentOut" when the contacts are not actuated ("idle current" principle). If you want to generate a FALSE signal on "S_AntivalentOut", connect a normally closed contact.

Function description

The signal connected to input parameter "S_ChannelNO" is processed by the function block.

Signal input NO processes the state of a signal (NO channel) from a safety sensor connected over 2 channels. This signal must indicate state FALSE in order to set enable output "S_AntivalentOut" to TRUE under consideration of the state on "S_ChannelNC".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

If a normally closed contact is connected, it is not actuated.
If a normally open contact is connected, it is actuated.

FALSE

If a normally closed contact is connected, it is actuated.
If a normally open contact is connected, it is not actuated.

6.6.4.4.4 DiscrepancyTime

General function

- Specification for the monitoring time of the switching operation on "S_ChannelNC" and "S_ChannelNO"

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the time within which one of the two signal inputs can change its state while the state of the other signal input is not changed without the function block detecting a fault.

A signal tolerance during the switching operation of the contacts is possible due to mechanical influences. The specified "DiscrepancyTime" value limits the duration within which the two signals can be switched successively without this being detected as not simultaneous.

The limits for input parameter "DiscrepancyTime" must be defined and validated based on your application.

6.6.4.5 Output parameters

6.6.4.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.4.5.2 S_AntivalentOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

Release signal "S_AntivalentOut" controls the safety request of downstream safety-relevant function blocks such as "SF_EmergencyStop".

The release signal is controlled based on the state of the safety sensor. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_AntivalentOut", this output is referred to as the "enable output".

> **Danger!**
>
> **The release signal is not permitted to control the process directly since this function block supports monitoring, not a safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The function block has detected the antivalence of both signal inputs.
- "S_ChannelNC" = TRUE and "S_ChannelNO" = FALSE.
- The function block did not detect any faults.

FALSE

The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block has detected no antivalence of both signal inputs.
- "S_ChannelNC" = FALSE and/or "S_ChannelNO" = TRUE.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

6.6.4.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

> ## Information:
>
> **In order to exit an error state ("Error" = TRUE), "S_ChannelNO" must be TRUE and "S_ChannelNC" must be FALSE.**

6.6.4.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.4.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	Initialization of the function block after the function block has been enabled and/or both signal inputs NC and NO have the opposite state of state definition NC and NO (NC = FALSE and NO = TRUE). If one or both input signals indicate the state corresponding to the definition of NC and NO, then this state is completed after one cycle of the safety controller.	• If the signal combination on the signal inputs is intended, set both input signals (NC and NO) of the connected peripheral to the defined state (NC = TRUE and NO = FALSE) to set the enable output to TRUE. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8004	Both signal inputs NC and NO indicate state TRUE. Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set signal input NO to state FALSE. • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
8005	Only one signal input (NC or NO) indicates the state opposite to the definition of NC and NO (NC = FALSE or NO = TRUE). Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set the other signal input to the state opposite to the definition (NC = FALSE or NO = TRUE). • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
8014	Both signal inputs NC and NO indicate state FALSE. Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set signal input NC to state TRUE. • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
C001	On a request to set the enable output to TRUE, the corresponding signal was only present on the NC signal input. Discrepancy time monitoring was started and has expired. When the discrepancy time expired, the NC signal input switched while the NO signal input did not.	• Check the connected peripheral (signal input NO) and correct any faults. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set the NC signal input to FALSE and the NO signal input to TRUE.
C002	On a request to set the enable output to TRUE, the corresponding signal was only present on the NO signal input. Discrepancy time monitoring was started and has expired. When the discrepancy time expired, the NO signal input switched while the NC signal input did not.	• Check the connected peripheral (signal input NC) and correct any faults. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set the NC signal input to FALSE and the NO signal input to TRUE.
C003	Both signal inputs did not change their state after the enable output was cut off. The discrepancy time started in this state is expired.	• Check the connected peripheral (signal input NC and signal input NO) and correct any faults. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set the NC signal input to FALSE and the NO signal input to TRUE.

Table 579: "SF_Antivalent": Diagnostic codes

6.6.4.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

Startup, normal operation

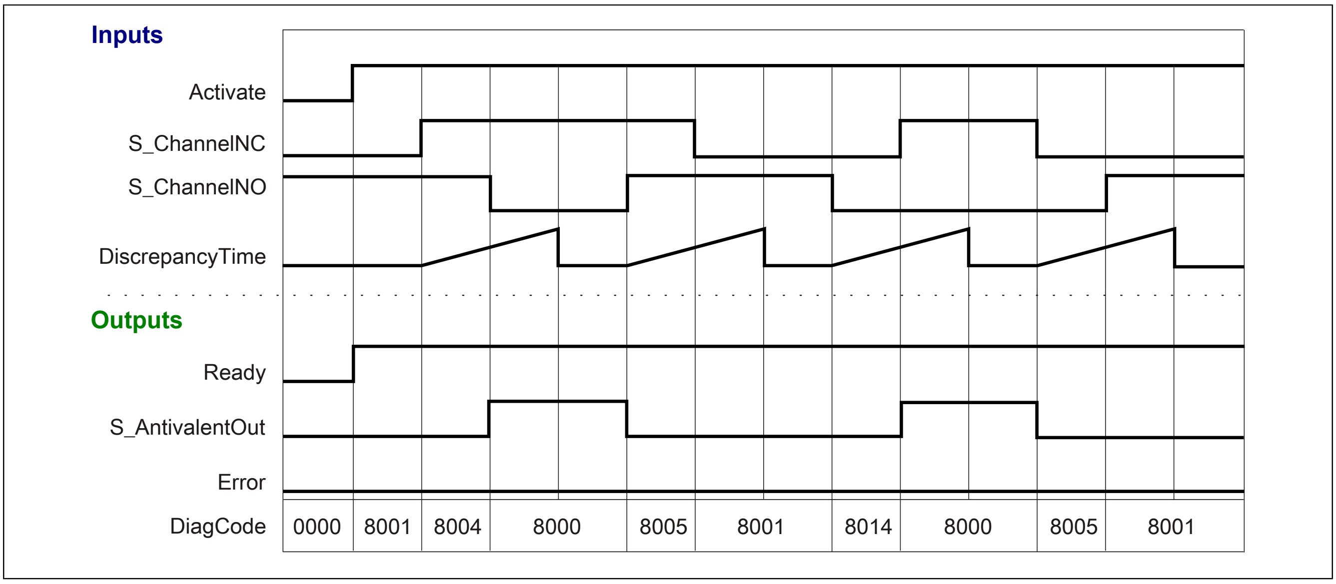


Figure 398: "SF_Antivalent": Signal sequence diagram 1

Signal sequence diagram 2

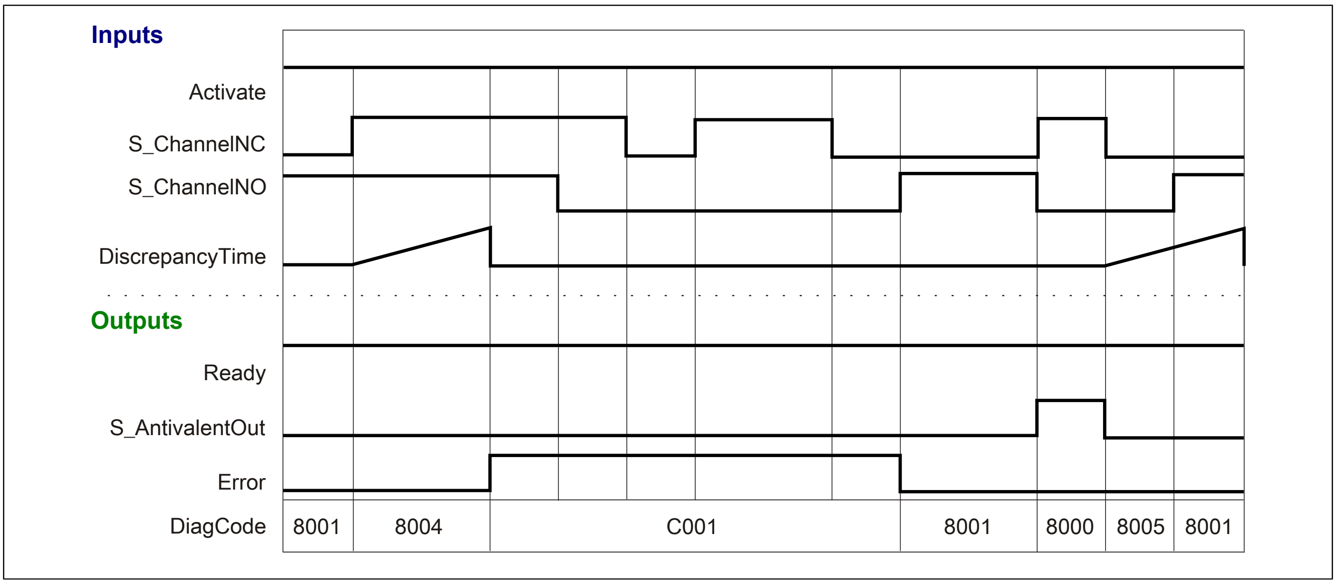


Figure 399: "SF_Antivalent": Signal sequence diagram 2

6.6.4.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement antivalent evaluation of signals.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.4.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "AV_S5".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

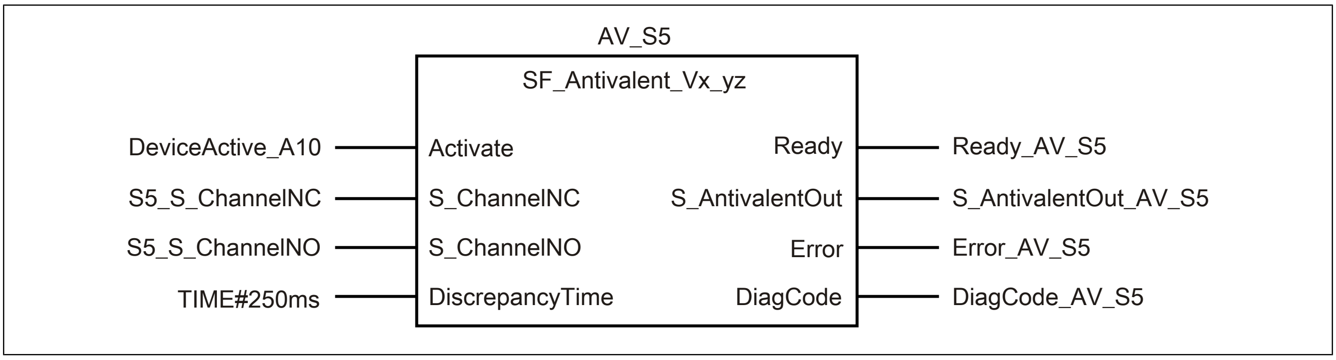


Figure 400: "SF_Antivalent": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the contacts of the safety switch are connected.
S5_S_ChannelNC	SAFEBOOL	NC input (normally closed). This input is connected to NC of a safety switch.
S5_S_ChannelNO	SAFEBOOL	NO input (normally open). This input is connected to NO of a safety switch.
TIME#250ms on "DiscrepancyTime"	TIME	Maximum permitted discrepancy time

Table 580: "SF_Antivalent": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_AV_S5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_AntivalentOut_AV_S5	SAFEBOOL	Release signal. The release signal is the resulting signal of both input signals and used for further processing in the safety application on the safety controller.
Error_AV_S5	BOOL	Error message from function block for further external processing
DiagCode_AV_S5	WORD	Diagnostic message from function block for further external processing

Table 581: "SF_Antivalent": Outputs connected to output parameters

6.6.4.7.2 Evaluation of 2 antivalent safe input signals

This example illustrates the connection of the function block when controlling the 1-channel antivalent signals of a safe input device.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.4.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "AV_S5".

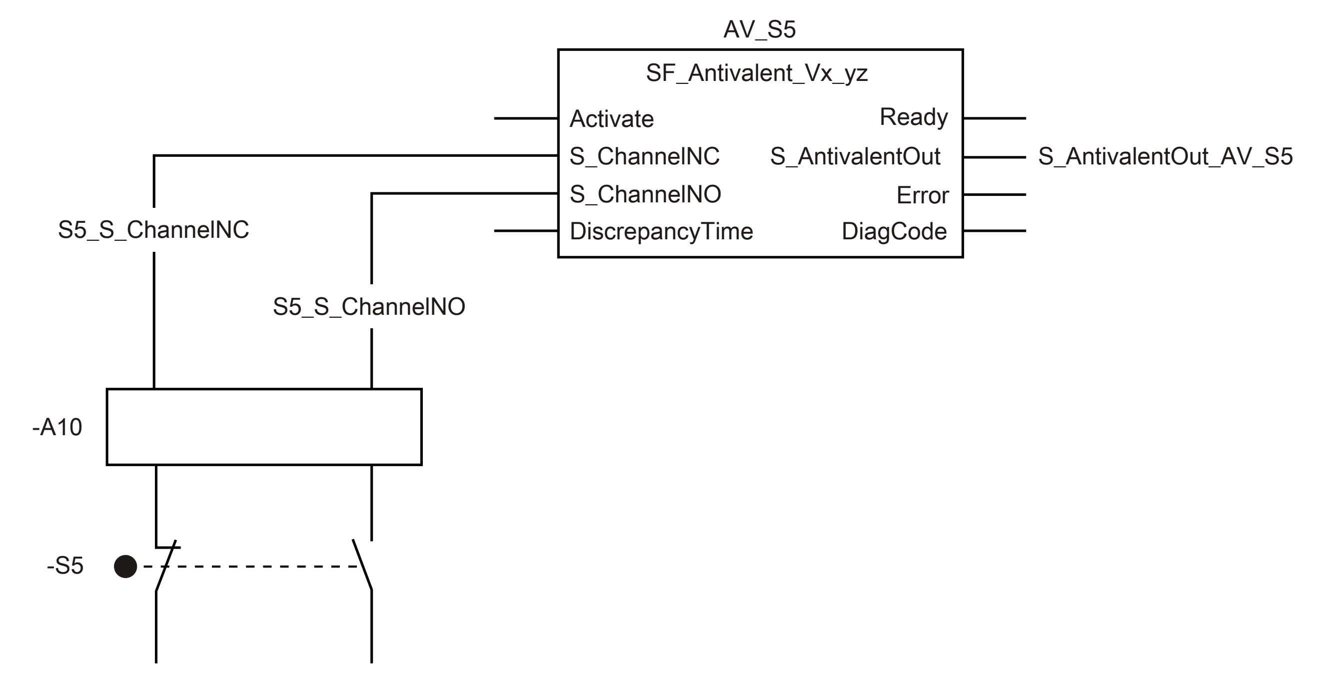


Figure 401: "SF_Antivalent": Evaluation of 2 antivalent safe input signals (1 input device)

List of equipment

-S5	Safety switch, 1x normally closed contact and 1x normally open contact (positively driven)
-A10	2x 1-channel safe inputs of a safe input device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S5_S_ChannelNC	Input on "S_ChannelNC"
S5_S_ChannelNO	Input on "S_ChannelNO"
S_AntivalentOut_AV_S5	Output on "S_AntivalentOut"

Description

In this example:

- The normally closed signal from the NC input of safe input device "-A10" is connected to input "S5_S_ChannelNC" and connected to function block input parameter "S_ChannelNC" for further processing.
- The normally open signal from input NO of safe input device "-A10" is connected to input "S5_S_ChannelNO" and connected to function block input parameter "S_ChannelNO" for further processing.
- Output parameter "S_AntivalentOut" is connected to output "S_AntivalentOut_AV_S5".
- Output "S_AntivalentOut_AV_S5" is the resulting signal of both input signals and used for further processing in the safety application on the safety controller.

6.6.4.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 954-1	General safety guidelines	Monitoring of antivalent switching behavior of 2 signals
EN 954-1	Error detection for CAT 3 and CAT 4	Monitoring of antivalent switching behavior of 2 signals

Table 582: "SF_Antivalent": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.5 SF_EDM

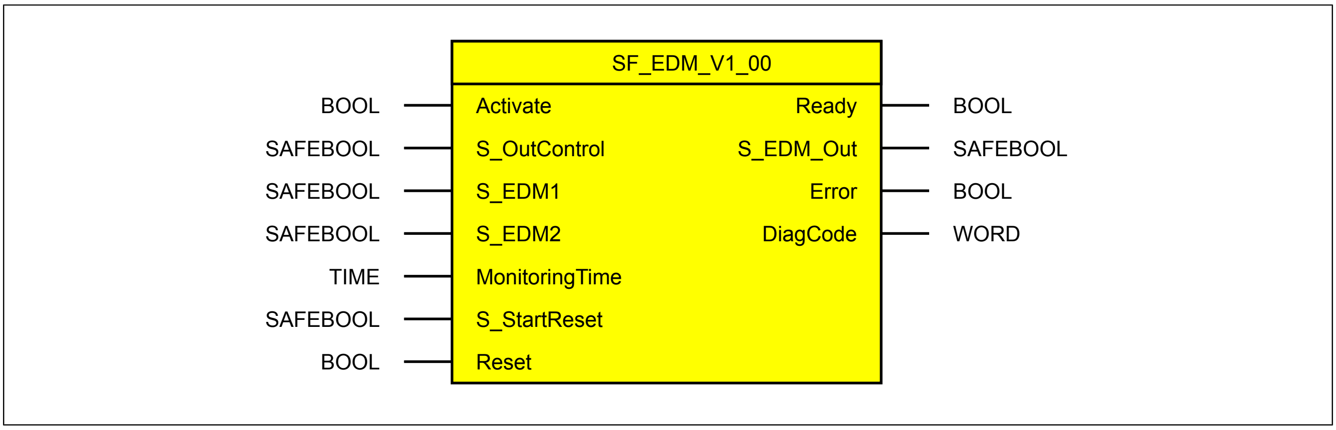


Figure 402: Function block "SF_EDM"

6.6.5.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_OutControl	SAFEBOOL	Variable	State	FALSE	Control signal. State of the connected safety function
S_EDM1	SAFEBOOL	Variable	State	FALSE	Input for feedback signal from switching amplifier 1
S_EDM2	SAFEBOOL	Variable	State	FALSE	Input for feedback signal from switching amplifier 2
MonitoringTime	TIME	Constant	State	#0ms	Specification for the monitoring time of the switching operation on "S_EDM1" and "S_EDM2"
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 583: "SF_EDM": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_EDM_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 584: "SF_EDM": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 585: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.5.2 Function

Function block "SF_EDM" checks the defined basic state and dynamic switching state of switching amplifiers that are controlled by safe output devices.

The outputs of safe devices that are used to control the switching amplifiers to be monitored are energized to do so by the enable output "S_EDM_Out" from this function block. The states of the connected switching amplifiers are fed back to the function block via safe input devices.

In the basic state, the fed back signals from the switching amplifiers being monitored must indicate state TRUE. In the switching state, the fed back signals from the switching amplifiers being monitored must indicate state FALSE after the specified time window expires (input parameter "MonitoringTime").

6.6.5.2.1 Monitoring the basic state

If the function block detects the basic state of the switching amplifiers (TRUE on "S_EDM1" and "S_EDM2") via the feedback signals present on its inputs during a switch-on request, then the function block will set its enable output ("S_EDM_Out") to TRUE and switch on the switching amplifiers.

If the basic state (TRUE) is not detected by the function block during a switch-on request, then the function block will set its enable output to FALSE and will not switch on the switching amplifiers.

6.6.5.2.2 Monitoring the switching state

After energizing / switching on the switching amplifiers, the function block checks the reaction of the switching amplifiers within a user-specified time window (input parameter "MonitoringTime"). If the function block does not detect the states of the feedback signals defined for the switching state (FALSE on "S_EDM1" and "S_EDM2"), then the function block will set its enable output ("S_EDM_Out") to FALSE and cut off the switching amplifiers again.

> **Danger!**
>
> **Select the switching amplifiers that you will use within the safety function according to your risk analysis.**

6.6.5.2.3 Switching amplifier control (1-channel and 2-channel)

Depending on the risk analysis and the resulting safety requirements of an application, 1- or 2-channel control must be configured for the switching amplifiers.

2-channel application / Safety function up to CAT 4

Feedback when controlling 2 switching amplifiers to be monitored

When feeding back 2 signals from switching amplifiers being monitored, each signal is connected via a safe input device to an intended function block input parameter. This makes it possible to perform exact channel-by-channel diagnostics for the connected switching amplifiers. In addition, it is possible when feeding back 2 signals for the function block to detect faults on the connected peripheral (e.g. feedback contact bypassed or decoupled from the switching amplifier).

When a signal from the switching amplifiers being monitored is fed back, the feedback contacts are connected in series. The signal resulting from the series connection is connected to the 2 intended function block input parameters ("S_EDM1" and "S_EDM2") via a safe input device. Both input parameters are connected via a graphical connection; as a result, they are set by one signal. Channel-by-channel diagnostics of the connected switching amplifiers is not possible with this connection. In addition, it is not possible for the function block to detect faults on the connected peripheral (e.g. feedback contact bypassed or decoupled from the switching amplifier).

1-channel application / Safety function up to CAT 2

Feedback when controlling 1 switching amplifier to be monitored

When feeding back the signal from 1 switching amplifier being monitored, the signal is connected via a safe input device to the two intended function block input parameters ("S_EDM1" and "S_EDM2"). Both input parameters are connected via a graphical connection; as a result, they are set by one signal.

If the function block is meant to be used with electromagnetic switching amplifiers for diagnostic purposes, then the switching amplifiers must be equipped with an auxiliary contact. To ensure unambiguous diagnostics, this auxiliary contact must be connected to the load contacts in a positively driven manner. This auxiliary contact is connected singly (1-channel) to a safe input of a safe device.

6.6.5.2.4 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.5.2.5 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.5.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.5.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.5.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret the signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Information:

Static TRUE signals on the formal parameters of the feedback signals for monitored switching amplifiers are detected by the function block as errors after a time window specified by the user has elapsed.

A static TRUE signal on the formal parameters of the feedback signals (caused by the removal of a contact block, jumpering of a contact, and/or a faulty contact, etc.) is detected by the function block as an error after the specified time window has elapsed for a 2-channel application if both feedback signals are connected to the function block individually.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller
- Contact block removed from the switching amplifier (user error)
- Sticking contacts (hardware fault)

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.5.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the status of the release signal depends on the status of the feedback signal(s) connected to the function block as well as the state of the upstream safety function ("S_OutControl").

6.6.5.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.5.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.5.4 Input parameters

6.6.5.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.5.4.2 S_OutControl

General function

- Control signal. State of the connected safety function.

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Please note that no hardware inputs are connected to input parameter "S_OutControl".**
>
> **You will usually connect "S_OutControl" to a release signal of a safe function block (e.g. "SF_TwoHandControlTypeII", "SF_OutControl").**
>
> **The state on input parameter "S_OutControl" corresponds to the state of the upstream function block / safety function (e.g. "SF_TwoHandControlTypeII", "SF_OutControl").**

Function description

The signal input processes the state of the safety function.

The state of "S_OutControl" controls the "S_EDM_Out" enable output under consideration of the states of the connected feedback signals on "S_EDM1" and "S_EDM2".

Input parameter "S_OutControl" is typically set by an upstream safety function, such as "emergency switch-off" or "2-hand control". In addition, a signal from the standard controller can be involved in the control process in order to set an output being monitored by the function block. The signal from the standard controller must be connected to signals from safety functions such as "emergency switch-off" (e.g. with "SF_OutControl") outside of the function block using an AND operator. The expression result then sets "S_OutControl".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream safety function was not triggered. A signal from the standard controller possibly involved in the process is indicating state TRUE (start of operation).

Output parameter "S_EDM_Out" is set to TRUE if the remaining input signal combination is valid for this; output parameter "DiagCode" is set accordingly.

FALSE

The upstream safety function was triggered or the safe device connected for this safety function is shut down or defective. A signal from the standard controller possibly involved in the process is indicating state FALSE (operation stop).

Output parameter "S_EDM_Out" is set to FALSE; output parameter "DiagCode" is set accordingly.

6.6.5.4.3 S_EDM1

General function

- Input for feedback signal from switching amplifier 1

Data type

- SAFEBOOL

> ## Information:
>
> **The achievable performance level of the implemented safety function is independent of the connected signal type (BOOL or SAFEBOOL). In this case, conversion block "BOOL_TO_SAFEBOOL" can be used without having to implement any additional measures.**
>
> **The safety function itself must be implemented using safety modules for this.**

Connection

- Variable

> ## Information:
>
> **Connect this input parameter with the input signal of a safe device to which the feedback signal of the switching amplifier being monitored is connected.**
>
> **If you are using feedback with only one signal (1-channel safety function or series connection with a 2-channel safety function), connect "S_EDM1" and "S_EDM2" using a graphical connection.**
>
> **Please see the information in section 6.6.5.3 "Fault avoidance" for applications with only one feedback signal. This information must be taken into account in the risk analysis, which is your responsibility.**

Function description

Using this signal, the function block monitors the basic state and switching states of connected switching amplifier 1.

TRUE

The feedback signal is present (switching amplifier 1 is de-energized / basic state).

In order to set enable output "S_EDM_Out" to TRUE, "S_EDM1" must also indicate state TRUE (basic state of switching amplifier 1).

Otherwise, an error message is generated and indicated by output parameter "DiagCode". Enable output "S_ED-M_Out" is set to FALSE.

FALSE

The feedback signal is not present (switching amplifier 1 is energized).

After "S_EDM_Out" is set to TRUE, state FALSE must be present on "S_EDM1" after the time specified on input parameter "MonitoringTime" has expired.

Switching states of the switching amplifier with release signal "S_EDM_Out"

"Activate"	= TRUE
"S_EDM_Out"	= FALSE → TRUE / permanently TRUE

After a signal change from FALSE to TRUE on "S_EDM_Out", input "S_EDM1" must indicate state FALSE within the time specified on input parameter "MonitoringTime". Otherwise, an error message is generated and indicated by output parameter "DiagCode". Enable output "S_EDM_Out" is set to FALSE.

6.6.5.4.4 S_EDM2

General function

- Input for feedback signal from switching amplifier 2

Data type

- SAFEBOOL

> ## Information:
>
> **The achievable performance level of the implemented safety function is independent of the connected signal type (BOOL or SAFEBOOL). In this case, conversion block "BOOL_TO_SAFEBOOL" can be used without having to implement any additional measures.**
>
> **The safety function itself must be implemented using safety modules for this.**

Connection

- Variable

> ## Information:
>
> **Connect this input parameter with the input signal of a safe device to which the feedback signal of the switching amplifier being monitored is connected.**
>
> **If you are using feedback with only one signal (1-channel safety function or series connection with a 2-channel safety function), connect "S_EDM1" and "S_EDM2" using a graphical connection.**
>
> **Please see the information in section 6.6.5.3 "Fault avoidance" for applications with only one feedback signal. This information must be taken into account in the risk analysis, which is your responsibility.**

Function description

Using this signal, the function block monitors the basic state and switching states of connected switching amplifier 2.

TRUE

The feedback signal is present (switching amplifier 2 is de-energized / basic state).

In order to set enable output "S_EDM_Out" to TRUE, "S_EDM2" must also indicate state TRUE (basic state of switching amplifier 2).

Otherwise, an error message is generated and indicated by output parameter "DiagCode". Enable output "S_EDM_Out" is set to FALSE.

FALSE

The feedback signal is not present (switching amplifier 2 is energized).

After "S_EDM_Out" is set to TRUE, state FALSE must be present on "S_EDM2" after the time specified on input parameter "MonitoringTime" has expired.

Switching state of the switching amplifier with release signal "S_EDM_Out"

"Activate"	= TRUE
"S_EDM_Out"	= FALSE → TRUE / permanently TRUE

After a signal change from FALSE to TRUE on "S_EDM_Out", input "S_EDM2" must indicate state FALSE within the time specified on input parameter "MonitoringTime". Otherwise, an error message is generated and indicated by output parameter "DiagCode". Enable output "S_EDM_Out" is set to FALSE.

6.6.5.4.5 MonitoringTime

General function

- Specification for the monitoring time of the switching operation on "S_EDM1" and "S_EDM2"

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the time within which the switching operations on signal inputs "S_EDM1" and "S_EDM2" must take place in order to be detected as valid.

The limits for input parameter "MonitoringTime" must be defined and validated based on your application.

6.6.5.4.6 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.5.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.5.5 Output parameters

6.6.5.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.5.5.2 S_EDM_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe signal used to control an output on a safe device, therefore also controlling the process.

The release signal is controlled depending on the state of the connected switching amplifier(s) as well as the start interlock of the upstream safety functions. If necessary, the release signal is controlled by the state of a signal from the standard controller if the signal is connected accordingly outside the function block and connected to "S_OutControl".

In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_EDM_Out", this output is referred to as the "enable output".

TRUE

The output of a safe device is set to TRUE. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The safety request has not been triggered ("S_OutControl" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The output of a safe device is set to FALSE. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- Or enabling the function block (depending on reset mode) was detected ("Activate" = FALSE → TRUE).
- The function block detected a safety request ("S_OutControl" = FALSE).
- The function block detected a cold restart of the safety controller (depending on the reset mode).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_EDM_Out" is set to TRUE only if input "S_OutControl" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_EDM_Out" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 586: "SF_EDM": Input parameter "S_StartReset"

6.6.5.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.5.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.5.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• No corrective measures are necessary if the signal is intended. • If the signal is unintended, check the program code and/or the "S_OutControl" control signal.
8001	The start interlock to be carried out after enabling the function block is specified ("S_StartReset" = FALSE). This state is completed after a safety controller cycle if "S_StartReset" = TRUE. The start interlock is no longer active after a safety controller cycle.	Reset the function block.
8010	No request is present on "S_OutControl" for setting enable output "S_EDM_Out" to TRUE.	"S_OutControl" must indicate state TRUE in order to set the enable output to TRUE.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	On the request to switch on the connected switching amplifiers, the switching amplifier signal looped back on "S_EDM1" was faulty. The function block detected "S_EDM1" = FALSE when checking the basic state of the switching amplifiers being controlled.	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C011	On the request to switch on the connected switching amplifiers, the switching amplifier signal looped back on "S_EDM1" was faulty. The function block detected "S_EDM1" = FALSE when checking the basic state of the switching amplifiers being controlled. At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C020	On the request to switch on the connected switching amplifiers, the switching amplifier signal looped back on "S_EDM2" was faulty. The function block detected "S_EDM2" = FALSE when checking the basic state of the switching amplifiers being controlled.	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C021	On the request to switch on the connected switching amplifiers, the switching amplifier signal looped back on "S_EDM2" was faulty. The function block detected "S_EDM2" = FALSE when checking the basic state of the switching amplifiers being controlled. At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.

Table 587: "SF_EDM": Diagnostic codes

Code (hex)	Description	Corrective measures
C030	On the request to switch on the connected switching amplifiers, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty. The function block detected "S_EDM1" = FALSE and "S_EDM2" = FALSE when checking the basic state of the switching amplifiers being controlled.	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C031	On the request to switch on the connected switching amplifiers, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty. The function block detected "S_EDM1" = FALSE and "S_EDM2" = FALSE when checking the basic state of the switching amplifiers being controlled. At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C040	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM1" was faulty (FALSE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C041	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM1" was faulty (FALSE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C050	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM2" was faulty (FALSE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C051	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM2" was faulty (FALSE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.

Table 587: "SF_EDM": Diagnostic codes

Code (hex)	Description	Corrective measures
C060	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty (FALSE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C061	When checking the basic state of the cut off, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty (FALSE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C070	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM1" was faulty (TRUE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.

Table 587: "SF_EDM": Diagnostic codes

Code (hex)	Description	Corrective measures
C071	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM1" was faulty (TRUE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C080	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM2" was faulty (TRUE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C081	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signal looped back on "S_EDM2" was faulty (TRUE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C090	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty (TRUE).	• Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C091	When checking the switching state of the switched on, connected switching amplifiers after the value specified on "MonitoringTime" has elapsed, the switching amplifier signals looped back on "S_EDM1" and "S_EDM2" were faulty (TRUE). At the same time, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the connected switching amplifiers. • If necessary, correct the error. • Both feedback signals from the cut-off, connected switching amplifiers must indicate state TRUE on the function block. • Reset the function block. This error message is not permitted in an application with only one feedback signal. • Check and, if necessary, correct the connection of the input parameters in your application.
C111	The function block detected a simultaneous signal change from FALSE to TRUE on "S_OutControl" and "Reset".	• Check the connection on "S_OutControl" and "Reset". "Reset" and "S_OutControl" must be controlled from 2 different signal sources. • If necessary, correct the connection error. • Set "Reset" to FALSE to exit the error state.

Table 587: "SF_EDM": Diagnostic codes

6.6.5.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_StartReset" = FALSE

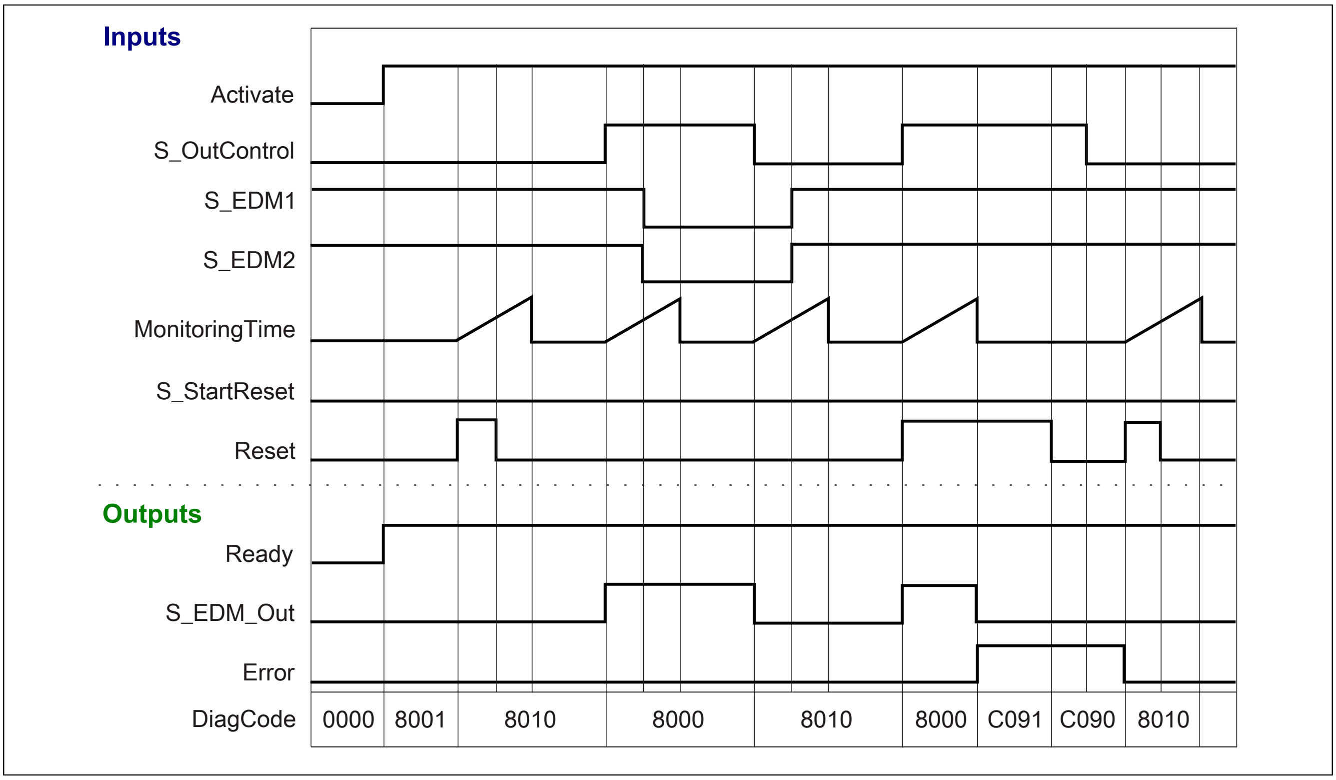


Figure 403: "SF_EDM": Signal sequence diagram 1

Signal sequence diagram 2

"S_StartReset" = TRUE

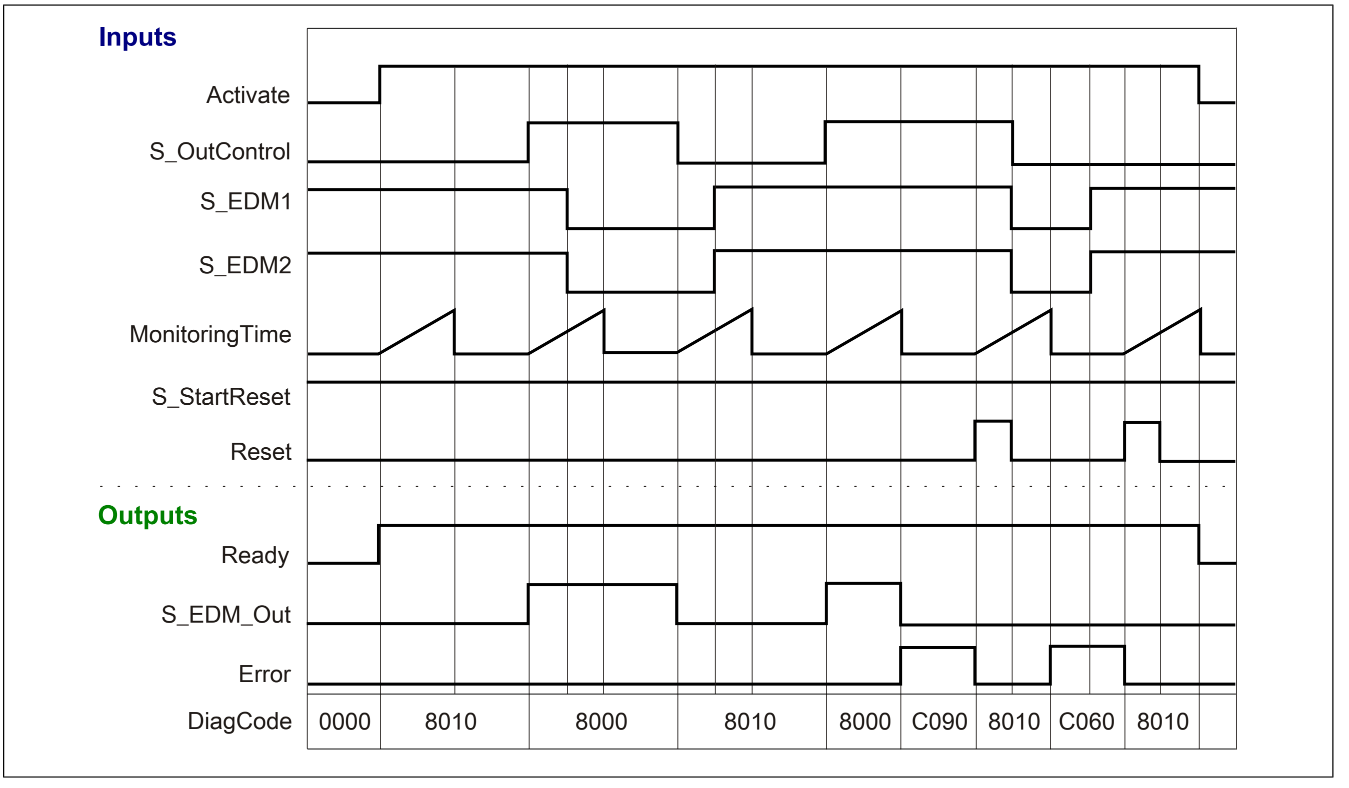


Figure 404: "SF_EDM": Signal sequence diagram 2

6.6.5.7 Application examples

This chapter illustrates possible applications in which the function block can be used to control monitored switching amplifiers monitored over 1 or 2 channels.

The following examples describe function block connections in the following cases:

- 1-channel switching amplifier feedback (see section 6.6.5.7.1 "Switching amplifier feedback over 1 channel")
- 2-channel switching amplifier feedback (see section 6.6.5.7.2 "Switching amplifier feedback over 2 channels (2 individual switching amplifier contacts)")
- 2-channel switching amplifier feedback with contacts connected in series for feedback (see section 6.6.5.7.3 "Switching amplifier feedback over 2 channels (2 switching amplifier contacts in series)")

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.5.7.1 Switching amplifier feedback over 1 channel

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "EDM_K1". Input parameters "S_EDM1" and "S_EDM2" are graphically connected to each other.

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to safe input signals "S_OutControl", "S_EDM1" and "S_EDM2", a rising edge on input parameter "Reset" is required to enable enable output "S_EDM_Out".

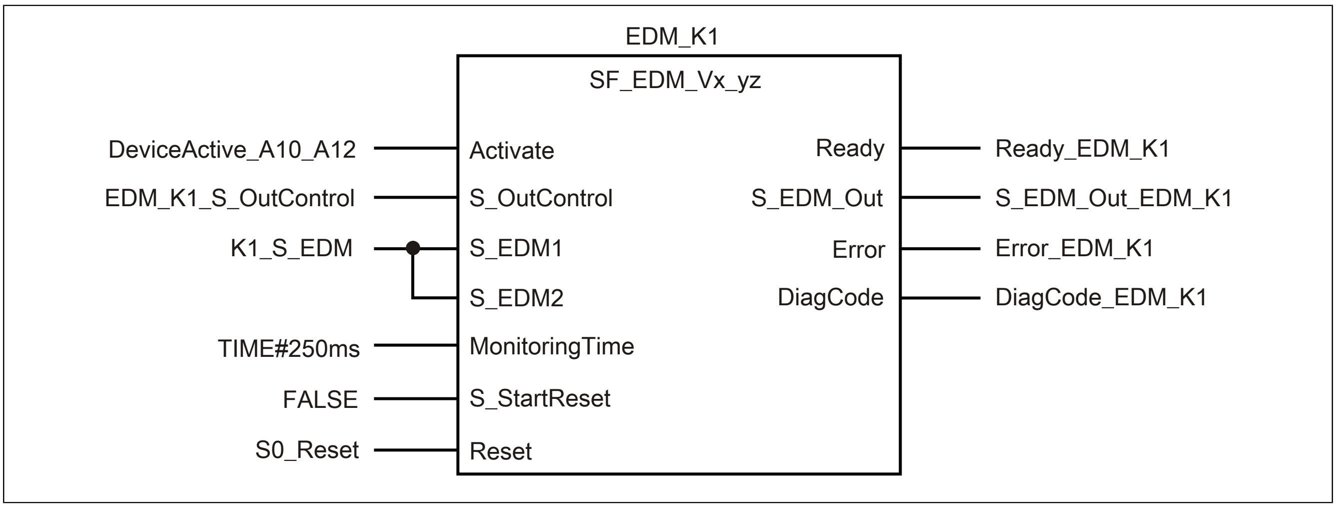


Figure 405: "SF_EDM": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input/output devices to which the switching amplifier or switching amplifiers are connected.
EDM_K1_S_OutControl	SAFEBOOL	Control from the safety application on the safety controller to set "S_EDM_Out" in consideration of "S_EDM1" and "S_EDM2"
K1_S_EDM	SAFEBOOL	Feedback signal from safe input device (switching amplifier "-K1"). This signal comes from a positively driven normally closed contact on switching amplifier "-K1".
TIME#250ms on "MonitoringTime"	TIME	Maximum permitted response time of the switching amplifier used in the example. This time monitoring applies both for switching the switching amplifier on as well as for switching it off.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 588: "SF_EDM": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_EDM_K1	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EDM_Out_EDM_K1	SAFEBOOL	Release signal. The release signal controls the monitored switching amplifier "-K1" via safe output devices.
Error_EDM_K1	BOOL	Error message from function block for further external processing
DiagCode_EDM_K1	WORD	Diagnostic message from function block for further external processing

Table 589: "SF_EDM": Outputs connected to output parameters

Description

In this example, function block enable output "S_EDM_Out" controls a switching amplifier via a 1-channel output of a safe output device. The switching amplifier provides 1-channel feedback to function block input "S_EDM1". For this, a contact of the switching amplifier is connected to input "S_EDM1" via a 1-channel input on the safe input device.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EDM_K1".

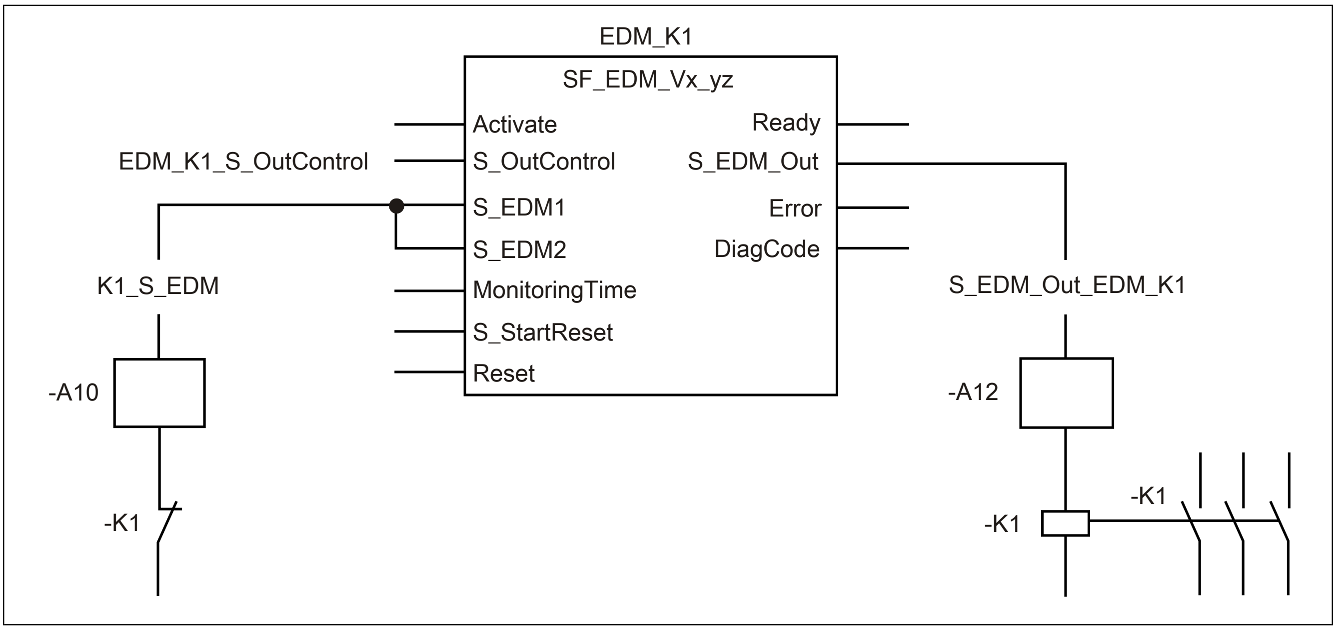


Figure 406: "SF_EDM": Switching amplifier feedback over 1 channel

List of equipment

-K1	Switching amplifier with positively driven contacts
-A10	1-channel safe input of a safe input device
-A12	1-channel safe output of a safe output device

Please note that, depending on your application, other combinations of safe devices can be used in place of the safe devices shown here.

Connected inputs and outputs

K1_S_EDM (graphically connected)	Input on "S_EDM1" and "S_EDM2"
EDM_K1_S_OutControl	Input on "S_OutControl"
S_EDM_Out_EDM_K1	Output on "S_EDM_Out"

Description

In this example:

- Release signal "EDM_K1_S_OutControl" from the safety application on the safety controller is connected to the function block input parameter "S_OutControl" for further processing.
- The signal of the normally closed contact of switching amplifier "-K1" on safe input device "-A10" is connected to input "K1_S_EDM". This input is connected to the graphically connected function block input parameters "S_EDM1" and "S_EDM2" for further processing.
- Signal "K1_S_EDM" is graphically connected to input parameters "S_EDM1" and "S_EDM2" since only one switching amplifier is being controlled and monitored in this example.
- Output parameter "S_EDM_Out" is connected to output "S_EDM_Out_EDM_K1". The output controls switching amplifier "-K1" and is output via the 1-channel output of safe output device "-A12".

6.6.5.7.2 Switching amplifier feedback over 2 channels (2 individual switching amplifier contacts)

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "EDM_K2_K3".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to safe input signals "S_OutControl", "S_EDM1" and "S_EDM2", a rising edge on input parameter "Reset" is required to enable enable output "S_EDM_Out".

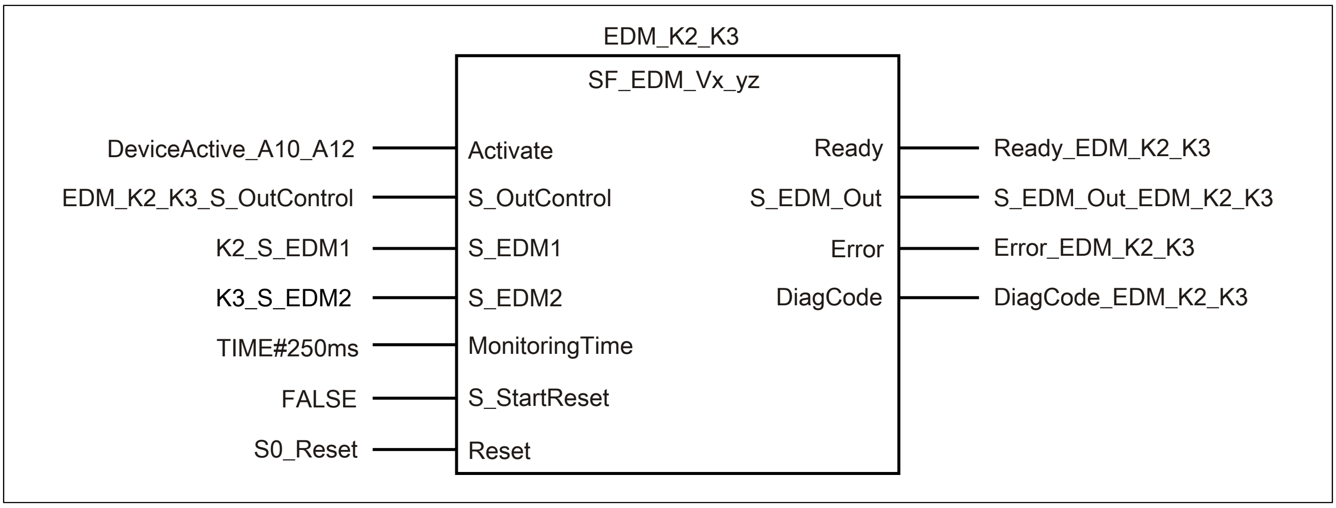


Figure 407: "SF_EDM": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input/output devices to which the switching amplifier or switching amplifiers are connected.
EDM_K2_K3_S_OutControl	SAFEBOOL	Control from the safety application on the safety controller to set "S_EDM_Out" in consideration of "S_EDM1" and "S_EDM2"
K2_S_EDM1	SAFEBOOL	Feedback signal from safe input device (switching amplifier "-K2"). This signal comes from a positively driven normally closed contact on switching amplifier "-K2".
K3_S_EDM2	SAFEBOOL	Feedback signal from safe input device (switching amplifier "-K3"). This signal comes from a positively driven normally closed contact on switching amplifier "-K3".
TIME#250ms on "MonitoringTime"	TIME	Maximum permitted response time of the switching amplifiers used in the example. This time monitoring applies both for switching the switching amplifier on as well as for switching it off.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 590: "SF_EDM": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_EDM_K2_K3	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EDM_Out_EDM_K2_K3	SAFEBOOL	Release signal. The release signal controls monitored switching amplifier "-K2" and "-K3" via the 2-channel output of a safe output device.
Error_EDM_K2_K3	BOOL	Error message from function block for further external processing
DiagCode_EDM_K2_K3	WORD	Diagnostic message from function block for further external processing

Table 591: "SF_EDM": Outputs connected to output parameters

Description

In this example, function block enable output "S_EDM_Out" controls the 2-channel output of a safe output device. One switching amplifier is controlled per output channel. The switching amplifiers provide 1-channel feedback to function block inputs "S_EDM1" and "S_EDM2". For this, a contact of a switching amplifier is each connected to a function block input via a 1-channel input on the safe input device.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EDM_K2_K3".

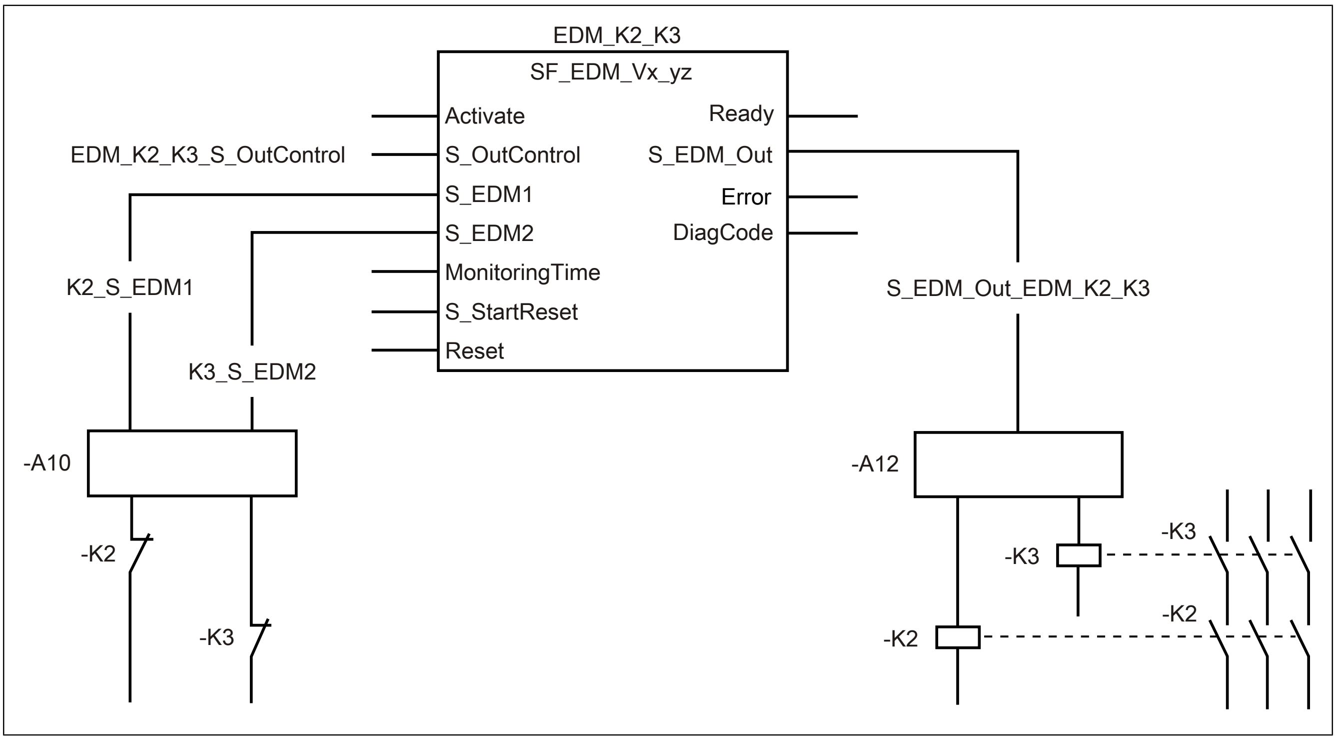


Figure 408: "SF_EDM": Switching amplifier feedback over 2 channels (2 individual switching amplifier contacts)

<u>List of equipment</u>

-K2, -K3	Switching amplifier with positively driven contacts
-A10	1-channel safe inputs of a safe input device
-A12	2-channel safe output of a safe output device

Please note that, depending on your application, other combinations of safe devices can be used in place of the safe devices shown here.

<u>Connected inputs and outputs</u>

K2_S_EDM1	Input on "S_EDM1"
K3_S_EDM2	Input on "S_EDM2"
EDM_K2_K3_S_OutControl	Input on "S_OutControl"
S_EDM_Out_EDM_K2_K3	Output on "S_EDM_Out"

<u>Description</u>

In this example:

- The resulting release signal "EDM_K2_K3_S_OutControl" from the safety application on the safety controller is connected to the function block input parameter "S_OutControl" for further processing.
- The signal of the normally closed contact of switching amplifier "-K2" on safe input device "-A10" is connected to input "K2_S_EDM1". This input is connected to function block input parameter "S_EDM1" for further processing.
- The signal of the normally closed contact of switching amplifier "-K3" on safe input device "-A10" is connected to input "K3_S_EDM2". This input is connected to function block input parameter "S_EDM2" for further processing.
- Output parameter "S_EDM_Out" is connected to output "S_EDM_Out_EDM_K2_K3". This output controls switching amplifiers "-K2" and "-K3" and is output via the 2-channel output of safe output device "-A12".

6.6.5.7.3 Switching amplifier feedback over 2 channels (2 switching amplifier contacts in series)

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "EDM_K4_K5". Input parameters "S_EDM1" and "S_EDM2" are graphically connected to each other.

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to safe input signals "S_OutControl", "S_EDM1" and "S_EDM2", a rising edge on input parameter "Reset" is required to enable enable output "S_EDM_Out".

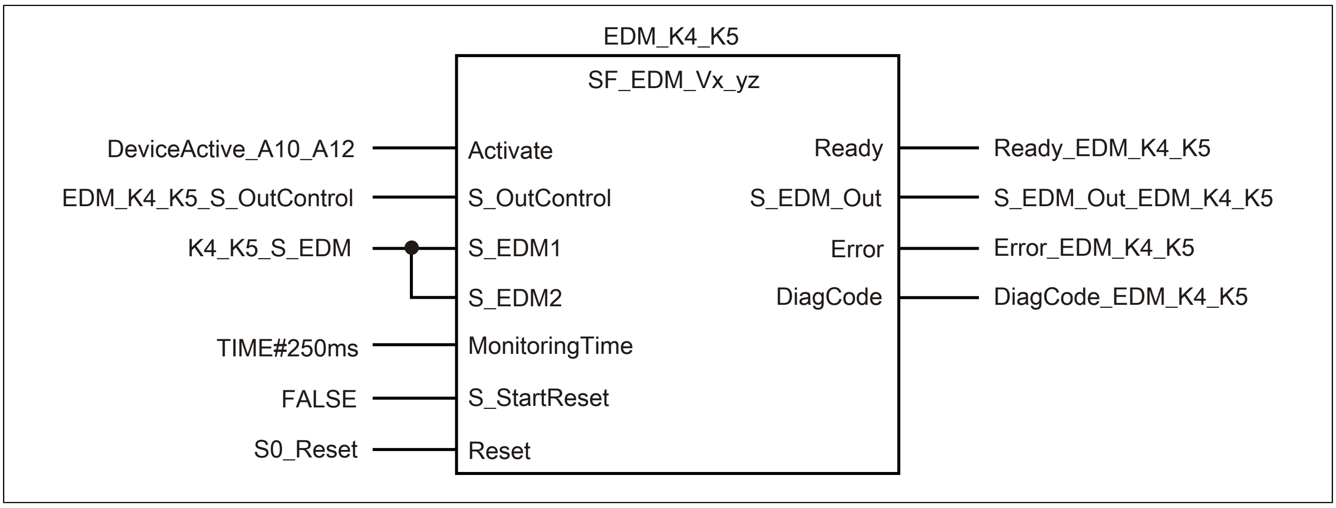


Figure 409: "SF_EDM": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input/output devices to which the switching amplifier or switching amplifiers are connected.
EDM_K4_K5_S_OutControl	SAFEBOOL	Control from the safety application on the safety controller to set "S_EDM_Out" in consideration of "S_EDM1" and "S_EDM2"
K4_K5_S_EDM	SAFEBOOL	Feedback signal from safe input device "-A10" (switching amplifiers "-K4" and "-K5"). This signal comes from the positively driven normally closed contacts connected in series on switching amplifiers "-K4" and "-K5".
TIME#250ms on "MonitoringTime"	TIME	Maximum permitted response time of the switching amplifiers used in the example. This time monitoring applies both for switching the switching amplifier on as well as for switching it off.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 592: "SF_EDM": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_EDM_K4_K5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EDM_Out_EDM_K4_K5	SAFEBOOL	Release signal. The release signal controls monitored switching amplifier "-K4" and "-K5" via the 2-channel output of a safe output device.
Error_EDM_K4_K5	BOOL	Error message from function block for further external processing
DiagCode_EDM_K4_K5	WORD	Diagnostic message from function block for further external processing

Table 593: "SF_EDM": Outputs connected to output parameters

Description

In this example, function block enable output "S_EDM_Out" controls the 2-channel output of a safe output device. One switching amplifier is controlled per output channel. The switching amplifiers provide 1-channel feedback to function block input "S_EDM1". For this, the contacts of the switching amplifiers connected in series are connected to input "S_EDM1" via a 1-channel input on the safe input device.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EDM_K4_K5".

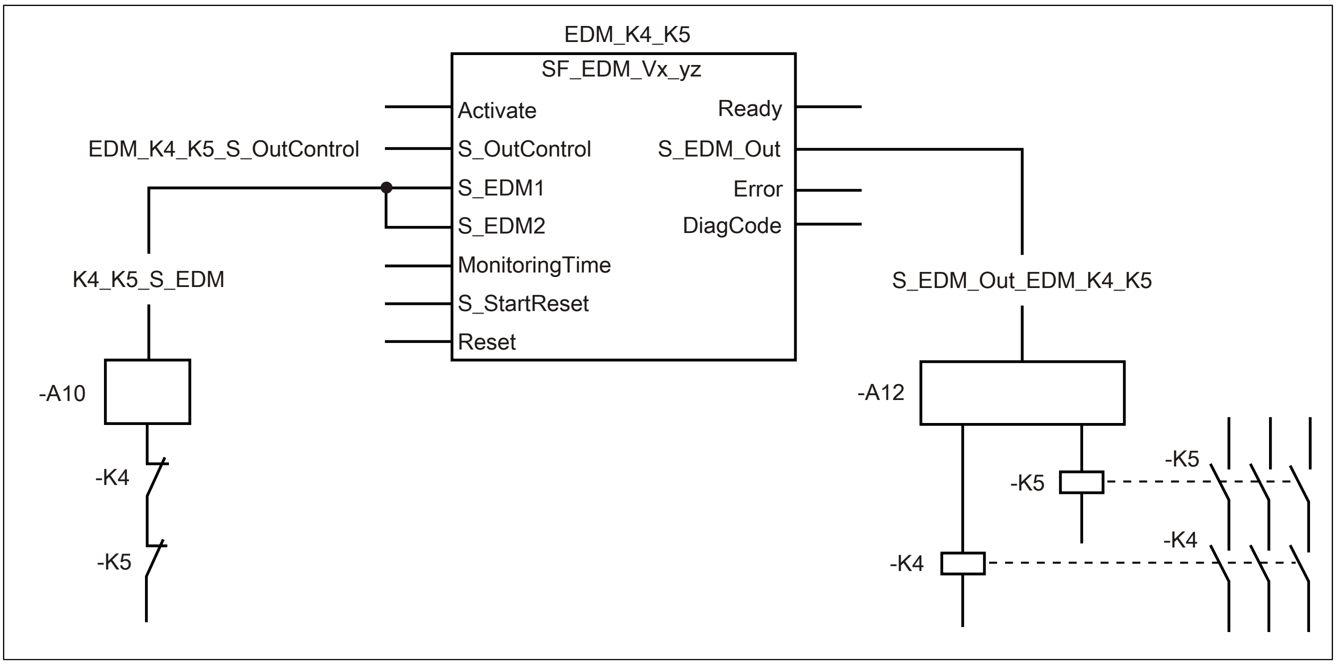


Figure 410: "SF_EDM": Switching amplifier feedback over 2 channels (2 switching amplifier contacts in series)

List of equipment

-K4, -K5	Switching amplifier with positively driven contacts
-A10	1-channel safe input of a safe input device
-A12	2-channel safe output of a safe output device

Please note that, depending on your application, other combinations of safe devices can be used in place of the safe devices shown here.

Connected inputs and outputs

K4_K5_S_EDM (graphically connected)	Input on "S_EDM1" and "S_EDM2"
EDM_K4_K5_S_OutControl	Input on "S_OutControl"
S_EDM_Out_EDM_K4_K5	Output on "S_EDM_Out"

Description

In this example:

- The resulting release signal "EDM_K4_K5_S_OutControl" from the safety application on the safety controller is connected to the function block input parameter "S_OutControl" for further processing.
- The signal of the normally closed contacts (connected in series) of switching amplifiers "-K4" and "-K5" on safe input device "-A10" are connected to input "K4_K5_S_EDM". This input is connected to the graphically connected function block input parameters "S_EDM1" and "S_EDM2" for further processing.
- Output parameter "S_EDM_Out" is connected to output "S_EDM_Out_EDM_K4_K5". This output controls switching amplifiers "-K4" and "-K5" and is output via the 2-channel output of safe output device "-A12".

6.6.5.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 60204	Stop functions	This function block (release signal "S_EDM_Out") executes a category 0 stop.
EN 60204	Start	Enable output "S_EDM_Out" is only set to TRUE if the combination of input signals is valid for this.
EN 954-1	Stop function	The function block sets its enable output to FALSE if it detects a safety request on "S_OutControl". A safety function must always be involved to control "S_OutControl". Operational starting and stopping from the standard controller must be controlled using an AND operator with the safety function.
EN 954-1	Specifications for fault detection categories	1-channel safety function: The function block checks the basic state and switching state of a feedback signal from a switching amplifier in order to safely diagnose the state of the switching amplifier. 2-channel safety function: The function block checks the basic state and switching state of the feedback signals from 2 switching amplifiers in order to detect a faulty switching amplifier, prevent it from being switched on or switch it off.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restarting the safety controller • Enabling the function block ("S_StartReset" = FALSE) If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.

Table 594: "SF_EDM": Implementation of requirements from standards

6.6.6 SF_EmergencyStop

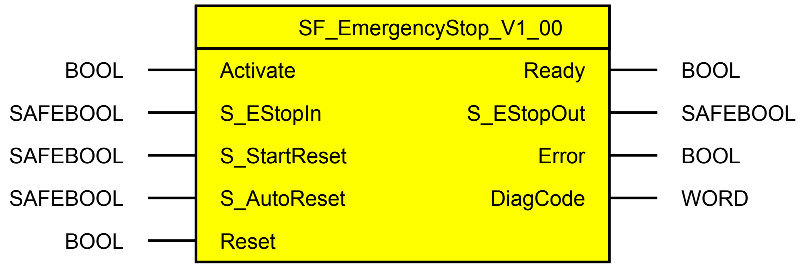


Figure 411: Function block "SF_EmergencyStop"

6.6.6.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_EStopIn	SAFEBOOL	Variable	State	FALSE	Control signal. State of the emergency switch-off control device
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after signal TRUE returns to "S_EStopIn"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 595: "SF_EmergencyStop": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_EStopOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 596: "SF_EmergencyStop": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 597: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.6.2 Function

Function block "SF_EmergencyStop" is used to support an emergency switch-off function in an application.

This function block is a safety-relevant function block for monitoring an emergency switch-off control device.

6.6.6.2.1 Actuating the emergency switch-off control device

If an emergency switch-off control device is actuated in the application, the function block ensures that function block enable output "S_EStopOut" is set to FALSE.

6.6.6.2.2 Resetting the emergency switch-off control device

If an actuated emergency switch-off control device is reset in the application, the function block can optionally (see start interlock) ensure within the safety control system that the enable output is not set to TRUE solely by this reset. Additional manual intervention on input parameter "Reset" is required for this (see start interlock).

6.6.6.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.6.2.4 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.6.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.6.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.6.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.6.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the state of the release signal depends on the state of the emergency switch-off control device connected to the function block as well as on how the optional start interlock is specified for when the TRUE signal returns to the emergency switch-off control device.

6.6.6.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.6.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.6.4 Input parameters

6.6.6.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.6.4.2 S_EStopIn

General function

- Control signal. State of the emergency switch-off control device.

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the emergency switch-off control device over 1 or 2 channels. Input parameter "S_EStopIn" is controlled via this signal.**

Function description

The signal connected to input parameter "S_EStopIn" is processed by the function block.

The signal input processes the state of the emergency switch-off control device.

Regardless whether the emergency switch-off control device is connected to the safe device over 1 or 2 channels, "S_EStopIn" is only connected to one signal of the emergency switch-off control device.

If an emergency switch-off control device is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on one signal to "S_EStopIn". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on one signal to "S_EStopIn". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The emergency switch-off control device is not actuated.

After a signal change from FALSE to TRUE on input parameter "S_EStopIn", the function block optionally supports a start interlock ("S_AutoReset" = FALSE). An active start interlock is indicated accordingly on output parameter "DiagCode". While the start interlock is active, enable output "S_EStopOut" remains set to FALSE. The start interlock is reset on a rising edge of "Reset" ("Reset": FALSE → TRUE). The reset changes enable output "S_EStopOut" from FALSE to TRUE.

FALSE

The emergency switch-off control device is actuated, the wiring to the emergency switch-off control device is interrupted or the safe device connected to the emergency switch-off control device is shut down or defective.

Enable output "S_EStopOut" is set to FALSE; output parameter "DiagCode" is set accordingly.

6.6.6.4.3 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

Automatic startup

Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.

The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.

6.6.6.4.4 S_AutoReset

General function

- Specification of the start interlock after signal TRUE returns to "S_EStopIn"

Data type

- SAFEBOOL

Connection

- Variable or constant

> **Information:**
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after the signal has returned to safe input parameter "S_EStopIn" (resetting the emergency switch-off control device).

TRUE

After signal TRUE returns to safe input parameter "S_EStopIn", the function block does not support start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> **Danger!**
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After signal TRUE returns to safe input parameter "S_EStopIn", the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> **Danger!**
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.6.4.5 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

> ## Danger!
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.6.5 Output parameters

6.6.6.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.6.5.2 S_EStopOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected emergency switch-off control device for the process being controlled.

The release signal is controlled based on the state of the emergency switch-off control device and start interlocks. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_EStopOut", this output is referred to as the "enable output".

Release signal "S_EStopOut" can be used for subsequent process control.

> ## Danger!
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The emergency switch-off control device is not actuated ("S_EStopIn" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected an actuated emergency switch-off control device ("S_EStopIn" = FALSE).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_EStopOut" is set to TRUE only if input "S_EStopIn" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	After signal TRUE returns on safe input "S_EStopIn", the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_EStopOut" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 598: "SF_EmergencyStop": Input parameter "S_AutoReset"/"S_StartReset"

6.6.6.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.6.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.6.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The emergency switch-off control device is not actuated.	No corrective measures are required.
8001	Initialization of the function block after it has been enabled. The initialization of the function block is completed after one safety controller cycle.	No corrective measures are required.
8002	The start interlock to be carried out after enabling the function block is active ("S_StartReset" = FALSE). The function block detected the following after being enabled: The emergency switch-off control device is actuated and locked, the wiring is faulty or the safe device connected to the emergency switch-off control device is shut down or defective.	• Reset the actuated and latched emergency switch-off control device. • For an unactuated emergency switch-off control device, check the wiring or the safe device connected to the emergency switch-off control device.
8003	The emergency switch-off control device is reset again. The function block's start interlock is active. ("S_StartReset" = FALSE)	Reset the function block.
8004	The start interlock to be carried out after enabling the function block is not active ("S_StartReset" = TRUE) and the function block was enabled, or enable output "S_EStopOut" was switched off by input parameter "S_EStopIn" = FALSE. The emergency switch-off control device is actuated and locked, the wiring is faulty or the safe device connected to the emergency switch-off control device is shut down or defective.	• Reset the actuated and latched emergency switch-off control device. • For an unactuated emergency switch-off control device, check the wiring or the safe device connected to the emergency switch-off control device.
8005	The emergency switch-off control device is reset again. The function block's start interlock is active. ("S_AutoReset" = FALSE)	Reset the function block.
C001	After resetting the emergency switch-off control device, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	After resetting the emergency switch-off control device, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 599: "SF_EmergencyStop": Diagnostic codes

6.6.6.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

Startup, reset, normal operation, safety request, restart

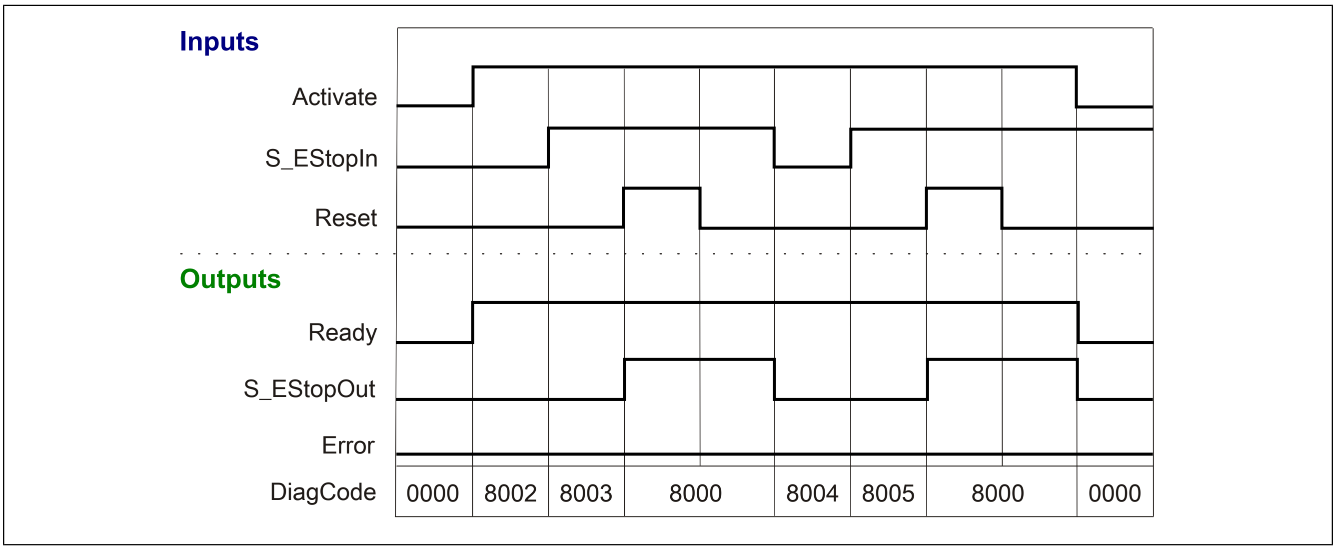


Figure 412: "SF_EmergencyStop": Signal sequence diagram 1

Signal sequence diagram 2

"S_StartReset" = TRUE
"S_AutoReset" = FALSE

Startup, normal operation, safety request, restart

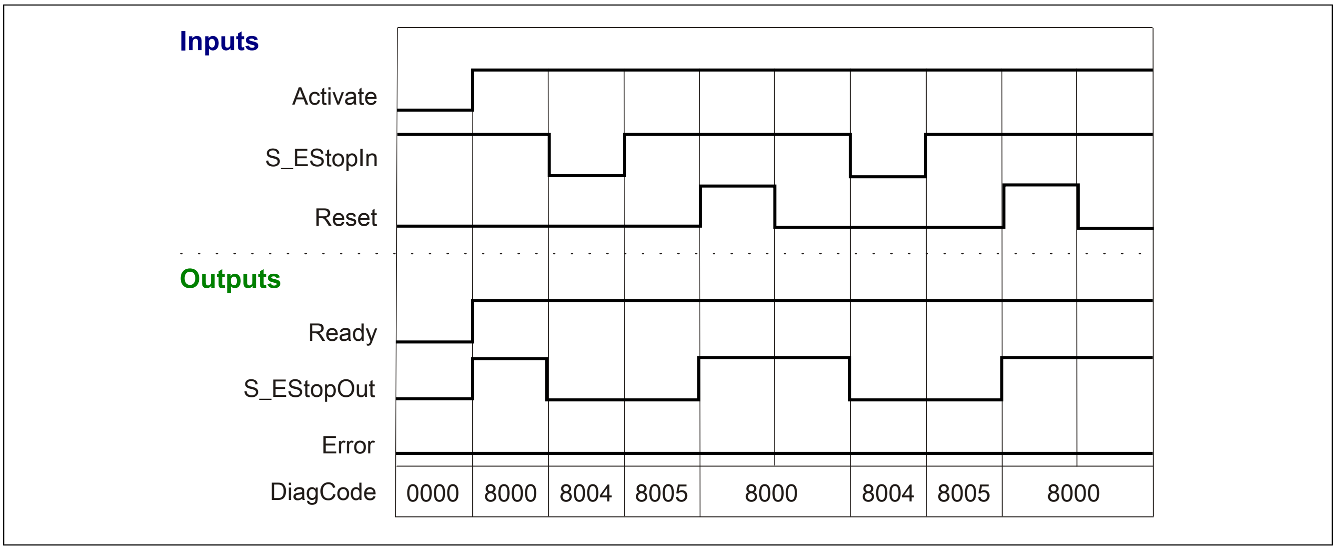


Figure 413: "SF_EmergencyStop": Signal sequence diagram 2

Signal sequence diagram 3

"S_StartReset" = FALSE
"S_AutoReset" = TRUE

Startup, normal operation, safety request, restart

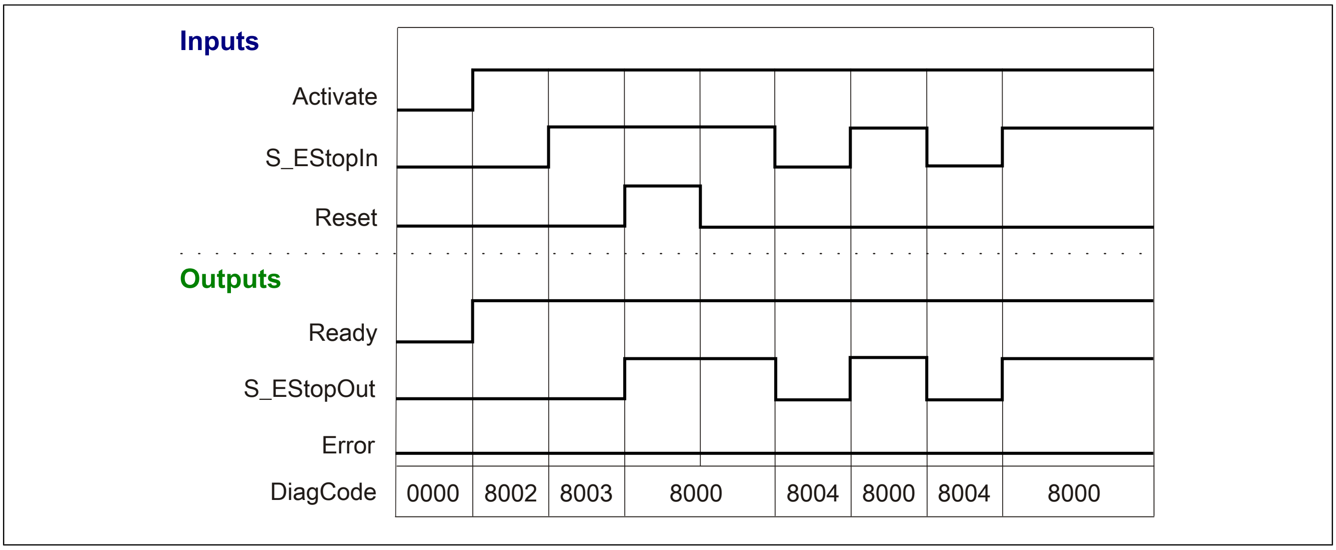


Figure 414: "SF_EmergencyStop": Signal sequence diagram 3

6.6.6.7 Application examples

This chapter illustrates possible applications in which the function block can be used to implement a 1-channel or 2-channel emergency switch-off device.

The following examples describe function block connections when controlling with the following:

- The signal from a emergency switch-off control device connected over 1 channel (see section 6.6.6.7.2 "Emergency switch-off control device connected over 1 channel")
- The signals from a 2-channel, equivalent connected emergency switch-off control device (see section 6.6.6.7.3 "Emergency switch-off control device connected over 2 channels, equivalent connection")
- The signals from a 2-channel, antivalent connected emergency switch-off control device (see section 6.6.6.7.4 "Emergency switch-off control device connected over 2 channels, antivalent connection")

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.6.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "EStop_S20".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signal on "S_EStopIn", a rising edge on input parameter "Reset" is required in order for enable output "S_EStopOut" to be enabled.

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the return of the safe input signal on "S_EStopIn". In addition to the safe input signal on "S_EStopIn", a rising edge on input parameter "Reset" is required in order for enable output "S_EStopOut" to be enabled.

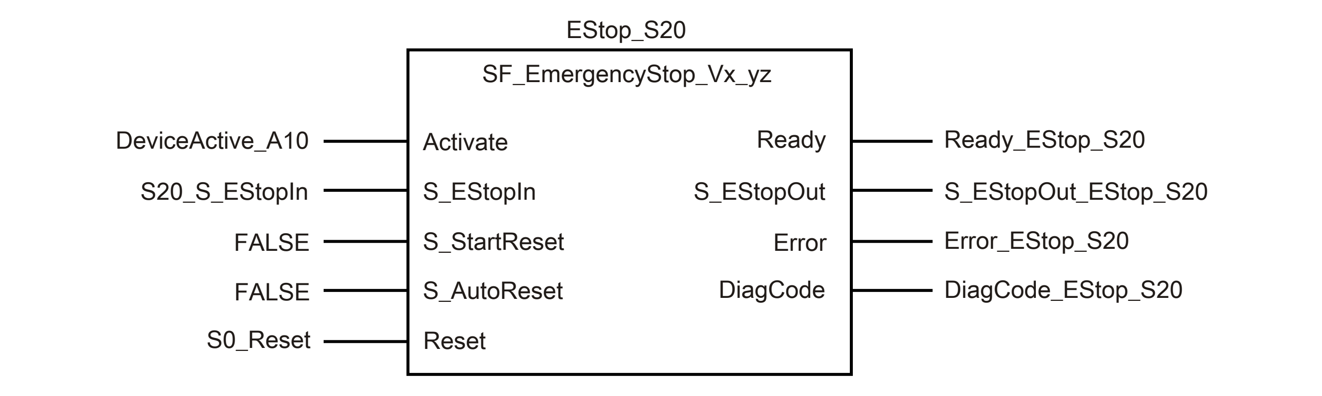


Figure 415: "SF_EmergencyStop": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the emergency switch-off control device is connected.
S20_S_EStopIn	SAFEBOOL	Control signal from a safe input device. The signal comes from a 1-channel or 2-channel input of a safe input device. The evaluation of dual-channel redundancy does not take place in the function block.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after resetting the emergency switch-off control device
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 600: "SF_EmergencyStop": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_EStop_S20	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EStopOut_EStop_S20	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
Error_EStop_S20	BOOL	Error message from function block for further external processing
DiagCode_EStop_S20	WORD	Diagnostic message from function block for further external processing

Table 601: "SF_EmergencyStop": Outputs connected to output parameters

6.6.6.7.2 Emergency switch-off control device connected over 1 channel

This example illustrates the connection of the function block when controlling using the signal from an emergency switch-off control device connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.6.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EStop_S20".

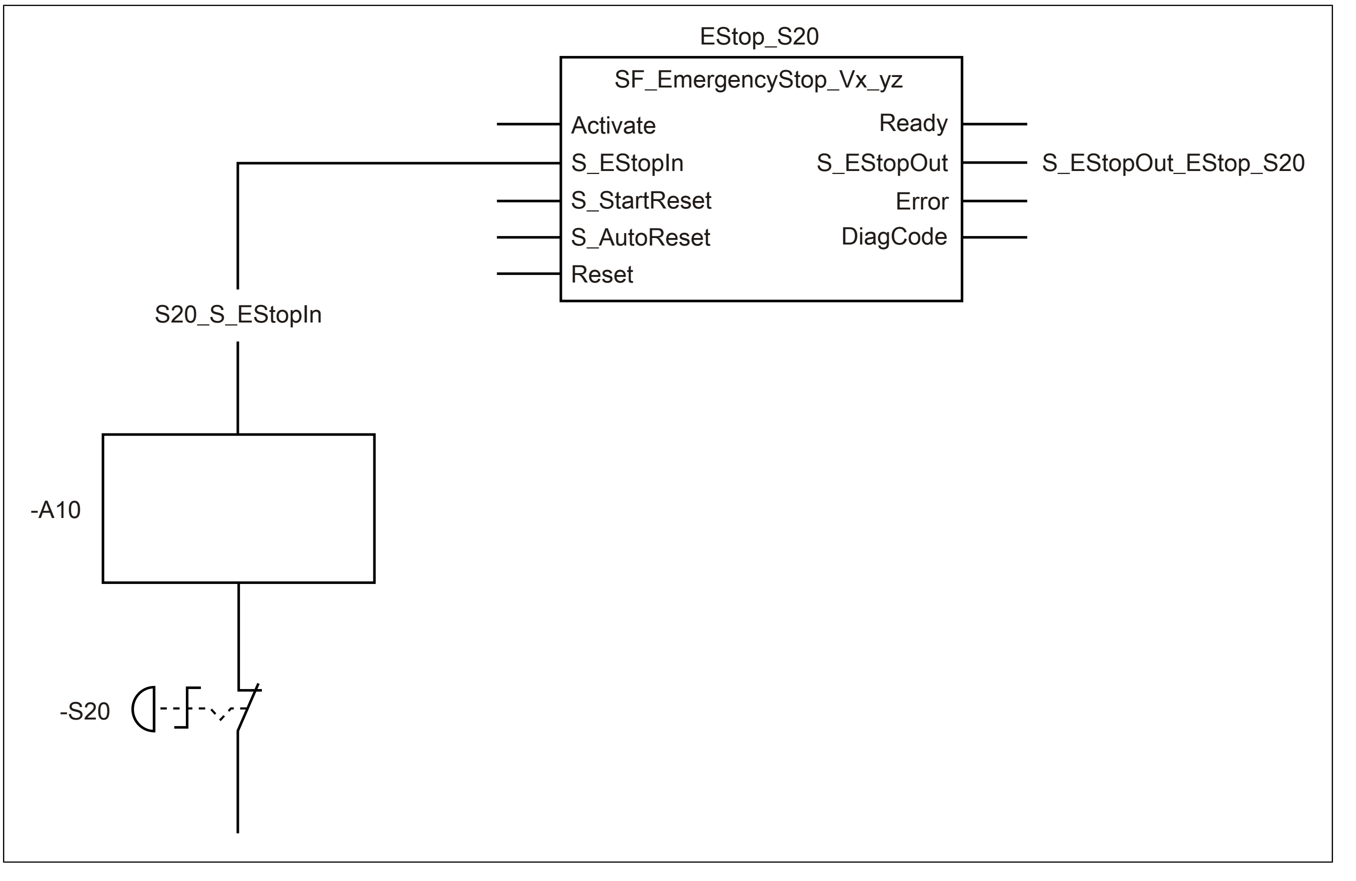


Figure 416: "SF_EmergencyStop": Emergency switch-off control device connected over 1 channel

List of equipment

-S20	Emergency switch-off control device (1-channel)
-A10	1-channel input of a safe input device

Connected inputs and outputs

S20_S_EStopIn	Input on "S_EStopIn"
S_EStopOut_EStop_S20	Output on "S_EStopOut"

Description

In this example:

- The signal from the 1-channel safe input of safe input device "-A10" is connected to input "S20_S_EStopIn".
- Input "S20_S_EStopIn" is connected to function block input parameter "S_EStopIn" for further processing.
- Output parameter "S_EStopOut" is connected to output "S_EStopOut_EStop_S20".
- Output "S_EStopOut_EStop_S20" is used as a release signal to control the process in consideration of other safety functions.

6.6.6.7.3 Emergency switch-off control device connected over 2 channels, equivalent connection

This example illustrates the connection of the function block when controlling using the signal from an emergency switch-off control device connected over 2 channels (equivalent).

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.6.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EStop_S20".

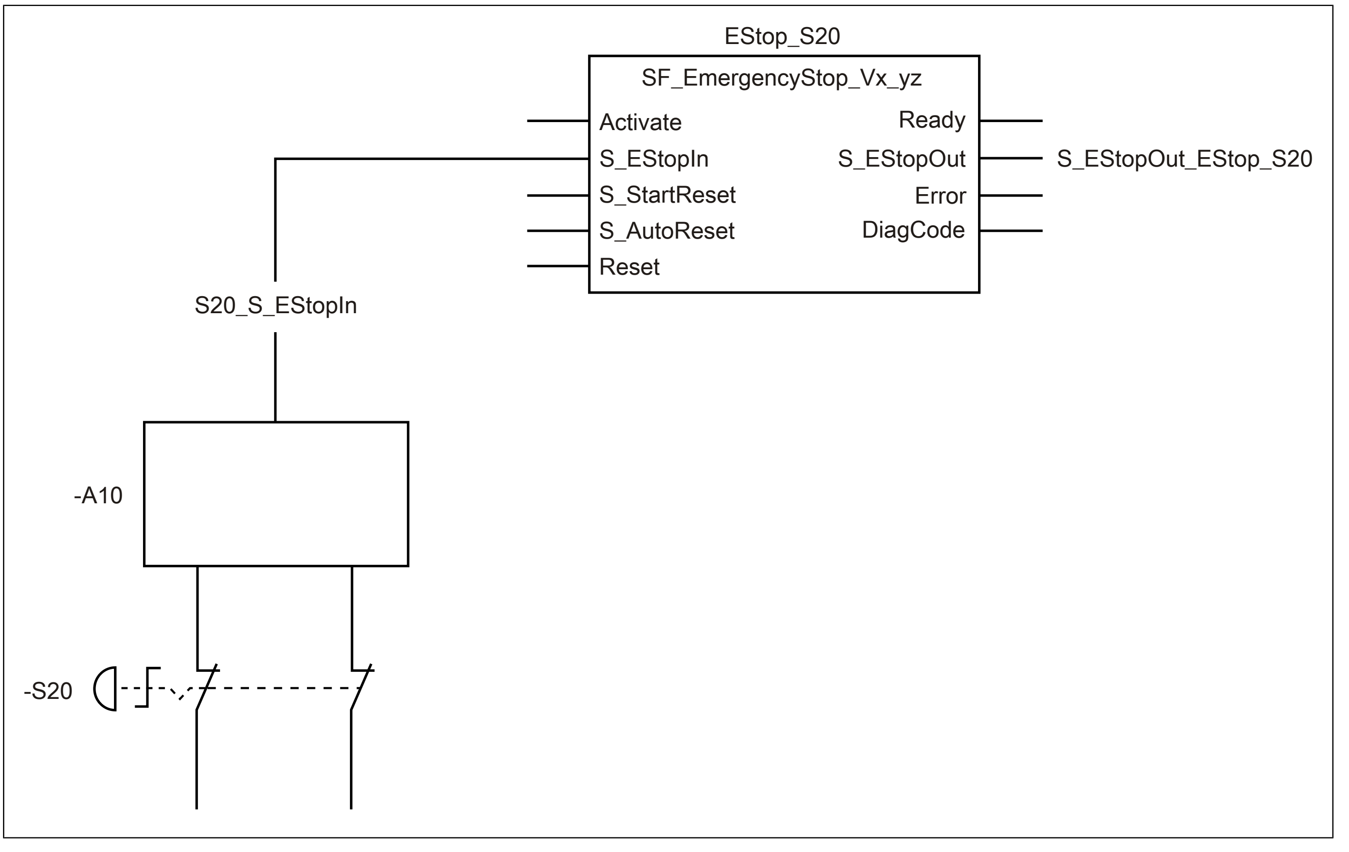


Figure 417: "SF_EmergencyStop": Emergency switch-off control device connected over 2 channels, equivalent connection

List of equipment

-S20	Emergency switch-off control device (2-channel, equivalent)
-A10	2-channel input of a safe input device (equivalent)

Connected inputs and outputs

S20_S_EStopIn	Input on "S_EStopIn"
S_EStopOut_EStop_S20	Output on "S_EStopOut"

Description

In this example:

- The resulting signal of the inputs from safe input device "-A10" are connected to input "S20_S_EStopIn".
- Input "S20_S_EStopIn" is connected to function block input parameter "S_EStopIn" for further processing.
- Output parameter "S_EStopOut" is connected to output "S_EStopOut_EStop_S20".
- Output "S_EStopOut_EStop_S20" is used as a release signal to control the process in consideration of other safety functions.

6.6.6.7.4 Emergency switch-off control device connected over 2 channels, antivalent connection

This example illustrates the connection of the function block when controlling using the signal from an emergency switch-off control device connected over 2 channels (antivalent).

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.6.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EStop_S20".

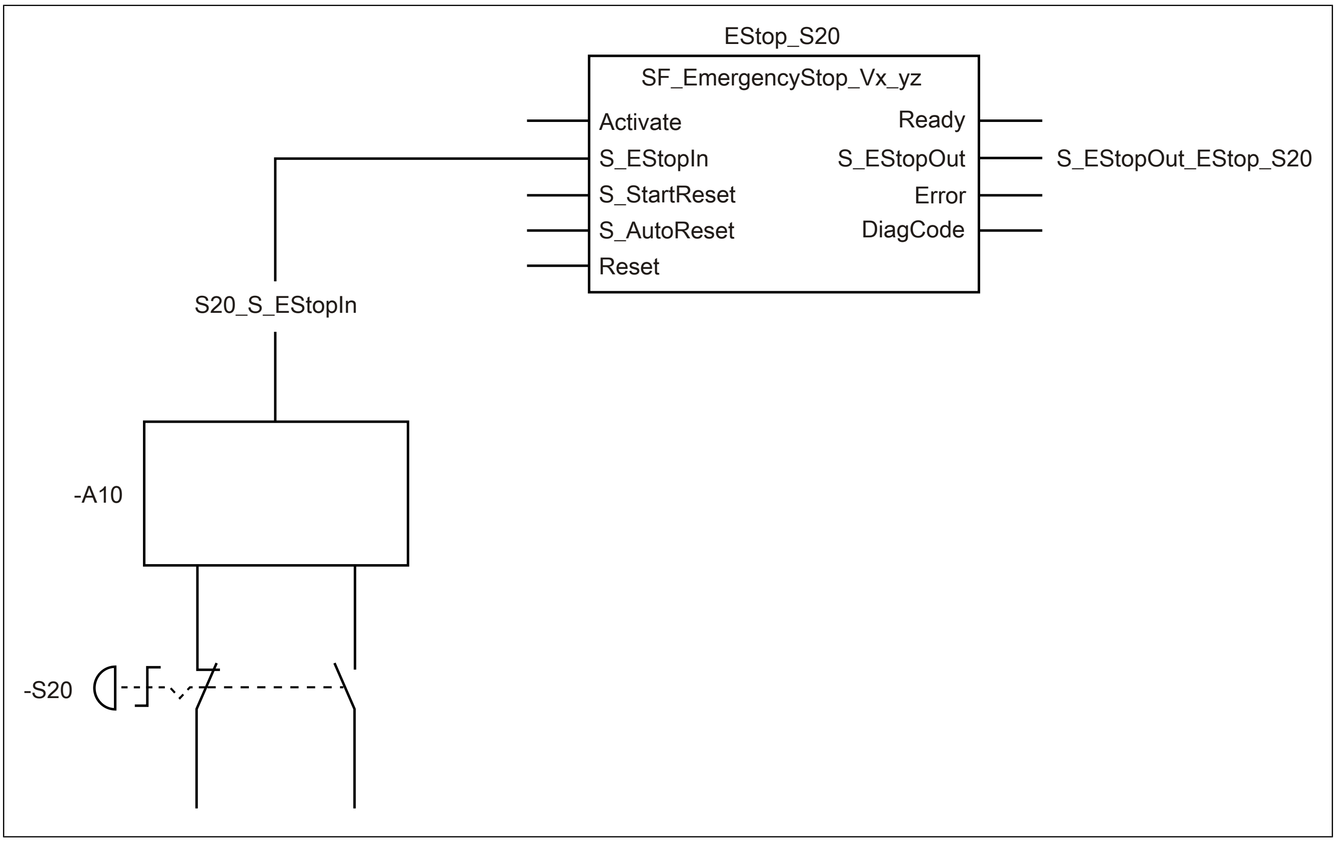


Figure 418: "SF_EmergencyStop": Emergency switch-off control device connected over 2 channels, antivalent connection

List of equipment

-S20	Emergency switch-off control device (2-channel, antivalent)
-A10	2-channel input of a safe input device (antivalent)

Connected inputs and outputs

S20_S_EStopIn	Input on "S_EStopIn"
S_EStopOut_EStop_S20	Output on "S_EStopOut"

Description

In this example:

- The resulting signal of the inputs from safe input device "-A10" are connected to input "S20_S_EStopIn".
- Input "S20_S_EStopIn" is connected to function block input parameter "S_EStopIn" for further processing.
- Output parameter "S_EStopOut" is connected to output "S_EStopOut_EStop_S20".
- Output "S_EStopOut_EStop_S20" is used as a release signal to control the process in consideration of other safety functions.

6.6.6.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 418	Positively driven actuation	Only use approved emergency switch-off control devices that comply with standard EN 418.
EN 418	Emergency switch-off precedence	It is your responsibility to ensure that emergency switch-off signals are connected so that the emergency switch-off command takes precedence over all other commands and is processed in every safety controller cycle. The order of execution of signals in the application program must be taken into account!
EN 418	Actuator operation / Resetting the operating element	The function block optionally supports a start interlock after resetting the emergency switch-off control device.
EN 418	Actuator operation / Function of the emergency switch-off device	After actuating the actuator, the emergency switch-off device must work to automatically avert or reduce hazards in the most effective way possible.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_AutoReset" = FALSE) • Enabling of the function block ("S_StartReset" = FALSE) • Resetting the emergency switch-off control device If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_EStopOut") executes a category 0 stop.

Table 602: "SF_EmergencyStop": Implementation of requirements from standards

Danger!

It is your responsibility to implement the "Actuator operation / Resetting the operating element" function outside of this function block in the safety control system.

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.7 SF_EnableSwitch

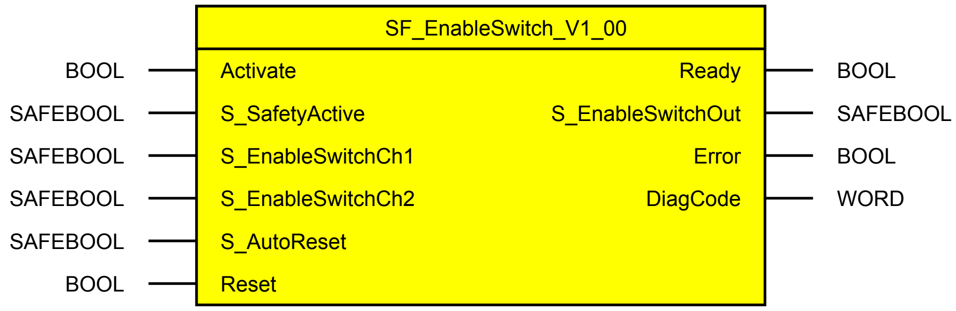


Figure 419: Function block "SF_EnableSwitch"

6.6.7.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_SafetyActive	SAFEBOOL	Variable/ Constant	State	FALSE	Control signal. Confirmation (feedback signal) that the selected safe operating mode is active
S_EnableSwitchCh1	SAFEBOOL	Variable	State	FALSE	Input for resulting signal of contacts E1 and E2 of the connected hand-operated, 3-position enable switch
S_EnableSwitchCh2	SAFEBOOL	Variable	State	FALSE	Input for resulting signal of contacts E3 and E4 of the connected hand-operated 3-position enable switch
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock if valid signals of the 3-position enable switch are present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" following an invalid signal sequence on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 603: "SF_EnableSwitch": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_EnableSwitchOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 604: "SF_EnableSwitch": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 605: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.7.2 Function

Function block "SF_EnableSwitch" supports the lifting of technical protective measures using a hand-operated 3-position enable switch if the corresponding operating mode (e.g. limiting the speed, energy or area of movement) is selected and active. The corresponding operating mode must be selected outside of the function block.

The function block evaluates the signals from a 3-position enable switch.

6.6.7.2.1 Enable switch requirements

The 3-position enable switch being used must support the following signal level for its 3 switch positions:

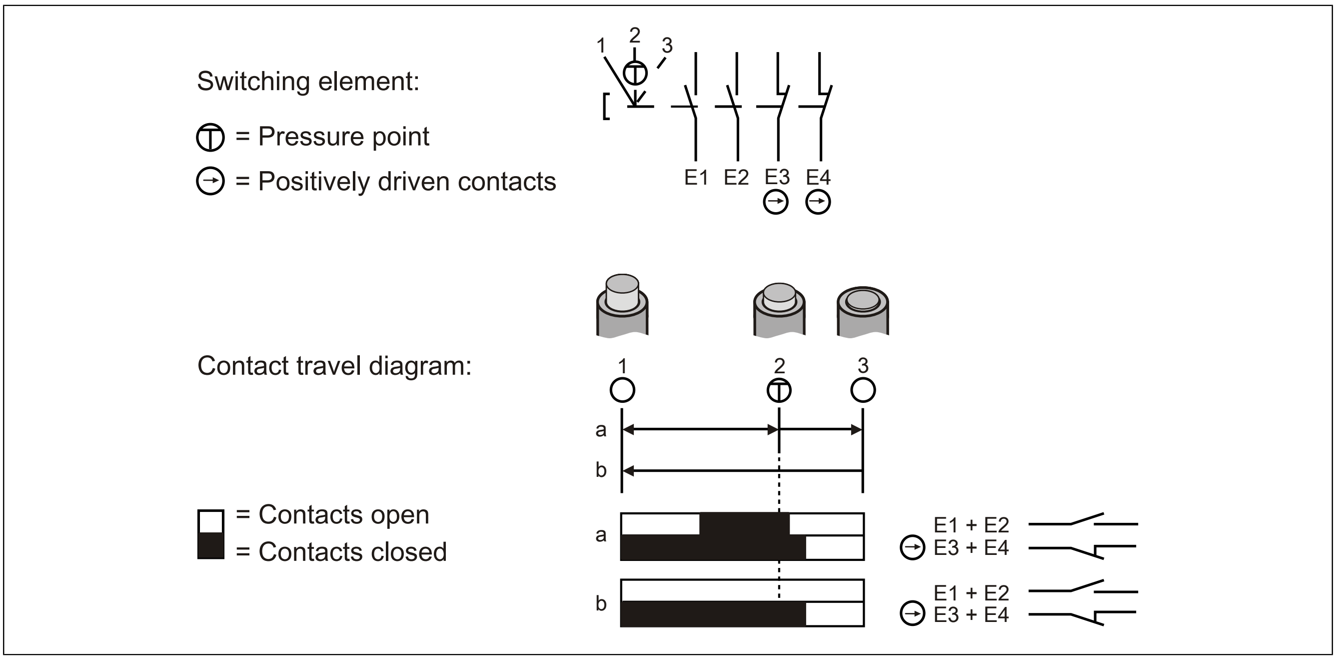


Figure 420: "SF_EnableSwitch": Enable switch - Switching elements and contact travel diagrams

Your enable switch must meet the criteria of the CAT/SIL required by your application.

6.6.7.2.2 Connecting conditions

The resulting signal from normally open contacts E1 and E2 must be connected to input parameter "S_EnableSwitchCh1". The resulting signal from positively driven normally closed contacts E3 and E4 must be connected to input parameter "S_EnableSwitchCh2". The function block can detect the switch position of the enable switch as well as its switching direction from this defined order of signals from the contacts (switch position 1 → switch position 2 / switch position 3 → switch position 2). The function block is only permitted to enable the lifting of the technical protective measure after changing from switch position 1 to switch position 2. The function block is not permitted to lift the technical protective measure in other switching directions or switch positions.

6.6.7.2.3 General requirements

> ## Information:
>
> **Use an appropriate switching device for the enable switch.**
>
> **Choose the corresponding safe operating mode in the safety application based on your risk analysis. In this operating mode, automatic operation must be absolutely prohibited using suitable means.**

You will typically specify the safe operating mode using an operating mode selector switch together with function blocks "SF_ModeSelector" and "SF_SafetyRequest".

Function block "SF_EnableSwitch" processes the confirmation feedback from the selected safe operating mode using input parameter "S_SafetyActive". If the safe operating mode is safely implemented without confirmation in your application, connect a static TRUE signal on input parameter "S_SafetyActive" .

6.6.7.2.4 Start interlock

Case 1: If the function block detects a selected safe operating mode on "S_SafetyActive", then input parameters "S_EnableSwitchCh1" and "S_EnableSwitchCh2" must indicate the signals for switch position 1 (enable switch not actuated). Otherwise (switch position 2 or 3), the function block outputs an error message and enable output "S_EnableSwitchOut" remains in the safe state (FALSE).

Case 2: If the function block detects the signals for switch position 2 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" even though the enable switch is set to switch position 3 (defective enable switch), then the function block outputs an error message and enable output "S_EnableSwitchOut" remains in the safe state (FALSE).

Even if the source of the error no longer exists in these two error cases, the start interlock remains intact if this behavior has been specified on input parameter "S_AutoReset". The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or guarantees are in place that start interlocks are implemented at another location or with other methods.**

6.6.7.2.5 Start interlock after cold restart of safety controller

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

After enabling the function block and/or cold restarting the safety controller, "S_EnableSwitchOut" is only set to TRUE if "S_SafetyActive" indicates state TRUE and the function block detects the signal sequence from switch position 1 to switch position 2 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2".

When cold restarting the safety controller and/or enabling the function block, the signal combination for switch position 1 (enable switch not actuated) must be present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" on feedback from the selected safe operating mode ("S_SafetyActive" = TRUE). Otherwise, the function block outputs an error message and remains in the safe state ("S_EnableSwitchOut" = FALSE).

6.6.7.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

Danger!

Always validate the entire safety function!

6.6.7.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

Danger!

You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!
Check the connections when validating the sub-application!

6.6.7.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.7.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.7.3.4 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.7.4 Input parameters

6.6.7.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to support a start interlock after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
  On feedback of the selected safe operating mode ("S_SafetyActive" = TRUE) during a cold restart of the safety controller and/or when enabling the function block, "S_EnableSwitchCh1" and "S_EnableSwitchCh2" must be set to the signal combination for switch position 1 (enable switch not activated). Otherwise, the function block outputs an error message and remains in the safe state.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.7.4.2 S_SafetyActive

General function

- Control signal. Confirmation (feedback signal) that the selected safe operating mode is active.

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

Connect this input parameter with a signal that reports the selected safe operating mode.

Output signal "S_SafetyActive" of function block "SF_SafetyRequest" and/or signals of connected safe peripherals are suitable for connection depending on requirements.

You will define requirements and/or corrective measures based on the risk analysis you have performed. If the safe operating mode is safely implemented without confirmation in your application in accordance with your risk analysis, connect a static TRUE signal on input parameter "S_SafetyActive".

Information:

The corresponding operating mode (limiting the speed, energy or area of movement) must be selected outside of function block "SF_EnableSwitch". It is also your responsibility to connect the feedback signal that specifies whether the selected operating mode is active.

Function description

The signal connected to input parameter "S_SafetyActive" is processed by the function block.

Regardless whether the feedback signal is connected to the safe device over 1 or 2 channels, "S_SafetyActive" is only connected to one signal.

If the feedback signal is wired to the safe device over 2 channels, then dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_SafetyActive". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_SafetyActive". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The selected safe operating mode is active. The function block sets "S_EnableSwitchOut" to TRUE if the signal combination for switch position 2 is present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2".

FALSE

A safe operating mode is not active. The function block sets "S_EnableSwitchOut" to the safe state (FALSE) and remains there.

6.6.7.4.3 S_EnableSwitchCh1

General function

- Input for resulting signal of contacts E1 and E2 of the connected hand-operated, 3-position enable switch

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter with the resulting signal of contacts E1 and E2 of the 3-position enable switch.

Function description

The function block evaluates the signals of contacts E1 and E2 for switch positions 1, 2 and 3 present on input "S_EnableSwitchCh1".

6.6.7.4.4 S_EnableSwitchCh2

General function

- Input for resulting signal of contacts E3 and E4 of the connected hand-operated 3-position enable switch

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter with the resulting signal of contacts E3 and E4 of the 3-position enable switch.**

Function description

The function block evaluates the signals of contacts E3 and E4 for switch positions 1, 2 and 3 present on input "S_EnableSwitchCh2".

6.6.7.4.5 S_AutoReset

General function

- Specification of the start interlock if valid signals of the 3-position enable switch are present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" following an invalid signal sequence on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2"

Data type

- SAFEBOOL

Connection

- Variable or constant

> **Information:**
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block following a faulty and/or invalid signal combination on "S_EnableSwitchCh1" and "S_EnableSwitchCh2".

TRUE

If "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate valid signal levels from the 3-position enable switch, then a signal change from FALSE to TRUE on "Reset" is not required after a faulty or invalid signal combination on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2" in order to exit the error state.

> **Danger!**
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to be assigned value TRUE if there are assurances that a hazardous situation cannot occur after the function block has detected switch position 1 of the 3-position enable switch on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" or startup is prevented by other measures.**

FALSE

If "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate valid signal levels from the 3-position enable switch, then a signal change from FALSE to TRUE on "Reset" is required after a faulty or invalid signal combination on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2" in order to exit the error state.

> **Danger!**
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if it is ensured that a dangerous situation cannot occur after the function block has detected switch position 1 of the 3-position enable switch on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" or startup is prevented by other measures.**

6.6.7.4.6 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

> **Danger!**
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

> **Information:**
>
> **If the signal level for switch position 1 of the 3-position enable switch is present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2", the formal parameter is used to exit an error state that was caused by an impermissible signal combination on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2".**
> **Diagnostic messages continue to be updated.**

6.6.7.5 Output parameters

6.6.7.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.7.5.2 S_EnableSwitchOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected enable switch for the process being controlled.

The release signal is controlled based on the state of the enable switch and start interlock. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_EnableSwitchOut", this output is referred to as the "enable output".

Only use this signal to control the downstream process if a selected safe operating mode is active (limiting the speed, energy or area of movement).

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The enable switch is actuated ("S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 2).
- A start interlock is not active.
- The function block did not detect any faults.

Danger!

A TRUE signal on "S_EnableSwitchOut" is not permitted to allow or initiate a dangerous state on its own. For this, a conscious startup command independent of the enable switch is required. Any person present in the danger zone must carry an enable switch that can be used to prevent a hazardous state.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected the non-actuated enable switch ("S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 1).
- The function block detected the signal level of switch position 3 of the 3-position enable switch on "S_EnableSwitchCh1" and "S_EnableSwitchCh2".
- A FALSE signal is present on "S_SafetyActive".
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_EnableSwitchOut" is set to TRUE after an error only if input "S_SafetyActive" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	If "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 1 of the 3-position enable switch, then a signal change from FALSE to TRUE on "Reset" is required after a faulty or invalid signal combination on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2" in order to exit the error state.	...inactive.	No action on "Reset" is required...	• ...to exit the error if "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 1 of the 3-position enable switch. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 606: "SF_EnableSwitch": Input parameter "S_AutoReset"

6.6.7.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.7.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.7.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The enable switch connected to "S_EnableSwitchCh1" (TRUE) and "S_EnableSwitchCh2" (TRUE) is in switch position 2. The safe state (e.g. safe reduced speed) of the monitored area is reported using the signal connected to "S_SafetyActive" (TRUE).	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8004	The safe state is not reported using the signal connected to "S_SafetyActive". "S_SafetyActive" therefore indicates state FALSE. To exit this state, control "S_SafetyActive" using a signal that reports back the safe state (e.g. safely reduced speed) of the monitored area with a TRUE signal.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8005	The safe state (e.g. safe reduced speed) is reported using the signal connected to "S_SafetyActive". This state is exited after a safety controller cycle.	Intended event: • No measures are necessary if this diagnostic code is exited after a safety controller cycle. Unintended event: • Please contact B&R Support if this diagnostic code is permanently output by the function block. • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8006	The enable switch connected to "S_EnableSwitchCh1" (FALSE) and "S_EnableSwitchCh2" (TRUE) is in switch position 1. The safe state (e.g. safe reduced speed) is reported using the signal connected to "S_SafetyActive". This state is exited after a safety controller cycle.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8007	The enable switch connected to "S_EnableSwitchCh1" (FALSE) and "S_EnableSwitchCh2" (FALSE) is in switch position 3. The safe state (e.g. safe reduced speed) is reported using the signal connected to "S_SafetyActive". This state is exited after a safety controller cycle.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 607: "SF_EnableSwitch": Diagnostic codes

Code (hex)	Description	Corrective measures
C010	The function block detected an illegal combination for switch position 1 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" after the safe operating mode was reported on "S_SafetyActive".	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • To exit this diagnostic code, set the signal combination on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" to the value that is valid for switch position 1. To do so, both "S_EnableSwitchCh1" and "S_EnableSwitchCh2" must indicate TRUE.
C020	After an invalid signal combination for switch position 1 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2", the pending signal combination for switch position 1 is now valid.	Reset the function block.
C030	When detecting the signal combination for switch position 3 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2", the function block instead detected the signal combination for switch position 2, which is not permitted.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • To exit this diagnostic code, set the signal combination on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" to the value that is valid for switch position 3 or 1. The valid signal combination for switch position 1 is "S_EnableSwitchCh1" = FALSE and "S_EnableSwitchCh2" = TRUE. The valid signal combination for switch position 3 is "S_EnableSwitchCh1" = FALSE and "S_EnableSwitchCh2" = FALSE.
C040	After an invalid signal combination for switch position 3 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2", the pending signal combination is now valid.	Reset the function block.

Table 607: "SF_EnableSwitch": Diagnostic codes

6.6.7.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_AutoReset" = FALSE

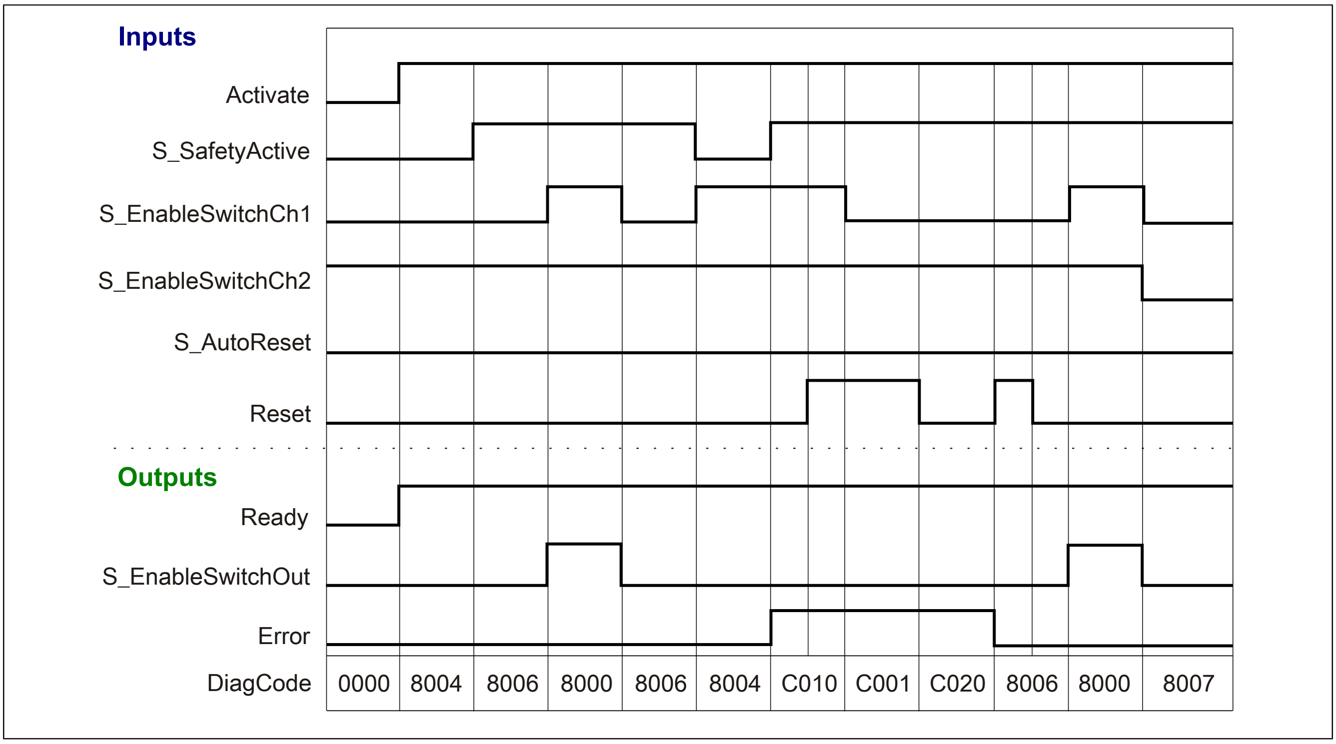


Figure 421: "SF_EnableSwitch": Signal sequence diagram 1

Signal sequence diagram 2

"S_AutoReset" = TRUE

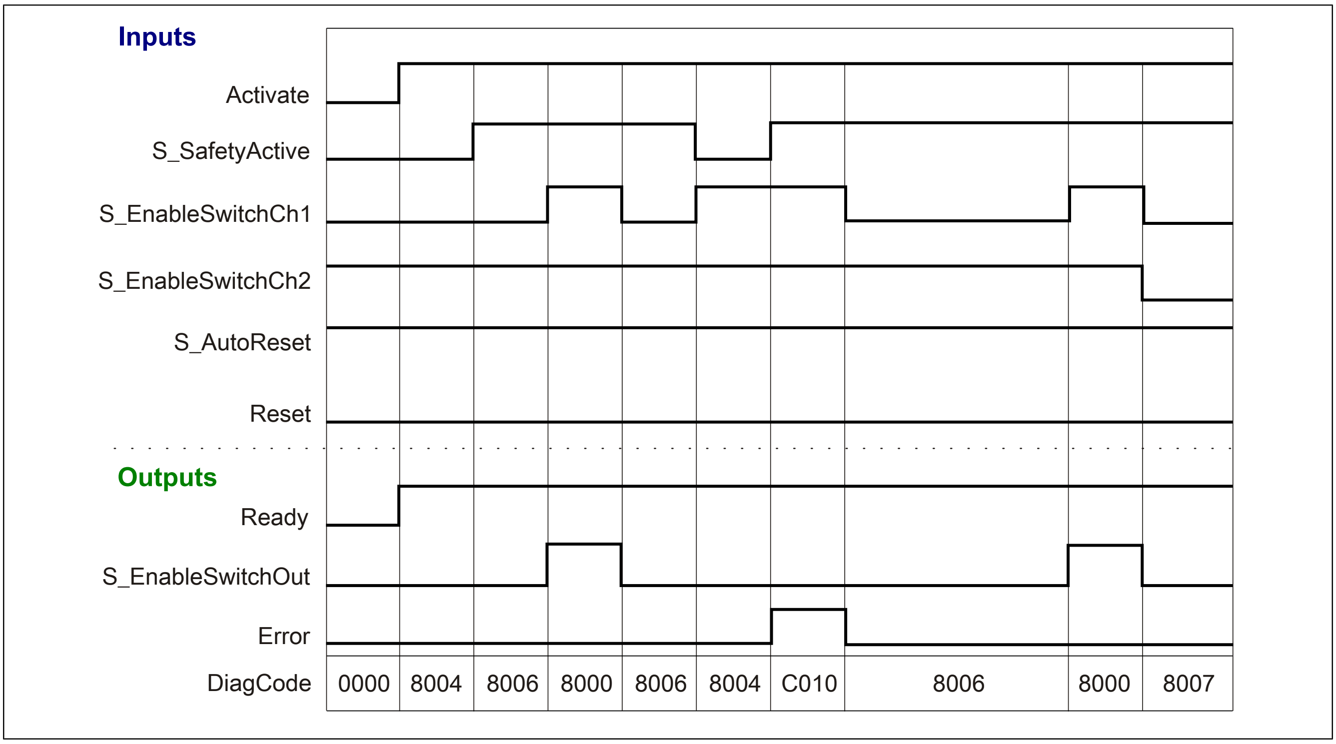


Figure 422: "SF_EnableSwitch": Signal sequence diagram 2

6.6.7.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement an enabling device.

The following example describes the function block connection for evaluating an enable switch.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.7.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "ES_S11".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant.

The start interlock is specified if the signals for switch position 1 of the 3-position enable switch are present on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" after an invalid signal sequence on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2".

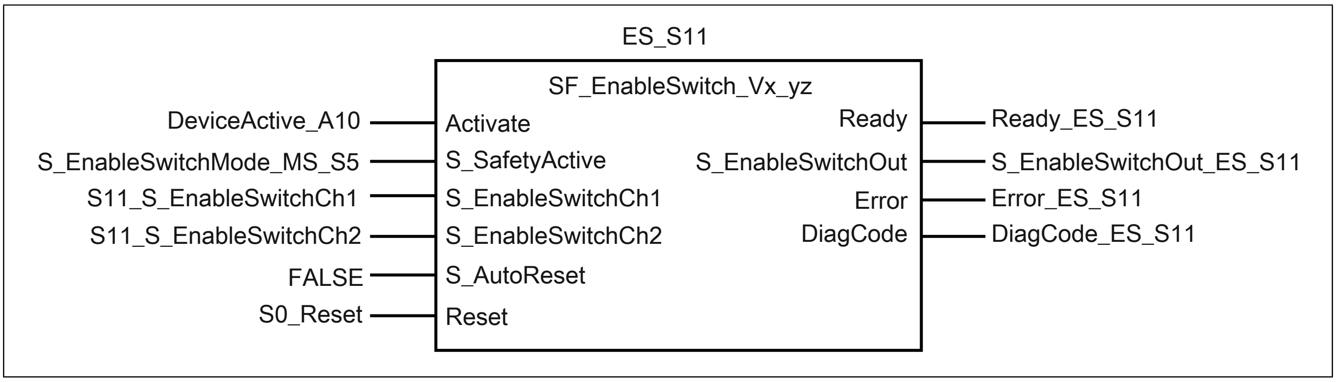


Figure 423: "SF_EnableSwitch": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device.
S_EnableSwitchMode_MS_S5	SAFEBOOL	Confirmation (feedback signal) that the selected safe operating mode is active
S11_S_EnableSwitchCh1	SAFEBOOL	Resulting signal of contacts E1 and E2 of the enable switch (TRUE in switch position 2)
S11_S_EnableSwitchCh2	SAFEBOOL	Resulting signal of contacts E3 and E4 of the enable switch (TRUE in switch position 1 and 2)
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after an invalid combination of signals from the enable switch
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 608: "SF_EnableSwitch": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_ES_S11	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EnableSwitchOut_ES_S11	SAFEBOOL	Release signal. The release signal is the safe enable signal of the connected enable switch for the process being controlled. The release signal is controlled based on the state of the enable switch and start interlock. In addition, the release signal controls the request for the stop function.
Error_ES_S11	BOOL	Error message from function block for further external processing
DiagCode_ES_S11	WORD	Diagnostic message from function block for further external processing

Table 609: "SF_EnableSwitch": Outputs connected to output parameters

6.6.7.7.2 Connecting the enable switch

This example illustrates the connection of the function block to the signals of an enable switch.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.7.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "ES_S11".

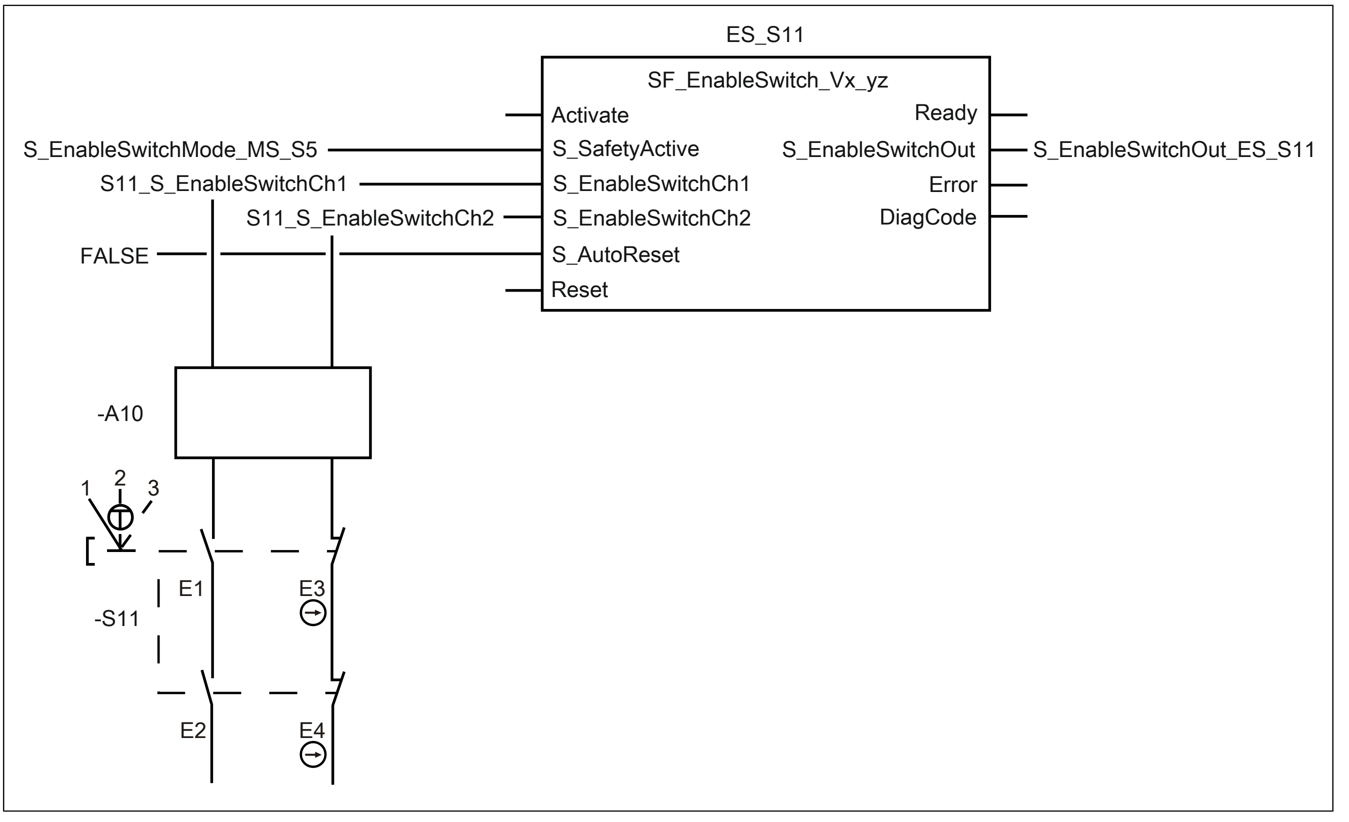


Figure 424: "SF_EnableSwitch": Connecting the enable switch

List of equipment

-S11	Enable switch
-A10	1-channel inputs of a safe input device

Connected inputs and outputs

S_EnableSwitchMode_MS_S5	Input on "S_SafetyActive"
S11_S_EnableSwitchCh1	Input on "S_EnableSwitchCh1"
S11_S_EnableSwitchCh2	Input on "S_EnableSwitchCh2"
S_EnableSwitchOut_ES_S11	Output on "S_EnableSwitchOut"

Description

In this example:

- Input "S_EnableSwitchMode_MS_S5" specifying that the selected safe operating mode is active and being monitored is connected to input parameter "S_SafetyActive".
- The resulting signal of normally open contacts E1 and E2 from the input of safe device "-A10" is connected to input "S11_S_EnableSwitchCh1".
- Input "S11_S_EnableSwitchCh1" is connected to input parameter "S_EnableSwitchCh1" for further processing.
- The resulting signal of normally closed contacts E3 and E4 from the input of safe device "-A10" is connected to input "S11_S_EnableSwitchCh2".
- Input "S11_S_EnableSwitchCh2" is connected to input parameter "S_EnableSwitchCh2" for further processing.
- Output parameter "S_EnableSwitchOut" is connected to output "S_EnableSwitchOut_ES_S11".
- Release signal "S_EnableSwitchOut_ES_S11" is the safe enable signal of the connected enable switch for the process being controlled. The release signal is controlled based on the state of the enable switch and start interlock. In addition, the release signal controls the request for the stop function.

6.6.7.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 60204	Enabling device	The function block evaluates the signals from a hand-operated 3-position enable switch. Switch position 1: • Off function of switch (actuator not actuated) Switch position 2: • Enable function (actuator actuated until the middle position) Switch position 3: • Off function of switch (actuator actuated past the middle position) Your enable switch must meet the specified requirements. Only when a signal change from switch position 1 to switch position 2 on "S_EnableSwitchCh1" and "S_EnableSwitchCh2" is detected does "S_EnableSwitchOut" switch from FALSE to TRUE and remain in this state until switch position 2 is no longer detected.
EN 60204	Lifting technical protective measures	The corresponding operating mode (limiting the speed, energy or area of movement) must be selected outside of function block "SF_EnableSwitch". The function block evaluates the selected safe operating mode on input parameter "S_SafetyActive". "S_EnableSwitchOut" can only be set to TRUE using "S_EnableSwitchCh1" and "S_EnableSwitchCh2" if "S_SafetyActive" indicates state TRUE.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restarting the safety controller • Enabling the function block "S_EnableSwitchCh1" and "S_EnableSwitchCh2" must indicate the signal level for switch position 1 of the connected enable switch (off function) after cold restarting the safety controller or enabling the function block. Otherwise, the function block will output an error message. If "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 1 after enabling the function block and/or cold restarting the safety controller, switch position 2 then triggers a TRUE signal on "S_EnableSwitchOut" if "S_SafetyActive" = TRUE. If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. The function block optionally supports (S_AutoReset = FALSE) a start interlock after a faulty or invalid signal combination on "S_EnableSwitchCh1" and/or "S_EnableSwitchCh2" if "S_EnableSwitchCh1" and "S_EnableSwitchCh2" indicate the signal level for switch position 1. To disable this optional start interlock, a signal change from FALSE to TRUE is required on "Reset". In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal. A TRUE signal on "S_EnableSwitchOut" is not permitted to allow or initiate a dangerous state on its own. For this, a conscious startup command independent of the enable switch is required. Any person present in the danger zone must carry an enable switch that can be used to prevent a hazardous state.
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_EnableSwitchOut") executes a category 0 stop.

Table 610: "SF_EnableSwitch": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.8 SF_Equivalent

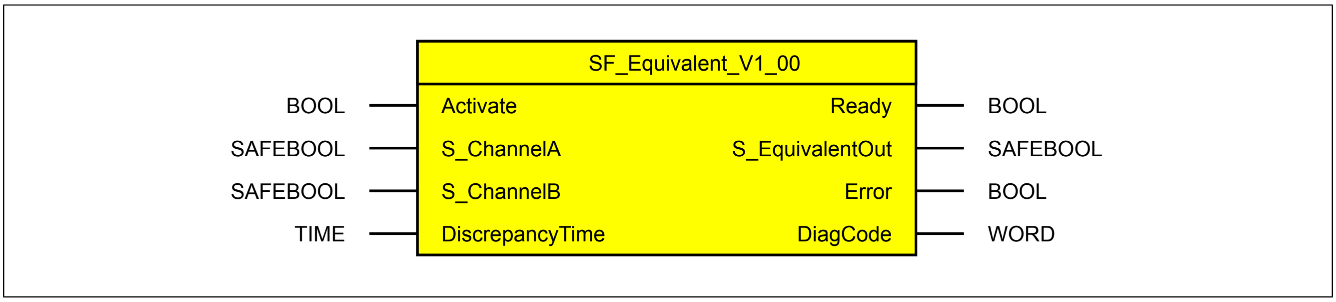


Figure 425: Function block "SF_Equivalent"

6.6.8.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_ChannelA	SAFEBOOL	Variable	State	FALSE	Input A (channel A)
S_ChannelB	SAFEBOOL	Variable	State	FALSE	Input B (channel B)
DiscrepancyTime	TIME	Constant	State	#0ms	Specification for the monitoring time of the switching operation on "S_ChannelA" and "S_ChannelB"

Table 611: "SF_Equivalent": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_EquivalentOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 612: "SF_Equivalent": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 613: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.8.2 Function

Function block "SF_Equivalent" supports a monitoring function with regard to the equivalence of 2 safe signal inputs.

This function block compares 2 safe signal inputs of a sensor connected over 2 channels for equivalent signal states. Discrepancy time monitoring starts after the state of one of the signal inputs has changed (input parameter "DiscrepancyTime"). During this time, the function block monitors whether a switching operation has taken place on the other signal input (switching function) and whether both signal inputs are switching symmetrically (switching symmetry).

If the signal switches for both signal inputs within the discrepancy time, then discrepancy time monitoring ends. An error message is reported if the discrepancy time is exceeded.

The two signal inputs are interdependent. For this reason, wire each of the two signal inputs either with normally closed or normally open contacts.

The result of the comparison is indicated on function block enable output "S_EquivalentOut". If the A signal input switches from TRUE to FALSE and/or the B signal input switches from TRUE to FALSE, then the enable output immediately switches to FALSE.

6.6.8.2.1 Discrepancy time monitoring

During the discrepancy time, both signal inputs are permitted to have different states. The function block does not detect this as an error.

Discrepancy time monitoring begins when the state of one of the signal inputs changes. The function block then detects an error if both signal inputs have different states after the discrepancy time elapses.

6.6.8.2.2 Switching states

Monitoring watches both signal inputs for a change from FALSE to TRUE as well as a change from TRUE to FALSE. For this reason, the signal inputs must be switched symmetrically.

6.6.8.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.8.3.1 Asymmetrical switching signals on signal inputs A and B

If the signals on signal inputs A and B are switched asymmetrically (due to faulty protective equipment, faulty switch, faulty cables, etc.), then the safe state (enable output "S_EquivalentOut" = FALSE) will be maintained once the protective equipment has been closed.

6.6.8.3.2 Different states on signal inputs A and B

Different states on signal inputs A and B (due to faulty protective equipment, faulty switch, faulty cables, etc.) after the specified discrepancy time has expired (input parameter "DiscrepancyTime") is reported as a fault by the function block.

6.6.8.3.3 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.8.3.4 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.8.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.8.4 Input parameters

6.6.8.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.8.4.2 S_ChannelA

General function

- Input A (channel A)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal (channel A) from a safety sensor connected over 2 channels. Input parameter "S_ChannelA" is then controlled via this signal.

The following combinations are allowed with interconnected signal inputs:

- **"S_ChannelA": Normally closed contact and "S_ChannelB": Normally closed contact, or**
- **"S_ChannelA": Normally open contact and "S_ChannelB": Normally open contact**

Information:

Connect a normally closed contact if you want to generate a TRUE signal on "S_EquivalentOut" when the contacts are not actuated ("idle current" principle). Please note that both signal inputs must be wired with the same type of contact.

Function description

The signal connected to input parameter "S_ChannelA" is processed by the function block.

Signal input A processes the state of a signal (channel A) from a safety sensor connected over 2 channels. This signal must indicate state TRUE in order to set enable output "S_EquivalentOut" to TRUE under consideration of the state on "S_ChannelB".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

If a normally closed contact is connected, it is not actuated.
If a normally open contact is connected, it is actuated.

FALSE

If a normally closed contact is connected, it is actuated.
If a normally open contact is connected, it is not actuated.

6.6.8.4.3 S_ChannelB

General function

- Input B (channel B)

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal (channel B) from a safety sensor connected over 2 channels. Input parameter "S_ChannelB" is then controlled via this signal.**
>
> **The following combinations are allowed with interconnected signal inputs:**
>
> - **"S_ChannelA": Normally closed contact and "S_ChannelB": Normally closed contact, or**
> - **"S_ChannelA": Normally open contact and "S_ChannelB": Normally open contact**

> ## Information:
>
> **Connect a normally closed contact if you want to generate a TRUE signal on "S_EquivalentOut" when the contacts are not actuated ("idle current" principle). Please note that both signal inputs must be wired with the same type of contact.**

Function description

The signal connected to input parameter "S_ChannelB" is processed by the function block.

Signal input B processes the state of a signal (channel B) from a safety sensor connected over 2 channels. This signal must indicate state TRUE in order to set enable output "S_EquivalentOut" to TRUE under consideration of the state on "S_ChannelA".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

If a normally closed contact is connected, it is not actuated.
If a normally open contact is connected, it is actuated.

FALSE

If a normally closed contact is connected, it is actuated.
If a normally open contact is connected, it is not actuated.

6.6.8.4.4 DiscrepancyTime

General function

- Specification for the monitoring time of the switching operation on "S_ChannelA" and "S_ChannelB"

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the time within which the switching operations on signal inputs "S_ChannelA" and "S_ChannelB" must take place in order to be detected as simultaneous and equivalent.

A signal tolerance during the switching operation of the contacts is possible due to mechanical influences. The specified "DiscrepancyTime" value limits the duration within which the two signals can be switched successively without this being detected as not simultaneous.

The limits for input parameter "DiscrepancyTime" must be defined and validated based on your application.

6.6.8.5 Output parameters

6.6.8.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.8.5.2 S_EquivalentOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal "S_EquivalentOut" controls the safety requirement of downstream safety-relevant function blocks such as "SF_EmergencyStop".

The release signal is controlled based on the state of the safety sensor. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_EquivalentOut", this output is referred to as the "enable output".

> **Danger!**
>
> **The release signal is not permitted to control the process directly since this function block supports monitoring, not a safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The function block has detected the equivalence of both signal inputs.
- "S_ChannelA" = TRUE and "S_ChannelB" = TRUE.
- The function block did not detect any faults.

FALSE

The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block has detected no equivalence of both signal inputs.
- "S_ChannelA" = FALSE and/or "S_ChannelB" = FALSE.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

6.6.8.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

> ## Information:
>
> **In order to exit an error state ("Error" = TRUE), "S_ChannelA" must be FALSE and "S_ChannelB" must be FALSE.**

6.6.8.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.8.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	• If this is a desired signal combination at the signal inputs, no action is required. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8001	Initialization of the function block after the function block has been enabled and/or both A and B signal inputs are FALSE. If one or both input signals indicate state TRUE, then this state is completed after one cycle of the safety controller.	• If the signal combination on the signal inputs is intended, set both input signals (A and B) of the connected peripheral to TRUE to set the enable output to TRUE. • If the signal combination on the signal inputs is unintended, check the connected peripheral and correct any faults.
8004	Only signal input A indicates state TRUE. Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set signal input B to state TRUE also. • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
8005	Only one signal input (A or B) indicates state TRUE. Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set the other signal input to state TRUE also. • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
8014	Only signal input B indicates state TRUE. Discrepancy time monitoring of the two signal inputs is started. If the discrepancy time is exceeded in this state, the function block transitions to an error state. Discrepancy time monitoring is stopped again when leaving this state.	• Set signal input A to state TRUE also. • Check the value on input parameter "DiscrepancyTime" and the connected peripheral.
C001	On a request to set the enable output to TRUE, the corresponding signal was only present on the A signal input. Discrepancy time monitoring was started and has expired. When the discrepancy time expired, the A signal input switched while the B signal input did not.	• Check the connected peripheral (signal input B) and correct any faults. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set both signal inputs to state FALSE.
C002	On a request to set the enable output to TRUE, the corresponding signal was only present on the B signal input. Discrepancy time monitoring was started and has expired. When the discrepancy time expired, the B signal input switched while the A signal input did not.	• Check the connected peripheral (signal input A) and remove any errors. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set both signal inputs to state FALSE.
C003	Both signal inputs did not change their state after the enable output was cut off. The discrepancy time started in this state is expired.	• Check the connected peripheral (signal input A and signal input B) and correct any faults. • Check the specified value on input parameter "DiscrepancyTime" and correct it if the values are too short. • Set both signal inputs to state FALSE.

Table 614: "SF_Equivalent": Diagnostic codes

6.6.8.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

Startup, normal operation

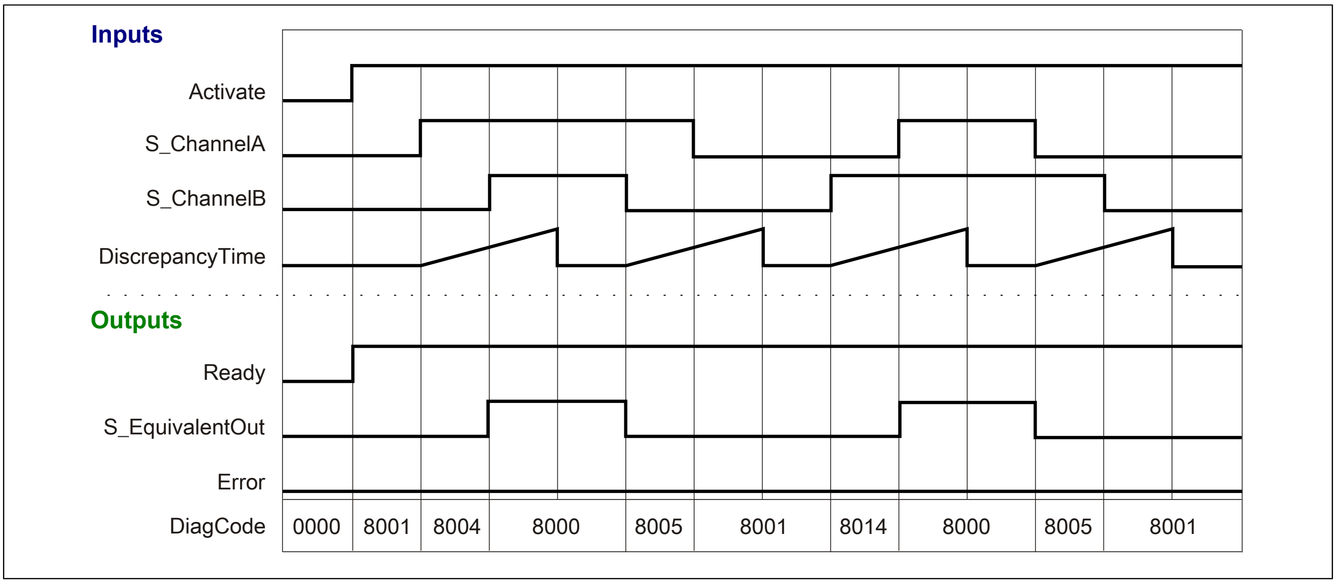


Figure 426: "SF_Equivalent": Signal sequence diagram 1

Signal sequence diagram 2

Discrepancy time expired, normal operation

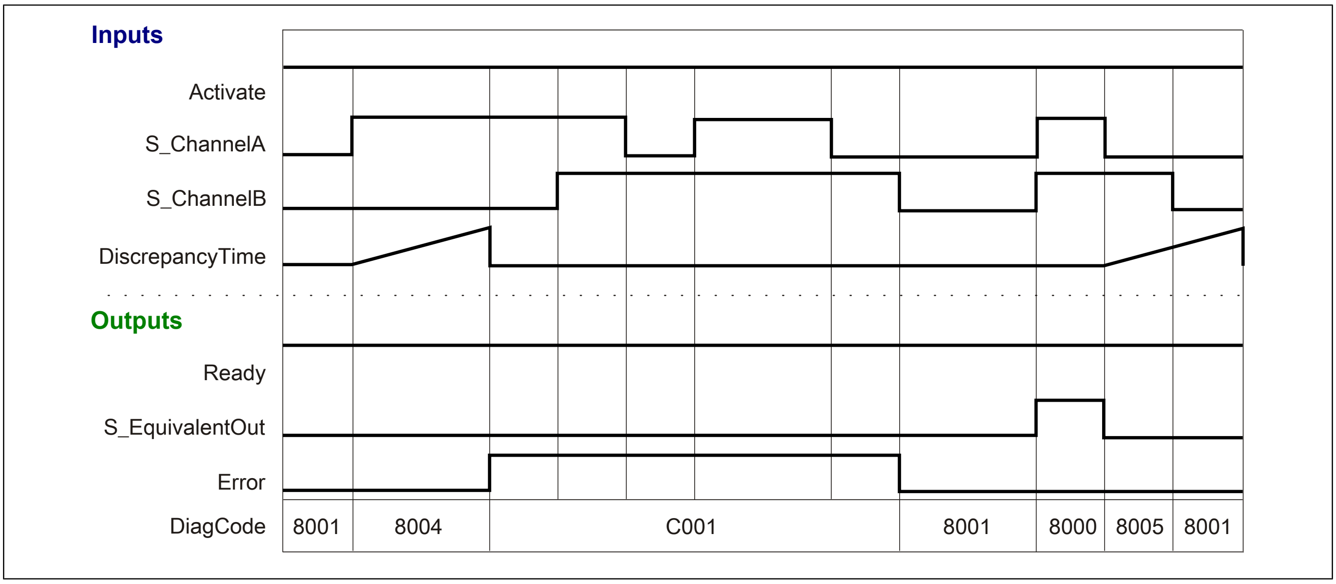


Figure 427: "SF_Equivalent": Signal sequence diagram 2

6.6.8.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement equivalent evaluation of signals.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.8.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "EV_S5".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each formal parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

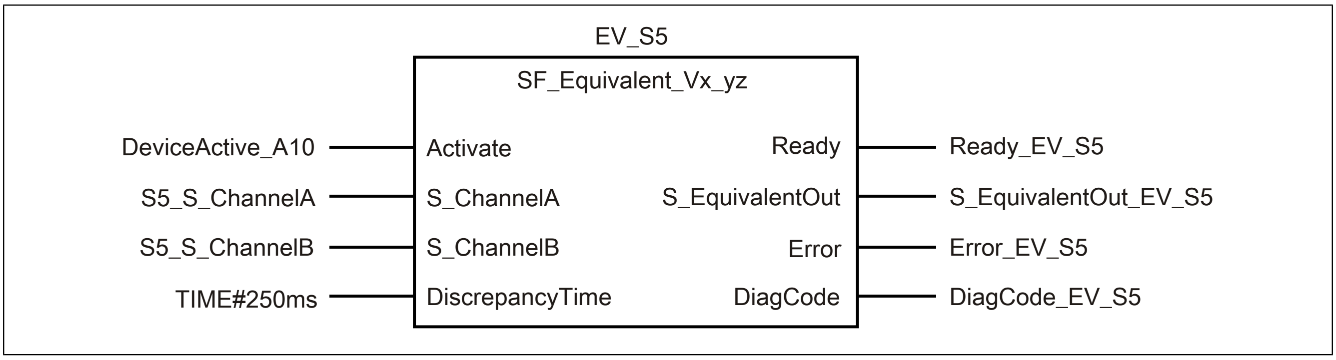


Figure 428: "SF_Equivalent": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the contacts of the safety switch are connected.
S5_S_ChannelA	SAFEBOOL	Input A (channel A). This input is connected to channel A of a safety switch.
S5_S_ChannelB	SAFEBOOL	Input B (channel B). This input is connected to channel B of a safety switch.
TIME#250ms on "DiscrepancyTime"	TIME	Maximum permitted discrepancy time

Table 615: "SF_Equivalent": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_EV_S5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_EquivalentOut_EV_S5	SAFEBOOL	Release signal. The release signal is the resulting signal of both input signals and used for further processing in the safety application on the safety controller.
Error_EV_S5	BOOL	Error message from function block for further external processing
DiagCode_EV_S5	WORD	Diagnostic message from function block for further external processing

Table 616: "SF_Equivalent": Outputs connected to output parameters

6.6.8.7.2 Evaluation of 2 equivalent safe input signals

This example illustrates the connection of the function block when controlling the 1-channel equivalent signals of a safe input device.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.8.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "EV_S5".

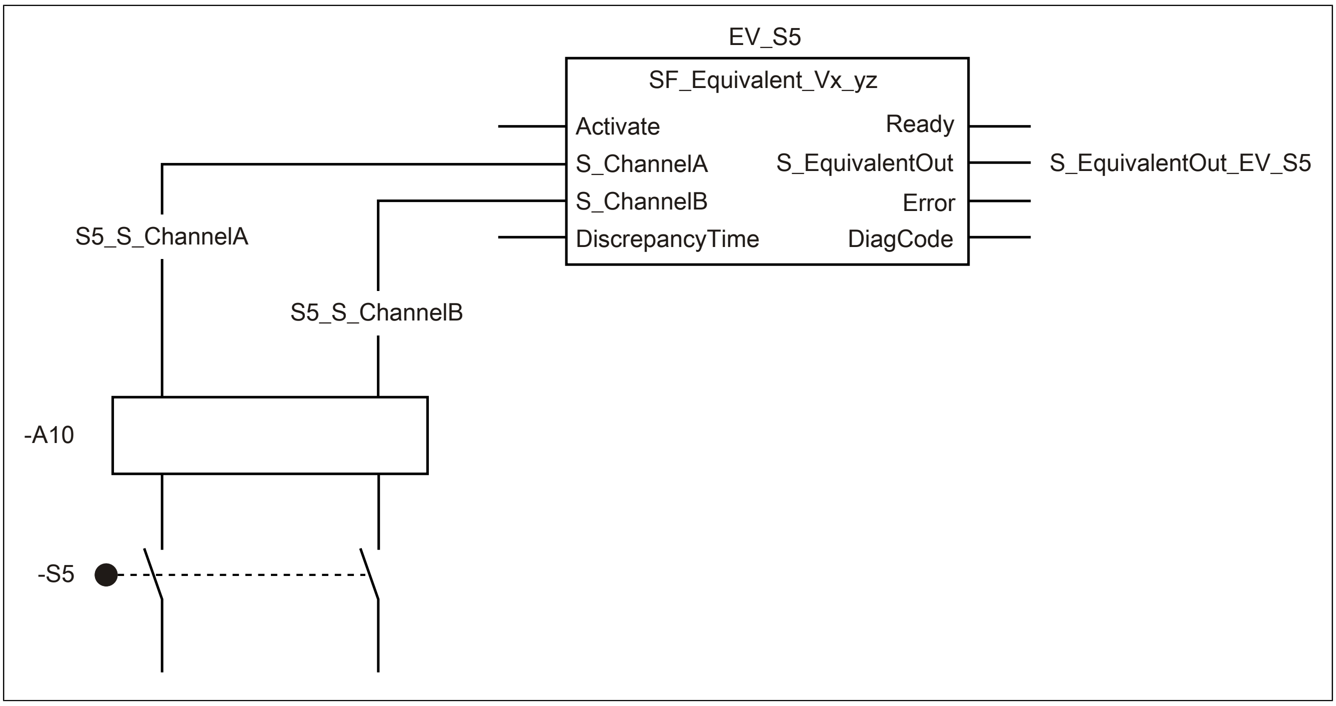


Figure 429: "SF_Equivalent": Evaluation of 2 equivalent safe input signals (1 input device)

List of equipment

-S5	Safety switch. 2x normally open contact.
-A10	2x 1-channel safe inputs of a safe input device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S5_S_ChannelA	Input on "S_ChannelA"
S5_S_ChannelB	Input on "S_ChannelB"
S_EquivalentOut_EV_S5	Output on "S_EquivalentOut"

Description

In this example:

- The signal from the input A of safe input device "-A10" is connected to input "S5_S_ChannelA" and connected to function block input parameter "S_ChannelA" for further processing.
- The signal from input B of safe input device "-A10" is connected to input "S5_S_ChannelB" and connected to function block input parameter "S_ChannelB" for further processing.
- Output parameter "S_EquivalentOut" is connected to output "S_EquivalentOut_EV_S5".
- Output "S_EquivalentOut_EV_S5" is the resulting signal of both input signals and used for further processing in the safety application on the safety controller.

6.6.8.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 954-1	General safety guidelines	Monitoring of equivalent switching behavior of 2 signals
EN 954-1	Error detection for CAT 3 and CAT 4	Monitoring of equivalent switching behavior of 2 signals

Table 617: "SF_Equivalent": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.9 SF_ESPE

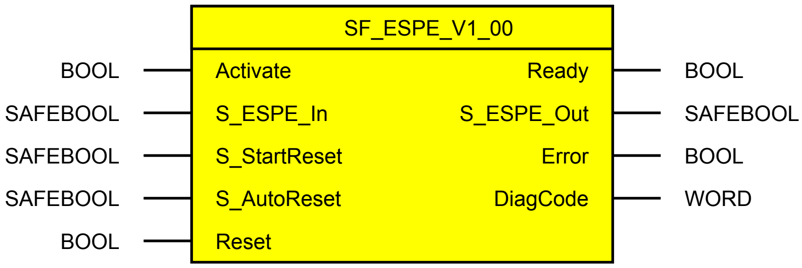


Figure 430: Function block "SF_ESPE"

6.6.9.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_ESPE_In	SAFEBOOL	Variable	State	FALSE	Control signal. State of the electro-sensitive protective equipment.
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after a signal change from FALSE to TRUE on "S_ESPE_In"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 618: "SF_ESPE": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_ESPE_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 619: "SF_ESPE": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 620: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.9.2 Function

Function block "SF_ESPE" is used to support an electro-sensitive protective equipment function in an application.

This function block is a safety-relevant function block for monitoring electro-sensitive protective equipment.

6.6.9.2.1 Triggering the electro-sensitive protective equipment

If electro-sensitive protective equipment is triggered in the application, the function block ensures that function block release signal "S_ESPE_Out" is set to FALSE.

6.6.9.2.2 Resetting the electro-sensitive protective equipment

If triggered electro-sensitive protective equipment has been reset in the application, the function block can optionally (see start interlock) ensure within the safety control system that release signal "S_ESPE_Out" is not set to TRUE solely by this reset. Additional manual intervention on input parameter "Reset" is required for this (see start interlock).

6.6.9.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after a signal change from FALSE to TRUE on the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.9.2.4 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.9.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.9.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.9.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.9.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the state of the release signal depends on the state of the electro-sensitive protective equipment connected to the function block as well as on how the optional start interlock is specified for when the signal of the electro-sensitive protective equipment changes (FALSE → TRUE).

6.6.9.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.9.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.9.4 Input parameters

6.6.9.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.9.4.2 S_ESPE_In

General function

- Control signal. State of the electro-sensitive protective equipment.

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the electro-sensitive protective equipment over 1 or 2 channels. Input parameter "S_ESPE_In" is then controlled via this signal.**

Function description

The signal connected to input parameter "S_ESPE_In" is processed by the function block.

The signal input processes the state of the electro-sensitive protective equipment.

Regardless whether the electro-sensitive protective equipment is connected to the safe device over 1 or 2 channels, "S_ESPE_In" is only connected to one signal of the electro-sensitive protective equipment.

If electro-sensitive protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on one signal to "S_ESPE_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on one signal to "S_ESPE_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The electro-sensitive protective equipment was not triggered.

After a signal change from FALSE to TRUE on input parameter "S_ESPE_In", the function block optionally supports a start interlock ("S_AutoReset" = FALSE). An active start interlock is indicated accordingly on output parameter "DiagCode". While the start interlock is active, enable output "S_ESPE_Out" remains set to FALSE. The start interlock is reset on a rising edge of "Reset" ("Reset": FALSE → TRUE). The reset changes enable output "S_ESPE_Out" from FALSE to TRUE.

FALSE

The electro-sensitive protective equipment was triggered, the wiring for the electro-sensitive protective equipment is interrupted or the safe device connected to the electro-sensitive protective equipment is shut down or defective.

Enable output "S_ESPE_Out" is set to FALSE; output parameter "DiagCode" is set accordingly.

6.6.9.4.3 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.9.4.4 S_AutoReset

General function

- Specification of the start interlock after a signal change from FALSE to TRUE on "S_ESPE_In"

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after a signal change from FALSE to TRUE on safe input parameter "S_ESPE_In" (resetting the electro-sensitive protective equipment).

TRUE

After a signal change from FALSE to TRUE on safe input parameter "S_ESPE_In", the function block does not support start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After a signal change from FALSE to TRUE on safe input parameter "S_ESPE_In", the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.9.4.5 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.9.5 Output parameters

6.6.9.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.9.5.2 S_ESPE_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the electro-sensitive protective equipment for the process being controlled.

The release signal is controlled based on the state of the electro-sensitive protective equipment and start interlock. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_ESPE_Out", this output is referred to as the "enable output".

Release signal "S_ESPE_Out" can be used for subsequent process control.

> **Danger!**
>
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The electro-sensitive protective equipment was not triggered ("S_ESPE_In" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected triggered electro-sensitive protective equipment ("S_ESPE_In" = FALSE).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_ESPE_Out" is set to TRUE only if input "S_ESPE_Int" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	After a signal change from FALSE to TRUE on safe input "S_ESPE_In", the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_ESPE_Out" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 621: "SF_ESPE": Input parameter "S_AutoReset"/"S_StartReset"

6.6.9.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.9.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.9.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The electro-sensitive protective equipment was not triggered.	No corrective measures are required.
8001	Initialization of the function block after it has been enabled. The initialization of the function block is completed after one safety controller cycle.	No corrective measures are required.
8002	The start interlock to be carried out after enabling the function block is active ("S_StartReset" = FALSE). The function block detected the following after being enabled: The electro-sensitive protective equipment was triggered, the wiring for the electro-sensitive protective equipment is faulty or the safe device connected to the electro-sensitive protective equipment is shut down or defective.	• Revoke the safety request of the electro-sensitive protective equipment. • For untriggered electro-sensitive protective equipment, check the wiring or the safe device connected to the electro-sensitive protective equipment.
8003	The safety request of the electro-sensitive protective equipment has been revoked. The function block's start interlock is active. ("S_StartReset" = FALSE)	Reset the function block.
8004	The start interlock to be carried out after enabling the function block is not active ("S_StartReset" = TRUE) and the function block was enabled, or enable output "S_ESPE_Out" was switched off by the state of signal "S_ESPE_In" = FALSE, which originates from the electro-sensitive protective equipment. The electro-sensitive protective equipment was triggered, the wiring for the electro-sensitive protective equipment is faulty or the safe device connected to the electro-sensitive protective equipment is shut down or defective.	• Revoke the safety request of the electro-sensitive protective equipment. • For untriggered electro-sensitive protective equipment, check the wiring or the safe device connected to the electro-sensitive protective equipment.
8005	The safety request of the electro-sensitive protective equipment has been revoked. The function block's start interlock is active. ("S_AutoReset" = FALSE)	Reset the function block.
C001	After resetting the safety request from the electro-sensitive protective equipment, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	After resetting the safety request from the electro-sensitive protective equipment, the function block detected a static TRUE signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 622: "SF_ESPE": Diagnostic codes

6.6.9.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

Startup, reset, normal operation, safety request, restart

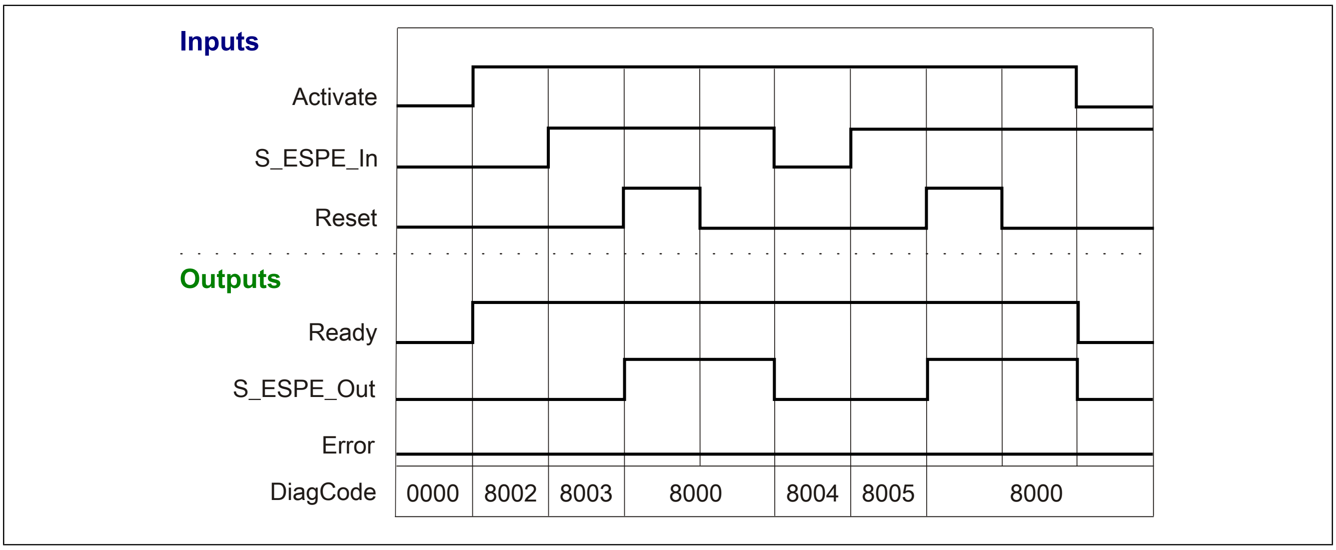


Figure 431: "SF_ESPE": Signal sequence diagram 1

Signal sequence diagram 2

"S_StartReset" = TRUE
"S_AutoReset" = FALSE

Startup, normal operation, safety request, restart

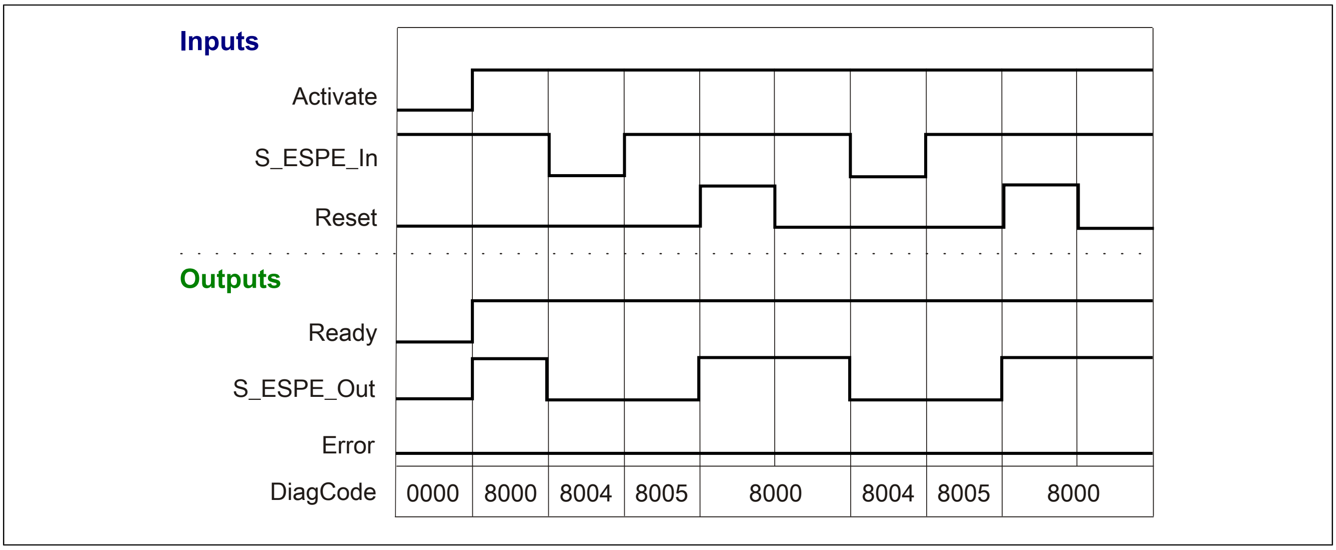


Figure 432: "SF_ESPE": Signal sequence diagram 2

Signal sequence diagram 3

"S_StartReset" = FALSE
"S_AutoReset" = TRUE

Startup, normal operation, safety request, restart

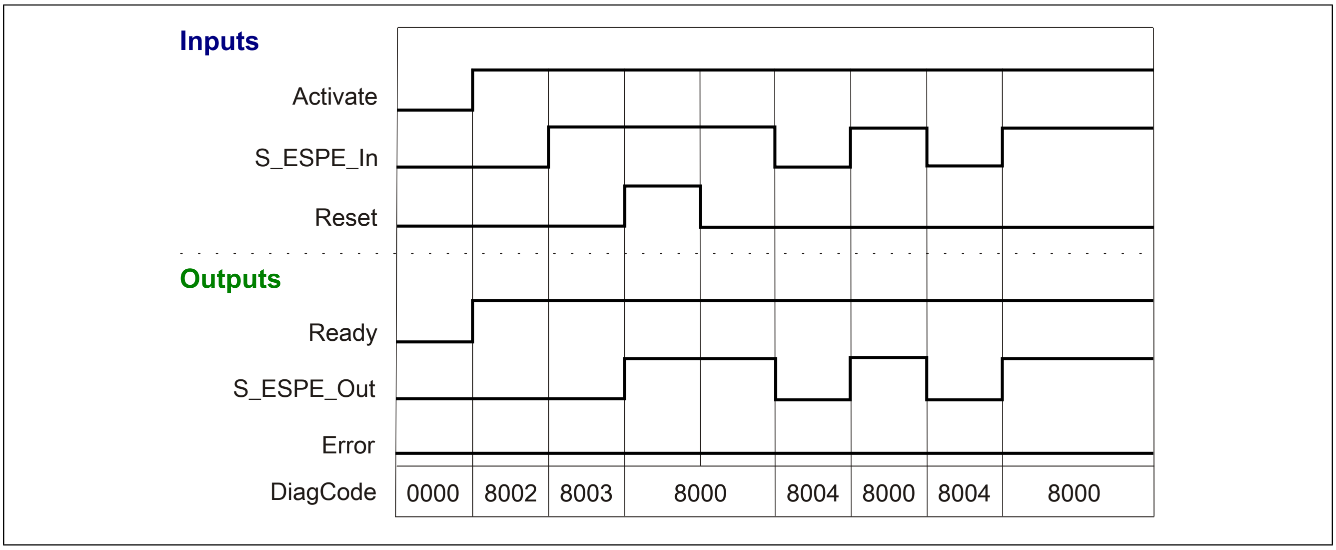


Figure 433: "SF_ESPE": Signal sequence diagram 3

6.6.9.7 Application examples

This chapter illustrates possible applications in which the function block can be used to implement a 1-channel or 2-channel evaluation of electro-sensitive protective equipment (ESPE).

The following examples describe function block connections when controlling with the following:

- The signal from EPSE connected over 1 channel (see section 6.6.9.7.2 "EPSE, connected over 1 channel")
- The signals from 2-channel, antivalent connected EPSE (see section 6.6.9.7.3 "EPSE connected over 2 channels, antivalent connection")

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.9.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "ESPE_S3".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signal on "S_ESPE_In", a rising edge on input parameter "Reset" is required in order for enable output "S_ESPE_Out" to be enabled.

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after a change of the safe input signal on "S_ESPE_In" (FALSE → TRUE). In addition to the safe input signal on "S_ESPE_In", a rising edge on input parameter "Reset" is required in order for enable output "S_ESPE_Out" to be enabled.

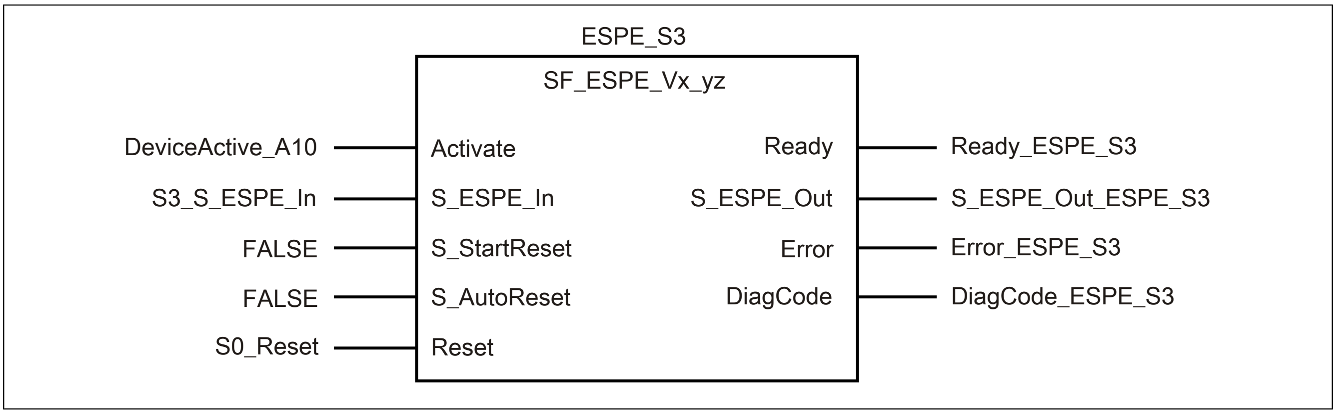


Figure 434: "SF_ESPE": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the electro-sensitive protective equipment is connected.
S3_S_ESPE_In	SAFEBOOL	Control signal of electro-sensitive protective equipment from a safe input device. The signal comes from a 1-channel or 2-channel input of a safe input device. The evaluation of dual-channel redundancy does not take place in the function block.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after resetting the electro-sensitive protective equipment
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 623: "SF_ESPE": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_ESPE_S3	BOOL	Function block active / Function block inactive. Message for further external processing.
S_ESPE_Out_ESPE_S3	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
Error_ESPE_S3	BOOL	Error message from function block for further external processing
DiagCode_ESPE_S3	WORD	Diagnostic message from function block for further external processing

Table 624: "SF_ESPE": Outputs connected to output parameters

6.6.9.7.2 EPSE, connected over 1 channel

This example illustrates the connection of the function block when controlling using the signal from electro-sensitive protective equipment connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.9.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "ESPE_S3".

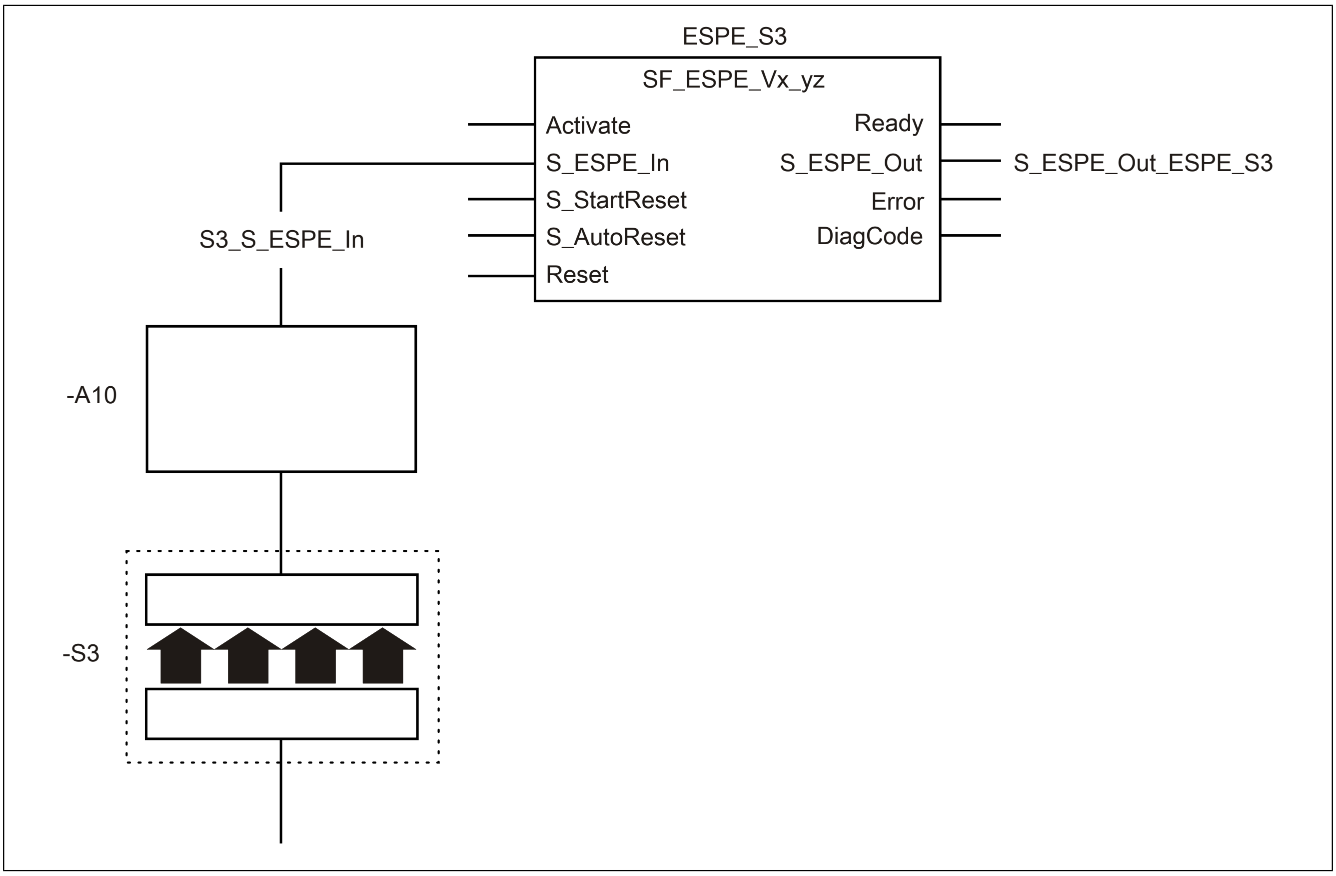


Figure 435: "SF_ESPE": EPSE, connected over 1 channel

List of equipment

-S3	Electro-sensitive protective equipment (1-channel)
-A10	1-channel input of a safe input device

Connected inputs and outputs

S3_S_ESPE_In	Input on "S_ESPE_In"
S_ESPE_Out_ESPE_S3	Output on "S_ESPE_Out"

Description

In this example:

- The signal from the 1-channel safe input of safe input device "-A10" is connected to input "S3_S_ESPE_In".
- Input "S3_S_ESPE_In" is connected to function block input parameter "S_ESPE_In" for further processing.
- Output parameter "S_ESPE_Out" is connected to output "S_ESPE_Out_ESPE_S3".
- Output "S_ESPE_Out_ESPE_S3" is used as a release signal to control the process in consideration of other safety functions.

6.6.9.7.3 EPSE connected over 2 channels, antivalent connection

This example illustrates the connection of the function block when controlling using the signal from electro-sensitive protective equipment connected over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.9.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "ESPE_S3".

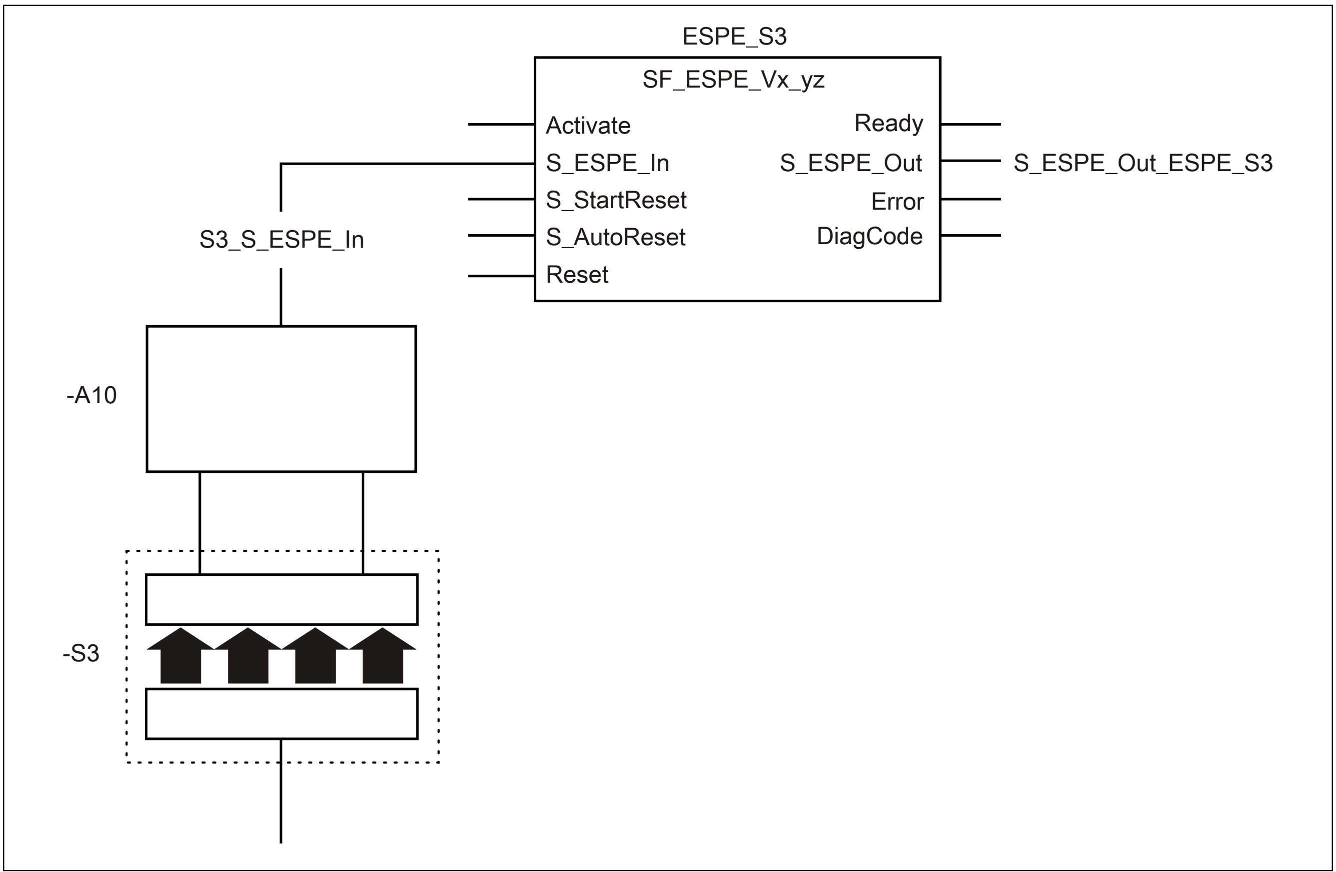


Figure 436: "SF_ESPE": EPSE connected over 2 channels, antivalent connection

List of equipment

-S3	Electro-sensitive protective equipment (2-channel, antivalent)
-A10	2-channel input of a safe input device (antivalent)

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S3_S_ESPE_In	Input on "S_ESPE_In"
S_ESPE_Out_ESPE_S3	Output on "S_ESPE_Out"

Description

In this example:

- The resulting signal of the inputs from safe input device "-A10" is connected to input "S3_S_ESPE_In".
- Input "S3_S_ESPE_In" is connected to function block input parameter "S_ESPE_In" for further processing.
- Output parameter "S_ESPE_Out" is connected to output "S_ESPE_Out_ESPE_S3".
- Output "S_ESPE_Out_ESPE_S3" is used as a release signal to control the process in consideration of other safety functions.

6.6.9.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 61496-1	Function of the start interlock	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_AutoReset" = FALSE) • Enabling of the function block ("S_StartReset" = FALSE) • A TRUE signal after the safety function is triggered You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_AutoReset" = FALSE) • Enabling of the function block ("S_StartReset" = FALSE) • A TRUE signal after the safety function is triggered If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_ESPE_Out") executes a category 0 stop.

Table 625: "SF_ESPE": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.10 SF_GuardLocking

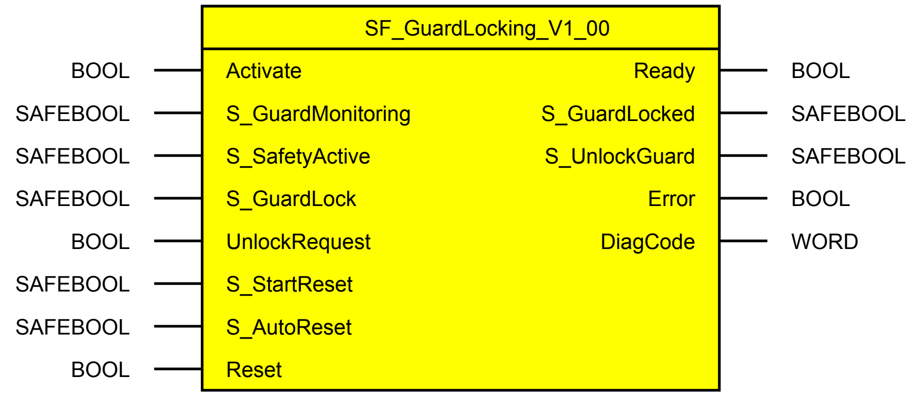


Figure 437: Function block "SF_GuardLocking"

6.6.10.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_GuardMonitoring	SAFEBOOL	Variable	State	FALSE	Input for the state of the protective equipment
S_SafetyActive	SAFEBOOL	Variable	State	FALSE	Input for the state of the safeguarded area
S_GuardLock	SAFEBOOL	Variable	State	FALSE	Input for the state of the guard locking device on the protective equipment
UnlockRequest	BOOL	Variable	State/ Edge	FALSE	Input for the request to unlatch the guard locking device on the protective equipment
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification for the start interlock after latching the guard locking device on the closed protective equipment
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 626: "SF_GuardLocking": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_GuardLocked	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
S_UnlockGuard	SAFEBOOL	Variable	State	FALSE	Unlatch signal for the guard locking device on the protective equipment
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 627: "SF_GuardLocking": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 628: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.10.2 Function

Function block "SF_GuardLocking" is used to support an interlocking guard with guard locking device function (safety door monitoring) in an application.

This function block is a safety-relevant function block for monitoring protective equipment with a 4-stage interlocking device.

6.6.10.2.1 Resetting the protective equipment's guard locking device

The protective equipment's guard locking device is only permitted to be reset upon detection that the hazard has been eliminated from the area safeguarded by the protective equipment. That the hazard has been eliminated (e.g. bringing the machine or system to standstill) is determined in the safety application. The function block evaluates this safe state via input parameter "S_SafetyActive".

The function block only sets output parameter "S_UnlockGuard" to TRUE when a TRUE signal is present on "S_SafetyActive" and the function block detects a rising edge on input parameter "UnlockRequest" (request to reset). This output parameter controls the coil on the guard locking device mechanism used to unlatch the guard locking device on the protective equipment.

In order to maintain the unlatched state, input parameter "UnlockRequest" must indicate a static TRUE signal after a rising edge.

6.6.10.2.2 Opening the protective equipment

After the guard locking device has been reset, the protective equipment can be opened. If the protective equipment is connected properly to the function block, the function block evaluates the opened state of the protective equipment using input parameter "S_GuardMonitoring". If the protective equipment properly connected to the function block is open, the function block sets enable output "S_GuardLocked" to FALSE (safe state).

The hazard elimination detected in the safety application must continue to exist in the open state of the protective equipment ("S_SafetyActive" = TRUE). Otherwise, the function block will switch to an error state. The safe state on enable output "S_GuardLocked" is maintained.

6.6.10.2.3 Closing the protective equipment

The protective equipment properly connected to the function block can be closed at any time.

Function block input parameter "S_GuardMonitoring" evaluates the closed state of the protective equipment properly connected to the function block.

If the reset request is still present (TRUE) on "UnlockRequest" after the protective equipment is closed, then the protective equipment can be opened again. Otherwise ("UnlockRequest" = FALSE), the function block requests the latching of the protective equipment's guard locking device ("S_UnlockGuard" = FALSE).

6.6.10.2.4 Latching the protective equipment's guard locking device

If the protective equipment is closed and the reset of the guard locking device is no longer being requested on input parameter "UnlockRequest" (i.e. request to latch), then output parameter "S_UnlockGuard" is set to FALSE. This output parameter must be used to control the coil on the guard locking device to latch the guard locking device. Input parameter "S_GuardLock" evaluates the state of the protective equipment properly connected to the function block.

If unlatched protective equipment is latched in the safety application, the function block can optionally (see start interlock) ensure within the safety control system that enable output "S_GuardLocked" is not set to TRUE solely by this latching. Additional manual intervention on function block input parameter "Reset" is required for this (see start interlock).

"S_GuardLock" must indicate state TRUE in the state of the function block in which the protective equipment's guard locking device is latched. If the function block detects FALSE on "S_GuardLock" in this state, however, then enable output "S_GuardLocked" is set to FALSE (safe state). This state is maintained and output as a fault by the function block.

6.6.10.2.5 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.10.2.6 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.10.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.10.3.1 Invalid switch or coil signal in latched state of protective equipment

In order to set enable output "S_GuardLocked" to TRUE, the protective equipment must be closed and the guard locking device latched.

In the closed and latched state, the signals on input parameters "S_GuardMonitoring" and "S_GuardLock" must indicate state TRUE. If the function block detects a FALSE signal on input parameter "S_GuardMonitoring" and/or input parameter "S_GuardLock", enable output "S_GuardLocked" is set to FALSE (safe state).

6.6.10.3.2 Invalid signal from hazard elimination determined in the safety application while protective equipment is open

The function block interprets a FALSE signal on input parameter "S_GuardMonitoring" as open protective equipment.

If there is a FALSE signal on this input parameter, the function block must detect the elimination of the hazard ("S_SafetyActive" = TRUE). If "S_SafetyActive" incorrectly indicates state FALSE, the function block initiates the error state, which is then maintained. Enable output "S_GuardLocked" is set to FALSE (safe state).

6.6.10.3.3 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.10.3.4 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.10.3.5 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the state of the release signal depends on the state of the protective equipment connected to the function block as well as on how the optional start interlock is specified for when the TRUE signal returns to the protective equipment.

6.6.10.3.6 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.10.3.7 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.10.4 Input parameters

6.6.10.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.10.4.2 S_GuardMonitoring

General function

- Input for the state of the protective equipment

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal of a safe input device that is connected to a switch on the protective equipment (e.g. safety door or standalone, possibly positively driven, 1-channel or 2-channel position switch). Input parameter "S_GuardMonitoring" is then controlled using this signal.**
>
> **Implement monitoring of dual-channel redundancy for the position switch in the safety application, e.g. in the safe input device. Input parameter "S_GuardMonitoring" is set using the resulting monitoring signal (e.g. from function block "SF_GuardMonitoring").**

Function description

The signal connected to input parameter "S_GuardMonitoring" is processed by the function block.

The signal input processes the state of the protective equipment (open/closed).

This signal must indicate state TRUE in order to set enable output "S_GuardLocked" to TRUE under consideration of the other input signal combination.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The protective equipment is closed.

FALSE

The protective equipment is open, the wiring for the protective equipment is interrupted or the safe input device connected to this protective equipment is shut down or defective.

6.6.10.4.3 S_SafetyActive

General function

- Input for the state of the safeguarded area (hazard eliminated / hazard not eliminated)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect the input parameter to a signal that indicates the state of the safeguarded area (e.g. safe standstill monitor connected to a safe input device or output on a safe timer).

Function description

Determine the state of the safeguarded area in the safety application, e.g. using a safe standstill monitor or timer.

It is only possible to set output parameter "S_UnlockGuard" to TRUE if the function block detects that the hazard has been eliminated in the safeguarded area and the other input signal combination is valid for this. The properly connected guard locking device is unlatched with output parameter "S_UnlockGuard" set to TRUE.

TRUE

The elimination of a hazard in the safeguarded area was determined outside the function block.

FALSE

The elimination of a hazard in the safeguarded area was not determined outside the function block, the wiring for the evaluating device (e.g. stall detection) is interrupted, the evaluating device is defective or the safe input device connected to the input parameter is shut down or defective.
If a timer is connected to determine the elimination of the hazard, then it has not yet expired.

6.6.10.4.4 S_GuardLock

General function

- Input for the state of the guard locking device on the protective equipment

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe input device that is connected to the latching/unlatching feedback signal of the protective equipment over 1 or 2 channels. Input parameter "S_GuardLock" is then controlled using this signal.

Implement monitoring of dual-channel redundancy for the feedback signal in the safety application, e.g. in the safe input device. Input parameter "S_GuardLock" is set using the resulting monitoring signal.

Function description

The signal connected to input parameter "S_GuardLock" is processed by the function block.

The signal input processes the state of the protective equipment's guard locking device (latched/unlatched).

This signal must indicate state TRUE in order to set enable output "S_GuardLocked" to TRUE under consideration of the other input signal combination.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The protective equipment is latched.

FALSE

The protective equipment is not locked, the wiring for the protective equipment is interrupted or the safe input device connected to the protective equipment is shut down or defective.

6.6.10.4.5 UnlockRequest

General function

- Input for the request to unlatch the guard locking device on the protective equipment

Data type

- BOOL

Connection

- Variable

Function description

If the protective equipment is closed and the protective equipment's guard locking device is latched, a signal change on "UnlockRequest" (FALSE → TRUE) will initiate a request for unlatching the guard locking device if the function block detects on "S_SafetyActive" the signal determined in the safety application that the hazard in the safeguarded area has been eliminated. Accordingly, output parameter "S_UnlockGuard" is set to TRUE, and enable output "S_GuardLocked" is set to FALSE. In order to maintain the request for unlatching, "UnlockRequest" must remain set to state TRUE.

TRUE

A request is made to unlatch the protective equipment's guard locking device.

Please note that the request is only made if input parameter "S_SafetyActive" indicates state TRUE in addition to a signal change taking place on "UnlockRequest" (FALSE → TRUE). A request is not made if a signal change takes place on "UnlockRequest" (FALSE → TRUE) while "S_SafetyActive" indicates state FALSE at the same time.

FALSE

A request is made to latch the protective equipment's guard locking device.

6.6.10.4.6 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

Automatic startup

Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.

The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.

6.6.10.4.7 S_AutoReset

General function

- Specification for the start interlock after latching the guard locking device on the closed protective equipment

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after the guard locking device on the closed protective equipment has been locked ("S_GuardLock" = TRUE).

TRUE

After signal TRUE returns on input parameter "S_GuardLock", the function block does not support a start interlock.

No action is required on "Reset" to set safe output parameter "S_GuardLocked" to TRUE if the input signal combination is valid for this.

> ## Danger!
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After signal TRUE returns on input parameter "S_GuardLock", the function block supports a start interlock.

"Reset" must be changed from FALSE to TRUE in order to set safe output parameter "S_GuardLocked" to TRUE if the input signal combination is valid for this.

> ## Danger!
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.10.4.8 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

In addition, this input parameter is used to set release signal "S_GuardLocked" to TRUE after a start interlock.

6.6.10.5 Output parameters

6.6.10.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.10.5.2 S_GuardLocked

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected protective equipment for the process being controlled.

The release signal is controlled based on the state of the protective equipment and start interlock. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_GuardLocked", this output is referred to as the "enable output".

Release signal "S_GuardLocked" can be used for subsequent process control.

> **Danger!**
>
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The protective equipment is closed and the guard locking device is latched. The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The protective equipment is closed ("S_GuardMonitoring" = TRUE).
- The protective equipment's guard locking device is latched ("S_GuardLock" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The guard locking device is unlatched and/or the protective equipment is open. The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected open protective equipment ("S_GuardMonitoring" = FALSE).
- The protective equipment's guard locking device is unlatched ("S_GuardLock" = FALSE).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_GuardLocked" is set to TRUE only if input "S_GuardLock" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	After the closed protective equipment's guard locking device is latched, the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_GuardLocked" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 629: "SF_GuardLocking": Input parameter "S_AutoReset" / "S_StartReset"

6.6.10.5.3 S_UnlockGuard

General function

- Unlatch signal for the guard locking device on the protective equipment

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Use a safe device to control the coil for the protective equipment's guard locking device.**

Function description

This output parameter requests the latching or unlatching of the protective equipment's guard locking device.

TRUE

A request is made to unlatch the protective equipment's guard locking device.

FALSE

A request is made to latch the protective equipment's guard locking device.

6.6.10.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.10.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.10.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. A safety function was not triggered. Normal operation is possible in the area safeguarded by the protective equipment. The unlatching of the protective equipment's guard locking device is requested using "UnlockRequest". Establish the safe state (e.g. safe standstill) for the area safeguarded by the protective equipment before you request the unlatching of the protective equipment's guard locking device. The function block evaluates the safe state on "S_SafetyActive".	Intended event: • No corrective measures are required. Unintended event: • Check the configuration of the function block. • Check the safety functions. • Check the safety program. • Check the protective equipment. Reset the protective equipment's guard locking device. • Request unlatching of the protective equipment's guard locking device on "UnlockRequest".
8001	The function block's start interlock is active. "S_StartReset" indicates state FALSE.	Execute a reset on the function block in order to end the active start interlock of the function block.
8003	The protective equipment is closed and the protective equipment's guard locking device is latched. The function block's start interlock is active. "S_AutoReset" indicates state FALSE.	Execute a reset on the function block in order to end the active start interlock of the function block and to set the release signal to TRUE.
8011	The protective equipment is closed and unlatching of the protective equipment's guard locking device has not been requested.	Request the unlatching of the protective equipment's guard locking device on "UnlockRequest".
8012	The protective equipment's guard locking device is unlatched and the protective equipment is open.	No corrective measures are required.
8013	The protective equipment's guard locking device is unlatched and the protective equipment is not open. You can open the protective equipment or revoke the request for unlatching in order to clear the area safeguarded by the protective equipment for normal operation.	No corrective measures are required.
8014	The function block detected the safe state again on "S_SafetyActive" after the area safeguarded by the protective equipment was previously no longer in a safe state.	Reset the function block.
C001	The function block detected a static reset signal on "Reset" ("S_StartReset" = FALSE).	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C003	The function block detected a static reset signal on "Reset" ("S_AutoReset" = FALSE).	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004	The safe state was not detected on "S_SafetyActive" while the protective equipment was open and the protective equipment's guard locking device was unlatched, or the protective equipment's guard locking device was unlatched in normal operation without a request and/or the protective equipment was opened.	Establish the safe state. "S_SafetyActive" must indicate state TRUE for this.

Table 630: "SF_GuardLocking": Diagnostic codes

6.6.10.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

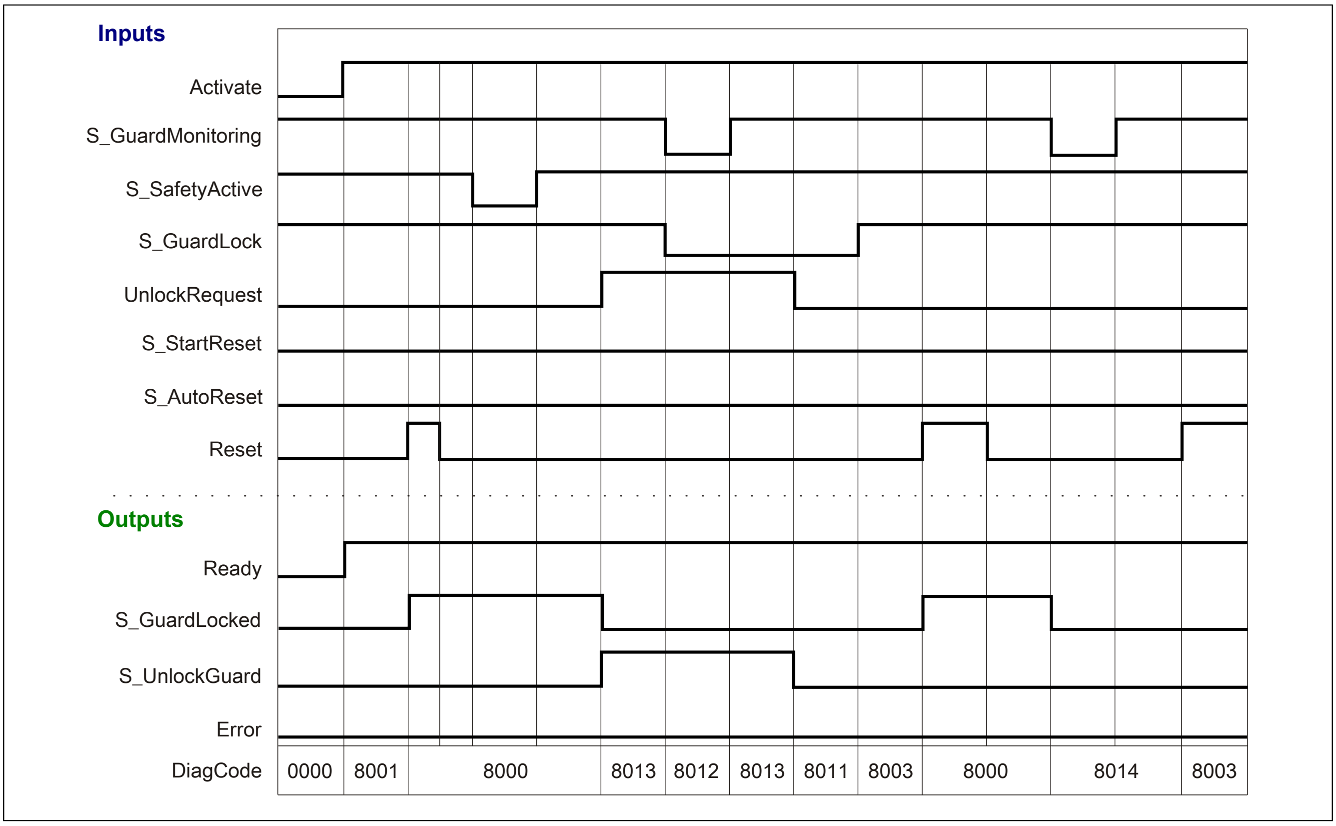


Figure 438: "SF_GuardLocking": Signal sequence diagram

6.6.10.7 Application examples

This chapter illustrates possible applications in which the function block can be used to implement 1-channel or 2-channel evaluation of guards.

Guard locking device / Interlocking device, 1-channel

Example "Safety door switch and interlock on guard locking device connected over 1 channel" illustrates the connection of the function block when controlling using the signal from a guard with protective equipment with guard locking device and interlock connected over 1 channel.

Guard locking device / Interlocking device, 2-channel

Example "Safety door switch and interlock on guard locking device connected over 2 channels" illustrates the connection of the function block when controlling using the signals from a guard with protective equipment with guard locking device and interlock connected over 2 channels.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.10.7.1 Protective equipment with guard locking device and interlock

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "GL_S8".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "S_GuardMonitoring", and "S_GuardLock", a rising edge on input parameter "Reset" is required in order for enable output "S_GuardLocked" to be enabled.

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the return of the safe input signals on "S_GuardMonitoring" and "S_GuardLock". In addition to the safe input signals on "S_GuardMonitoring", and "S_GuardLock", a rising edge on input parameter "Reset" is required in order for enable output "S_GuardLocked" to be enabled.

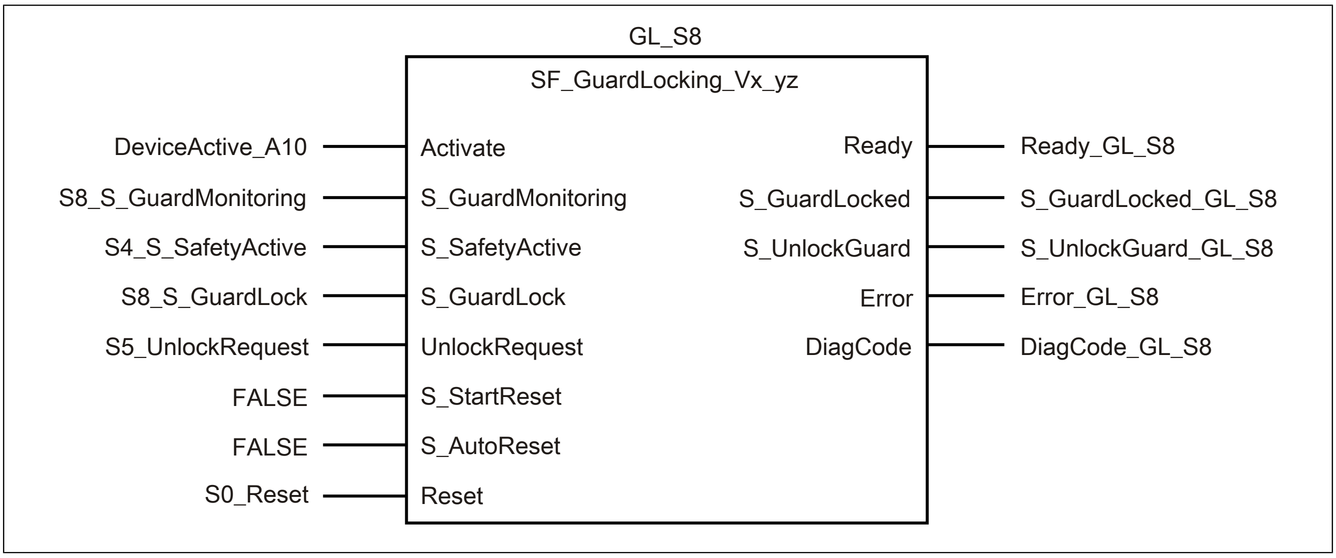


Figure 439: "SF_GuardLocking": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In the example, the signal is connected to a signal that indicates the state of the safe devices involved in the safety function. This prevents a safety function from being reported even if the safe devices are inactive.
S8_S_GuardMonitoring	SAFEBOOL	Signal of protective equipment from a safe input device. The signal indicates whether the protective equipment is open or closed. The signal comes from a 1-channel or 2-channel input of a safe input device (protective equipment). The evaluation of dual-channel redundancy does not take place in the function block. TRUE = The protective equipment is closed.
S4_S_SafetyActive	SAFEBOOL	This input is connected to a signal that specifies whether the hazard in a safeguarded area has been eliminated (the signal may come from a safe standstill monitor, for example). TRUE = The hazard has been eliminated.
S8_S_GuardLock	SAFEBOOL	Signal of the guard locking device from a safe input device. The signal indicates whether the protective equipment is open/closed and/or whether the protective equipment's guard locking device is latched/unlatched. The signal comes from a 1-channel or 2-channel input of a safe input device (protective equipment). The evaluation of dual-channel redundancy does not take place in the function block. TRUE = The protective equipment is closed and the protective equipment's guard locking device is latched.
S5_UnlockRequest	BOOL	Request signal for resetting the protective equipment's guard locking device
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after latching the guard locking device on the closed protective equipment
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause is no longer pending.

Table 631: "SF_GuardLocking": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_GL_S8	BOOL	Function block active / Function block inactive. Message for further external processing.
S_GuardLocked_GL_S8	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
S_UnlockGuard_GL_S8	SAFEBOOL	Control signal for unlatching the protective equipment's guard locking device This signal is used to control the coils for unlatching the guard locking device. TRUE = The guard locking device is unlatched and the protective equipment can be opened.
Error_GL_S8	BOOL	Error message from function block for further external processing
DiagCode_GL_S8	WORD	Diagnostic message from function block for further external processing

Table 632: "SF_GuardLocking": Outputs connected to output parameters

6.6.10.7.1.1 Safety door switch and interlock on guard locking device connected over 1 channel

This example illustrates the connection of the function block when controlling using the signal from a mechanically actuated position switch (safety door switch) connected over 1 channel. The latch signal for the protective equipment's guard locking device is connected to the function block over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GL_S8".

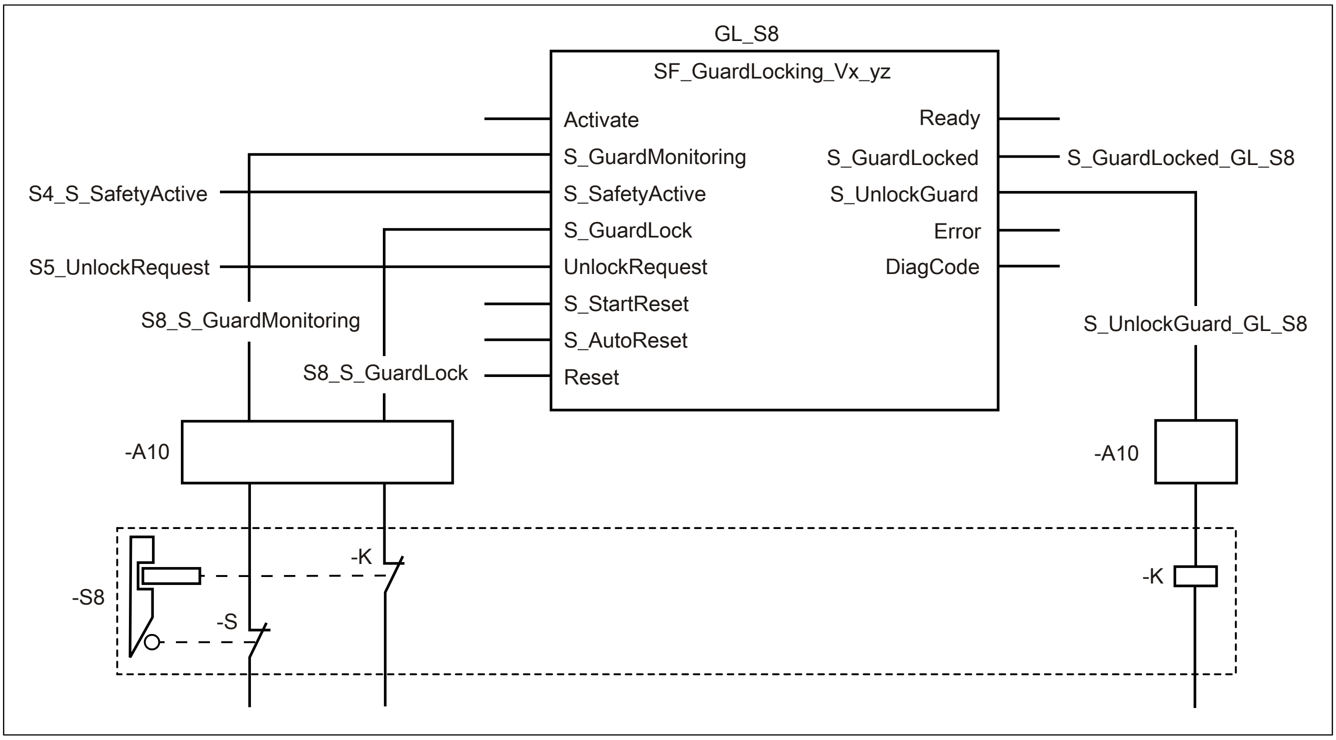


Figure 440: "SF_GuardLocking": Safety door switch and interlock on guard locking device connected over 1 channel

<u>List of equipment</u>

-S8	Safety switch with guard locking device. With integrated safety door switch "-S" (1-channel). Latch monitoring for the guard locking device (contact "-K": feedback regarding latching/unlatching and coil "-K" for opening the interlock)
-A10	1-channel safe inputs and 1-channel safe output of a safe device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

<u>Connected inputs and outputs</u>

S8_S_GuardMonitoring	Input on "S_GuardMonitoring"
S4_S_SafetyActive	Input on "S_SafetyActive"
S8_S_GuardLock	Input on "S_GuardLock"
S5_UnlockRequest	Input on "UnlockRequest"
S_GuardLocked_GL_S8	Output on "S_GuardLocked"
S_UnlockGuard_GL_S8	Output on "S_UnlockGuard"

Description

In this example:

- "-S" is the safety door switch that is integrated in the safety switch with guard locking device "-S8".
- The signal of safety door switch "-S" from an input of safe device "-A10" is connected to input "S8_S_GuardMonitoring". This input is connected to function block input parameter "S_GuardMonitoring" for further processing.
- "-K" is the coil for latching that is integrated in the safety switch with guard locking device "-S8".
- The signal of feedback contact for latching "-K" from another input of safe device "-A10" is connected to input "S8_S_GuardLock". This input is connected to function block input parameter "S_GuardLock" for further processing.
- The safe signal connected to input "S4_S_SafetyActive". This signal specifies whether the safe state exists in the safeguarded area. This input is connected to function block input parameter "S_SafetyActive" for further processing.
- The signal from the standard controller is connected to input "S5_UnlockRequest". This signal is used to request latching/unlatching of the protective equipment's guard locking device. This input is connected to function block input parameter "UnlockRequest" for further processing.
- Output parameter "S_UnlockGuard" is connected to output "S_UnlockGuard_GL_S8". This output is used to unlatch the guard locking device with coil "-K". The output is output via the 1-channel output of safe device "-A10".
- Output parameter "S_GuardLocked" is connected to output "S_GuardLocked_GL_S8".
- Output "S_GuardLocked_GL_S8" is used as a release signal to control the process in consideration of other safety functions.

6.6.10.7.1.2 Safety door switch and interlock on guard locking device connected over 2 channels

This example illustrates the connection of the function block when controlling using the signals from a mechanically actuated position switch (safety door switch, 2-channel antivalent) connected over 2 channels. The latch signals for the protective equipment's guard locking device (antivalent contacts of coil "-K") are connected to the function block over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GL_S8".

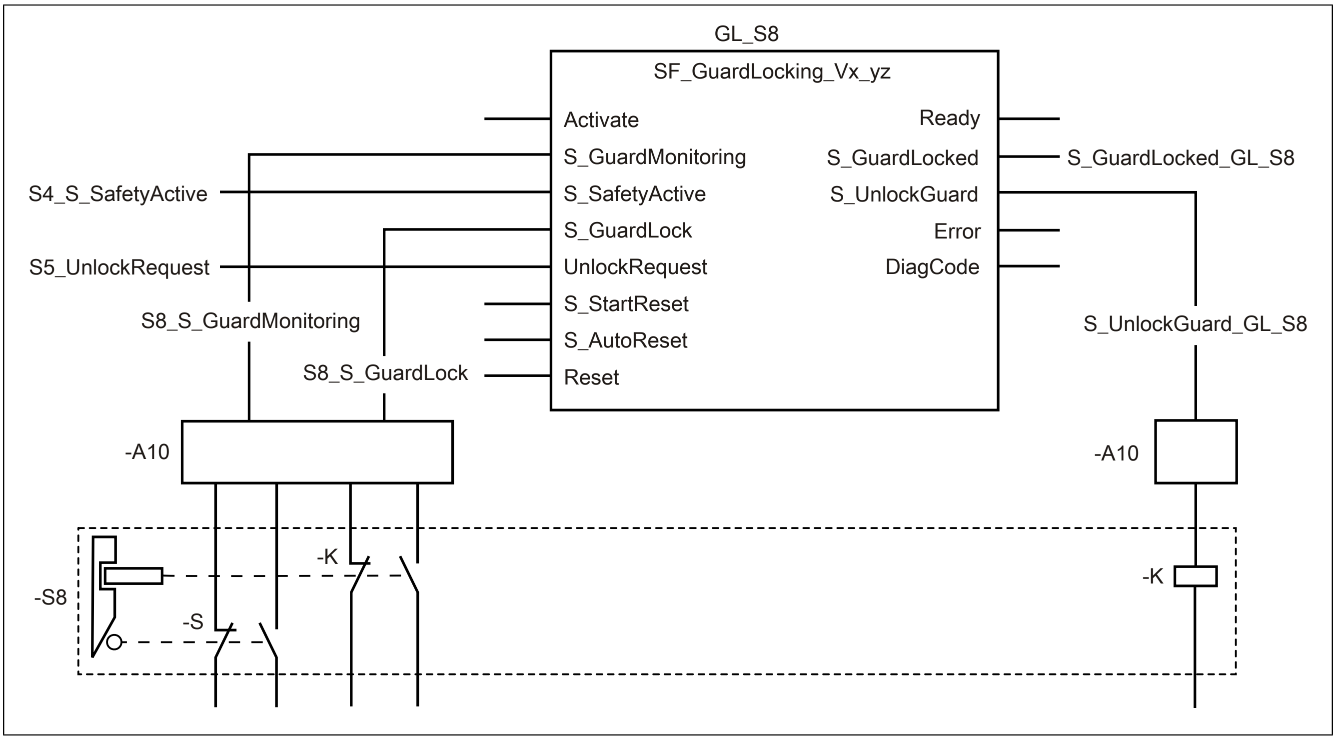


Figure 441: "SF_GuardLocking": Safety door switch and interlock on guard locking device connected over 2 channels

List of equipment

-S8	Safety switch with guard locking device. With integrated safety door switch "-S" (2-channel). Latch monitoring for the guard locking device ("-K" contacts: feedback regarding latching/unlatching (2-channel, antivalent) and coil "-K" for opening the interlock)
-A10	2-channel safe inputs and 1-channel safe output of a safe device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S8_S_GuardMonitoring	Input on "S_GuardMonitoring"
S4_S_SafetyActive	Input on "S_SafetyActive"
S8_S_GuardLock	Input on "S_GuardLock"
S5_UnlockRequest	Input on "UnlockRequest"
S_GuardLocked_GL_S8	Output on "S_GuardLocked"
S_UnlockGuard_GL_S8	Output on "S_UnlockGuard"

Description

In this example:

- "-S" is the safety door switch that is integrated in the safety switch with guard locking device "-S8".
- The resulting signal of the 2-channel safety door switch "-S" from an input of safe device "-A10" is connected to input "S8_S_GuardMonitoring". This input is connected to function block input parameter "S_GuardMonitoring" for further processing.
- "-K" is the coil for latching that is integrated in the safety switch with guard locking device "-S8".
- The resulting signal of 2-channel feedback contact for latching "-K" from another input of safe device "-A10" is connected to input "S8_S_GuardLock". This input is connected to function block input parameter "S_GuardLock" for further processing.
- The safe signal connected to input "S4_S_SafetyActive". This signal specifies whether the safe state exists in the safeguarded area. This input is connected to function block input parameter "S_SafetyActive" for further processing.
- The signal from the standard controller is connected to input "S5_UnlockRequest". This signal is used to request latching/unlatching of the protective equipment's guard locking device. This input is connected to function block input parameter "UnlockRequest" for further processing.
- Output parameter "S_UnlockGuard" is connected to output "S_UnlockGuard_GL_S8". This output is used to unlatch the guard locking device with coil "-K". The output is output via the 1-channel output of safe device "-A10".
- Output parameter "S_GuardLocked" is connected to output "S_GuardLocked_GL_S8".
- Output "S_GuardLocked_GL_S8" is used as a release signal to control the process in consideration of other safety functions.

6.6.10.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 1088	Interlocking guard with guard locking device	"S_GuardMonitoring" must be connected to the switch on the protective equipment in order for the function block to be able to evaluate the state of the connected protective equipment (open/closed). If "S_GuardMonitoring" = FALSE, the function block sets enable output "S_GuardLocked" to FALSE. "S_GuardLock" must be connected to a feedback signal from the protective equipment's guard locking device in order for the function block to be able to evaluate the state of the protective equipment's guard locking device (latched/unlatched). If "S_SafetyActive" = FALSE, the function block sets enable output "S_GuardLocked" to FALSE. "S_SafetyActive" must be connected to a signal generated outside of the function block that reports that a hazard occurring in an area safeguarded by the protective equipment has been eliminated. If "S_SafetyActive" = TRUE, the request can be made with "UnlockRequest" to reset the protective equipment's guard locking device. The signal you generated for reporting the elimination of a hazard is only permitted to indicate state TRUE if no risk of injury exists in the safeguarded area. It is your responsibility to implement control of "S_SafetyActive". After the closed protective equipment has been latched, the function block optionally supports a start interlock depending on the specification on "S_AutoReset". After the function block is enabled, the function block optionally supports a start interlock depending on the specification on "S_StartReset".
EN 1088	Interlocking device with guard locking device / Conditional unlatching (4-stage interlocking device)	The properly connected function block supports 4-stage interlocking devices. It is your responsibility to implement the connection.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	If state FALSE is indicated on "S_StartReset" and TRUE is indicated on both safe function blocks inputs ("S_GuardMonitoring" and "S_GuardLock"), then the enable output is set to TRUE when the function block is enabled after a signal change from FALSE to TRUE takes place on "Reset".
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.

Table 633: "SF_GuardLocking": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.11 SF_GuardMonitoring

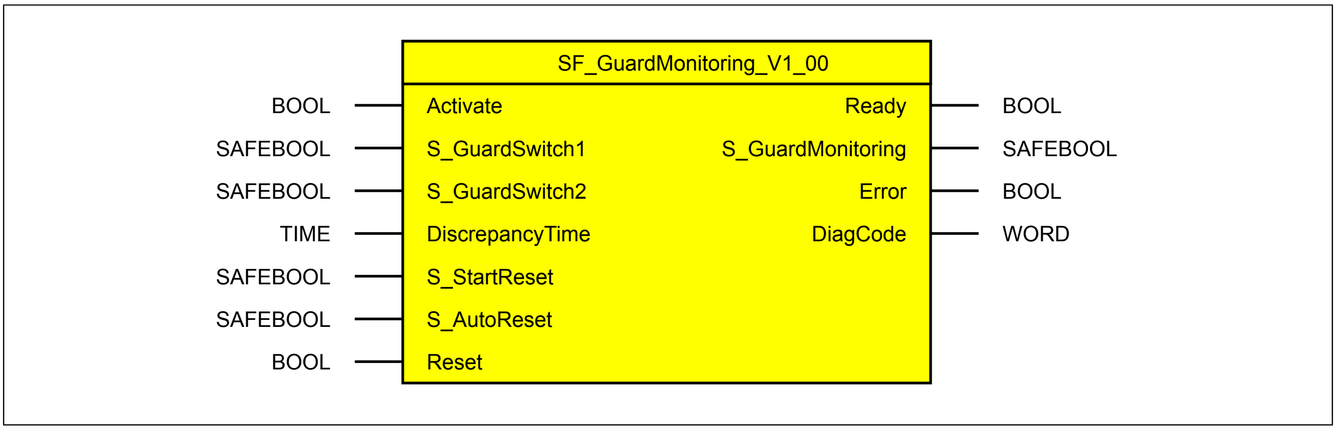


Figure 442: Function block "SF_GuardMonitoring"

6.6.11.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_GuardSwitch1	SAFEBOOL	Variable	State	FALSE	Signal input 1 / Position switch 1. State of the protective equipment
S_GuardSwitch2	SAFEBOOL	Variable	State	FALSE	Signal input 2 / Position switch 2. State of the protective equipment
DiscrepancyTime	TIME	Constant	State	#0ms	Specification for the monitoring time of the switching operation on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2"
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the return of TRUE signals on safe inputs "S_GuardSwitch1" and "S_GuardSwitch2"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 634: "SF_GuardMonitoring": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_GuardMonitoring	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 635: "SF_GuardMonitoring": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 636: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.11.2 Function

Function block "SF_GuardMonitoring" is used to support a protective equipment without guard locking device function (safety door monitoring) in an application.

This function block is a safety-relevant function block for monitoring protective equipment with a 2-stage interlocking device.

6.6.11.2.1 Opening the protective equipment

If the protective equipment properly connected to the function block is opened, the function block sets enable output "S_GuardMonitoring" to FALSE (safe state).

6.6.11.2.2 Closing the protective equipment

If opened protective equipment in the application is closed, the function block can optionally (see start interlock) ensure within the safety control system that the release signal is not set to TRUE solely by this closing. Additional manual intervention on input parameter "Reset" is required for this (see start interlock).

6.6.11.2.3 Interlocking device

The function block fully supports a 2-stage interlocking device.

On a 3-stage interlocking device, the function block supports the transition from the second to the third stage and back again. On a 4-stage interlocking device, the function block supports the transition from the third to the fourth stage and back again. All other stages must be implemented in the safety application.

6.6.11.2.4 Guard locking device

Program the protective equipment's guard locking device in a separate safety function block or in the safety application since this function block simply grants or revokes its approval. Maintaining the approval in conjunction with the guard locking device and standstill monitoring must be viewed in close connection with the safety application.

6.6.11.2.5 Application with 1 position switch

1 position switch (1-channel)

If you are using 1 position switch (1-channel) in the protective equipment, you must connect the signal to function block input parameter "S_GuardSwitch1" via a safe input device. In addition, you have to establish a graphic connection between both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") in the safety application. This allows both signal inputs to be controlled by the same signal.

The function block enable output is only set to TRUE if both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") indicate state TRUE and the remaining input signal combination is valid for this as well.

1 position switch (2-channel)

If you are using 1 position switch (2-channel) in the protective equipment, you must individually connect the signals using a safe input device. In the safety application, you must check the signals of the position switch for dual-channel redundancy (e.g. using safe devices or other function blocks such as "SF_Antivalent" or "SF_Equivalent"). You must connect the signal resulting from this check to function block input parameter "S_GuardSwitch1". In addition, you have to establish a graphic connection between both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") in the safety application. This allows both signal inputs to be controlled by the same signal.

The function block enable output is only set to TRUE if both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") indicate state TRUE and the remaining input signal combination is valid for this as well.

6.6.11.2.6 Application with 2 position switches

2 position switches (1-channel)

If you are using 2 position switches (1-channel) in the protective equipment, you must individually connect the signals using one or more safe input devices. You must connect one of the signals to function block input parameter "S_GuardSwitch1". The other signal must be connected to the second function block input parameter "S_GuardSwitch2".

The function block enable output is only set to TRUE if both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") indicate state TRUE and the remaining input signal combination is valid for this as well.

2 position switches (2-channel)

If you are using 2 position switches (2-channel) in the protective equipment, you must individually connect the 4 signals using one or more safe input devices. In the safety application, you must check the respective signals of the position switches for dual-channel redundancy (e.g. using safe devices or other function blocks such as "SF_Antivalent" or "SF_Equivalent"). A signal must result from this check for each of the 2 position switches. You must connect one of these signals to function block input parameter "S_GuardSwitch1". The other signal must be connected to the second function block input parameter "S_GuardSwitch2".

The function block enable output is only set to TRUE if both input parameters ("S_GuardSwitch1" and "S_GuardSwitch2") indicate state TRUE and the remaining input signal combination is valid for this as well.

6.6.11.2.7 Quantity and design of supported position switches

The function block supports protective equipment with 1 or 2 mechanical or non-mechanical position switches. Depending on the CAT or SIL requirements, 1-channel or 2-channel position switches with equivalent or antivalent arrangement of switching contacts are required.

The function block processes one signal per position switch, which results from checking the dual-channel redundancy on 2-channel position switches.

The function block does not monitor the dual-channel redundancy (line control) or equivalence/antivalence of the signals. You must implement this monitoring in the safety application using safe devices or other function blocks ("SF_Antivalent" or "SF_Equivalent"), for example.

6.6.11.2.8 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.11.2.9 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.11.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.11.3.1 Asymmetrical switching signals on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2"

If the signals on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2" are switched asymmetrically (due to faulty protective equipment, faulty switch, faulty cables, etc.), then the safe state (enable output "S_GuardMonitoring" = FALSE) will be maintained once the protective equipment has been closed.

6.6.11.3.2 Non-simultaneous signals on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2"

The function block reports non-simultaneous signals present on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2" (e.g. due to faulty protective equipment, faulty switch, faulty lines) as a fault after the specified discrepancy time (input parameter "DiscrepancyTime") has elapsed.

6.6.11.3.3 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.11.3.4 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.11.3.5 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the state of the release signal depends on the state of the protective equipment connected to the function block as well as on how the optional start interlock is specified for when the TRUE signal returns to the protective equipment.

6.6.11.3.6 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.11.3.7 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.11.4 Input parameters

6.6.11.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.11.4.2 S_GuardSwitch1

General function

- Signal input 1 / Position switch 1. State of the protective equipment.

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe input device that is connected to a switch on the protective equipment (e.g. safety door or standalone, possibly positively driven, 1-channel or 2-channel position switch). Input parameter "S_GuardSwitch1" is then controlled using this signal.

Implement monitoring of dual-channel redundancy for the position switches in the safety application, e.g. in the safe device or using other function blocks such as "SF_Antivalent" or "SF_Equivalent". Input parameter "S_GuardSwitch1" is set using the resulting monitoring signal.

Information:

For detailed information about applications with 1 position switch, see section 6.6.11.2.5 "Application with 1 position switch".
For detailed information about applications with 2 position switches, see section 6.6.11.2.6 "Application with 2 position switches".

Function description

The signal connected to input parameter "S_GuardSwitch1" is processed by the function block.

The signal input processes the state of a position switch on protective equipment.

This signal must indicate state TRUE in order to set enable output "S_GuardMonitoring" to TRUE under consideration of the other input signal combination.

Regardless whether the protective equipment's position switch is connected to the safe input device over 1 or 2 channels, "S_GuardSwitch1" is only connected to one signal of the protective equipment.

If a position switch on protective equipment is wired to the safe input device over 2 channels, then monitoring the dual-channel redundancy must be handled by the safe input device or implemented using other means (according to the performed risk analysis). This device passes on a signal to "S_GuardSwitch1". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The protective equipment is closed.
Function block enable output "S_GuardMonitoring" is set to TRUE under consideration of the other input signal combination.

FALSE

The protective equipment is open, the wiring for the protective equipment is interrupted or the safe input device connected to this protective equipment is shut down or defective.
Function block enable output "S_GuardMonitoring" is set to FALSE.

6.6.11.4.3 S_GuardSwitch2

General function

- Signal input 2 / Position switch 2. State of the protective equipment.

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe input device that is connected to a switch on the protective equipment (e.g. safety door or standalone, possibly positively driven, 1-channel or 2-channel position switch). Input parameter "S_GuardSwitch2" is then controlled using this signal.

Implement monitoring of dual-channel redundancy for the position switches in the safety application, e.g. in the safe device or using other function blocks such as "SF_Antivalent" or "SF_Equivalent". Input parameter "S_GuardSwitch2" is set using the resulting monitoring signal.

Information:

For detailed information about applications with 1 position switch, see section 6.6.11.2.5 "Application with 1 position switch".
For detailed information about applications with 2 position switches, see section 6.6.11.2.6 "Application with 2 position switches".

Function description

The signal connected to input parameter "S_GuardSwitch2" is processed by the function block.

The signal input processes the state of a position switch on protective equipment.

This signal must indicate state TRUE in order to set enable output "S_GuardMonitoring" to TRUE under consideration of the other input signal combination.

Regardless whether the protective equipment's position switch is connected to the safe input device over 1 or 2 channels, "S_GuardSwitch2" is only connected to one signal of the protective equipment.

If a position switch on protective equipment is wired to the safe input device over 2 channels, then monitoring the dual-channel redundancy must be handled by the safe input device or implemented using other means (according to the performed risk analysis). This device passes on a signal to "S_GuardSwitch2". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The protective equipment is closed.
Function block enable output "S_GuardMonitoring" is set to TRUE under consideration of the other input signal combination.

FALSE

The protective equipment is open, the wiring for the protective equipment is interrupted or the safe input device connected to this protective equipment is shut down or defective.
Function block enable output "S_GuardMonitoring" is set to FALSE.

6.6.11.4.4 DiscrepancyTime

General function

- Specification for the monitoring time of the switching operation on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2"

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the time within which the switching operations on signal inputs "S_GuardSwitch1" and "S_GuardSwitch2" must take place when closing the protective equipment in order to be detected as simultaneous.

A signal tolerance during the closing of the protective equipment is possible due to mechanical influences. The specified "DiscrepancyTime" value limits the duration within which the two signals can be switched successively without this being detected as not simultaneous.

This input parameter must be connected to constant TIME#0ms if you connect both signal inputs ("S_GuardSwitch1" and "S_GuardSwitch2") to the same signal in your application. If time value TIME#0ms was specified in an application with only 1 signal, the function block will detect an illegal time offset of the signals on inputs "S_GuardSwitch1" and "S_GuardSwitch2" and report it as an error. If you connect the signal inputs to different signals, you must connect "DiscrepancyTime" to a constant not equal to 0.

You must define and validate the time value for input parameter "DiscrepancyTime" based on your application and risk analysis.

6.6.11.4.5 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.11.4.6 S_AutoReset

General function

- Specification of the start interlock after the return of TRUE signals on safe inputs "S_GuardSwitch1" and "S_GuardSwitch2"

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after TRUE signals have returned to the safe input parameters "S_GuardSwitch1" and "S_GuardSwitch2" (closing the guard).

TRUE

After TRUE signals return to the safe input parameters "S_GuardSwitch1" and "S_GuardSwitch2", the function block does not support a start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After TRUE signals return to the safe input parameters "S_GuardSwitch1" and "S_GuardSwitch2", the function block supports a start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.11.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

> **Danger!**
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.11.5 Output parameters

6.6.11.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.11.5.2 S_GuardMonitoring

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected guard for the process being controlled. The release signal is controlled based on the state of the guard and start interlock. In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_GuardMonitoring", this output is referred to as the "enable output".

Release signal "S_GuardMonitoring" can be used for subsequent process control.

> ## Danger!
>
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The guard is not open ("S_GuardSwitch1" = TRUE and "S_GuardSwitch2" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected an open guard ("S_GuardSwitch1" = FALSE and "S_GuardSwitch2" = FALSE).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_GuardMonitoring" is set to TRUE only if inputs "S_GuardSwitch1" and "S_GuardSwitch2" indicate state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	After signal TRUE returns on safe inputs "S_GuardSwitch1" and "S_GuardSwitch2", the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_GuardMonitoring" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 637: "SF_GuardMonitoring": Input parameter "S_AutoReset" / "S_StartReset"

6.6.11.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.11.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.11.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. A triggered safety function is not present.	Intended event: • No corrective measures are required. Unintended event: • Check the configuration of the function block. • Check the safety functions. • Check the safety program. • Check the position switches connected to the function block. • Check the configuration of the safe devices connected to the position switches.
8001	Initialization of the function block after it has been enabled. The initialization of the function block is completed after one safety controller cycle.	No corrective measures are required.
8002	Applications with 2 position switches on the protective equipment:	
	The function block detected the opening of the connected protective equipment. One of the two switching signals indicates state FALSE. If the protective equipment is completely open, the wiring is faulty or the safe device connected to the protective equipment is shut down or defective.	• Open the protective equipment completely. • If the protective equipment is completely open, check the wiring, the protective equipment's position switches or the safe device connected to the protective equipment and its configuration.
	Applications with 1 position switch on the protective equipment:	
	In applications with only 1 position switch, the state described above is completed after one cycle of the safety controller. Otherwise, there is an error.	• Check the connection of function block input parameters. • For applications with only one position switch, a graphic connection must be added between "S_GuardSwitch1" and "S_GuardSwitch2". This allows both input parameters to be set by one switching signal. • Please note that "S_GuardSwitch2" is not permitted to be connected to any other input!
8003	The function block's start interlock is active. ("S_AutoReset" = FALSE)	Execute a reset on the function block in order to end the active start interlock of the function block and to set the release signal to TRUE.
8004	Applications with 2 position switches on the protective equipment:	
	After being completely open, the protective equipment is not yet completely closed. The signal on "S_GuardSwitch1" is set back to TRUE while the signal on "S_GuardSwitch2" still indicates state FALSE. The specified monitoring time on "DiscrepancyTime" is started in this state.	• Close the protective equipment completely. • Check the specified time value on "DiscrepancyTime". • If the protective equipment is completely closed, check the position switches, wiring, safe device (including configuration) and the safety program.
	Applications with 1 position switch on the protective equipment:	
	In applications with only 1 position switch, the state described above is not permitted to occur.	• Check the connection of function block input parameters. • For applications with only one position switch, a graphic connection must be added between "S_GuardSwitch1" and "S_GuardSwitch2". This allows both input parameters to be set by one switching signal. • Please note that "S_GuardSwitch2" is not permitted to be connected to any other input!
8005	The opened protective equipment is completely closed again. This state is completed after one cycle of the safety controller.	No corrective measures are required.
8012	The protective equipment is completely open.	Intended event: • Close the protective equipment in order to revoke the triggered safety function. Unintended event: • Check the safety functions. • Check the safety program. • Check the position switches connected to the function block. • Check the configuration of the safe devices connected to the position switches.
8014	Applications with 2 position switches on the protective equipment:	
	After being completely open, the protective equipment is not yet completely closed. The signal on "S_GuardSwitch2" is set back to TRUE while the signal on "S_GuardSwitch1" still indicates state FALSE. The specified monitoring time on "DiscrepancyTime" is started in this state.	• Close the protective equipment completely. • Check the specified time value on "DiscrepancyTime". • If the protective equipment is completely closed, check the position switches, wiring, safe device (including configuration) and the safety program.
	Applications with 1 position switch on the protective equipment:	

Table 638: "SF_GuardMonitoring": Diagnostic codes

Code (hex)	Description	Corrective measures
	In applications with only 1 position switch, the state described above is not permitted to occur.	• Check the connection of function block input parameters. • For applications with only one position switch, a graphic connection must be added between "S_GuardSwitch1" and "S_GuardSwitch2". This allows both input parameters to be set by one switching signal. • Please note that "S_GuardSwitch2" is not permitted to be connected to any other input!
C001	The function block detected a static reset signal on "Reset". ("S_AutoReset" = FALSE)	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C011	Applications with 2 position switches on the protective equipment:	
	The specified time on "DiscrepancyTime" has expired. The cause is a TRUE signal on "S_GuardSwitch1" and FALSE signal on "S_GuardSwitch2".	• Check the position switches on the protective equipment and correct the error if necessary. • While the protective equipment is closed and the position switches are in an error-free state, check the wiring and/or the safe device connected to the protective equipment, including its configuration. • Check the specified value on "DiscrepancyTime".
	Applications with 1 position switch on the protective equipment:	
	In applications with only 1 position switch, the state described above is not permitted to occur.	• Check the connection of function block input parameters. • For applications with only one position switch, a graphic connection must be added between "S_GuardSwitch1" and "S_GuardSwitch2". This allows both input parameters to be set by one switching signal. • Please note that "S_GuardSwitch2" is not permitted to be connected to any other input!
C012	Applications with 2 position switches on the protective equipment:	
	The specified time on "DiscrepancyTime" has expired. The cause is a TRUE signal on "S_GuardSwitch2" and FALSE signal on "S_GuardSwitch1".	• Check the position switches on the protective equipment and correct the error if necessary. • While the protective equipment is closed and the position switches are in an error-free state, check the wiring and/or the safe device connected to the protective equipment, including its configuration. • Check the specified value on "DiscrepancyTime".
	Applications with 1 position switch on the protective equipment:	
	In applications with only 1 position switch, the state described above is not permitted to occur.	• Check the connection of function block input parameters. • For applications with only one position switch, a graphic connection must be added between "S_GuardSwitch1" and "S_GuardSwitch2". This allows both input parameters to be set by one switching signal. • Please note that "S_GuardSwitch2" is not permitted to be connected to any other input!

Table 638: "SF_GuardMonitoring": Diagnostic codes

6.6.11.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

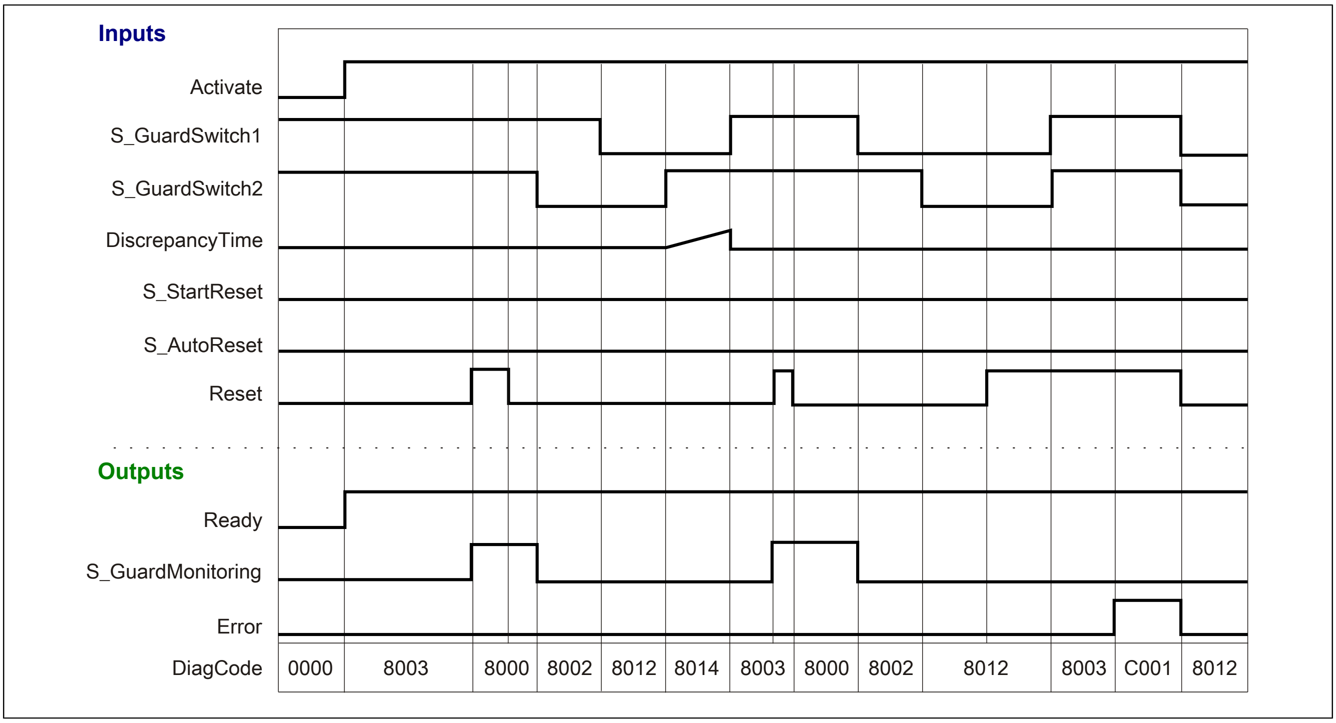


Figure 443: "SF_GuardMonitoring": Signal sequence diagram 1

Signal sequence diagram 2

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

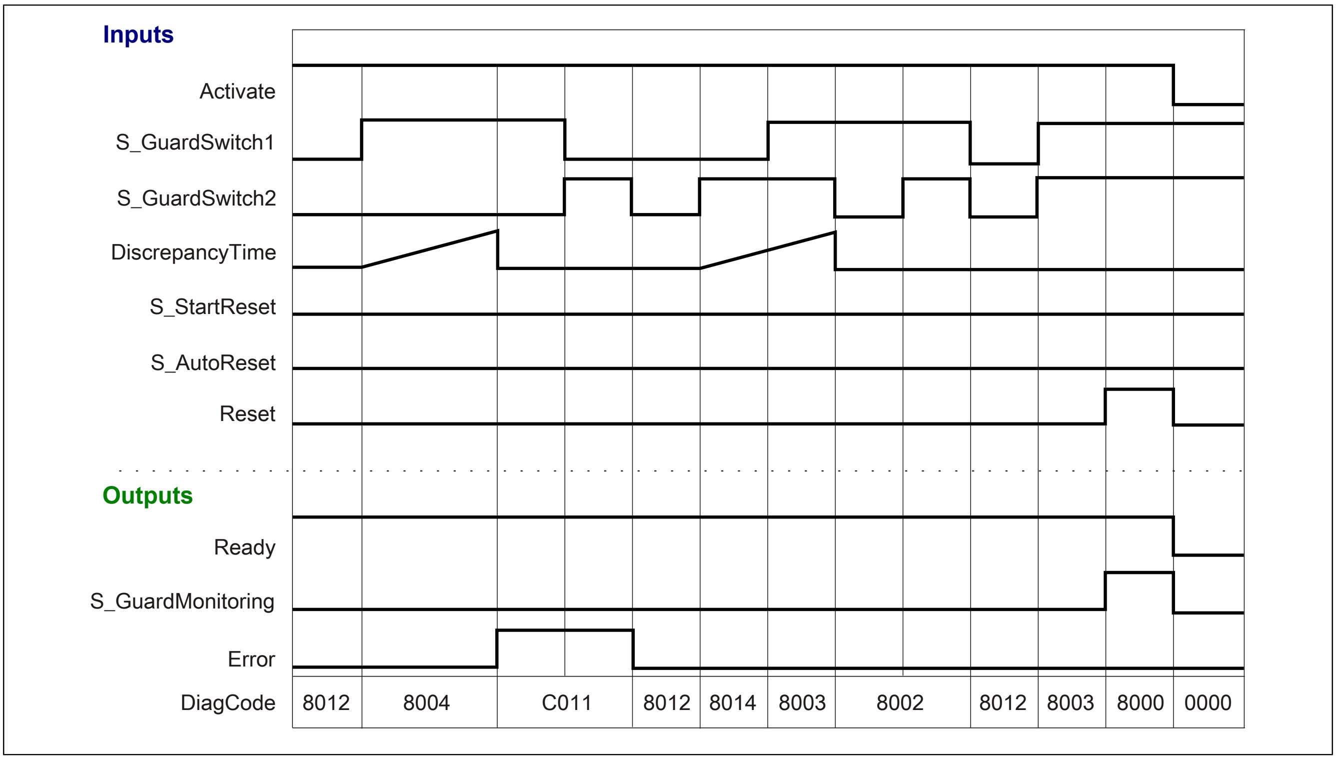


Figure 444: "SF_GuardMonitoring": Signal sequence diagram 2

6.6.11.7 Application examples

This chapter illustrates possible applications in which the function block can be used to implement 1-channel or 2-channel evaluation of guards.

Protective equipment with 1 signal

The following examples describe function block connections when controlling with the signal from the following:

- 1 mechanically actuated, 1-channel position switch (see section 6.6.11.7.1.1 "Interlocking device with 1 mechanically actuated position switch (1-channel)")
- 1 mechanically actuated, 2-channel equivalent position switch (see section 6.6.11.7.1.2 "Interlocking device with 1 mechanically actuated position switch (2-channel, equivalent)")
- 1 mechanically actuated, 2-channel antivalent position switch (see section 6.6.11.7.1.3 "Interlocking device with 1 mechanically actuated position switch (2-channel, antivalent)")

Protective equipment with 2 signals

The following examples describe function block connections when controlling with the signals from the following:

- 2 mechanically actuated position switches connected over 1 channel (see section 6.6.11.7.2.1 "Interlocking device with 2 mechanically actuated position switches (1-channel)")
- 2 non-mechanically actuated position switches connected over 1 channel (see section 6.6.11.7.2.2 "Interlocking device with 2 non-mechanically actuated position switches (1-channel)")
- 2 mechanically actuated position switches connected over 2 channels (see section 6.6.11.7.2.3 "Interlocking device with 2 mechanically actuated position switches (2-channel, antivalent)")

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.11.7.1 Protective equipment with 1 signal

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "GM_S5".

In this example, input parameters "S_GuardSwitch1" and "S_GuardSwitch2" are graphically connected to each other. Input parameter "DiscrepancyTime" must have value TIME#0ms.

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2", a rising edge on input parameter "Reset" is required to enable enable output "S_GuardMonitoring".

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the return of the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2". In addition to the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2", a rising edge on input parameter "Reset" is required to enable enable output "S_GuardMonitoring".

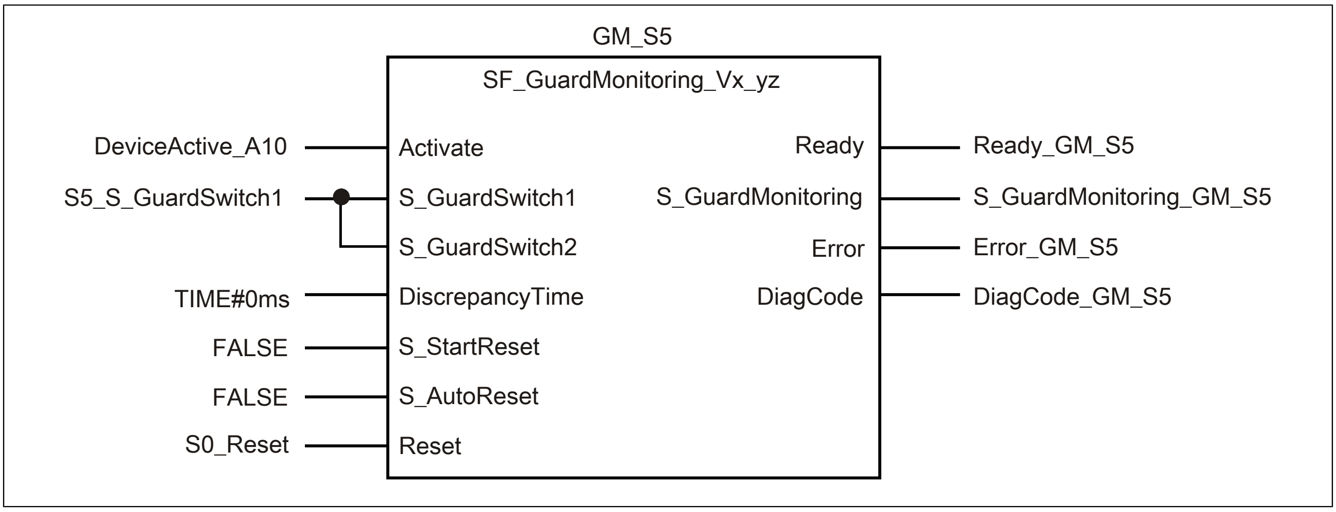


Figure 445: "SF_GuardMonitoring": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the safety switch(es) of the guard is/are connected.
S5_S_GuardSwitch1	SAFEBOOL	Input signal 1 from the safe input device / guard. Depending on the safety application, the signal comes from one or both contacts on the guard. The signal is read in over 1 channel in this example.
TIME#0ms on "DiscrepancyTime"	TIME	Because inputs "S_GuardSwitch1" and "S_GuardSwitch2" are graphically connected, this input parameter must have the value 0 ms.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after closing the guard
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 639: "SF_GuardMonitoring": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_GM_S5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_GuardMonitoring_GM_S5	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
Error_GM_S5	BOOL	Error message from function block for further external processing
DiagCode_GM_S5	WORD	Diagnostic message from function block for further external processing

Table 640: "SF_GuardMonitoring": Outputs connected to output parameters

6.6.11.7.1.1 Interlocking device with 1 mechanically actuated position switch (1-channel)

This example illustrates the connection of the function block when controlling using the signal from a mechanically actuated position switch (safety door switch) connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5".

Input parameters "S_GuardSwitch1" and "S_GuardSwitch2" are graphically connected to each other.

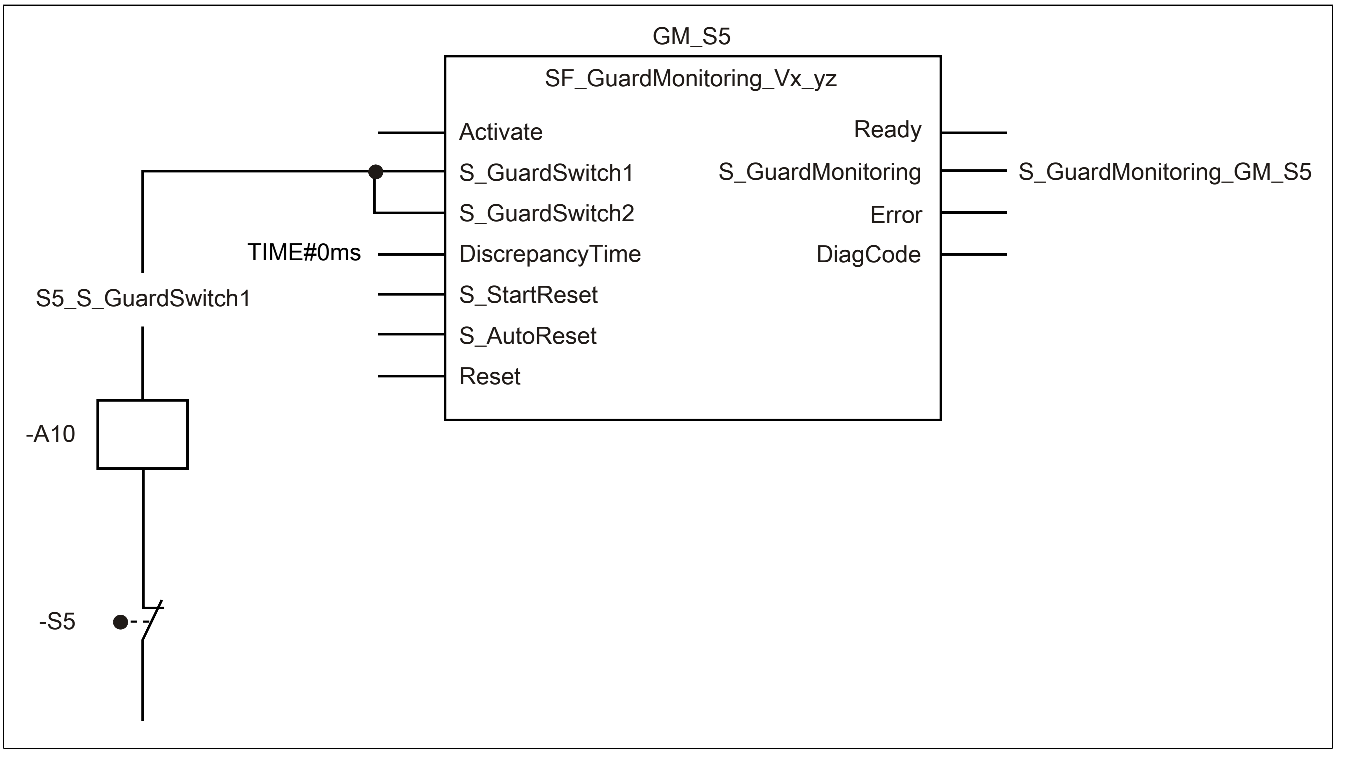


Figure 446: "SF_GuardMonitoring": Interlocking device with 1 mechanically actuated position switch (1-channel)

<u>List of equipment</u>

-S5	Safety door switch, 1-channel, positively driven with direct opening action (TRUE when protective equipment closed)
-A10	1-channel input of a safe input device

<u>Connected inputs and outputs</u>

S5_S_GuardSwitch1 (graphically connected)	Input on "S_GuardSwitch1" and "S_GuardSwitch2"
S_GuardMonitoring_GM_S5	Output on "S_GuardMonitoring"

<u>Description</u>

In this example:

- The signal from the 1-channel safe input of safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameters "S_GuardSwitch1" and "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5".
- Output "S_GuardMonitoring_GM_S5" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.7.1.2 Interlocking device with 1 mechanically actuated position switch (2-channel, equivalent)

This example illustrates the connection of the function block when controlling using the signal from a mechanically actuated position switch (safety door switch) connected equivalently over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5".

Input parameters "S_GuardSwitch1" and "S_GuardSwitch2" are graphically connected to each other.

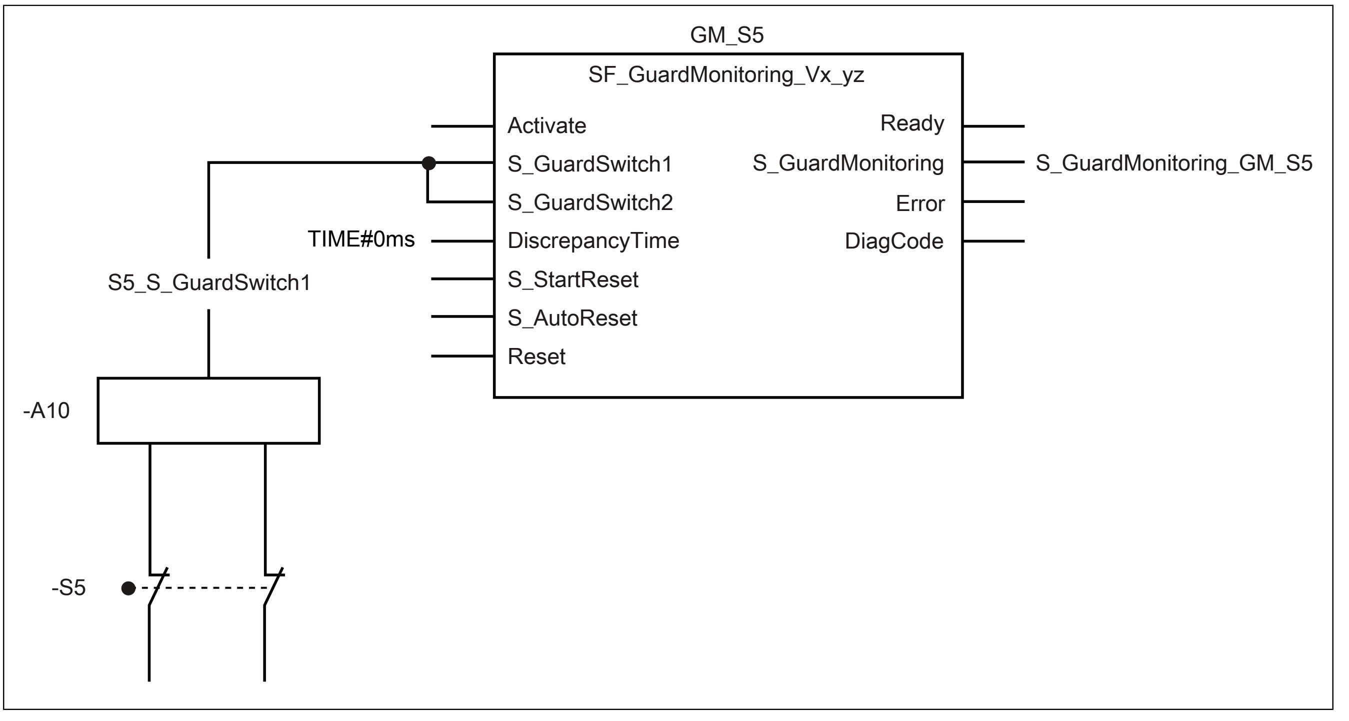


Figure 447: "SF_GuardMonitoring": Interlocking device with 1 mechanically actuated position switch (2-channel, equivalent)

List of equipment

-S5	Safety door switch, 2-channel (equivalent), positively driven with direct opening action (TRUE when protective equipment closed)
-A10	2-channel input of a safe input device with line control and equivalent evaluation

Connected inputs and outputs

S5_S_GuardSwitch1 (graphically connected)	Input on "S_GuardSwitch1" and "S_GuardSwitch2"
S_GuardMonitoring_GM_S5	Output on "S_GuardMonitoring"

Description

In this example:

- The resulting signal of the inputs from safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameters "S_GuardSwitch1" and "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5".
- Output "S_GuardMonitoring_GM_S5" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.7.1.3 Interlocking device with 1 mechanically actuated position switch (2-channel, antivalent)

This example illustrates the connection of the function block when controlling using the signal from a mechanically actuated position switch (safety door switch) connected antivalently over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5".

Input parameters "S_GuardSwitch1" and "S_GuardSwitch2" are graphically connected to each other.

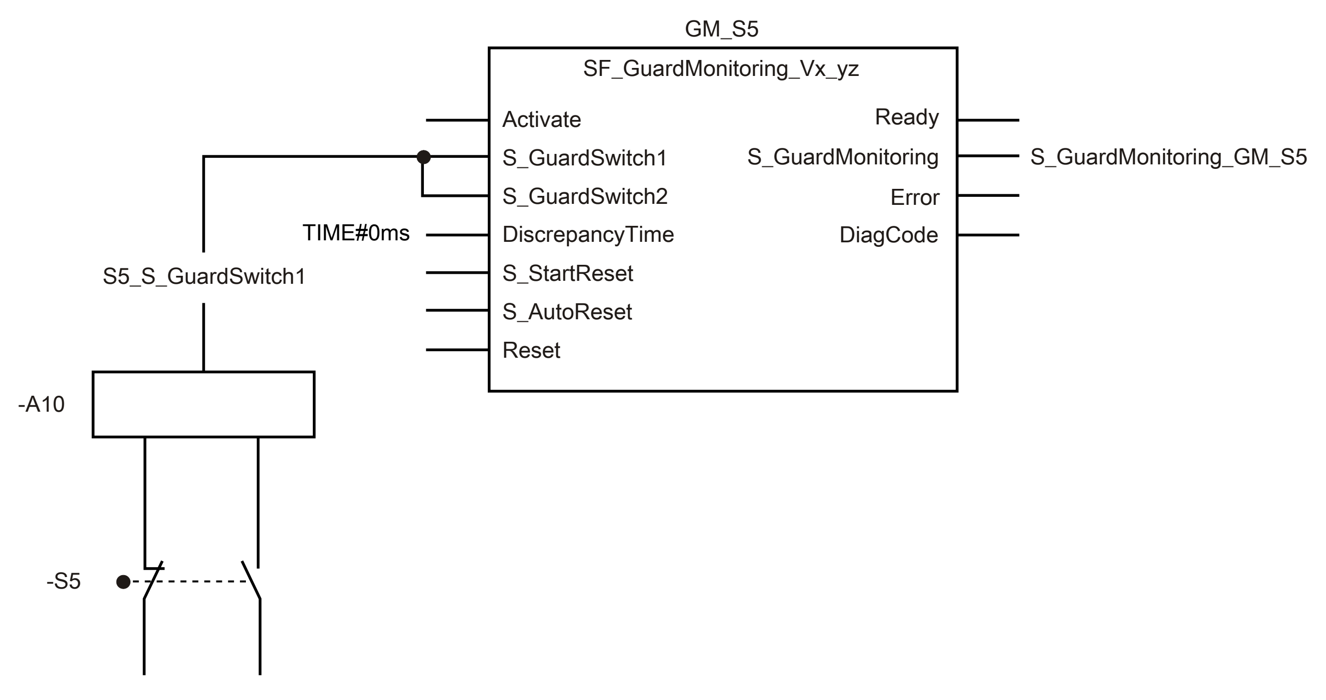


Figure 448: "SF_GuardMonitoring": Interlocking device with 1 mechanically actuated position switch (2-channel, antivalent)

List of equipment

-S5	Safety door switch, 2-channel (antivalent), positively driven with direct opening action (TRUE when protective equipment closed)
-A10	2-channel input of a safe input device with line control and antivalent evaluation

Connected inputs and outputs

S5_S_GuardSwitch1 (graphically connected)	Input on "S_GuardSwitch1" and "S_GuardSwitch2"
S_GuardMonitoring_GM_S5	Output on "S_GuardMonitoring"

Description

In this example:

- The resulting signal of the inputs from safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameters "S_GuardSwitch1" and "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5".
- Output "S_GuardMonitoring_GM_S5" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.7.2 Protective equipment with 2 signals

Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "GM_S5_S6".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2", a rising edge on input parameter "Reset" is required to enable enable output "S_GuardMonitoring".

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the return of the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2". In addition to the safe input signals on "S_GuardSwitch1" and "S_GuardSwitch2", a rising edge on input parameter "Reset" is required to enable enable output "S_GuardMonitoring".

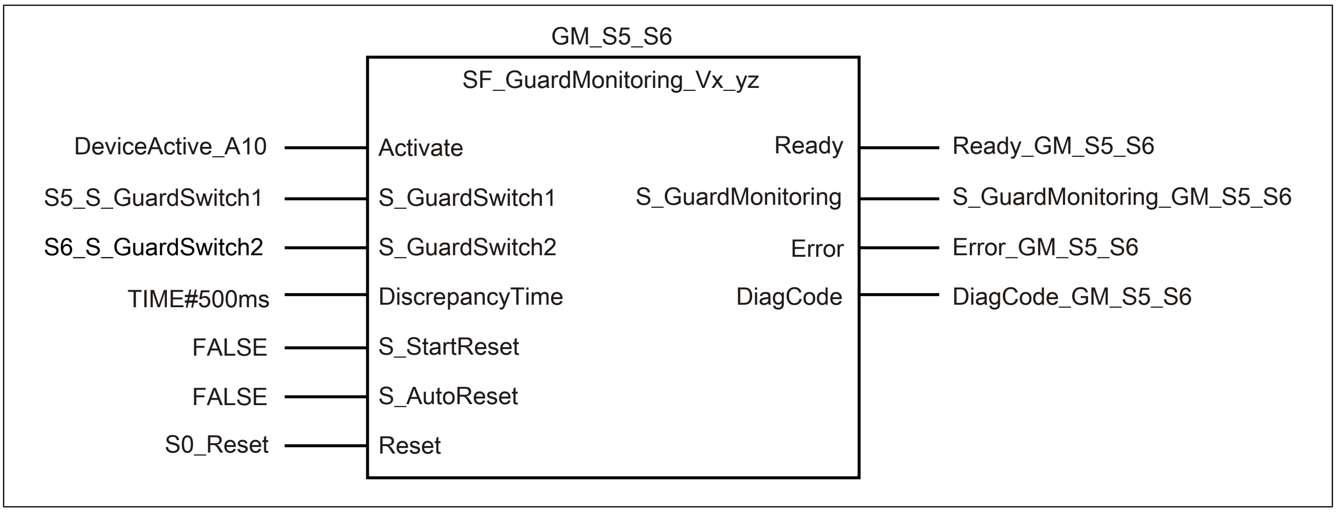


Figure 449: "SF_GuardMonitoring": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the safety switches of the guard are connected.
S5_S_GuardSwitch1	SAFEBOOL	Input signal 1 from the safe input device / guard. Depending on the safety application, the signal comes from a 1-channel or 2-channel switch on the guard. The signal is read in over 1 channel in this example.
S6_S_GuardSwitch2	SAFEBOOL	Input signal 2 from the safe input device / guard. Depending on the safety application, the signal comes from a 1-channel or 2-channel switch on the guard. The signal is read in over 1 channel in this example.
TIME#500ms on "DiscrepancyTime"	TIME	Maximum permitted discrepancy time
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after closing the guard
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 641: "SF_GuardMonitoring": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_GM_S5_S6	BOOL	Function block active / Function block inactive. Message for further external processing.
S_GuardMonitoring_GM_S5_S6	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
Error_GM_S5_S6	BOOL	Error message from function block for further external processing
DiagCode_GM_S5_S6	WORD	Diagnostic message from function block for further external processing

Table 642: "SF_GuardMonitoring": Outputs connected to output parameters

6.6.11.7.2.1 Interlocking device with 2 mechanically actuated position switches (1-channel)

This example illustrates the connection of the function block when controlling using the signals from 2 mechanically actuated position switches (safety door switches), each connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5_S6".

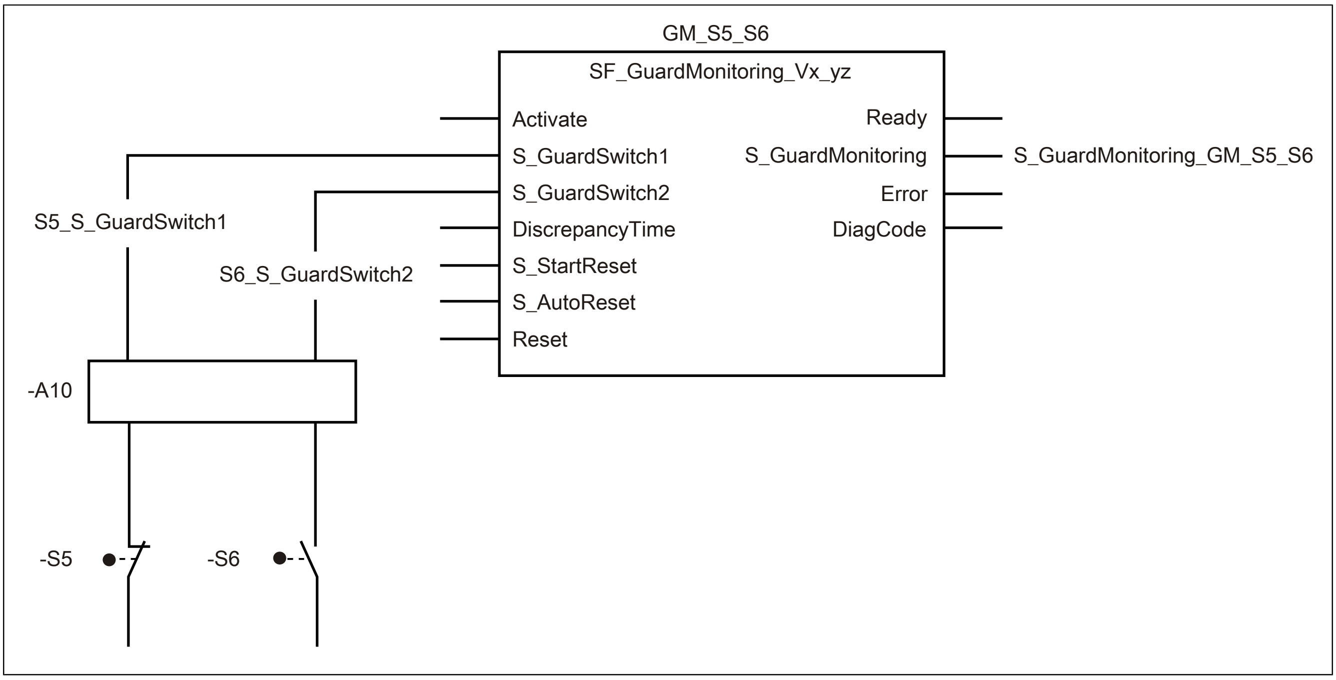


Figure 450: "SF_GuardMonitoring": Interlocking device with 2 mechanically actuated position switches (1-channel)

List of equipment

-S5	Safety door switch, 1-channel, positively driven when opening the protective equipment (TRUE when protective equipment closed)
-S6	Safety door switch, 1-channel, positively driven and positive opening operation with closed protective equipment
-A10	1-channel inputs of a safe input device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S5_S_GuardSwitch1	Input on "S_GuardSwitch1"
S6_S_GuardSwitch2	Input on "S_GuardSwitch2"
S_GuardMonitoring_GM_S5_S6	Output on "S_GuardMonitoring"

Description

In this example:

- The signal of safety door switch "-S5" from the input of safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameter "S_GuardSwitch1" for further processing.
- The signal of safety door switch "-S6" from the input of safe input device "-A10" is connected to input "S6_S_GuardSwitch2".
- Input "S6_S_GuardSwitch2" is connected to function block input parameter "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5_S6".
- Output "S_GuardMonitoring_GM_S5_S6" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.7.2.2 Interlocking device with 2 non-mechanically actuated position switches (1-channel)

This example illustrates the connection of the function block when controlling using the signals from 2 non-mechanically actuated position switches (safety door switches), each connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5_S6".

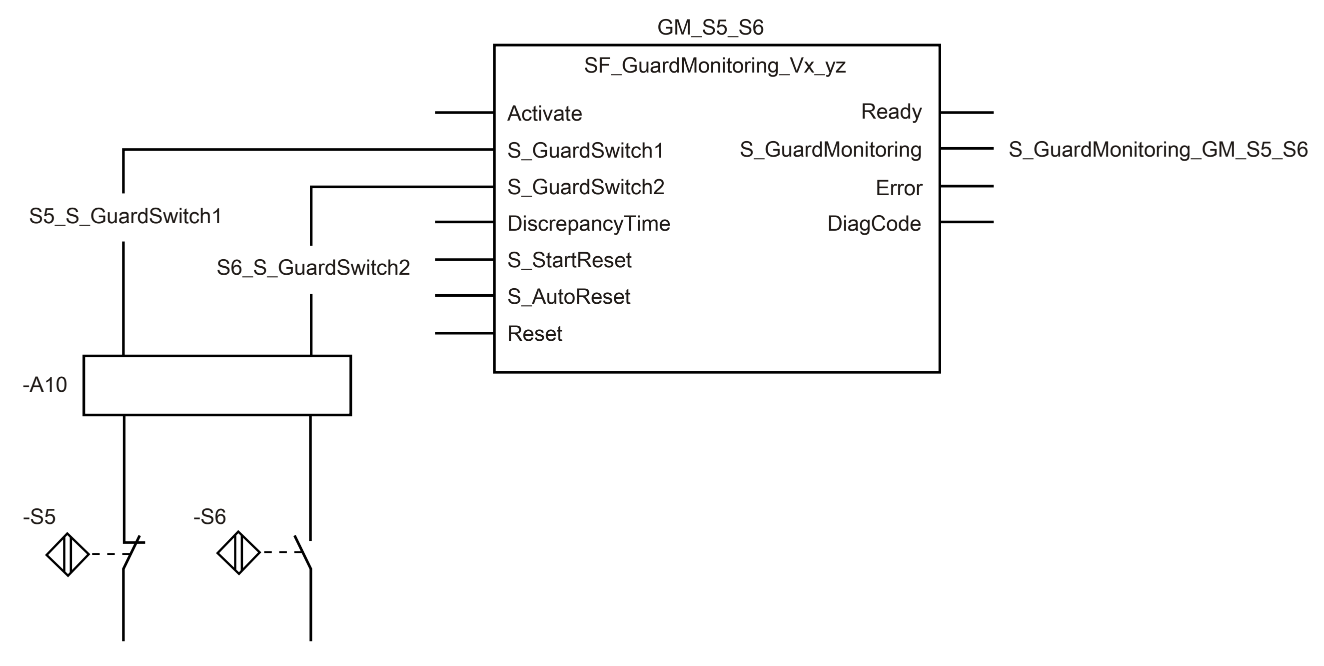


Figure 451: "SF_GuardMonitoring": Interlocking device with 2 non-mechanically actuated position switches (1-channel)

<u>List of equipment</u>

-S5	Electronic safety door switch, 1-channel (semiconductor outputs, positive switching, protected against short circuit)
-S6	Electronic safety door switch, 1-channel (semiconductor outputs, positive switching, protected against short circuit)
-A10	1-channel inputs of a safe input device

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

<u>Connected inputs and outputs</u>

S5_S_GuardSwitch1	Input on "S_GuardSwitch1"
S6_S_GuardSwitch2	Input on "S_GuardSwitch2"
S_GuardMonitoring_GM_S5_S6	Output on "S_GuardMonitoring"

Description

In this example:

- The signal of safety door switch "-S5" from the input of safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameter "S_GuardSwitch1" for further processing.
- The signal of safety door switch "-S6" from the input of safe input device "-A10" is connected to input "S6_S_GuardSwitch2".
- Input "S6_S_GuardSwitch2" is connected to function block input parameter "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5_S6".
- Output "S_GuardMonitoring_GM_S5_S6" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.7.2.3 Interlocking device with 2 mechanically actuated position switches (2-channel, antivalent)

This example illustrates the connection of the function block when controlling using the signals from 2 mechanically actuated position switches (safety door switches), each connected antivalently over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "GM_S5_S6".

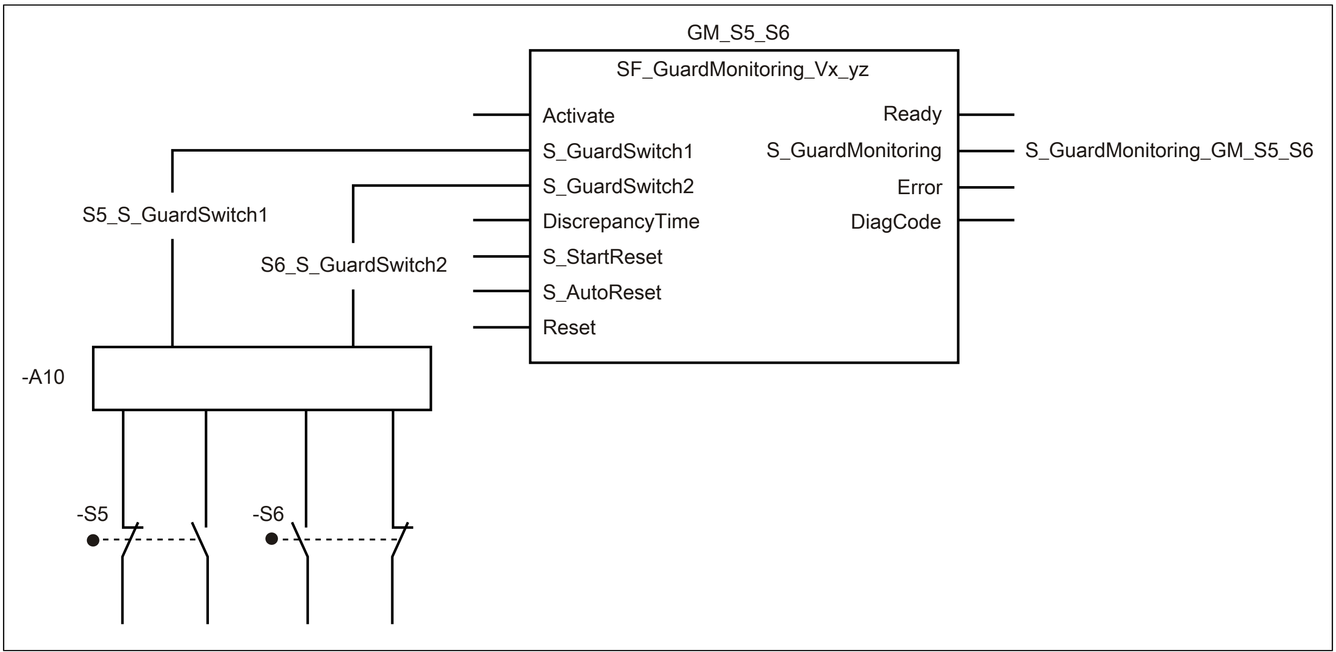


Figure 452: "SF_GuardMonitoring": Interlocking device with 2 mechanically actuated position switches (2-channel, antivalent)

<u>List of equipment</u>

-S5	Safety door switch, 2-channel antivalent, positively driven when opening the safety door
-S6	Safety door switch, 2-channel antivalent, positively driven when opening the safety door (contacts connected antivalently to switch "-S5")
-A10	2-channel inputs each of a safe input device with line control and antivalent evaluation per safety door switch

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

<u>Connected inputs and outputs</u>

S5_S_GuardSwitch1	Input on "S_GuardSwitch1"
S6_S_GuardSwitch2	Input on "S_GuardSwitch2"
S_GuardMonitoring_GM_S5_S6	Output on "S_GuardMonitoring"

Description

In this example:

- The resulting signal of safety door switch "-S5" from the 2-channel input of safe input device "-A10" is connected to input "S5_S_GuardSwitch1".
- Input "S5_S_GuardSwitch1" is connected to function block input parameter "S_GuardSwitch1" for further processing.
- The resulting signal of safety door switch "-S6" from the 2-channel input of safe input device "-A10" is connected to input "S6_S_GuardSwitch2".
- Input "S6_S_GuardSwitch2" is connected to function block input parameter "S_GuardSwitch2" for further processing.
- Output parameter "S_GuardMonitoring" is connected to output "S_GuardMonitoring_GM_S5_S6".
- Output "S_GuardMonitoring_GM_S5_S6" is used as a release signal to control the process in consideration of other safety functions.

6.6.11.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 953	Control guard	If state FALSE exists on one or both safe function block inputs ("S_GuardSwitch1" / "S_GuardSwitch2"), then the enable output will be set to FALSE. If state TRUE exists on "S_AutoReset" and both safe inputs ("S_GuardSwitch1" and "S_GuardSwitch2") have been set symmetrically from FALSE to TRUE, then the enable output will be set to TRUE when the function block is active. If state TRUE exists on "S_StartReset" and state TRUE exists on both safe inputs ("S_GuardSwitch1" and "S_GuardSwitch2"), then the enable output is set to TRUE when the function block is enabled. You are solely responsible for planning and implementing all requirements related to using control guards.
EN 1088	Interlocking guard	If state FALSE exists on one or both safe function block inputs ("S_GuardSwitch1" and "S_GuardSwitch2"), then the enable output will be set to FALSE. If state FALSE exists on "S_AutoReset" and both safe inputs ("S_GuardSwitch1" and "S_GuardSwitch2") have been set symmetrically from FALSE to TRUE, then the enable output will be set to TRUE when the function block is active and a signal change from FALSE to TRUE has taken place on "Reset". If state FALSE exists on "S_StartReset" and state TRUE exists on both signal inputs, then the enable output is set to TRUE when the function block is enabled after a signal change from FALSE to TRUE takes place on "Reset".
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	If state FALSE exists on "S_StartReset" and TRUE exists on both safe function blocks inputs ("S_GuardSwitch1" and "S_GuardSwitch2"), then the enable output is set to TRUE when the function block is enabled after a signal change from FALSE to TRUE takes place on "Reset" .
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 1088	Direct opening action of a contact element	The position switches you are using must comply with EN 1088 requirements. In addition, you are responsible for everything involved in designing and implementing the protective equipment.
EN 1088	Two-stage interlocking device Interlocking device without guard locking	If state FALSE exists on one or both safe function block inputs ("S_GuardSwitch1" and "S_GuardSwitch2"), then the enable output will be set to FALSE. If both safe function block inputs ("S_GuardSwitch1" and "S_GuardSwitch2") are set symmetrically from FALSE to TRUE, then the enable output is set to TRUE when the function block is active. Depending on the specification on "S_AutoReset", a signal change from FALSE to TRUE on "Reset" may additionally be required. If state TRUE exists on both safe function block inputs ("S_GuardSwitch1" and "S_GuardSwitch2"), then the enable output will be set to TRUE when the function block is enabled. Depending on the specification on "S_StartReset", a signal change from FALSE to TRUE on "Reset" may additionally be required.

Table 643: "SF_GuardMonitoring": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.12 SF_ModeSelector

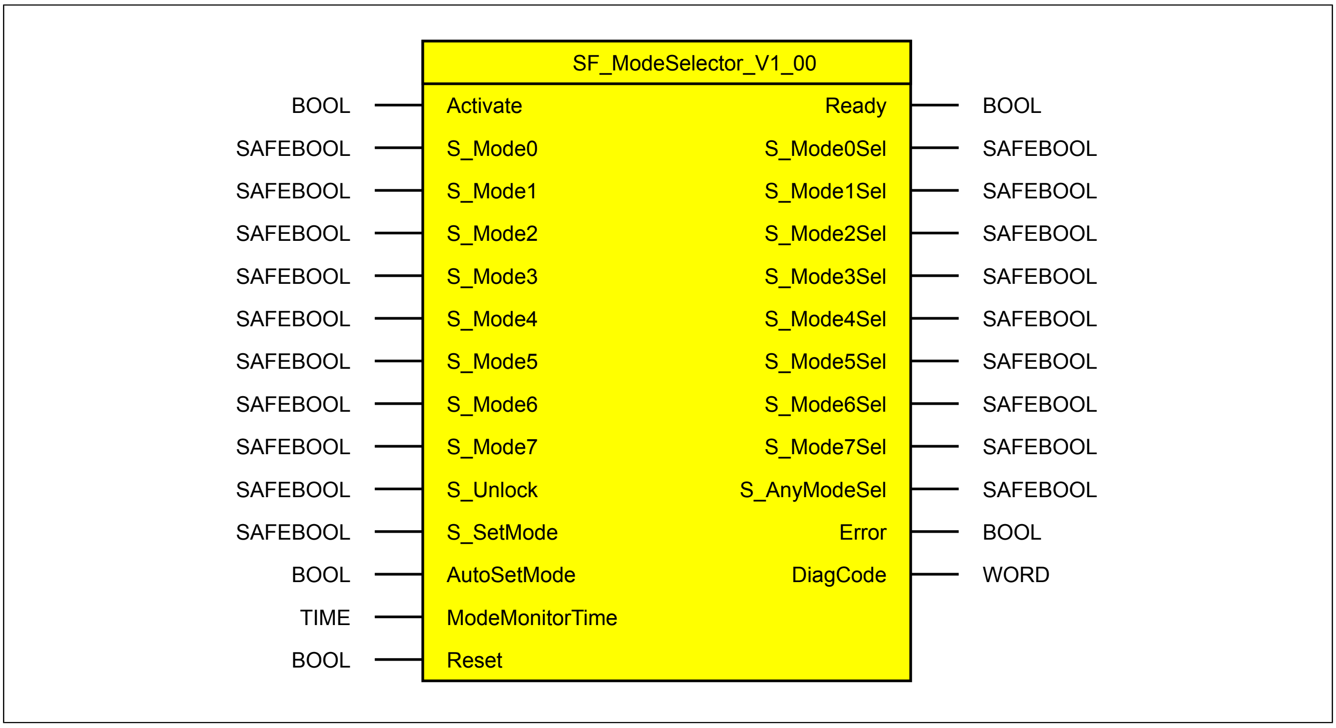


Figure 453: Function block "SF_ModeSelector"

6.6.12.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_ModeX (X = 0 to 7)	SAFEBOOL	Variable/ Constant	State	FALSE	Signal inputs X (X = 0 to 7) for selecting the operating mode of the mode selector switch. TRUE: Operating mode X is selected.
S_Unlock	SAFEBOOL	Variable/ Constant	State	FALSE	Input for configuring the operating mode lock. TRUE: Changing the operating mode is possible.
S_SetMode	SAFEBOOL	Variable/ Constant	State	FALSE	Input for acknowledgment/release of the configured operating mode
AutoSetMode	BOOL	Constant	State	FALSE	Specification for initiating an operating mode change automatically on "S_ModeXSel". TRUE: Automatic operating mode change is active.
ModeMonitorTime	TIME	Constant	State	#0ms	Specification for the maximum permitted time for an operating mode change (state change on inputs "S_ModeX")
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 644: "SF_ModeSelector": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_ModeXSel (X = 0 to 7)	SAFEBOOL	Variable	State	FALSE	Release signal for the selected operating mode
S_AnyModeSel	SAFEBOOL	Variable	State	FALSE	Message that a signal combination is being output on outputs "S_ModeXSel"
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 645: "SF_ModeSelector": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 646: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.12.2 Function

Function block "SF_ModeSelector" is used to support a mode selector switch function with up to 8 signals in an application.

You specify an operating mode for a defined safety level with a TRUE signal on function block input parameters "S_Mode0" to "S_Mode7" using a connected mode selector switch. The operating mode (e.g. service mode, clean mode, jog mode, setup mode or automatic mode) is meant to be executed by the safety application.

Due to the influence of the mechanical contacts in the mode selector switch, state TRUE may exist for more than one signal or no signal at all when switching the mode selector switch. You specify a time frame on input parameter "ModeMonitorTime", within which these states are permitted when switching. Outside of this time frame, the function block detects these states as an error.

The signal states on "S_Mode0" to "S_Mode7" are output on the corresponding output parameters "S_Mode0Sel" to "S_Mode7Sel". This output takes place either automatically ("AutoSetMode" = TRUE) or manually ("AutoSetMode" = FALSE) when enabled on a rising edge of input parameter "S_SetMode".

Danger!

Optionally specify automatic output on outputs "S_ModeXSel" (apply values on operating mode change) only if there are assurances that no dangerous situation can occur in the safeguarded area when the safety controller starts or if there are other measures in place preventing startup.

A signal change on "S_Mode0" to "S_Mode7" has no effect on TRUE signals on output parameters "S_Mode0Sel" to "S_Mode7Sel" if these TRUE signals are locked against changes during the execution time of the function block.

While the TRUE signals on "S_Mode0Sel" to "S_Mode7Sel" are locked against changes, the function block does not check the signals on input parameters "S_Mode0" to "S_Mode7" for plausibility. In addition, the values on the input parameters have no effect on output parameters "S_Mode0Sel" to "S_Mode7Sel".

Connect output parameters "S_Mode0Sel" to "S_Mode7Sel" to the safety application in such a way that the application downstream from the safety application implements the selected operating mode with the corresponding connections based on the values outputs on "S_Mode0Sel" to "S_Mode7Sel".

Danger!

The function block outputs a signal to outputs "S_Mode0Sel" to "S_Mode7Sel" according to the operating mode selected on the inputs ("S_Mode0" to "S_Mode7"). This function block does not ensure that the selected operating mode is executed by the downstream application, however.

Based on the requirements of the safety level for the selected operating mode, make sure that the signal output by the function block is connected properly to the downstream application and that it is processed properly by the downstream application.

The active function block only checks the plausibility of the states on "S_Mode0" to "S_Mode7" if the signals on outputs "S_Mode0Sel" to "S_Mode7Sel" are not locked against changes.

6.6.12.2.1 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameter ("AutoSetMode").

A start interlock is active after a signal change on "S_ModeX" (X = 0 to 7) when the output signal is not locked and/or after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "S_SetMode" if the input signal combination is permitted for this.

You can reset a start interlock that was enabled after a detected error with a rising edge on input parameter "Reset".

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.12.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.12.3.1 More than one TRUE signal on input parameters "S_ModeX"

Multiple TRUE signals on input parameters "S_Mode0" to "S_Mode7" are detected by the enabled function block as an error if output parameters "S_Mode0Sel" to "S_Mode7Sel" are not locked against changes.

In the locked state, no plausibility check takes place for input signals "S_Mode0" to "S_Mode7", and these input signals have no effect on the operating mode output by the function block on "S_Mode0Sel" to "S_Mode7Sel".

Possible causes:

- The same or incorrect variables are connected to input or output parameters in the safety application (programming error, user error).
- Cross fault in cables (wiring error, user error)
- Defective mode selector switch (hardware fault)

6.6.12.3.2 No TRUE signal on input parameters "S_ModeX"

No TRUE signal existing on input parameters "S_Mode0" to "S_Mode7" is detected by the enabled function block as an error if output parameters "S_Mode0Sel" to "S_Mode7Sel" are not locked against changes.

In the locked state, no plausibility check takes place for input signals "S_Mode0" to "S_Mode7", and these input signals have no effect on the operating mode output by the function block on "S_Mode0Sel" to "S_Mode7Sel".

Possible causes:

- Incorrect variables are connected to input or output parameters in the safety application (user error).
- Incorrect device parameterization (user error)
- Wiring error (user error)
- Open circuit in cables (user error, wiring error)
- Defective mode selector switch (hardware fault)

6.6.12.3.3 Time frame on "ModeMonitorTime" not dimensioned correctly

A time frame that is too short is recognized as an error by the function block when the operating mode changes. If the time frame is too long, a missing signal on "S_ModeX" is not recognized as an error.

Possible causes:

- Programming error (user error)
- Incorrectly calculated value for time frame (user error)

6.6.12.3.4 Static TRUE signals on "S_SetMode"

If a static TRUE signal exists on input parameter "S_SetMode", it is not possible to represent an operating mode requested with a mode selector switch on the output parameters. In this case, no operating mode is active. This results in a loss of plant availability.

Possible causes:

- Unintentional specification of a static TRUE signal on input parameter "S_SetMode" in the safety application (user error)
- Wiring error (user error)

6.6.12.3.5 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> ## Danger!
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.12.3.6 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.12.3.7 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "S_SetMode" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled.

If this start interlock is not specified for a cold restart of the safety controller ("AutoSetMode" = TRUE), then the state of "S_SetMode" is irrelevant. In this case, the status of the release signal depends on the status of "S_ModeX" (X = 0 to 7).

6.6.12.3.8 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.12.3.9 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.12.4 Input parameters

6.6.12.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (depending on the specification on "AutoSetMode") after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock if "AutoSetMode" is set to state FALSE. The start interlock is reset by a rising edge on "S_SetMode" if "S_Unlock" and a signal on "S_Mode0" to "S_Mode7" has state TRUE. An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.12.4.2 S_ModeX (X = 0 to 7)

General function

- Signal inputs X (X = 0 to 7) for selecting the operating mode of the mode selector switch

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **Connect these input parameters to the signals of a safe input device that is connected to signals X of the mode selector switch (bit X). Inputs "S_ModeX" are then controlled using these signals.**

Function description

The signal combination connected to input parameters "S_ModeX" is processed by the function block and output to output parameters "S_ModeXSel" accordingly.

Multiple TRUE signals on input parameters "S_Mode0" to "S_Mode7" are detected by the enabled function block as an error if output parameters "S_Mode0Sel" to "S_Mode7Sel" are not locked against changes.

No TRUE signal existing on input parameters "S_Mode0" to "S_Mode7" is detected by the enabled function block as an error if output parameters "S_Mode0Sel" to "S_Mode7Sel" are not locked against changes.

When the operating mode changes, a state change on function block signal inputs "S_ModeX" is not necessarily indicated on outputs "S_ModeXSel". The previously configured operating mode must be unlocked for this ("S_Unlock" = TRUE). If the configured operating mode is unlocked, a state change on input parameters "S_ModeX" is output to output parameters "S_ModeXSel" only when a signal change takes place on "S_SetMode" (FALSE → TRUE). Optionally, a state change on inputs "S_ModeX" can be output automatically to outputs "S_ModeXSel" (if "AutoSetMode" = TRUE).

Note that the signals on "S_Mode0" to "S_Mode7" are not checked for plausibility if the signal combination set on outputs "S_ModeXSel" is locked against changes. The values on "S_Mode0" to "S_Mode7" are irrelevant for the output signal in this case.

The signal inputs are state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

Operating mode X is requested by the connected mode selector switch.

FALSE

Operating mode X is not requested by the connected mode selector switch. The mode selector switch is not actuated, the wiring for the mode selector switch is interrupted or the safe device connected to the mode selector switch is shut down or defective.

6.6.12.4.3 S_Unlock

General function

- Input for configuring the operating mode lock

Data type

- SAFEBOOL

Connection

- Variable or constant

Danger!

Note that it is not possible to control the operating mode latch dynamically if you connect this input parameter to a literal. In this case, a static TRUE signal causes outputs "S_ModeXSel" to not be controlled by the function block.

A static FALSE signal on "S_Unlock" causes outputs "S_ModeXSel" to be able to be controlled by the function block. A TRUE signal output on a "S_ModeXSel" output cannot be locked against changes, however.

You will typically control input parameter "S_Unlock" with a key switch or defined access code.

Function description

The signal connected to input parameter "S_Unlock" is processed by the function block.

A TRUE signal output on a "S_ModeXSel" output can be locked against changes with a FALSE signal on "S_Unlock". This prevents a signal change on inputs "S_ModeX" from being passed on to the outputs at runtime.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The signal combination output on "S_ModeXSel" (operating mode X) is not locked.
Changing the operating mode is possible.

FALSE

The signal combination output on "S_ModeXSel" (operating mode X) is locked.
Changing the operating mode is not possible.

A change to inputs "S_ModeX" and/or a rising edge on input parameter "S_SetMode" does not cause a change to outputs "S_ModeXSel".

In the locked state, no plausibility check takes place for input signals "S_Mode0" to "S_Mode7". These input signals have no effect on the operating mode output by the function block on "S_Mode0Sel" to "S_Mode7Sel".

6.6.12.4.4 S_SetMode

General function

- Input for acknowledgment/release of the configured operating mode

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
> **You can specify a static FALSE signal on "S_SetMode" in order to fully connect the function block input interface when "AutoSetMode" = TRUE. The function block detects a static TRUE signal on "S_SetMode" at runtime as an error.**

Function description

The signal connected to input parameter "S_SetMode" is processed by the function block. The edges of the input parameter are monitored internally by the function block. Acknowledgment only takes place on a rising edge. An additional static TRUE signal following a rising edge does not trigger acknowledgment again.

The function block detects a static TRUE signal on "S_SetMode" as an error in states where a rising edge on "S_SetMode" is required. Set "S_SetMode" to FALSE to exit the error state.

Input parameter "S_SetMode" supports a start interlock when the function block is enabled and/or after a change in operating mode.

The following settings must be taken into account for this:

Signal changes on inputs "S_ModeX" (operating mode change) are not applied to outputs "S_ModeXSel" if you specify a FALSE signal on "S_Unlock". All "S_ModeXSel" outputs indicate state FALSE in this case. In this case, the function block outputs diagnostic code WORD#16#8005 on "DiagCode".

The function block outputs the changes made on "S_ModeX" to outputs "S_ModeXSel" if:

- "S_Unlock": TRUE and
- "S_SetMode": FALSE → TRUE and
- "AutoSetMode": FALSE

> ## Information:
> **If state TRUE exists for "AutoSetMode", a signal change on "S_SetMode" is not required to apply the values on "S_ModeXSel".**

6.6.12.4.5 AutoSetMode

General function

- Specification for initiating an operating mode change automatically on "S_ModeXSel"

Data type

- BOOL

Connection

- Constant

Function description

The signal connected to input parameter "AutoSetMode" is processed by the function block.

Signal changes on inputs "S_ModeX" (operating mode change) are applied to outputs "S_ModeXSel" automatically only if you specify a TRUE signal on "S_Unlock" and "AutoSetMode".

TRUE

If state TRUE exists for "AutoSetMode", the values on "S_ModeXSel" are applied automatically without additional acknowledgment.

> **Danger!**
>
> **Optionally specify automatic output on outputs "S_ModeXSel" (apply values on operating mode change) only if there are assurances that no dangerous situation can occur in the safeguarded area when the safety controller starts or if there are other measures in place preventing startup.**

FALSE

If state FALSE exists for "AutoSetMode", the values on "S_ModeXSel" are not applied automatically without additional acknowledgment. Acknowledgment takes place on a rising edge of input parameter "S_SetMode".

6.6.12.4.6 ModeMonitorTime

General function

- Specification for the maximum permitted time for an operating mode change (state change on inputs "S_ModeX")

Data type

- TIME

Connection

- Constant

Function description

An impermissible input signal combination on inputs "S_ModeX" (all signals = FALSE) is detected by the enabled function block as an error after the time specified with "ModeMonitorTime" expires if the signals on "S_ModeXSel" are not locked against changes ("S_Unlock" = TRUE).

Due to mechanical effects when switching the mode selector switch connected to the function block or due to a fault (e.g. programming error, wiring error), it is possible that state TRUE does not exist on any "S_ModeX" inputs. These states are only permitted in the time frame specified on "ModeMonitorTime". Outside of this time frame, the function block detects these states as an error if the signals on "S_ModeXSel" are not locked against changes and a change takes place.

You must define and validate the time value for input parameter "ModeMonitorTime" based on your application and risk analysis.

6.6.12.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting manual reset of a start interlock

Data type

- BOOL

> ## Danger!
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

In addition, this input parameter is used after a start interlock to output to the "S_ModeXSel" outputs the signal combination that corresponds to the operating mode requested on the "S_ModeX" inputs (output "S_ModeXSel" = TRUE).

6.6.12.5 Output parameters

6.6.12.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.12.5.2 S_ModeXSel (X = 0 to 7)

General function

- Release signal for the selected operating mode

Data type

- SAFEBOOL

Connection

- Variable

Information:

Make logical connections to the safety application with the signal combination on outputs "S_ModeXSel" in order to implement (program/control) the requested operating mode in the safety application. Each output signal combination contains maximum one TRUE signal of an output.

You must implement these connections in the program on the safety controller in such a way that the function you have defined (e.g. manual mode, automatic mode) are locked or blocked with a FALSE signal. With a TRUE signal and appropriate AND operator, the defined functions are unlocked in the program.

Plan, implement and validate the connections according to the results of your risk analysis.

Function description

Signal changes on inputs "S_ModeX" (operating mode change) are only applied to outputs "S_ModeXSel" if you specify a TRUE signal on "S_Unlock".
If state TRUE exists for "AutoSetMode", the values on outputs "S_ModeXSel" are applied automatically without additional acknowledgment.
If state FALSE exists for "AutoSetMode", the values on outputs "S_ModeXSel" are not applied without additional acknowledgment.

Danger!

Optionally specify automatic output on outputs "S_ModeXSel" (apply values on operating mode change) only if there are assurances that no dangerous situation can occur in the safeguarded area when the safety controller starts or if there are other measures in place preventing startup.

Danger!

The function block outputs on outputs "S_ModeXSel" a signal combination for the operating mode requested on inputs "S_ModeX".

Note that the function block does not ensure that the output operating mode is executed by the downstream application.

Based on the requirements of the safety level for the selected operating mode, make sure that the signal output by the function block is processed properly by the downstream application.

TRUE

Operating mode X is requested in the safety application (program on the safety controller).

FALSE

Operating mode X is not requested in the safety application (program on the safety controller).

6.6.12.5.3 S_AnyModeSel

General function

- Message that a signal combination is being output on outputs "S_ModeXSel" to set an operating mode in the downstream application

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this signal to evaluate whether a signal combination is output on outputs "S_ModeXSel" in order to set an operating mode in the downstream application.

Danger!

Note that this signal does not provide any information about on which of the "S_Mode0Sel" to "S_Mode7Sel" outputs a TRUE signal is output.

Function description

This output parameter indicates that one of the configurable operating modes is being output to outputs "S_ModeXSel".

TRUE

Signal TRUE is output on one of the "S_ModeXSel" outputs (X = 0 to 7) in order to set to operating mode X in the downstream application.

FALSE

Only FALSE signals are output to outputs "S_ModeXSel" (X = 0 to 7).

6.6.12.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.12.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.12.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The operating mode requested on signal inputs "S_Mode0" to "S_Mode7" is output on the corresponding output parameter "S_Mode0Sel" to "S_Mode7Sel". The operating mode configured on one of the "S_Mode0Sel" to "S_Mode7Sel" output parameters is not locked against changes.	Locking the operating mode output by the function block: • Set "S_Unlock" to FALSE to lock the operating mode output on "S_Mode0Sel" to "S_Mode7Sel" against changes. Changing the operating mode output by the function block: • Set the corresponding input parameter ("S_Mode0" to "S_Mode7") to TRUE to specify the necessary operating mode on the function block. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the mode selector switch connected to the function block. • Check the configuration of the safe device(s) connected to the mode selector switch.
8004	The operating mode requested on signal inputs "S_Mode0" to "S_Mode7" is output on the corresponding output parameter "S_Mode0Sel" to "S_Mode7Sel". The operating mode configured on one of the "S_Mode0Sel" to "S_Mode7Sel" output parameters is locked against changes. A signal change on "S_Mode0" to "S_Mode7" has no effect on the state of output parameters "S_Mode0Sel" to "S_Mode7Sel".	Unlocking the operating mode output by the function block: • Set "S_Unlock" to TRUE to unlock the operating mode output on "S_Mode0Sel" to "S_Mode7Sel" against changes. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8005	The function block was enabled or the operating mode requested on "S_Mode0" to "S_Mode7" was changed. Signal FALSE is output to each output parameter "S_Mode0Sel" to "S_Mode7Sel".	Outputting the selected operating mode: • Connect a constant with value TRUE to "AutoSetMode" or set "S_SetMode" from FALSE to TRUE. • Note that state TRUE must exist on "S_Unlock". • Note that the operating mode output in the following state of the function block is not locked against changes. To do so, you must set input parameter "S_Unlock" from TRUE to FALSE after the operating mode requested on inputs "S_Mode0" to "S_Mode7" is output on outputs "S_Mode0Sel" to "S_Mode7Sel". Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
C001	The function block detected an invalid signal combination on input parameters "S_Mode0" to "S_Mode7". More than one of these input parameters were set to TRUE at the same time outside of the time frame specified on "ModeMonitorTime".	• Check the safety functions. • Check the safety program. • Check the safe peripheral.
C002	The function block detected an invalid signal combination on input parameters "S_Mode0" to "S_Mode7". All of these input parameters were set to FALSE at the same time outside of the time frame specified on "ModeMonitorTime".	• Check the safety functions. • Check the safety program. • Check the safe peripheral.
C003	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C100	The function block detected a static TRUE signal on "S_SetMode".	• Check the operating element connected to "S_SetMode" and associated wiring. • The signal on "S_SetMode" must indicate state FALSE. • Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error. • Check the safety functions. • Check the safety program. • Check the safe peripheral.

Table 647: "SF_ModeSelector": Diagnostic codes

6.6.12.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

The following image shows a valid change in operating mode with acknowledgment/release on "S_SetMode".

"AutoSetMode" = FALSE

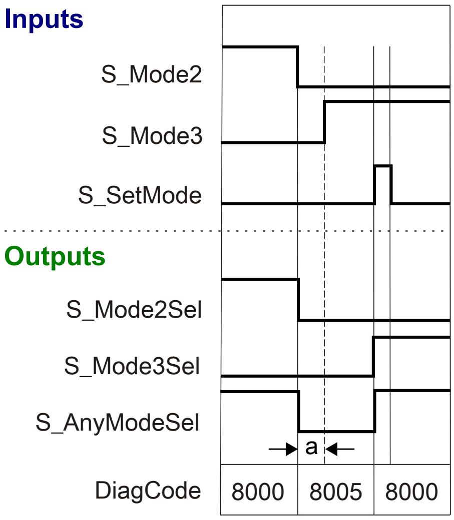


Figure 454: "SF_ModeSelector": Signal sequence diagram 1

a Configured "ModeMonitorTime"

Signal sequence diagram 2

The following image shows the behavior of the function block when all inputs "S_Mode0" to "S_Mode7" indicate state FALSE for longer than the time specified on "ModeMonitorTime".

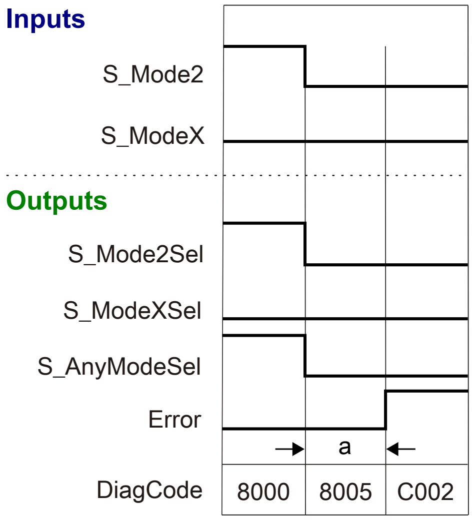


Figure 455: "SF_ModeSelector": Signal sequence diagram 2

a Configured "ModeMonitorTime"

Signal sequence diagram 3

The following image shows the behavior of the function block when an existing error has been reset.

"AutoSetMode" = FALSE

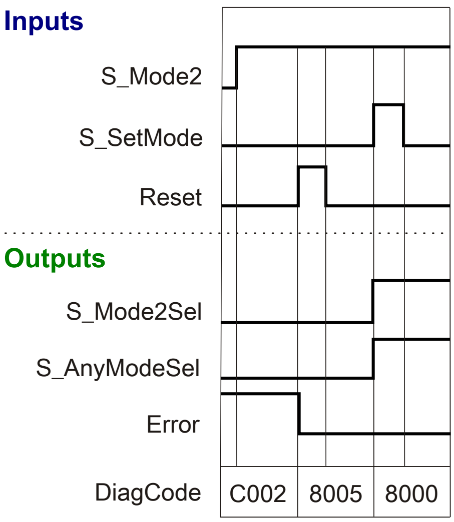


Figure 456: "SF_ModeSelector": Signal sequence diagram 3

6.6.12.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement selection of a safe operating mode.

The following example describes function block connections when controlling with the signals from a mode selector switch (see section 6.6.12.7.2 "Mode selection without locking of the mode selector switch with enable button").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.12.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "MS_S5".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Locking the mode selector switch

State TRUE on input parameter "S_Unlock" is required to switch the operating mode. This signal usually comes from a key switch connected to a safe input device. The signal from the key switch is used to control locking for the mode selector switch, i.e. whether changing the operating mode is permitted or not.

Enabling the operating mode

Input parameter "AutoSetMode" determines the behavior of the function block when changes are made to the position of the mode selector switch. This input parameter is connected to the FALSE constant. After the switch position has changed, a rising edge on "S_SetMode" is required to enable the configured operating mode.

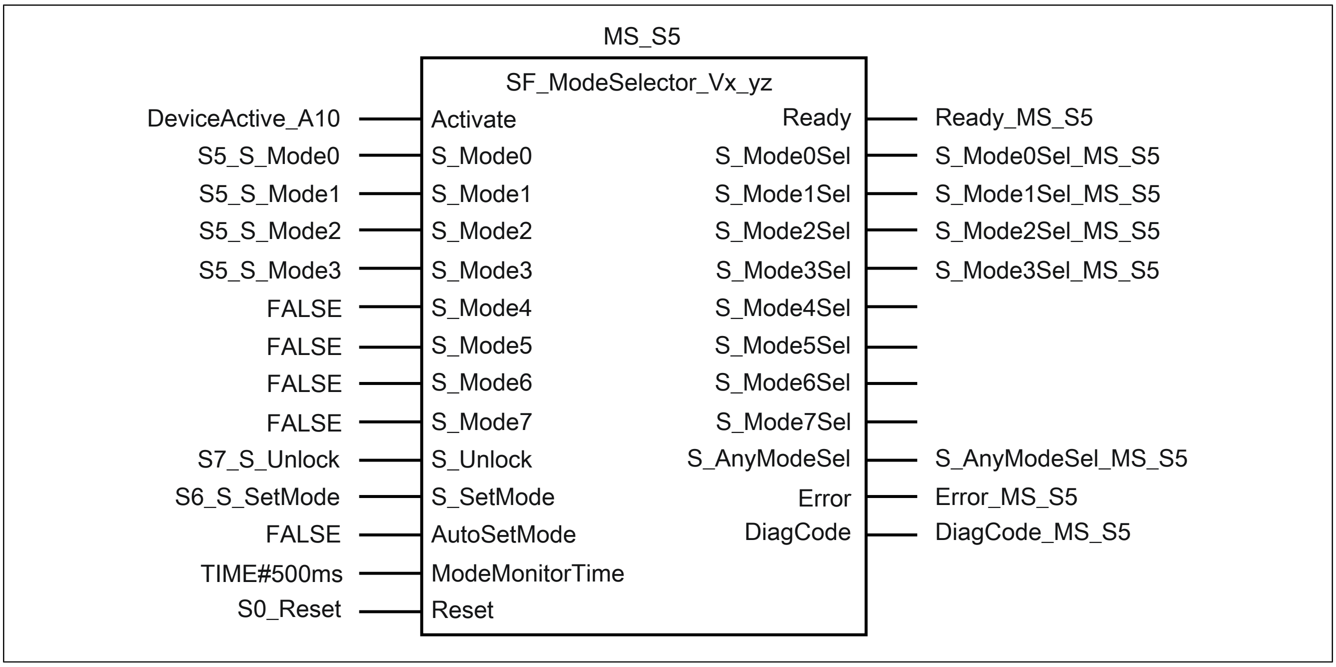


Figure 457: "SF_ModeSelector": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the mode selector switch is connected.
S5_S_Mode0	SAFEBOOL	Signal combination from the connected mode selector switch
S5_S_Mode1	SAFEBOOL	
S5_S_Mode2	SAFEBOOL	
S5_S_Mode3	SAFEBOOL	
FALSE on "S_Mode4"	SAFEBOOL	Inputs "S_Mode4" to "S_Mode7" are each connected to constant FALSE (not used).
FALSE on "S_Mode5"	SAFEBOOL	
FALSE on "S_Mode6"	SAFEBOOL	
FALSE on "S_Mode7"	SAFEBOOL	
S7_S_Unlock	SAFEBOOL	Signal for locking the configured operating mode. The states that are configured on outputs "S_Mode0Sel" to "S_Mode3Sel" are locked against changes. In this example, "S7_S_Unlock" must be controlled by the application in such a way that the states on outputs "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5" are locked against changes when a valid signal combination is output.
S6_S_SetMode	SAFEBOOL	This external signal controls the application of states on inputs "S5_S_Mode0" to "S5_S_Mode3" for outputs "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5". This applies if a valid signal combination exists on the inputs and a FALSE signal is output on each output "S_Mode4Sel_MS_S5" to "S_Mode7Sel_MS_S5" ("DiagCode" = WORD#16#8005). In this example, "S6_S_SetMode" must be controlled by the application in such a way that the function block applies the states on inputs "S5_S_Mode0" to "S5_S_Mode3" for outputs "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5".
FALSE on "AutoSetMode"	BOOL	The automatic application of input signals "S5_S_Mode0" to "S5_S_Mode3" to outputs "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5" is locked.
TIME#500ms on "ModeMonitorTime"	TIME	Time value during which the states on inputs "S5_S_Mode0" to "S5_S_Mode3" are not checked for plausibility.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 648: "SF_ModeSelector": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_MS_S5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_Mode0Sel_MS_S5	SAFEBOOL	Signal combination for implementing and controlling the requested operating mode by the downstream safety application (safety program) using logical operators. You must implement these connections in the program on the safety controller in such a way that the function you have defined (e.g. manual mode, automatic mode) are locked or blocked with a FALSE signal. With a TRUE signal and appropriate AND operator, the defined functions are unlocked in the program. **Danger!** **Note that the links required in the safe program are not illustrated since this example only shows the connection of the function block in principle. To implement this example, the required links must be planned in accordance with the results of a performed risk analysis and implemented in accordance with the application.**
S_Mode1Sel_MS_S5	SAFEBOOL	
S_Mode2Sel_MS_S5	SAFEBOOL	
S_Mode3Sel_MS_S5	SAFEBOOL	
S_AnyModeSel_MS_S5	SAFEBOOL	This signal controls whether the function block outputs a signal combination on "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5" in order to set an operating mode in the downstream application. The signal is evaluated in the downstream application.
Error_MS_S5	BOOL	Error message from function block for further external processing
DiagCode_MS_S5	WORD	Diagnostic message from function block for further external processing

Table 649: "SF_ModeSelector": Outputs connected to output parameters

6.6.12.7.2 Mode selection without locking of the mode selector switch with enable button

This example illustrates the connection of the function block when controlling using the signal from a mode selector switch.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.12.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "MS_S5".

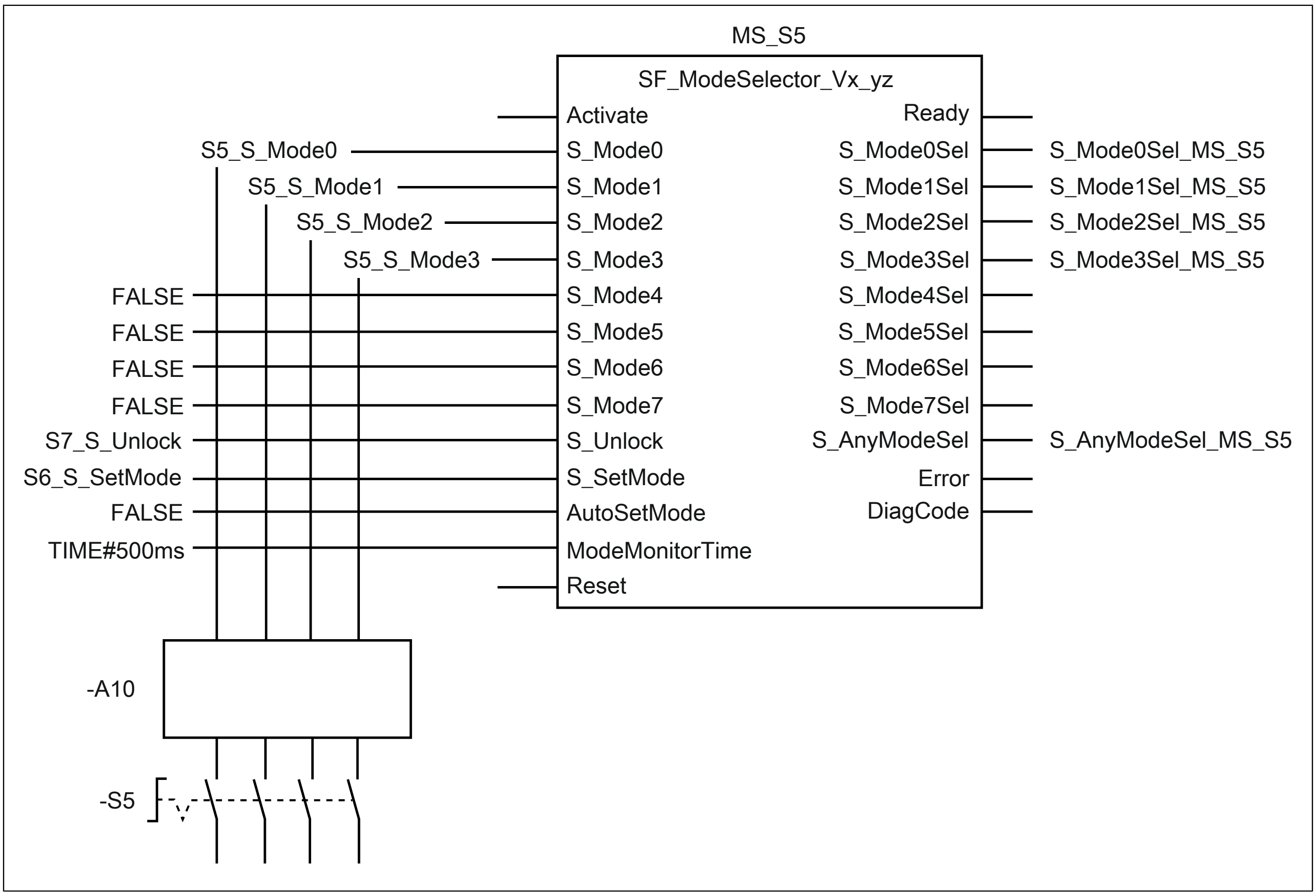


Figure 458: "SF_ModeSelector": Mode selection without locking of the mode selector switch with enable button

List of equipment

-S5	Mode selector switch (e.g. with 4 switch positions)
-A10	Inputs of a safe input device (the safe inputs are each configured for 1 channel)

Connected inputs and outputs

S5_S_Mode0	Input on "S_Mode0"
S5_S_Mode1	Input on "S_Mode1"
S5_S_Mode2	Input on "S_Mode2"
S5_S_Mode3	Input on "S_Mode3"
S7_S_Unlock	Input on "S_Unlock"
S6_S_SetMode	Input on "S_SetMode"
S_Mode0Sel_MS_S5	Output on "S_Mode0Sel"
S_Mode1Sel_MS_S5	Output on "S_Mode1Sel"
S_Mode2Sel_MS_S5	Output on "S_Mode2Sel"
S_Mode3Sel_MS_S5	Output on "S_Mode3Sel"
S_AnyModeSel_MS_S5	Output on "S_AnyModeSel"

Description

In this example:

- The signals of mode selector switch "-S5" from safe input device "-A10" are connected to inputs "S5_S_Mode0" to "S5_S_Mode3".
- Inputs "S5_S_Mode0" to "S5_S_Mode3" are connected to function block input parameters "S_Mode0" to "S_Mode3" for further processing. Signal "S7_S_Unlock" (TRUE) disables locking of the operating mode via the safety application.
- The signal of variable "S6_S_SetMode" is connected to function block input parameter "S_SetMode" for further processing.
- Input parameter "AutoSetMode" is connected to constant FALSE.
- Output parameter "S_AnyModeSel" is connected to output "S_AnyModeSel_MS_S5". The state of this parameter is processed further in the program on the safety controller.
- Output parameters "S_Mode0Sel" to "S_Mode3Sel" are connected to outputs "S_Mode0Sel_MS_S5" to "S_Mode3Sel_MS_S5". These outputs are used for further processing in the program on the safety controller.

6.6.12.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
MD 98/38/EC	Startup	The enabled function block monitors the signals of a connected mode selector switch for plausibility after a specified time frame ("ModeMonitorTime") has passed if the output parameters are not locked against changes ("S_Unlock" = TRUE). A maximum of one signal is permitted to indicate state TRUE for a signal input combination on "S_ModeX". In addition, all "S_ModeX" input parameters are not permitted to indicate state FALSE. Signals that deviate from these specifications after the time frame has expired are detected by the function block as faults. In addition, the function block monitors adjustments to the connected mode selector switch if the output parameters are not locked against changes ("S_Unlock" = TRUE = no locking). Optionally ("AutoSetMode" = FALSE), a signal change from FALSE to TRUE on "S_SetMode" is required in this state to apply the modified signal combination on "S_ModeXSel". You can lock the values output by output parameters "S_ModeXSel" against changes on the function block by setting "S_Unlock" from TRUE to FALSE if the output signal combination is output on "S_ModeXSel".
EN ISO 12100-2	Selecting control and operating modes	The function block provides you the option of locking a signal combination output on "S_ModeXSel" against changes. This can be done by setting "S_Unlock" from TRUE to FALSE if the output signal combination is output on "S_ModeXSel". A change to inputs "S_ModeX" then has no effect on "S_ModeXSel". Make logical connections to the safety application with the signal combination on outputs "S_ModeXSel" in order to implement (program/control) the requested operating mode in the safety application. You must implement these connections in the program on the safety controller in such a way that the function you have defined (e.g. manual mode, automatic mode) are locked or blocked with a FALSE signal. With a TRUE signal and appropriate AND operator, the defined functions are unlocked in the program. Plan, implement and validate the connections according to the results of your risk analysis.
EN 60204-1	Operating modes	The function block provides you the option of locking a signal combination output on "S_ModeXSel" against changes. This can be done by setting "S_Unlock" from TRUE to FALSE if the output signal combination is output on "S_ModeXSel". A change to inputs "S_ModeX" then has no effect on "S_ModeXSel". In addition, the function block monitors adjustments to the connected mode selector switch if the output parameters are not locked against changes ("S_Unlock" = TRUE = no locking). Optionally ("AutoSetMode" = FALSE), a signal change from FALSE to TRUE on "S_SetMode" is required in this state to apply the modified signal combination on "S_ModeXSel".
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("AutoSetMode" = FALSE) • Enabling the function block ("AutoSetMode" = FALSE) • Changing the signal combination on "S_ModeX" ("AutoSetMode" = FALSE) • Error detection by the function block If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal and/or the signal on "S_SetMode".
EN 954-1	Category B to 4	Error detection on "S_ModeX": The enabled function block monitors the signals of a connected mode selector switch for plausibility after a specified time frame ("ModeMonitorTime") has passed if the output parameters are not locked against changes ("S_Unlock" = TRUE). A maximum of one signal is permitted to indicate state TRUE for a signal input combination on "S_ModeX". A possible cross fault (more than one TRUE signal on inputs "S_ModeX") is therefore detected by the function block. In addition, all "S_ModeX" input parameters are not permitted to indicate state FALSE. Signals that deviate from these specifications after the time frame has expired are detected by the function block as faults. A possible open circuit (all signals on inputs "S_ModeX" = TRUE) is therefore detected by the function block if the request for the operating mode on inputs "S_ModeX" is interrupted as a result of this open circuit. The following applies to all other signals: 1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_EStopOut") executes a category 0 stop.

Table 650: "SF_ModeSelector": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.13 SF_MutingPar

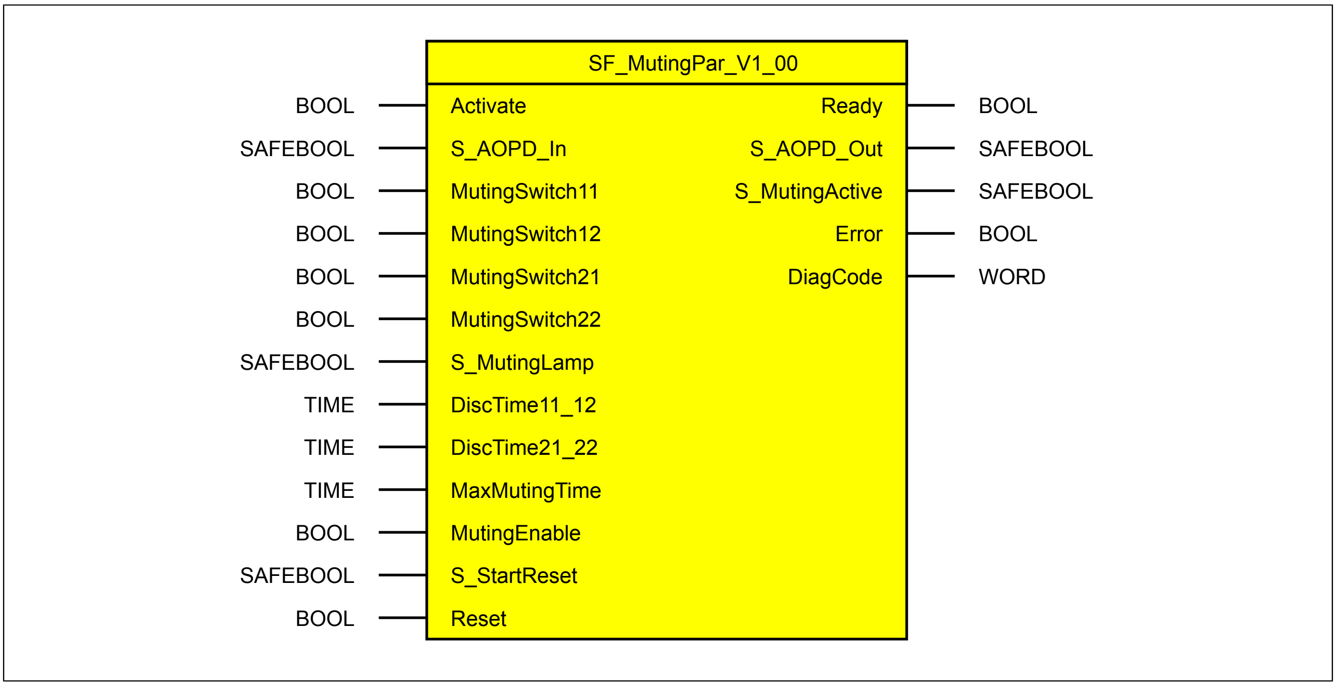


Figure 459: Function block "SF_MutingPar"

6.6.13.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_AOPD_In	SAFEBOOL	Variable	State	FALSE	Signal input of protective equipment (light curtain)
MutingSwitch11	BOOL	Variable	State	FALSE	Signal input of muting sensor 1
MutingSwitch12	BOOL	Variable	State	FALSE	Signal input of muting sensor 2
MutingSwitch21	BOOL	Variable	State	FALSE	Signal input of muting sensor 3
MutingSwitch22	BOOL	Variable	State	FALSE	Signal input of muting sensor 4
S_MutingLamp	SAFEBOOL	Variable/ Constant	State	FALSE	Feedback signal of the muting lamp
DiscTime11_12	TIME	Constant	State	#0ms	Specification for the maximum discrepancy time for muting sensor 1 and muting sensor 2
DiscTime21_22	TIME	Constant	State	#0ms	Specification for the maximum discrepancy time for muting sensor 3 and muting sensor 4
MaxMutingTime	TIME	Constant	State	#0ms	Specification for the maximum time for the complete muting process
MutingEnable	BOOL	Variable/ Constant	State	FALSE	Startup specification for the muting process
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 651: "SF_MutingPar": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_AOPD_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
S_MutingActive	SAFEBOOL	Variable	State	FALSE	State of the muting process
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 652: "SF_MutingPar": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 653: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.13.2 Function

Function block "SF_MutingPar" is used to support a parallel muting function with 4 muting sensors ("MS_11", "MS_12" and "MS_21", "MS_22") in an application.

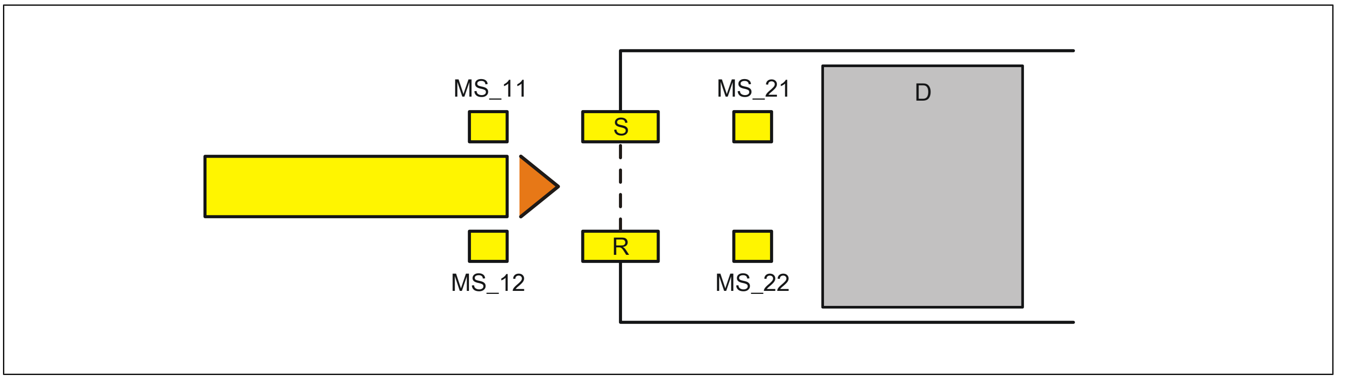


Figure 460: "SF_MutingPar": Parallel muting with 4 muting sensors

Key:

MS_11, MS_12, MS_21, MS_22	Muting sensors
S, R	Transmitter (send) and receiver (receive) of the light curtain
D	Danger zone (danger)

6.6.13.2.1 Connecting conditions

The signals of muting sensors "MS_11" and "MS_12" are connected to input parameters "MutingSwitch11" and "MutingSwitch12" in a muting application. The signals of muting sensors "MS_21" and "MS_22" are connected to input parameters "MutingSwitch21" and "MutingSwitch22". The signal from the protective equipment (light curtain) is connected to input parameter "S_AOPD_In".

Function block signal "S_AOPD_Out" is the release signal for the entire process. To process the release or request of the safe state in the safety application, the signal from "S_AOPD_Out" must be logically connected in the safety application so that FALSE on "S_AOPD_Out" results in a cutoff of the application in the safeguarded danger zone.

6.6.13.2.2 Muting process

The entire muting process is divided up into different muting sequences. The following describes the individual muting sequences.

Note that the following only describes the direction of material flow from muting sensors "MS_11"/"MS_12" to muting sensors "MS_21"/"MS_22". The function block also supports the reverse direction of material flow from muting sensors "MS_21"/"MS_22" to muting sensors "MS_11"/"MS_12". The functional sequence is identical.

Safeguarding the danger zone (muting is inactive, the protective equipment is active)

If the function block does not detect the active muting process on its inputs "MutingSwitch11" and "MutingSwitch12", then signal FALSE on "S_AOPD_In" (light curtain) results in a safe state of output parameter "S_AOPD_Out" (FALSE).

Enabling the muting process (the protective equipment is disabled)

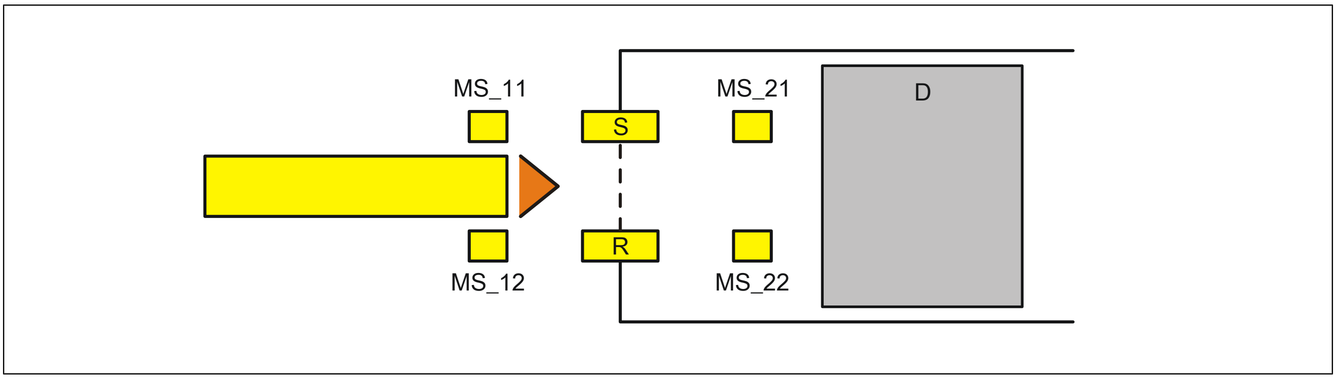


Figure 461: "SF_MutingPar": Parallel muting with 4 muting sensors - Muting sequence 1

The muting process is enabled if a state change from FALSE to TRUE takes place on "MutingSwitch11" and "MutingSwitch12" within the specified time frame set on "DiscTime11_12".

Active muting process (the protective equipment is inactive)

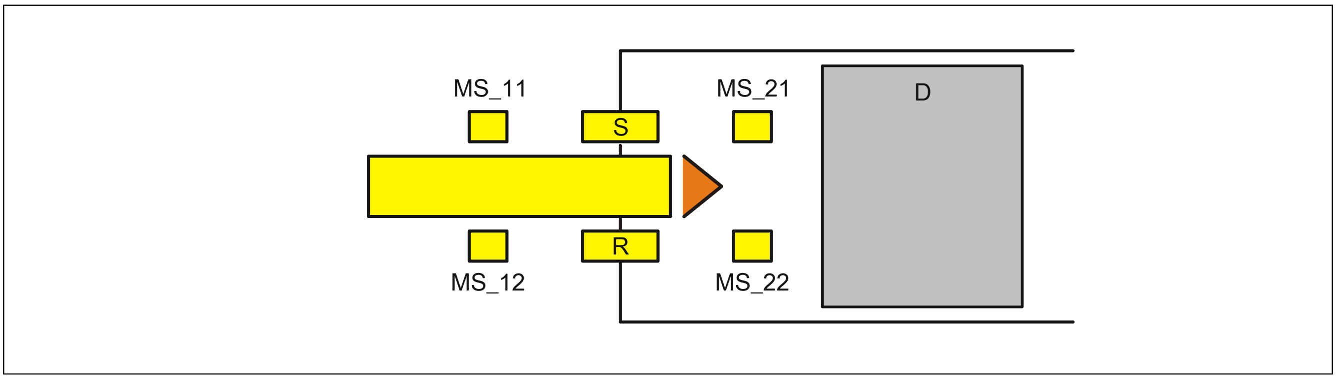


Figure 462: "SF_MutingPar": Parallel muting with 4 muting sensors - Muting sequence 2

While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out".

The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE).

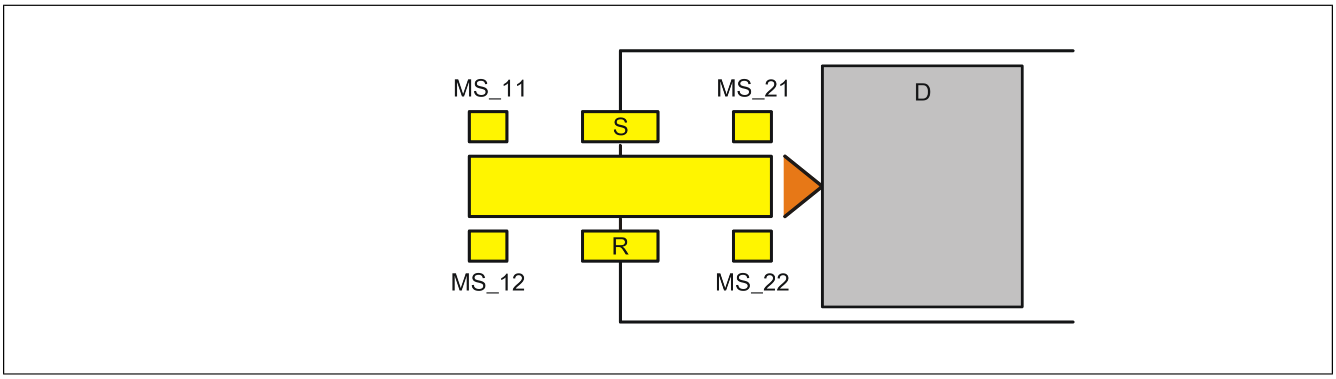


Figure 463: "SF_MutingPar": Parallel muting with 4 muting sensors - Muting sequence 3

"MutingSwitch11" and "MutingSwitch12" must be TRUE if a state change from FALSE to TRUE takes place on "MutingSwitch21" and "MutingSwitch22" within the specified time frame set on "DiscTime21_22". A state change from TRUE to FALSE on "MutingSwitch11" and "MutingSwitch12" is only permitted afterwards. Signal FALSE on "S_AOPD_In" does not result in the safe state of output parameter "S_AOPD_Out" in this phase of the muting sequence.

Completing the muting process (the protective equipment is enabled again)

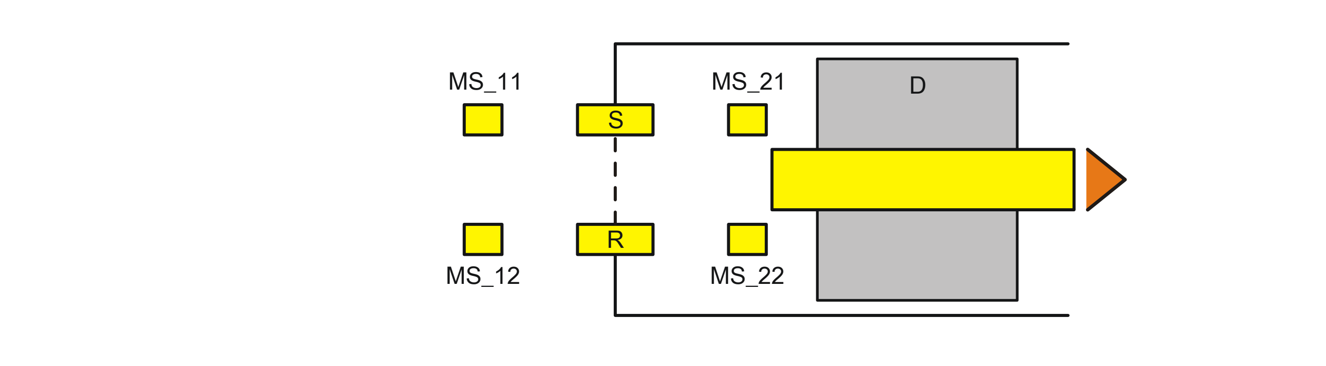


Figure 464: "SF_MutingPar": Parallel muting with 4 muting sensors - Muting sequence 4

The muting process is disabled if "MutingSwitch21" and/or "MutingSwitch22" is set from TRUE to FALSE.

Signal FALSE on "S_AOPD_In" then sets enable output "S_AOPD_Out" to FALSE.

Invalid muting sequences

Invalid states for the muting process on input parameters "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" result in enable output "S_AOPD_Out" taking on the safe state (FALSE) and output "S_MutingActive" switching to FALSE.

6.6.13.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.13.2.4 Start interlock

A start interlock is active after the safety function is no longer being requested and/or after error messages (e.g. invalid muting sequence detected).

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

6.6.13.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.13.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.13.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.13.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

6.6.13.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.13.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.13.4 Input parameters

6.6.13.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.13.4.2 S_AOPD_In

General function

- Signal input of protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to the protective equipment (e.g. safety light curtain) in the muting application over 1 or 2 channels. Input parameter "S_AOPD_In" is then controlled via this signal.

Function description

The function block evaluates the state of the connected protective equipment via the signal connected on input parameter "S_AOPD_In".

Regardless whether the protective equipment is connected to the safe device over 1 or 2 channels, "S_AOPD_In" is only connected to one signal.

If protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected protective equipment is not damped.

FALSE

The connected protective equipment is damped.

If muting is active, "S_AOPD_Out" does not initiate the safe state.
If muting is inactive, "S_AOPD_Out" initiates the safe state.

6.6.13.4.3 MutingSwitch11

General function

- Signal input of muting sensor 1
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_11")

Data type

- BOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal of a device that is connected to muting sensor "MS_11" from the muting application. Input parameter "MutingSwitch11" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch11". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch11" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch11". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch11". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.13.4.4 MutingSwitch12

General function

- Signal input of muting sensor 2
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_12")

Data type

- BOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal of a device that is connected to muting sensor "MS_12" from the muting application. Input parameter "MutingSwitch12" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch12". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch12" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch12". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch12". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.13.4.5 MutingSwitch21

General function

- Signal input of muting sensor 3
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_21")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_21" from the muting application. Input parameter "MutingSwitch21" is then controlled via this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch21". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch21" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch21". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch21". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.13.4.6 MutingSwitch22

General function

- Signal input of muting sensor 4
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_22")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_22" from the muting application. Input parameter "MutingSwitch22" is then controlled via this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch22". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch22" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch22". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch22". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.13.4.7 S_MutingLamp

General function

- Feedback signal of the muting lamp

Data type

- SAFEBOOL

Connection

- Variable or constant

> **Information:**
>
> **Connect this input parameter to the signal of a safe device that is connected to the feedback signal of the muting lamp in the muting application over 1 or 2 channels. Input parameter "S_MutingLamp" is then controlled using this signal.**

> **Information:**
>
> **If your risk analysis determines that a muting lamp is not required in your muting application, note that you can specify constant TRUE on this input parameter.**

Function description

The function block evaluates the state of the connected muting lamp (lamp operational / not operational) using the signal connected on input parameter "S_MutingLamp".

Note that the feedback signal of the muting lamp must permanently indicate state TRUE if its function is not impaired. If the lamp function is impaired, the feedback signal must permanently indicate state FALSE.

Regardless of whether the muting lamp is connected to the safe device over 1 or 2 channels, "S_MutingLamp" is only connected to one signal.

If the muting lamp is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The function of the muting lamp is not impaired.

FALSE

The function of the muting lamp is impaired.

6.6.13.4.8 DiscTime11_12

General function

- Specification for the maximum discrepancy time for muting sensor 1 and muting sensor 2

Data type

- TIME

Connection

- Constant

Function description

The function block starts the muting process if "MutingSwitch11" and "MutingSwitch12" switch from FALSE to TRUE within the specified time on "DiscTime11_12". If the second required signal change on "MutingSwitch11" and "MutingSwitch12" does not take place during this time frame, enable output "S_AOPD_Out" initiates the safe state (FALSE).

You must define and validate the time value for input parameter "DiscTime11_12" based on your application and risk analysis.

Range of values: 0 to 4 seconds

6.6.13.4.9 DiscTime21_22

General function

- Specification for the maximum discrepancy time for muting sensor 3 and muting sensor 4

Data type

- TIME

Connection

- Constant

Function description

The function block starts the muting process if "MutingSwitch21" and "MutingSwitch22" switch from FALSE to TRUE within the specified time on "DiscTime21_22". If the second required signal change on "MutingSwitch21" and "MutingSwitch22" does not take place during this time frame, enable output "S_AOPD_Out" initiates the safe state (FALSE).

You must define and validate the time value for input parameter "DiscTime21_22" based on your application and risk analysis.

Range of values: 0 to 4 seconds

6.6.13.4.10 MaxMutingTime

General function

- Specification for the maximum time for the complete muting process

Data type

- TIME

Connection

- Constant

Function description

The maximum time for the complete muting process is specified using this input parameter. This time starts when the signals on "MutingSwitch11" and/or "MutingSwitch12" change from FALSE to TRUE. The muting process is completed when "MutingSwitch21" and "MutingSwitch22" have afterwards changed from TRUE to FALSE.

You must define and validate the time value for input parameter "MaxMutingTime" based on your application and risk analysis.

Range of values: 0 to 10 minutes

6.6.13.4.11 MutingEnable

General function

- Startup specification for the muting process

Data type

- BOOL

Connection

- Variable or constant

Information:

Control this input parameter with a signal from the standard application that releases the muting process. If the result of your risk analysis is that you do not need a release signal from the standard application, you can alternatively assign constant TRUE.

Function description

Input parameter "MutingEnable" receives the starting signal from the standard application to enable the muting process. This is a measure for reducing the risk of an unintended muting process.

TRUE

Starting the muting function is possible.

FALSE

Starting the muting function is not possible.

6.6.13.4.12 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.13.4.13 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.13.5 Output parameters

6.6.13.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.13.5.2 S_AOPD_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this output parameter to the safety application in such a way that the safety application takes on and maintains the safe state when signal FALSE is output.

Function description

The release signal is the safe enable signal for the safeguarded area to control an output on a safe device and therefore the process. This output parameter represents the state of the protective device in the muting application. The release signal is controlled based on the state of the protective device and start interlock.

In addition, the release signal controls the request for the stop function. Control the stop function of the connected safety application by appropriately connecting "S_AOPD_Out".

Since the release signal is present on output "S_AOPD_Out", this output is referred to as the "enable output".

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = TRUE (light curtain)
- or: The muting process is active and the function block did not detect an invalid muting sequence.
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = FALSE (light curtain)
- or: The muting process is active and the function block detected an invalid muting sequence.
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_AOPD_Out" is set to TRUE only if input "S_AOPD_In" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_AOPD_Out" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 654: "SF_MutingPar": Input parameter "S_StartReset"

6.6.13.5.3 S_MutingActive

General function

- State of the muting process

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **If the result of your risk analysis is that you must use a muting lamp in the muting application, connect this output parameter to a safe output device that is connected to the muting lamp.**
>
> **In addition, connect this output parameter to the safety application in such a way that the safety application is controlled according to the muting state.**

Function description

This output parameter indicates whether a muting process is enabled and being executed or not enabled at all.

TRUE

The function block has been enabled ("Activate" = TRUE).

The muting process is enabled ("MutingEnable" = TRUE) and being executed. "S_AOPD_In" = FALSE does not result in the safe state on "S_AOPD_Out" (FALSE).

FALSE

The muting process is not enabled ("MutingEnable" = FALSE). "S_AOPD_In" = FALSE results in the safe state on "S_AOPD_Out" (FALSE).

6.6.13.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.13.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.13.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The muting process is not active. A safety request is not active. The function block assesses signal FALSE on "S_AOPD_In" as a safety function request. Normal operation is possible in the area safeguarded by the protective equipment.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	The function block has been enabled. The function block's start interlock is active.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The function block is in the safe state. The function block detected a safety request ("S_AOPD_In" = FALSE) while a muting process was not active. The start interlock is active after a safety request on the function block. To revoke the safety request, set "S_AOPD_In" to TRUE in your application.	To end the function block's active start interlock, perform a reset on the function block after you have revoked the safety request on "S_AOPD_In".
8005	The function block is in the safe state. This diagnostic message is only displayed for one cycle of the safety controller. Depending on the state on "S_AOPD_In" and "S_MutingLamp", the diagnostic message switches from WORD#16#8005 to WORD#16#8002, WORD#16#8000 or WORD#16#C003.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8011	Conveying direction 1: The muting process was requested on "MutingSwitch11". State FALSE exists on "MutingSwitch12". Time monitoring until damping of the second muting sensor ("DiscTime11_12") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. To exit this sequence, signal TRUE is required on "MutingSwitch12". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8012	Conveying direction 1: The muting process is active. State TRUE exists on "MutingSwitch11" and "MutingSwitch12". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, signal TRUE is required on "MutingSwitch21" and/or "MutingSwitch22".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8014	Conveying direction 1: The muting process is active. State TRUE exists on "MutingSwitch11", "MutingSwitch12" and "MutingSwitch21". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. Time monitoring until damping of the second muting sensor ("DiscTime21_22") is started. To exit this sequence, signal TRUE is required on "MutingSwitch22".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8021	Conveying direction 1: The muting process is active. State TRUE exists on "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, state FALSE is required on "MutingSwitch11" and "MutingSwitch12". State FALSE is then required on "MutingSwitch21" and/or "MutingSwitch22".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
8112	Conveying direction 2: The muting process is active. State TRUE exists on "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, state FALSE is required on "MutingSwitch21" and "MutingSwitch22". State FALSE is then required on "MutingSwitch11" and/or "MutingSwitch12".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8114	Conveying direction 2: The muting process is active. State TRUE exists on "MutingSwitch21", "MutingSwitch22" and "MutingSwitch11". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. Time monitoring until damping of the second muting sensor ("DiscTime11_12") is started. To exit this sequence, signal TRUE is required on "MutingSwitch12".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8121	Conveying direction 2: The muting process is active. State TRUE exists on "MutingSwitch21" and "MutingSwitch22". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, signal TRUE is required on "MutingSwitch11" and/or "MutingSwitch12".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8122	Conveying direction 2: The muting process was requested on "MutingSwitch21". State FALSE exists on "MutingSwitch22". Time monitoring until damping of the second muting sensor ("DiscTime21_22") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. To exit this sequence, signal TRUE is required on "MutingSwitch22". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8311	Conveying direction 1: The muting process was requested on "MutingSwitch12". State FALSE exists on "MutingSwitch11". Time monitoring until damping of the second muting sensor ("DiscTime11_12") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. To exit this sequence, signal TRUE is required on "MutingSwitch11". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8314	Conveying direction 1: The muting process is active. State TRUE exists on "MutingSwitch11", "MutingSwitch12" and "MutingSwitch22". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. Time monitoring until damping of the second muting sensor ("DiscTime21_22") is started. To exit this sequence, signal TRUE is required on "MutingSwitch21".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8414	Conveying direction 2: The muting process is active. State TRUE exists on "MutingSwitch21", "MutingSwitch22" and "MutingSwitch12". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. Time monitoring until damping of the second muting sensor ("DiscTime11_12") is started. To exit this sequence, signal TRUE is required on "MutingSwitch11".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
8422	Conveying direction 2: The muting process was requested on "MutingSwitch22". State FALSE exists on "MutingSwitch21". Time monitoring until damping of the second muting sensor ("DiscTime21_22") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. To exit this sequence, signal TRUE is required on "MutingSwitch21". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C003	The function block detected an impermissible FALSE signal on "S_MutingLamp".	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signal connected to "S_MutingLamp" must indicate state TRUE. • Reset the function block.
C005	The function block detected an invalid value on "DiscTime11_12" and/or "DiscTime21_22" and/or "MaxMutingTime".	• Check the values connected to "DiscTime11_12", "DiscTime21_22" and "MaxMutingTime". • Correct the faulty value(s). • Values must correspond to the results of your risk analysis. • Input parameter limit values must be taken into account when performing your risk analysis. • Compile the project in your safe programming environment and transfer the project to the safety controller you are using. • Perform a cold restart of the safety controller you are using. • Reset the function block.
C006	The muting process was active. The muting process was not yet completed when time monitoring of the maximum muting time expired.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "MaxMutingTime". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" indicate state FALSE.
C007	The muting process was requested on "MutingSwitch11" or "MutingSwitch12". Only one muting sensor indicated state TRUE from the expiration of time monitoring until damping of the second muting sensor ("DiscTime11_12").	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "DiscTime11_12". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" indicate state FALSE.
C008	The muting process was requested on "MutingSwitch21" or "MutingSwitch22". Only one muting sensor indicated state TRUE from the expiration of time monitoring until damping of the second muting sensor ("DiscTime21_22").	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "DiscTime21_22". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" indicate state FALSE.
C010	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C020	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C030	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C050	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C060	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004 to C0F4	Conveying direction 1 and 2: The function block detected an impermissible muting sequence when requesting the muting process. In the last valid muting sequence, TRUE signals were permitted for muting sensors only for one conveying direction. TRUE signals on muting sensors for both conveying directions are therefore impermissible. Static TRUE signals on "MutingSwitch11" to "MutingSwitch22" are impermissible in this sequence.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C004	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C014	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C024	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C034	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C044	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C054	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C064	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C074	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C084	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C094	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C0A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C0B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C0C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C104 to C1F4	Conveying direction 1: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch12" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch11" was impermissible. In addition, TRUE signals on "MutingSwitch21" and/or "MutingSwitch22" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C104	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C114	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C124	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C134	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C144	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C154	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C164	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C174	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C184	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C194	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C1D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C1E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C1F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C204 to C2F4	Conveying direction 1: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch12" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch11" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch12" was impermissible. In addition, TRUE signals on "MutingSwitch21" and/or "MutingSwitch22" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C204	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C214	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C224	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C234	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C244	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C254	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C264	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C274	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C284	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C294	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C2D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C2E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C2F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C304 to C3F4	Conveying direction 1: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" and "MutingSwitch12" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch21" and/or "MutingSwitch22" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch11" and/or "MutingSwitch12" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C304	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C314	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C324	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C334	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C344	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C354	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C364	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C374	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C384	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C394	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C404 to C4F4	Conveying direction 1: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" and "MutingSwitch12" was requested. In the last valid muting sequence, "MutingSwitch21" indicated state TRUE and "MutingSwitch22" indicated state FALSE. The next valid muting sequence would have been reached with "MutingSwitch22" = TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch11" and/or "MutingSwitch12" and/or "MutingSwitch21" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C404	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C414	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C424	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C434	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C444	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C454	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C464	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C474	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C484	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C494	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C4A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C4B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C4C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C504 to C5F4	Conveying direction 1: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" and "MutingSwitch12" was requested. In the last valid muting sequence, "MutingSwitch11", "MutingSwitch12" and "MutingSwitch22" indicated state TRUE. "MutingSwitch21" indicated state FALSE. The next valid muting sequence would have been reached with "MutingSwitch21" = TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch11" and/or "MutingSwitch12" and/or "MutingSwitch22" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C504	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C514	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C524	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C534	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C544	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C554	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C564	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C574	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C584	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C594	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C5A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C5B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C5C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C5D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C5E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C5F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C604 to C6F4	Conveying direction 1: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" and "MutingSwitch12" was requested. To reach the last valid muting sequence, "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" indicated state TRUE. In the last valid muting sequence, a signal change from FALSE to TRUE on one or more of these 4 input parameters is impermissible. To exit this muting sequence in a valid way, 3 of the 4 signals in the conveying direction would have needed to change from TRUE to FALSE. In addition, reversing the conveying direction is impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C604	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C614	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C624	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C634	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C644	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C654	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C664	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C674	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C684	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C694	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C6A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C6B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C6C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C6D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C6E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C6F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C704 to C7F4	Conveying direction 2: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch21" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch22" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch21" was impermissible. In addition, TRUE signals on "MutingSwitch11" and/or "MutingSwitch12" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C704	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C714	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C724	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C734	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C744	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C754	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C764	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C774	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C784	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C794	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C7A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C7B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C7C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C7D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C7E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C7F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C804 to C8F4	Conveying direction 2: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch22" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch21" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch22" was impermissible. In addition, TRUE signals on "MutingSwitch11" and/or "MutingSwitch12" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C804	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C814	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C824	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C834	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C844	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C854	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C864	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C874	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C884	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C894	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C8A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C8B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C8C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C8D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C8E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C8F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C904 to C9F4	Conveying direction 2: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch21" and "MutingSwitch22" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch11" and/or "MutingSwitch12" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch21" and/or "MutingSwitch22" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C904	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C914	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C924	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C934	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
C944	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C954	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C964	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C974	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C984	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C994	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C9A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C9B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C9C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C9D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C9E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C9F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CA04 to CAF4	Conveying direction 2: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch21" and "MutingSwitch22" was requested. In the last valid muting sequence, "MutingSwitch11" indicated state TRUE and "MutingSwitch12" indicated state FALSE. The next valid muting sequence would have been reached with "MutingSwitch12" = TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch21" and/or "MutingSwitch22" and/or "MutingSwitch11" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
CA04	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CA14	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CA24	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CA34	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CA44	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CA54	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CA64	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CA74	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CA84	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CA94	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CAA4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CAB4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CAC4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CAD4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CAE4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CAF4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CB04 to CBF4	Conveying direction 2: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch21" and "MutingSwitch22" was requested. In the last valid muting sequence, "MutingSwitch21", "MutingSwitch22" and "MutingSwitch12" indicated state TRUE. "MutingSwitch11" indicated state FALSE. The next valid muting sequence would have been reached with "MutingSwitch11" = TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch21" and/or "MutingSwitch22" and/or "MutingSwitch12" was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
CB04	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CB14	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CB24	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CB34	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CB44	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CB54	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CB64	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CB74	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CB84	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CB94	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CBA4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CBB4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CBC4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CBD4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CBE4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CBF4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CC04 to CCF4	Conveying direction 2: During an active muting process, the function block detected an impermissible muting sequence after the muting process on "MutingSwitch21" and "MutingSwitch22" was requested. To reach the last valid muting sequence, "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" indicated state TRUE. In the last valid muting sequence, a signal change from FALSE to TRUE on one or more of these 4 input parameters is impermissible. To exit this muting sequence in a valid way, 3 of the 4 signals in the conveying direction would have needed to change from TRUE to FALSE. In addition, reversing the conveying direction is impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
CC04	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CC14	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CC24	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CC34	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CC44	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CC54	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CC64	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CC74	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CC84	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CC94	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CCA4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CCB4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CCC4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CCD4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CCE4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CCF4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CF04 to CFF4	Conveying direction 1 and 2: The function block detected an impermissible FALSE signal on "MutingEnable" when requesting the muting process or during the active muting process.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
CF04	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF14	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF24	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF34	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF44	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF54	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF64	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF74	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF84	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

Code (hex)	Description	Corrective measures
CF94	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFA4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFB4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFC4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFD4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFE4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFF4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 655: "SF_MutingPar": Diagnostic codes

6.6.13.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

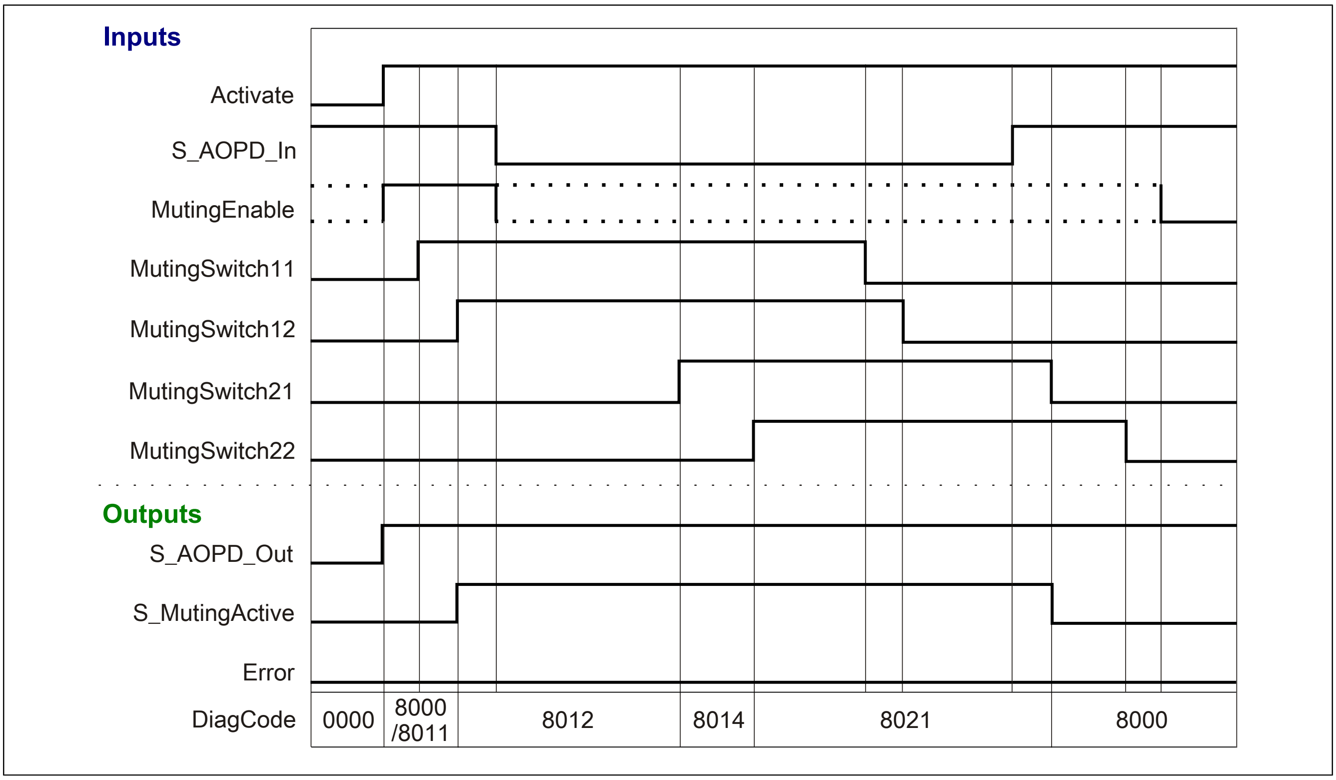


Figure 465: "SF_MutingPar": Signal sequence diagram

6.6.13.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement parallel muting with 4 sensors.

The following example describes function block connections in a muting application when controlling with the signals from 4 muting sensors (parallel muting, see section 6.6.13.7.2 "Parallel muting with 4 sensors").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.13.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "PM4_S15".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22", a rising edge on input parameter "Reset" is required to enable enable output "S_AOPD_Out".

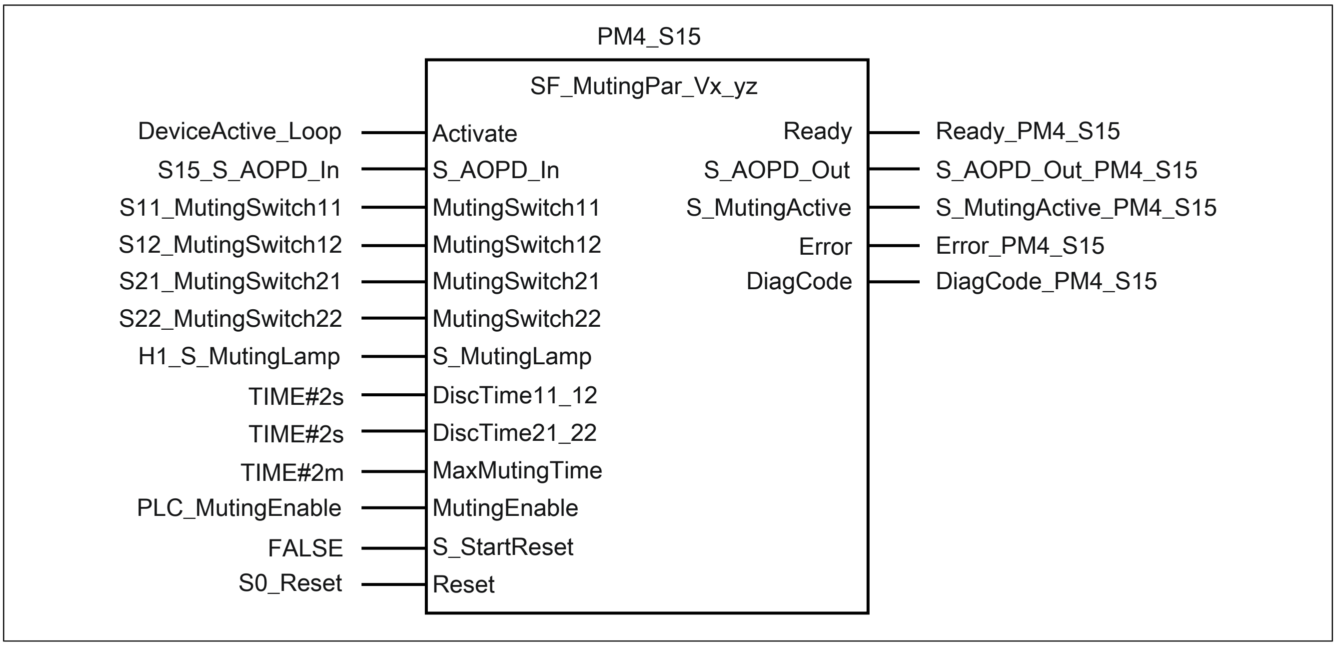


Figure 466: "SF_MutingPar": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_Loop	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the states of the safe and non-safe input and output devices to which the muting sensors, light curtain and muting lamp involved in the safety function are connected.
S15_S_AOPD_In	SAFEBOOL	In this example, this signal comes from a 2-channel light curtain.
S11_MutingSwitch11	BOOL	Signal input of muting sensor 1
S12_MutingSwitch12	BOOL	Signal input of muting sensor 2
S21_MutingSwitch21	BOOL	Signal input of muting sensor 3
S22_MutingSwitch22	BOOL	Signal input of muting sensor 4
H1_S_MutingLamp	SAFEBOOL	Feedback signal of the muting lamp
TIME#2s on "DiscTime11_12"	TIME	Specification of the maximum discrepancy time for muting sensors 1 and 2
TIME#2s on "DiscTime21_22"	TIME	Specification of the maximum discrepancy time for muting sensors 3 and 4
TIME#2m on "MaxMutingTime"	TIME	Specification for the maximum time for the complete muting process
PLC_MutingEnable	BOOL	Startup specification for the muting process
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 656: "SF_MutingPar": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_PM4_S15	BOOL	Function block active / Function block inactive. Message for further external processing.
S_AOPD_Out_PM4_S15	SAFEBOOL	Release signal. This output is used for further processing in the program on the safety controller.
S_MutingActive_PM4_S15	SAFEBOOL	Muting active. This output is output via a safe output device. This output is used to control the muting lamp.
Error_PM4_S15	BOOL	Error message from function block for further external processing
DiagCode_PM4_S15	WORD	Diagnostic message from function block for further external processing

Table 657: "SF_MutingPar": Outputs connected to output parameters

6.6.13.7.2 Parallel muting with 4 sensors

This example illustrates function block connections in a muting application when controlling with the signals from 4 muting sensors. Sensors are arranged in parallel in groups of 2 (see chapter 6.6.13.2 "Function").

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.13.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "PM4_S15".

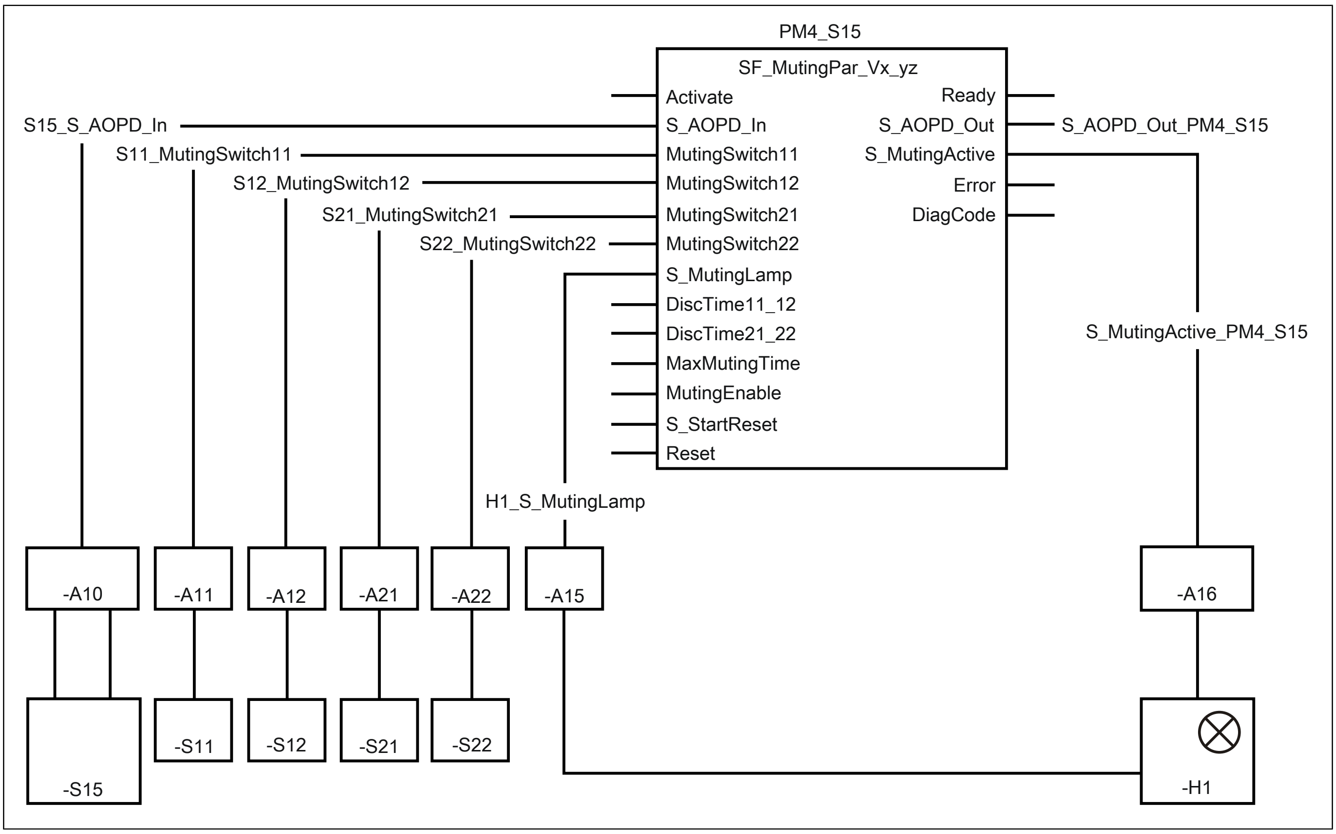


Figure 467: "SF_MutingPar": Parallel muting with 4 sensors

List of equipment

-A10	2-channel input of a safe input device
-A11, -A12, -A21, -A22	1-channel inputs of the input devices
-A15	1-channel input of a safe input device
-A16	Outputs of a safe output device
-S15	Light curtain, 2-channel
-S11, -S12	Muting sensors arranged in front of the light curtain, see Fig. 460
-S21, -S22	Muting sensors arranged behind the light curtain, see Fig. 460
-H1	Muting lamp monitored by logic

Connected inputs and outputs

S15_S_AOPD_In	Input on "S_AOPD_In"
S11_MutingSwitch11	Input on "MutingSwitch11"
S12_MutingSwitch12	Input on "MutingSwitch12"
S21_MutingSwitch21	Input on "MutingSwitch21"
S22_MutingSwitch22	Input on "MutingSwitch22"
H1_S_MutingLamp	Input on "S_MutingLamp"
S_AOPD_Out_PM4_S15	Output on "S_AOPD_Out"
S_MutingActive_PM4_S15	Output on "S_MutingActive"

Description

In this example:

- Protective equipment "-S15" (light curtain) is connected to an input of safe device "-A10" via terminals over 2 channels.
- The resulting signal of the 2-channel protective equipment "-S15" from an input of safe device "-A10" is connected to input "S15_S_AOPD_In". This input is connected to function block input parameter "S_AOPD_In" for further processing.
- Muting sensor "-S11" is connected to an input of device "-A11" via terminals over 1 channel.
- The resulting signal of muting sensor "-S11" from an input of device "-A11" is connected to input "S11_MutingSwitch11". This input is connected to function block input parameter "MutingSwitch11" for further processing.
- Muting sensor "-S12" is connected to an input of device "-A12" via terminals over 1 channel.
- The resulting signal of muting sensor "-S12" from an input of device "-A12" is connected to input "S12_MutingSwitch12". This input is connected to function block input parameter "MutingSwitch12" for further processing.
- Muting sensor "-S21" is connected to an input of device "-A21" via terminals over 1 channel.
- The resulting signal of muting sensor "-S21" from an input of device "-A21" is connected to input "S21_MutingSwitch21". This input is connected to function block input parameter "MutingSwitch21" for further processing.
- Muting sensor "-S22" is connected to an input of device "-A22" via terminals over 1 channel.
- The resulting signal of muting sensor "-S22" from an input of device "-A22" is connected to input "S22_MutingSwitch22". This input is connected to function block input parameter "MutingSwitch22" for further processing.
- Feedback signal of muting lamp "-H1" is connected to an input of safe device "-A15" via terminals over 1 channel.
- The resulting signal of muting lamp "-H1" from an input of safe device "-A15" is connected to input "H1_S_MutingLamp". This input is connected to function block input parameter "S_MutingLamp" for further processing.
- Output parameter "S_MutingActive" is connected to output "S_MutingActive_PM4_S15".
- Output "S_MutingActive_PM4_S15" controls a 1-channel output of safe device "-A16". The muting lamp is connected to this output of safe device "-A16" via terminals over 1 channel.
- Output "S_AOPD_Out_PM4_S15" connected to output parameter "S_AOPD_Out" controls the stop request to the application.

6.6.13.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 61496-1	Muting appendix	**Muting process:** Note that the following only describes the direction of material flow from muting sensors "MS_11"/"MS_12" to muting sensors "MS_21"/"MS_22". The function block also supports the reverse direction of material flow from muting sensors "MS_21"/"MS_22" to muting sensors "MS_11"/"MS_12". The functional sequence is identical. **Safeguarding the danger zone:** If the function block does not detect the active muting process on its inputs "MutingSwitch11" and "MutingSwitch12", then signal FALSE on "S_AOPD_In" (light curtain) results in a safe state of output parameter "S_AOPD_Out" (FALSE). **Enabling the muting process:** The muting process is enabled if a state change from FALSE to TRUE takes place on "MutingSwitch11" and "MutingSwitch12" within the specified time frame set on "DiscTime11_12". **Active muting process:** While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out". The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE). "MutingSwitch11" and "MutingSwitch12" must be TRUE if a state change from FALSE to TRUE takes place on "MutingSwitch21" and "MutingSwitch22" within the specified time frame set on "DiscTime21_22". A state change from TRUE to FALSE on "MutingSwitch11" and "MutingSwitch12" is only permitted afterwards. Signal FALSE on "S_AOPD_In" does not result in the safe state of output parameter "S_AOPD_Out" in this phase of the muting sequence. **Completing the muting process:** • The muting process is disabled if "MutingSwitch11" and/or "MutingSwitch12" is set from TRUE to FALSE. • The muting process is disabled if "MutingSwitch21" and/or "MutingSwitch22" is set from TRUE to FALSE. Signal FALSE on "S_AOPD_In" then sets enable output "S_AOPD_Out" to FALSE. **Invalid muting sequences:** Invalid states for the muting process on input parameters "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22" result in "S_AOPD_Out" taking on the safe state (FALSE). **Start interlocks:** After the function block is enabled, the function block optionally supports a start interlock depending on the specification on "S_StartReset". A start interlock is active after the function block has detected an invalid muting sequence or other error. **State of the muting process:** The function block outputs the state of the muting process on "S_MutingActive".
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_StartReset" = FALSE) • Enabling the function block ("S_StartReset" = FALSE) The function block supports a start interlock after the following: • Invalid muting sequence • Error detection by the function block • The safety function is no longer being requested If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal and/or the signal on "S_StartReset".
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_AOPD_Out") executes a category 0 stop.

Table 658: "SF_MutingPar": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.14 SF_MutingPar_2Sensor

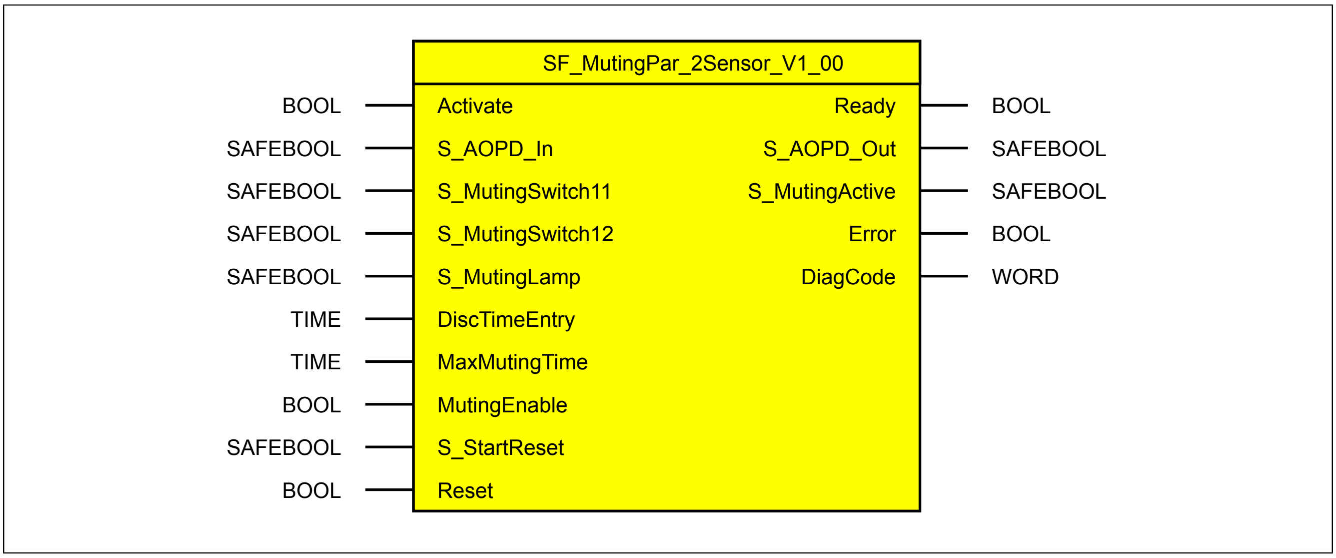


Figure 468: Function block "SF_MutingPar_2Sensor"

6.6.14.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_AOPD_In	SAFEBOOL	Variable	State	FALSE	Signal input of protective equipment (light curtain)
S_MutingSwitch11	SAFEBOOL	Variable	State	FALSE	Signal input of muting sensor 1
S_MutingSwitch12	SAFEBOOL	Variable	State	FALSE	Signal input of muting sensor 2
S_MutingLamp	SAFEBOOL	Variable/ Constant	State	FALSE	Feedback signal of the muting lamp
DiscTimeEntry	TIME	Constant	State	#0ms	Specification for the maximum discrepancy time for muting sensor 1 and muting sensor 2
MaxMutingTime	TIME	Constant	State	#0ms	Specification for the maximum time for the complete muting process
MutingEnable	BOOL	Variable/ Constant	State	FALSE	Startup specification for the muting process
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 659: "SF_MutingPar_2Sensor": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type 1)	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_AOPD_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
S_MutingActive	SAFEBOOL	Variable	State	FALSE	State of the muting process
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 660: "SF_MutingPar_2Sensor": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 661: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.14.2 Function

Function block "SF_MutingPar_2Sensor" is used to support a parallel muting function with 2 muting sensors ("MS_11" and "MS_12") in an application.

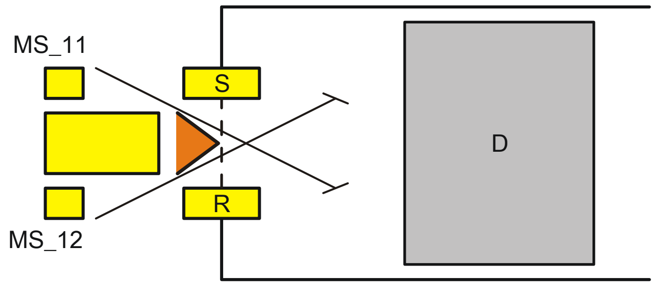


Figure 469: "SF_MutingPar_2Sensor": Parallel muting with 2 muting sensors

Key:

MS_11, MS_12	Muting sensors
S, R	Transmitter (send) and receiver (receive) of the light curtain
D	Danger zone (danger)

6.6.14.2.1 Connecting conditions

The signals of muting sensors "MS_11" and "MS_12" are connected to input parameters "S_MutingSwitch11" and "S_MutingSwitch12" in a muting application. The signal from the protective equipment (light curtain) is connected to input parameter "S_AOPD_In".

Function block signal "S_AOPD_Out" is the release signal for the entire process. To process the release or request of the safe state in the safety application, the signal from "S_AOPD_Out" must be logically connected in the safety application so that FALSE on "S_AOPD_Out" results in a cutoff of the application in the safeguarded danger zone.

6.6.14.2.2 Muting process

The entire muting process is divided up into different muting sequences. The following describes the individual muting sequences.

Safeguarding the danger zone (muting is inactive, the protective equipment is active)

If the function block does not detect the active muting process on its inputs "S_MutingSwitch11" and "S_MutingSwitch12", then signal FALSE on "S_AOPD_In" (light curtain) results in a safe state of output parameter "S_AOPD_Out" (FALSE).

Enabling the muting process (the protective equipment is disabled)

The muting process is enabled if a state change from FALSE to TRUE takes place on "S_MutingSwitch11" and "S_MutingSwitch12" within the specified time frame set on "DiscTimeEntry".

Active muting process (the protective equipment is inactive)

While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out".

The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE).

Completing the muting process (the protective equipment is enabled again)

The muting process is disabled if "S_MutingSwitch11" and/or "S_MutingSwitch12" is set from TRUE to FALSE.

Signal FALSE on "S_AOPD_In" then sets enable output "S_AOPD_Out" to FALSE.

Invalid muting sequences

Invalid states for the muting process on input parameters "S_MutingSwitch11" and/or "S_MutingSwitch12" result in enable output "S_AOPD_Out" taking on the safe state (FALSE) and output "S_MutingActive" switching to FALSE.

6.6.14.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.14.2.4 Start interlock

A start interlock is active after the safety function is no longer being requested and/or after error messages (e.g. invalid muting sequence detected).

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

6.6.14.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.14.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.14.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.14.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

6.6.14.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.14.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.14.4 Input parameters

6.6.14.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.14.4.2 S_AOPD_In

General function

- Signal input of protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the protective equipment (e.g. safety light curtain) in the muting application over 1 or 2 channels. Input parameter "S_AOPD_In" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected protective equipment via the signal connected on input parameter "S_AOPD_In".

Regardless whether the protective equipment is connected to the safe device over 1 or 2 channels, "S_AOPD_In" is only connected to one signal.

If protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected protective equipment is not damped.

FALSE

The connected protective equipment is damped.

If muting is active, "S_AOPD_Out" does not initiate the safe state.
If muting is inactive, "S_AOPD_Out" initiates the safe state.

6.6.14.4.3 S_MutingSwitch11

General function

- Signal input of muting sensor 1
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_11")

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal of a safe device that is connected to muting sensor "MS_11" from the muting application. Input parameter "S_MutingSwitch11" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "S_MutingSwitch11". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "S_MutingSwitch11" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingSwitch11". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingSwitch11". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.14.4.4 S_MutingSwitch12

General function

- Signal input of muting sensor 2
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_12")

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **Connect this input parameter to the signal of a safe device that is connected to muting sensor "MS_12" from the muting application. Input parameter "S_MutingSwitch12" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "S_MutingSwitch12". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "S_MutingSwitch12" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingSwitch12". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingSwitch12". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.14.4.5 S_MutingLamp

General function

- Feedback signal of the muting lamp

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

Connect this input parameter to the signal of a safe device that is connected to the feedback signal of the muting lamp in the muting application over 1 or 2 channels. Input parameter "S_MutingLamp" is then controlled using this signal.

Information:

If your risk analysis determines that a muting lamp is not required in your muting application, note that you can specify constant TRUE on this input parameter.

Function description

The function block evaluates the state of the connected muting lamp (lamp operational / not operational) using the signal connected on input parameter "S_MutingLamp".

Note that the feedback signal of the muting lamp must permanently indicate state TRUE if its function is not impaired. If the lamp function is impaired, the feedback signal must permanently indicate state FALSE.

Regardless of whether the muting lamp is connected to the safe device over 1 or 2 channels, "S_MutingLamp" is only connected to one signal.

If the muting lamp is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The function of the muting lamp is not impaired.

FALSE

The function of the muting lamp is impaired.

6.6.14.4.6 DiscTimeEntry

General function

- Specification for the maximum discrepancy time for muting sensor 1 and muting sensor 2

Data type

- TIME

Connection

- Constant

Function description

The function block starts the muting process if "S_MutingSwitch11" and "S_MutingSwitch12" switch from FALSE to TRUE within the specified time on "DiscTimeEntry". If the second required signal change on "S_MutingSwitch11" and "S_MutingSwitch12" does not take place during this time frame, enable output "S_AOPD_Out" initiates the safe state (FALSE).

You must define and validate the time value for input parameter "DiscTimeEntry" based on your application and risk analysis.

Range of values: 0 to 4 seconds

6.6.14.4.7 MaxMutingTime

General function

- Specification for the maximum time for the complete muting process

Data type

- TIME

Connection

- Constant

Function description

The maximum time for the complete muting process is specified using this input parameter. This time starts when the signals on "S_MutingSwitch11" and/or "S_MutingSwitch12" change from FALSE to TRUE. The muting process is completed when "S_MutingSwitch11" and/or "S_MutingSwitch12" again indicate FALSE.

You must define and validate the time value for input parameter "MaxMutingTime" based on your application and risk analysis.

Range of values: 0 to 10 minutes

6.6.14.4.8 MutingEnable

General function

- Startup specification for the muting process

Data type

- BOOL

Connection

- Variable or constant

> ## Information:
>
> **Control this input parameter with a signal from the standard application that releases the muting process. If the result of your risk analysis is that you do not need a release signal from the standard application, you can alternatively assign constant TRUE.**

Function description

Input parameter "MutingEnable" receives the starting signal from the standard application to enable the muting process. This is a measure for reducing the risk of an unintended muting process.

TRUE

Starting the muting function is possible.

FALSE

Starting the muting function is not possible.

6.6.14.4.9 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.14.4.10 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.14.5 Output parameters

6.6.14.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.14.5.2 S_AOPD_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this output parameter to the safety application in such a way that the safety application takes on and maintains the safe state when signal FALSE is output.

Function description

The release signal is the safe enable signal for the safeguarded area to control an output on a safe device and therefore the process. This output parameter represents the state of the protective device in the muting application. The release signal is controlled based on the state of the protective device and start interlock.

In addition, the release signal controls the request for the stop function. Control the stop function of the connected safety application by appropriately connecting "S_AOPD_Out".

Since the release signal is present on output "S_AOPD_Out", this output is referred to as the "enable output".

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = TRUE (light curtain)
- or: The muting process is active and the function block did not detect an invalid muting sequence.
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = FALSE (light curtain)
- or: The muting process is active and the function block detected an invalid muting sequence.
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_AOPD_Out" is set to TRUE only if input "S_AOPD_In" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

<table>
<tr><th colspan="2">Input parameters</th><th>Action</th><th>Start interlock</th><th>Reset</th><th>Enable output</th></tr>
<tr><td rowspan="2">S_StartReset</td><td>TRUE</td><td rowspan="2">After the function block is enabled / cold restart of the safety controller, the start interlock is...</td><td>...inactive.</td><td>No action on "Reset" is required...</td><td rowspan="2">• ...to set enable output "S_AOPD_Out" to TRUE on a valid input signal combination.
• ...to end the start interlock.</td></tr>
<tr><td>FALSE</td><td>...active.</td><td>"Reset" must be set from FALSE to TRUE...</td></tr>
</table>

Table 662: "SF_MutingPar_2Sensor": Input parameter "S_StartReset"

6.6.14.5.3 S_MutingActive

General function

- State of the muting process

Data type

- SAFEBOOL

Connection

- Variable

Information:

If the result of your risk analysis is that you must use a muting lamp in the muting application, connect this output parameter to a safe output device that is connected to the muting lamp.

In addition, connect this output parameter to the safety application in such a way that the safety application is controlled according to the muting state.

Function description

This output parameter indicates whether a muting process is enabled and being executed or not enabled at all.

TRUE

The function block has been enabled ("Activate" = TRUE).

The muting process is enabled ("MutingEnable" = TRUE) and being executed. "S_AOPD_In" = FALSE does not result in the safe state on "S_AOPD_Out" (FALSE).

FALSE

The muting process is not enabled ("MutingEnable" = FALSE). "S_AOPD_In" = FALSE results in the safe state on "S_AOPD_Out" (FALSE).

6.6.14.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.14.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.14.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The muting process is not active. A safety request is not active. The function block assesses signal FALSE on "S_AOPD_In" as a safety function request. Normal operation is possible in the area safeguarded by the protective equipment.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	The function block has been enabled. The function block's start interlock is active.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The function block is in the safe state. The function block detected a safety request ("S_AOPD_In" = FALSE) while a muting process was not active. The start interlock is active after a safety request on the function block. To revoke the safety request, set "S_AOPD_In" to TRUE in your application.	To end the function block's active start interlock, perform a reset on the function block after you have revoked the safety request on "S_AOPD_In".
8005	The function block is in the safe state. This diagnostic message is only displayed for one cycle of the safety controller. Depending on the state on "S_AOPD_In" and "S_MutingLamp", the diagnostic message switches from WORD#16#8005 to WORD#16#8002, WORD#16#8000 or WORD#16#C003.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8011	The muting process was requested on "S_MutingSwitch11". State FALSE exists on "S_MutingSwitch12". Time monitoring until damping of the second muting sensor ("DiscTimeEntry") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8012	The muting process is active. State TRUE exists on "S_MutingSwitch11" and "S_MutingSwitch12". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. The muting process is completed if "S_MutingSwitch11" and/or "S_MutingSwitch12" switches from TRUE to FALSE.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8311	The muting process was requested on "S_MutingSwitch12". State FALSE exists on "S_MutingSwitch11". Time monitoring until damping of the second muting sensor ("DiscTimeEntry") and time monitoring of the maximum muting duration ("MaxMutingTime") are started. When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C003	The function block detected an impermissible FALSE signal on "S_MutingLamp".	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signal connected to "S_MutingLamp" must indicate state TRUE. • Reset the function block.
C005	The function block detected an invalid value on "DiscTimeEntry" and/or "MaxMutingTime".	• Check the values connected to "DiscTimeEntry" and "MaxMutingTime". • Correct the faulty value(s). • Values must correspond to the results of your risk analysis. • Input parameter limit values must be taken into account when performing your risk analysis. • Compile the project in your safe programming environment and transfer the project to the safety controller you are using. • Perform a cold restart of the safety controller you are using.

Table 663: "SF_MutingPar_2Sensor": Diagnostic codes

Code (hex)	Description	Corrective measures
C006	The muting process was active. The muting process was not yet completed when time monitoring of the maximum muting time expired.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "MaxMutingTime". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
C007	The muting process was requested on "S_MutingSwitch11" or "S_MutingSwitch12". Only one muting sensor indicated state TRUE from the expiration of time monitoring until damping of the second muting sensor ("DiscTimeEntry").	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "DiscTimeEntry". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
C010	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C020	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C030	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C050	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004 to C034	The function block detected an impermissible muting sequence when requesting the muting process on "S_MutingSwitch11" and/or "S_MutingSwitch12". Static TRUE signals on "S_MutingSwitch11" and/or "S_MutingSwitch12" are impermissible for the request.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "S_MutingSwitch11" and "S_MutingSwitch12" must indicate state FALSE. • In addition, the 2 muting sensors connected to "S_MutingSwitch11" and "S_MutingSwitch12" must be undamped. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
C004	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = FALSE.	
C014	At the moment of error detection: • "S_MutingSwitch11" = TRUE and • "S_MutingSwitch12" = FALSE.	
C024	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = TRUE.	
C034	At the moment of error detection: • "S_MutingSwitch11" = TRUE and • "S_MutingSwitch12" = TRUE.	
C104 to C124	The function block detected an impermissible muting sequence after the muting process on "S_MutingSwitch11" was requested. Signal FALSE on "S_MutingSwitch11" is impermissible in this muting sequence.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "S_MutingSwitch11" and "S_MutingSwitch12" must indicate state FALSE. • In addition, the 2 muting sensors connected to "S_MutingSwitch11" and "S_MutingSwitch12" must be undamped. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
C104	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = FALSE.	

Table 663: "SF_MutingPar_2Sensor": Diagnostic codes

Code (hex)	Description	Corrective measures
C124	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = TRUE.	
C204 to C214	The function block detected an impermissible muting sequence after the muting process on "S_MutingSwitch12" was requested. Signal FALSE on "S_MutingSwitch12" is impermissible in this muting sequence.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "S_MutingSwitch11" and "S_MutingSwitch12" must indicate state FALSE. • In addition, the 2 muting sensors connected to "S_MutingSwitch11" and "S_MutingSwitch12" must be undamped. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
C204	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = FALSE.	
C214	At the moment of error detection: • "S_MutingSwitch11" = TRUE and • "S_MutingSwitch12" = FALSE.	
CF04 to CF34	The function block detected an impermissible FALSE signal on "MutingEnable" when requesting the muting process or during the active muting process.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "S_MutingSwitch11" and "S_MutingSwitch12" must indicate state FALSE. • In addition, the 2 muting sensors connected to "S_MutingSwitch11" and "S_MutingSwitch12" must be undamped. • Perform a reset on the function block if "S_MutingSwitch11" and "S_MutingSwitch12" indicate state FALSE.
CF04	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = FALSE.	
CF14	At the moment of error detection: • "S_MutingSwitch11" = TRUE and • "S_MutingSwitch12" = FALSE.	
CF24	At the moment of error detection: • "S_MutingSwitch11" = FALSE and • "S_MutingSwitch12" = TRUE.	
CF34	At the moment of error detection: • "S_MutingSwitch11" = TRUE and • "S_MutingSwitch12" = TRUE.	

Table 663: "SF_MutingPar_2Sensor": Diagnostic codes

6.6.14.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

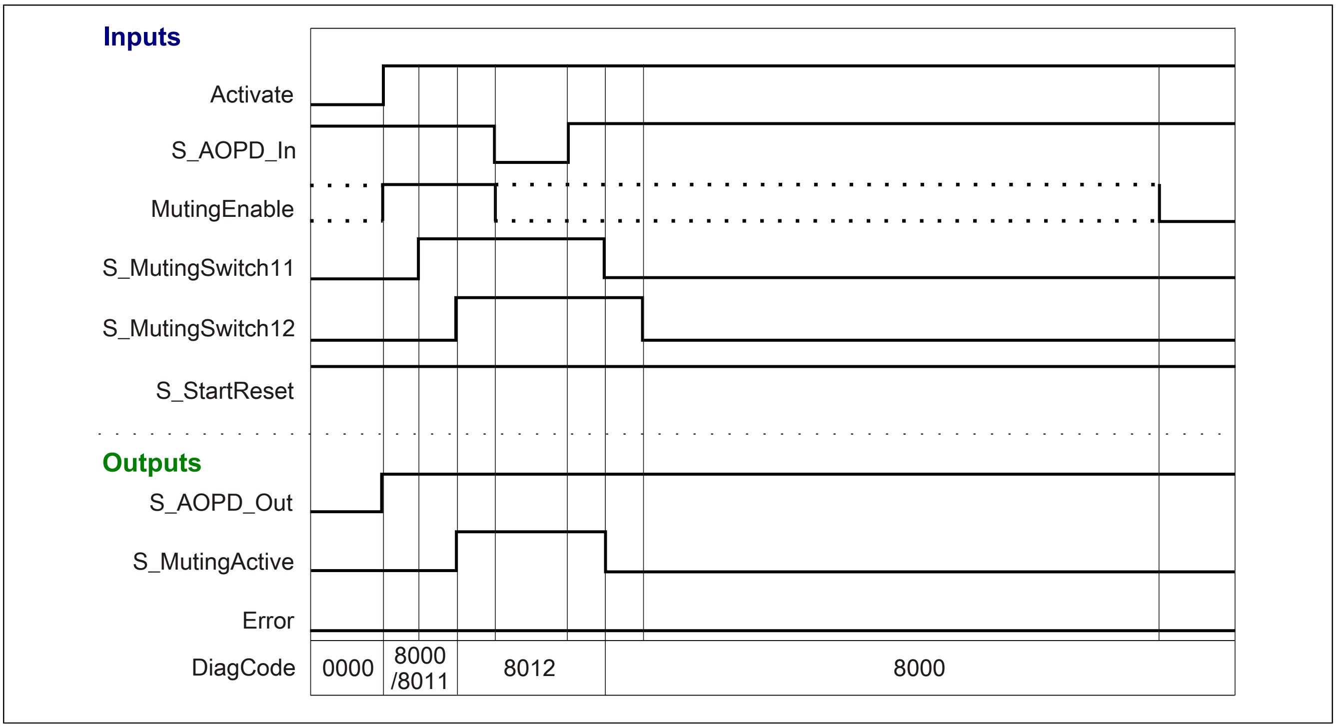


Figure 470: "SF_MutingPar_2Sensor": Signal sequence diagram

6.6.14.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement parallel muting with 2 sensors.

The following example describes function block connections in a muting application when controlling with the signals from 2 parallel muting sensors (see section 6.6.14.7.2 "Parallel muting with 2 sensors").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.14.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "PM2_S15".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "S_MutingSwitch11" and "S_MutingSwitch12", a rising edge on input parameter "Reset" is required to enable enable output "S_AOPD_Out".

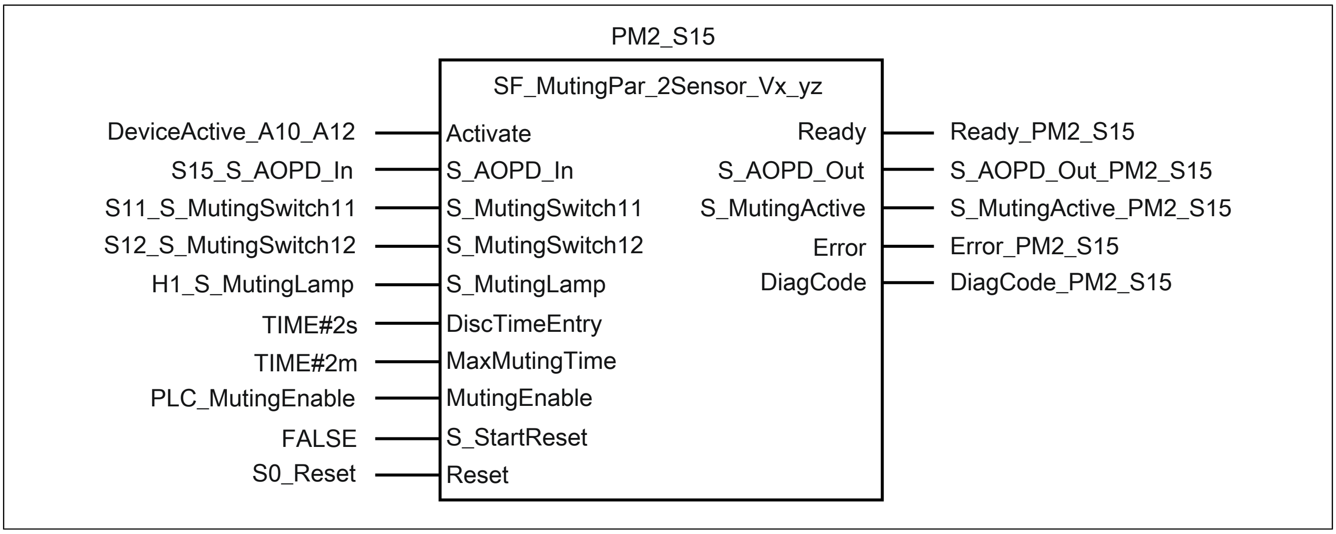


Figure 471: "SF_MutingPar_2Sensor": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the states of the safe input and output devices to which the muting sensors, light curtain and muting lamp involved in the safety function are connected.
S15_S_AOPD_In	SAFEBOOL	In this example, this signal comes from a 2-channel light curtain.
S11_S_MutingSwitch11	SAFEBOOL	Signal input of muting sensor 1
S12_S_MutingSwitch12	SAFEBOOL	Signal input of muting sensor 2
H1_S_MutingLamp	SAFEBOOL	Feedback signal of the muting lamp
TIME#2s on "DiscTimeEntry"	TIME	Specification of the maximum discrepancy time for the muting sensors
TIME#2m on "MaxMutingTime"	TIME	Specification for the maximum time for the complete muting process
PLC_MutingEnable	BOOL	Startup specification for the muting process
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 664: "SF_MutingPar_2Sensor": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_PM2_S15	BOOL	Function block active / Function block inactive. Message for further external processing.
S_AOPD_Out_PM2_S15	SAFEBOOL	Release signal. This output is used for further processing in the program on the safety controller.
S_MutingActive_PM2_S15	SAFEBOOL	Muting active. This output is output via a safe output device. This output is used to control the muting lamp.
Error_PM2_S15	BOOL	Error message from function block for further external processing
DiagCode_PM2_S15	WORD	Diagnostic message from function block for further external processing

Table 665: "SF_MutingPar_2Sensor": Outputs connected to output parameters

6.6.14.7.2 Parallel muting with 2 sensors

This example illustrates function block connections in a muting application when controlling with the signals from 2 parallel muting sensors (see chapter 6.6.14.2 "Function").

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.14.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "PM2_S15".

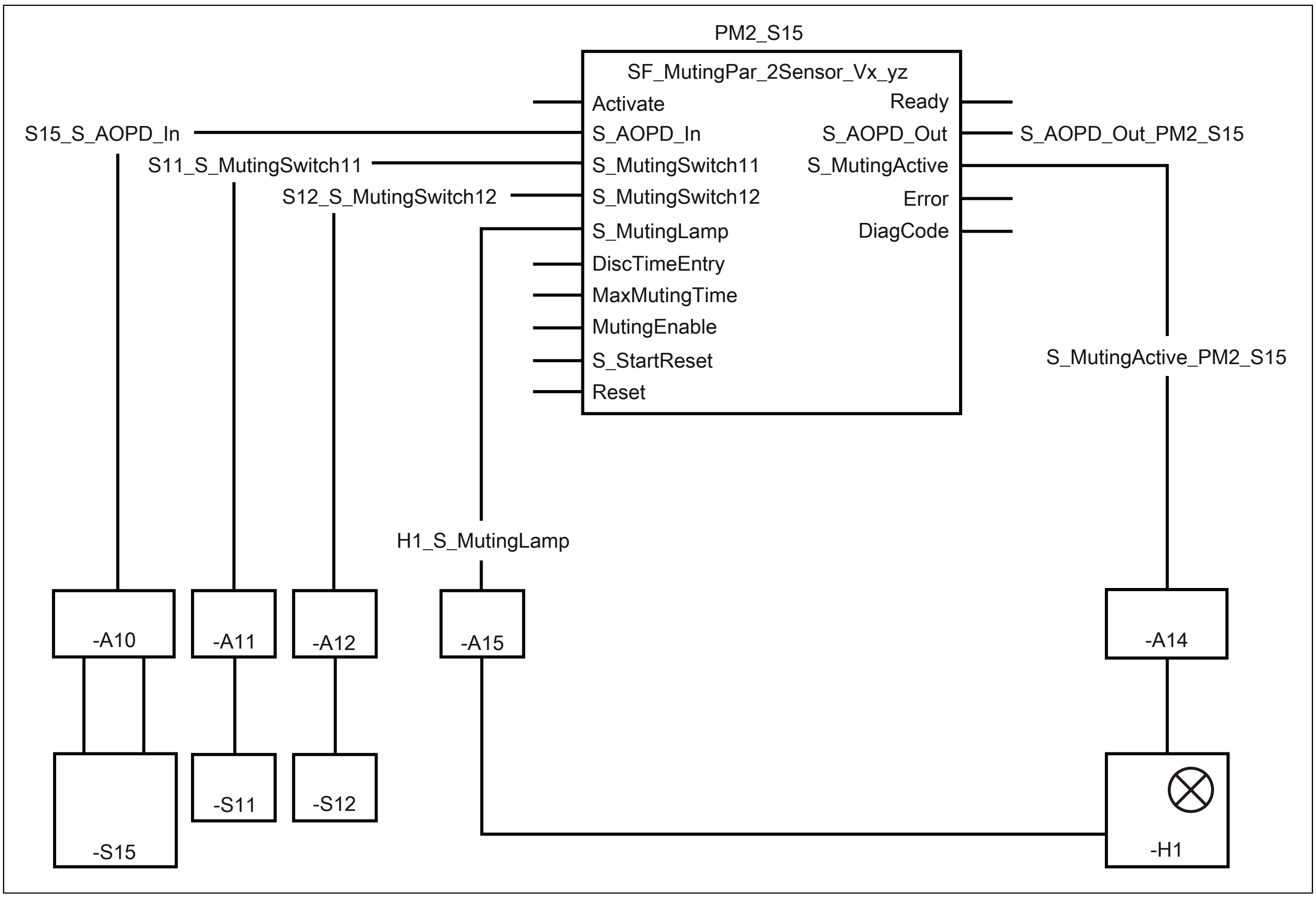


Figure 472: "SF_MutingPar_2Sensor": Parallel muting with 2 sensors

List of equipment

-A10	2-channel input of a safe input device
-A11, -A12	1-channel inputs of the safe input devices
-A15	1-channel input of a safe input device
-A14	1-channel output of a safe output device
-S15	Light curtain, 2-channel
-S11, -S12	Muting sensors arranged in front of the light curtain, see Fig. 469
-H1	Muting lamp monitored by logic

Connected inputs and outputs

S15_S_AOPD_In	Input on "S_AOPD_In"
S11_S_MutingSwitch11	Input on "S_MutingSwitch11"
S12_S_MutingSwitch12	Input on "S_MutingSwitch12"
H1_S_MutingLamp	Input on "S_MutingLamp"
S_AOPD_Out_PM2_S15	Output on "S_AOPD_Out"
S_MutingActive_PM2_S15	Output on "S_MutingActive"

Description

In this example:

- Protective equipment "-S15" (light curtain) is connected to an input of safe device "-A10" via terminals over 2 channels.
- The resulting signal of the 2-channel protective equipment "-S15" from an input of safe device "-A10" is connected to input "S15_S_AOPD_In". This input is connected to function block input parameter "S_AOPD_In" for further processing.
- Muting sensor "-S11" is connected to an input of safe device "-A11" via terminals over 1 channel.
- The resulting signal of muting sensor "-S11" from an input of safe device "-A11" is connected to input "S11_S_MutingSwitch11". This input is connected to function block input parameter "S_MutingSwitch11" for further processing.
- Muting sensor "-S12" is connected to an input of safe device "-A12" via terminals over 1 channel.
- The resulting signal of muting sensor "-S12" from an input of safe device "-A12" is connected to input "S12_S_MutingSwitch12". This input is connected to function block input parameter "S_MutingSwitch12" for further processing.
- Feedback signal of muting lamp "-H1" is connected to an input of safe device "-A15" via terminals over 1 channel.
- The resulting signal of muting lamp "-H1" from an input of safe device "-A15" is connected to input "H1_S_MutingLamp". This input is connected to function block input parameter "S_MutingLamp" for further processing.
- Output parameter "S_MutingActive" is connected to output "S_MutingActive_PM2_S15".
- Output "S_MutingActive_PM2_S15" controls a 1-channel output of safe device "-A14". The muting lamp is connected to this output of safe device "-A14" via terminals over 1 channel.
- Output "S_AOPD_Out_PM2_S15" connected to output parameter "S_AOPD_Out" controls the stop request to the application.

6.6.14.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 61496-1	Muting appendix	**Safeguarding the danger zone (muting is inactive):** If the function block does not detect the active muting process on its inputs "S_MutingSwitch11" and "S_MutingSwitch12", then signal FALSE on "S_AOPD_In" results in a safe state of output parameter "S_AOPD_Out" (FALSE). **Enabling the muting process:** The state on "S_MutingSwitch11" and "S_MutingSwitch12" switches from FALSE to TRUE within the time frame specified on "DiscTimeEntry". **Active muting process:** While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out". The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE). **Completing the muting process:** "S_MutingSwitch11" and/or "S_MutingSwitch12" is set from TRUE to FALSE. **Invalid muting sequences:** Invalid states for the muting process on "S_MutingSwitch11" and/or "MutingSwitch12" result in "S_AOPD_Out" taking on the safe state (FALSE). **Start interlocks:** After the function block is enabled, the function block optionally supports a start interlock depending on the specification on "S_StartReset". A start interlock is active after the function block has detected an invalid muting sequence or other error. **State of the muting process:** The function block outputs the state of the muting process on "S_MutingActive".
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_StartReset" = FALSE) • Enabling the function block ("S_StartReset" = FALSE) The function block supports a start interlock after the following: • Invalid muting sequence • Error detection by the function block • The safety function is no longer being requested If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal and/or the signal on "S_StartReset".
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_AOPD_Out") executes a category 0 stop.

Table 666: "SF_MutingPar_2Sensor": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.15 SF_MutingSeq

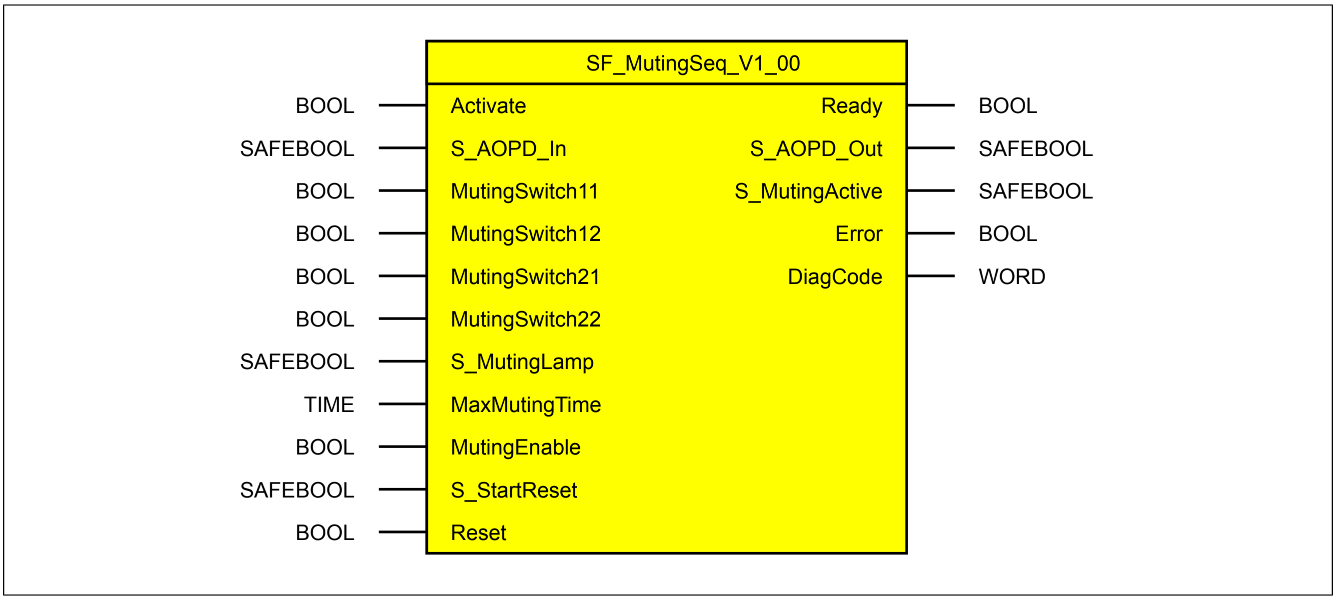


Figure 473: Function block "SF_MutingSeq"

6.6.15.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_AOPD_In	SAFEBOOL	Variable	State	FALSE	Signal input of protective equipment (light curtain)
MutingSwitch11	BOOL	Variable	State	FALSE	Signal input of muting sensor 1
MutingSwitch12	BOOL	Variable	State	FALSE	Signal input of muting sensor 2
MutingSwitch21	BOOL	Variable	State	FALSE	Signal input of muting sensor 3
MutingSwitch22	BOOL	Variable	State	FALSE	Signal input of muting sensor 4
S_MutingLamp	SAFEBOOL	Variable/ Constant	State	FALSE	Feedback signal of the muting lamp
MaxMutingTime	TIME	Constant	State	#0ms	Specification for the maximum time for the complete muting process
MutingEnable	BOOL	Variable/ Constant	State	FALSE	Startup specification for the muting process
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 667: "SF_MutingSeq": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_AOPD_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
S_MutingActive	SAFEBOOL	Variable	State	FALSE	State of the muting process
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 668: "SF_MutingSeq": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 669: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.15.2 Function

Function block "SF_MutingSeq" is used to support a sequential muting function with 4 muting sensors ("MS_11", "MS_12" and "MS_21", "MS_22") in an application.

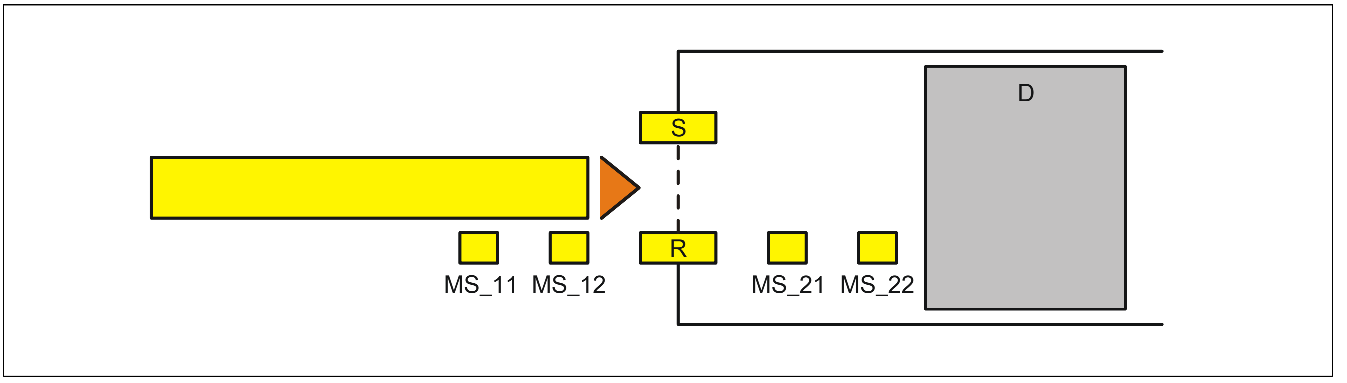


Figure 474: "SF_MutingSeq": Sequential muting with 4 muting sensors

Key:

MS_11, MS_12, MS_21, MS_22	Muting sensors
S, R	Transmitter (send) and receiver (receive) of the light curtain
D	Danger zone (danger)

6.6.15.2.1 Connecting conditions

The signals of muting sensors "MS_11" and "MS_12" are connected to input parameters "MutingSwitch11" and "MutingSwitch12" in a muting application. The signals of muting sensors "MS_21" and "MS_22" are connected to input parameters "MutingSwitch21" and "MutingSwitch22". The signal from the protective equipment (light curtain) is connected to input parameter "S_AOPD_In".

Function block signal "S_AOPD_Out" is the release signal for the entire process. To process the release or request of the safe state in the safety application, the signal from "S_AOPD_Out" must be logically connected in the safety application so that FALSE on "S_AOPD_Out" results in a cutoff of the application in the safeguarded danger zone.

6.6.15.2.2 Muting process

The entire muting process is divided up into different muting sequences. The following describes the individual muting sequences.

Note that the following only describes the direction of material flow from muting sensors "MS_11"/"MS_12" to muting sensors "MS_21"/"MS_22". The function block also supports the reverse direction of material flow from muting sensors "MS_21"/"MS_22" to muting sensors "MS_11"/"MS_12". The functional sequence is identical.

Safeguarding the danger zone (muting is inactive, the protective equipment is active)

If the function block does not detect the active muting process on its inputs "MutingSwitch11" and "MutingSwitch12", then signal FALSE on "S_AOPD_In" (light curtain) results in a safe state of output parameter "S_AOPD_Out" (FALSE).

Enabling the muting process (the protective equipment is disabled)

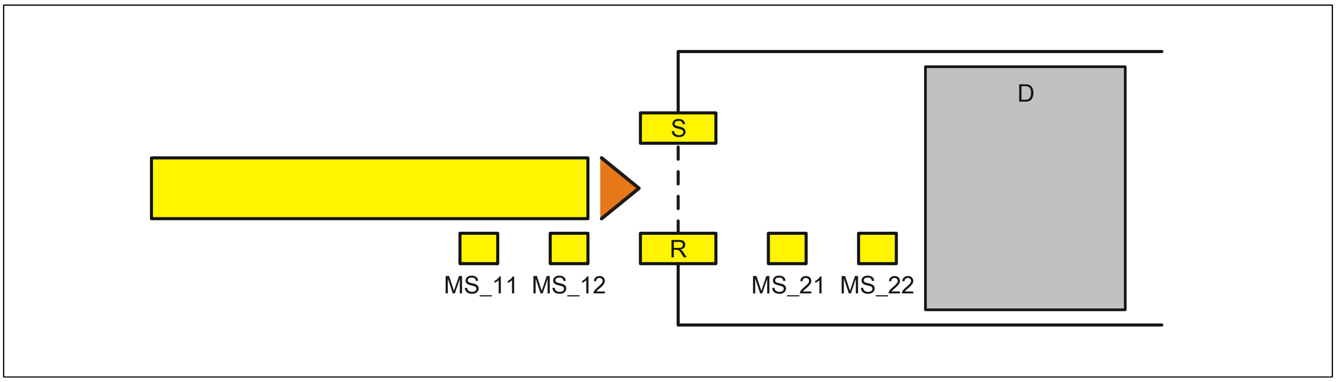


Figure 475: "SF_MutingSeq": Sequential muting with 4 muting sensors - Muting sequence 1

The muting process is active if a signal change from FALSE to TRUE takes place on "MutingSwitch12" after the signal change from FALSE to TRUE on "MutingSwitch11".

Time "MaxMutingTime" is started on a signal change from FALSE to TRUE on "MutingSwitch11".

The entire muting process is not permitted to exceed the time specified on "MaxMutingTime". If this time is exceeded, the function block generates an error message. Enable output "S_AOPD_Out" takes on the safe state (FALSE).

Active muting process (the protective equipment is inactive)

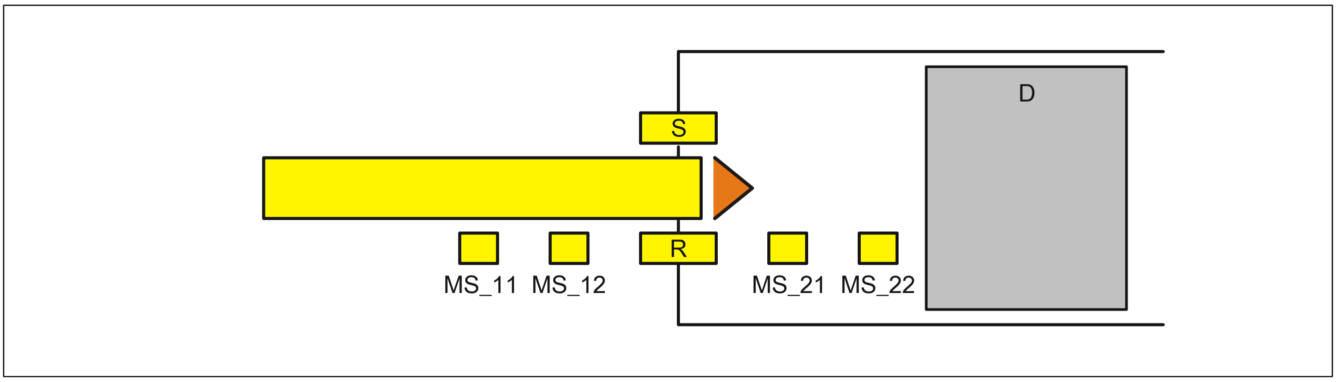


Figure 476: "SF_MutingSeq": Sequential muting with 4 muting sensors - Muting sequence 2

While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out".

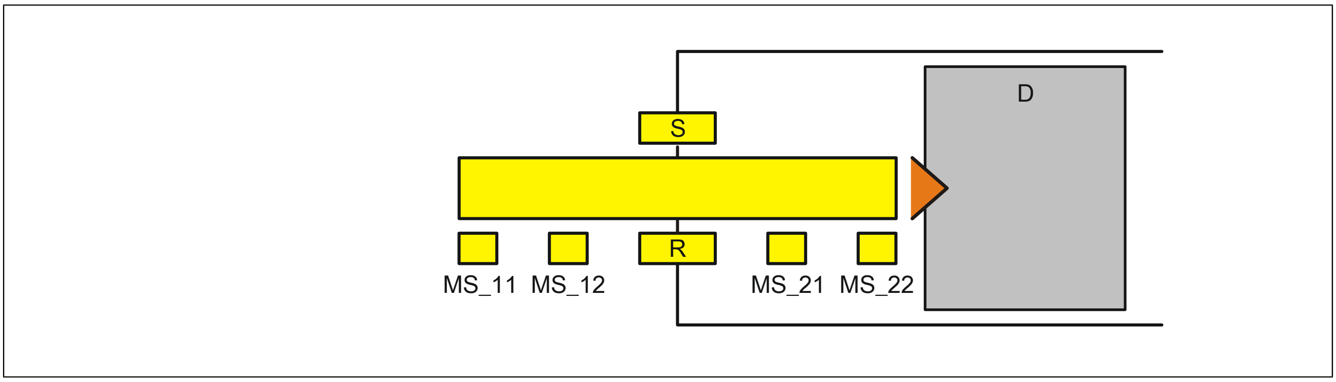


Figure 477: "SF_MutingSeq": Sequential muting with 4 muting sensors - Muting sequence 3

All 4 muting sensors detect the material. The muting process only remains active if muting sensors "MS_21" and "MS_22" are enabled before muting sensors "MS_11" and "MS_12" are disabled and the time specified on input parameter "MaxMutingTime" has not been exceeded.

The active muting process is indicated by signal TRUE on "S_MutingActive".

The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE).

Completing the muting process (the protective equipment is enabled again)

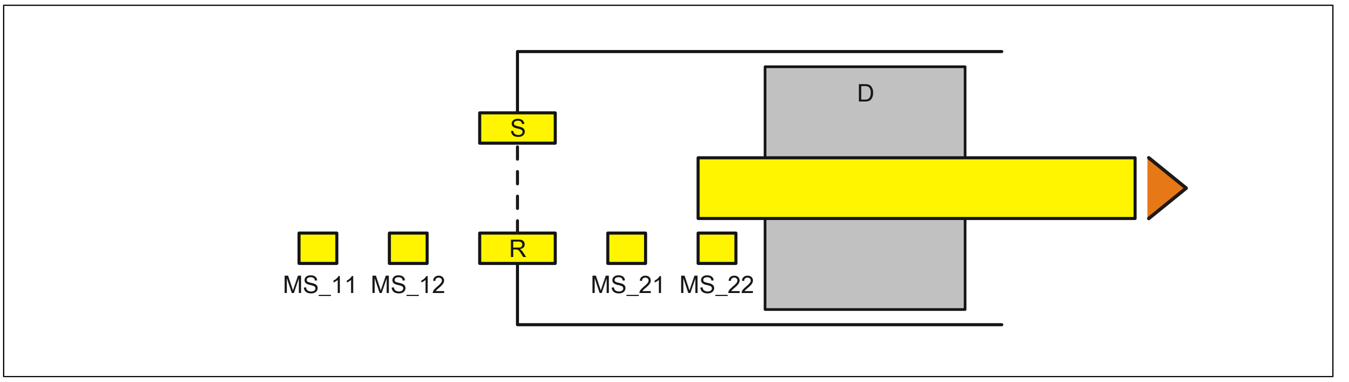


Figure 478: "SF_MutingSeq": Sequential muting with 4 muting sensors - Muting sequence 4

If "MutingSwitch11" and "MutingSwitch12" indicate state FALSE, a state change from TRUE to FALSE on "MutingSwitch21" disables the muting process. The output signal on "S_MutingActive" switches from TRUE to FALSE. In this muting phase, signal FALSE on "S_AOPD_In" (light curtain) results in enable output "S_AOPD_Out" taking on the safe state (FALSE).

Invalid muting sequences

Invalid states for the muting process on "MutingSwitch11" and/or "MutingSwitch12" and/or "MutingSwitch21" and/or "MutingSwitch22" result in enable output "S_AOPD_Out" taking on the safe state (FALSE) and output "S_MutingActive" switching to FALSE.

6.6.15.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.15.2.4 Start interlock

A start interlock is active after the safety function is no longer being requested and/or after error messages (e.g. invalid muting sequence detected).

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

6.6.15.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.15.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.15.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.15.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

6.6.15.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.15.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.15.4 Input parameters

6.6.15.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.15.4.2 S_AOPD_In

General function

- Signal input of protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to the protective equipment (e.g. safety light curtain) in the muting application over 1 or 2 channels. Input parameter "S_AOPD_In" is then controlled via this signal.

Function description

The function block evaluates the state of the connected protective equipment via the signal connected on input parameter "S_AOPD_In".

Regardless whether the protective equipment is connected to the safe device over 1 or 2 channels, "S_AOPD_In" is only connected to one signal.

If protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected protective equipment is not damped.

FALSE

The connected protective equipment is damped.

If muting is active, "S_AOPD_Out" does not initiate the safe state.
If muting is inactive, "S_AOPD_Out" initiates the safe state.

6.6.15.4.3 MutingSwitch11

General function

- Signal input of muting sensor 1
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_11")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_11" from the muting application. Input parameter "MutingSwitch11" is then controlled via this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch11". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch11" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch11". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch11". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.15.4.4 MutingSwitch12

General function

- Signal input of muting sensor 2
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_12")

Data type

- BOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a device that is connected to muting sensor "MS_12" from the muting application. Input parameter "MutingSwitch12" is then controlled via this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch12". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch12" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch12". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch12". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.15.4.5 MutingSwitch21

General function

- Signal input of muting sensor 3
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_21")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_21" from the muting application. Input parameter "MutingSwitch21" is then controlled via this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch21". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch21" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch21". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch21". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.15.4.6 MutingSwitch22

General function

- Signal input of muting sensor 4
  (in the example, see chapter "Function" and chapter "Application example": Muting-Sensor "MS_22")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_22" from the muting application. Input parameter "MutingSwitch22" is then controlled via this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch22". The result of the evaluation results in starting or stopping the muting process. In addition, it also checks whether the connected muting sensor is in an invalid state.

Take the following points into account if you are using a safe input device to evaluate the muting sensors.

Regardless whether the muting sensor is connected to the safe device over 1 or 2 channels, "MutingSwitch22" is only connected to one signal.

If a muting sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "MutingSwitch22". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "MutingSwitch22". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.15.4.7 S_MutingLamp

General function

- Feedback signal of the muting lamp

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the feedback signal of the muting lamp in the muting application over 1 or 2 channels. Input parameter "S_MutingLamp" is then controlled using this signal.**

> ## Information:
> **If your risk analysis determines that a muting lamp is not required in your muting application, note that you can specify constant TRUE on this input parameter.**

Function description

The function block evaluates the state of the connected muting lamp (lamp operational / not operational) using the signal connected on input parameter "S_MutingLamp".

Note that the feedback signal of the muting lamp must permanently indicate state TRUE if its function is not impaired. If the lamp function is impaired, the feedback signal must permanently indicate state FALSE.

Regardless of whether the muting lamp is connected to the safe device over 1 or 2 channels, "S_MutingLamp" is only connected to one signal.

If the muting lamp is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_MutingLamp". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The function of the muting lamp is not impaired.

FALSE

The function of the muting lamp is impaired.

6.6.15.4.8 MaxMutingTime

General function

- Specification for the maximum time for the complete muting process

Data type

- TIME

Connection

- Constant

Function description

The maximum time for the complete muting process is specified using this input parameter. This time starts when the signals on "MutingSwitch11" and/or "MutingSwitch12" change from FALSE to TRUE. The muting process is completed when "MutingSwitch21" and "MutingSwitch22" have afterwards changed from TRUE to FALSE.

You must define and validate the time value for input parameter "MaxMutingTime" based on your application and risk analysis.

Range of values: 0 to 10 minutes

6.6.15.4.9 MutingEnable

General function

- Startup specification for the muting process

Data type

- BOOL

Connection

- Variable or constant

Information:

Control this input parameter with a signal from the standard application that releases the muting process. If the result of your risk analysis is that you do not need a release signal from the standard application, you can alternatively assign constant TRUE.

Function description

Input parameter "MutingEnable" receives the starting signal from the standard application to enable the muting process. This is a measure for reducing the risk of an unintended muting process.

TRUE

Starting the muting function is possible.

FALSE

Starting the muting function is not possible.

6.6.15.4.10 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.15.4.11 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.15.5 Output parameters

6.6.15.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.15.5.2 S_AOPD_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **Connect this output parameter to the safety application in such a way that the safety application takes on and maintains the safe state when signal FALSE is output.**

Function description

The release signal is the safe enable signal for the safeguarded area to control an output on a safe device and therefore the process. This output parameter represents the state of the protective device in the muting application. The release signal is controlled based on the state of the protective device and start interlock.

In addition, the release signal controls the request for the stop function. Control the stop function of the connected safety application by appropriately connecting "S_AOPD_Out".

Since the release signal is present on output "S_AOPD_Out", this output is referred to as the "enable output".

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = TRUE (light curtain)
- or: The muting process is active and the function block did not detect an invalid muting sequence.
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- and: The muting process is not active ("MutingEnable" = FALSE) and "S_AOPD_In" = FALSE (light curtain)
- or: The muting process is active and the function block detected an invalid muting sequence.
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_AOPD_Out" is set to TRUE only if input "S_AOPD_In" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

<table>
<tr><th colspan="2">Input parameters</th><th>Action</th><th>Start interlock</th><th>Reset</th><th>Enable output</th></tr>
<tr><td rowspan="2">S_StartReset</td><td>TRUE</td><td rowspan="2">After the function block is enabled / cold restart of the safety controller, the start interlock is...</td><td>...inactive.</td><td>No action on "Reset" is required...</td><td rowspan="2">• ...to set enable output "S_AOPD_Out" to TRUE on a valid input signal combination.
• ...to end the start interlock.</td></tr>
<tr><td>FALSE</td><td>...active.</td><td>"Reset" must be set from FALSE to TRUE...</td></tr>
</table>

Table 670: "SF_MutingSeq": Input parameter "S_StartReset"

6.6.15.5.3 S_MutingActive

General function

- State of the muting process

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **If the result of your risk analysis is that you must use a muting lamp in the muting application, connect this output parameter to a safe output device that is connected to the muting lamp.**
>
> **In addition, connect this output parameter to the safety application in such a way that the safety application is controlled according to the muting state.**

Function description

This output parameter indicates whether a muting process is enabled and being executed or not enabled at all.

TRUE

The function block has been enabled ("Activate" = TRUE).

The muting process is enabled ("MutingEnable" = TRUE) and being executed. "S_AOPD_In" = FALSE does not result in the safe state on "S_AOPD_Out" (FALSE).

FALSE

The muting process is not enabled ("MutingEnable" = FALSE). "S_AOPD_In" = FALSE results in the safe state on "S_AOPD_Out" (FALSE).

6.6.15.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.15.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.15.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. The muting process is not active. A safety request is not active. The function block assesses signal FALSE on "S_AOPD_In" as a safety function request. Normal operation is possible in the area safeguarded by the protective equipment.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	The function block has been enabled. The function block's start interlock is active.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The function block is in the safe state. The function block detected a safety request ("S_AOPD_In" = FALSE) while a muting process was not active. The start interlock is active after a safety request on the function block. To revoke the safety request, set "S_AOPD_In" to TRUE in your application.	To end the function block's active start interlock, perform a reset on the function block after you have revoked the safety request on "S_AOPD_In".
8005	The function block is in the safe state. This diagnostic message is only displayed for one cycle of the safety controller. Depending on the state on "S_AOPD_In" and "S_MutingLamp", the diagnostic message switches from WORD#16#8005 to WORD#16#8002, WORD#16#8000 or WORD#16#C003.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8011	Conveying direction 1: The muting process was requested on "MutingSwitch11". State FALSE exists on "MutingSwitch12". Time monitoring of the maximum muting duration ("MaxMutingTime") is started. To exit this sequence, signal TRUE is required on "MutingSwitch12". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8012	Conveying direction 1: The muting process is active. State TRUE exists on "MutingSwitch11" and "MutingSwitch12". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, "MutingSwitch11" and "MutingSwitch12" and then "MutingSwitch21" must take on state FALSE.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8112	Conveying direction 2: The muting process is active. State TRUE existed on "MutingSwitch21" and "MutingSwitch22". Time monitoring of the maximum muting duration ("MaxMutingTime") is running. "S_AOPD_In" can indicate state FALSE until the maximum muting duration ("MaxMutingTime") expires without the function block taking on the safe state. To exit this sequence, "MutingSwitch21" and "MutingSwitch22" and then "MutingSwitch12" must take on state FALSE.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8122	Conveying direction 2: The muting process was requested on "MutingSwitch22". State FALSE exists on "MutingSwitch21". Time monitoring of the maximum muting duration ("MaxMutingTime") is started. To exit this sequence, signal TRUE is required on "MutingSwitch21". When the safety function is requested ("S_AOPD_In" = FALSE), the function block takes on the safe state.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
C001	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C003	The function block detected an impermissible FALSE signal on "S_MutingLamp".	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signal connected to "S_MutingLamp" must indicate state TRUE. • Reset the function block.

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C005	The function block detected an invalid value on "MaxMutingTime".	• Check the values connected to "MaxMutingTime". • Correct the faulty value(s). • Values must correspond to the results of your risk analysis. • Input parameter limit values must be taken into account when performing your risk analysis. • Compile the project in your safe programming environment and transfer the project to the safety controller you are using. • Perform a cold restart of the safety controller you are using. • Reset the function block.
C006	The muting process was active. The muting process was not yet completed when time monitoring of the maximum muting time expired.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the muting process. • Check the specified time value on "MaxMutingTime". This value must correspond to the results of your risk analysis. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C010	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C020	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C030	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C040	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004 to C0F4	Conveying direction 1 and 2: The function block detected an impermissible muting sequence when requesting the muting process. In the last valid muting sequence, only one TRUE signal of the external muting sensors ("MutingSwitch11" and "MutingSwitch22") was permitted for one conveying direction only. Signal combinations of the muting sensor signals that deviate from this are therefore impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C004	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C014	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C024	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C034	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C044	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C054	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C064	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C074	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C084	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C094	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C0A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C0B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C0C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C0F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C104 to C1F4	Conveying direction 1: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch11" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch12" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch11" was impermissible. In addition, TRUE signals on "MutingSwitch21" and/or "MutingSwitch22" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C104	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C114	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C124	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C134	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C144	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C154	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C164	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C174	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C184	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C194	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C1C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C1D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C1E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C1F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C204 to C2F4	Conveying direction 1: During an active muting process, the function block detected an impermissible muting sequence after the muting process was enabled by "MutingSwitch11". To exit the last valid muting sequence in a valid way, "MutingSwitch11", "MutingSwitch12" and then "MutingSwitch21" would have needed to take on state FALSE. In the last valid muting sequence, a signal change from FALSE to TRUE on "MutingSwitch11" and/or "MutingSwitch12" was impermissible. In addition, a signal change from TRUE to FALSE on "MutingSwitch22" as well as reversing the conveying direction was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C204	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C214	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C224	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C234	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C244	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C254	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C264	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C274	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C284	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C294	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C2C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C2D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C2E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C2F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C304 to C3F4	Conveying direction 2: The function block detected an impermissible muting sequence after the muting process on "MutingSwitch22" was requested. To exit the last valid muting sequence in a valid way, "MutingSwitch21" would have needed to take on state TRUE. In the last valid muting sequence, signal FALSE on "MutingSwitch22" was impermissible. In addition, TRUE signals on "MutingSwitch11" and/or "MutingSwitch12" were impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C304	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C314	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C324	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C334	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C344	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C354	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C364	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C374	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C384	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C394	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C3C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C3F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C404 to C4F4	Conveying direction 2: During an active muting process, the function block detected an impermissible muting sequence after the muting process was requested on "MutingSwitch22". To exit the last valid muting sequence in a valid way, "MutingSwitch22", "MutingSwitch21" and then "MutingSwitch12" would have needed to take on state FALSE. In the last valid muting sequence, a signal change from FALSE to TRUE on "MutingSwitch22" and/or "MutingSwitch21" was impermissible. In addition, a signal change from TRUE to FALSE on "MutingSwitch11" as well as reversing the conveying direction was impermissible.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
C404	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C414	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C424	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C434	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
C444	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C454	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C464	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C474	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
C484	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C494	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C4A4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
C4B4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
C4C4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4D4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4E4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
C4F4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CF04 to CFF4	Conveying direction 1 and 2: The function block detected an impermissible FALSE signal on "MutingEnable" when requesting the muting process or during the active muting process.	• Check the safety functions. • Check the safety program. • Check the safe peripheral. • The signals connected to the muting sensors on "MutingSwitch11" to "MutingSwitch22" must indicate state FALSE. In addition, the 4 muting sensors connected to "MutingSwitch11" to "MutingSwitch22" must be undamped. • Perform a reset on the function block if "MutingSwitch11" to "MutingSwitch22" indicate state FALSE.
CF04	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF14	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF24	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF34	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = FALSE.	
CF44	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF54	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	

Table 671: "SF_MutingSeq": Diagnostic codes

Code (hex)	Description	Corrective measures
CF64	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF74	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = FALSE.	
CF84	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CF94	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFA4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFB4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = FALSE and • "MutingSwitch22" = TRUE.	
CFC4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFD4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = FALSE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFE4	At the moment of error detection: • "MutingSwitch11" = FALSE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	
CFF4	At the moment of error detection: • "MutingSwitch11" = TRUE and • "MutingSwitch12" = TRUE and • "MutingSwitch21" = TRUE and • "MutingSwitch22" = TRUE.	

Table 671: "SF_MutingSeq": Diagnostic codes

6.6.15.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

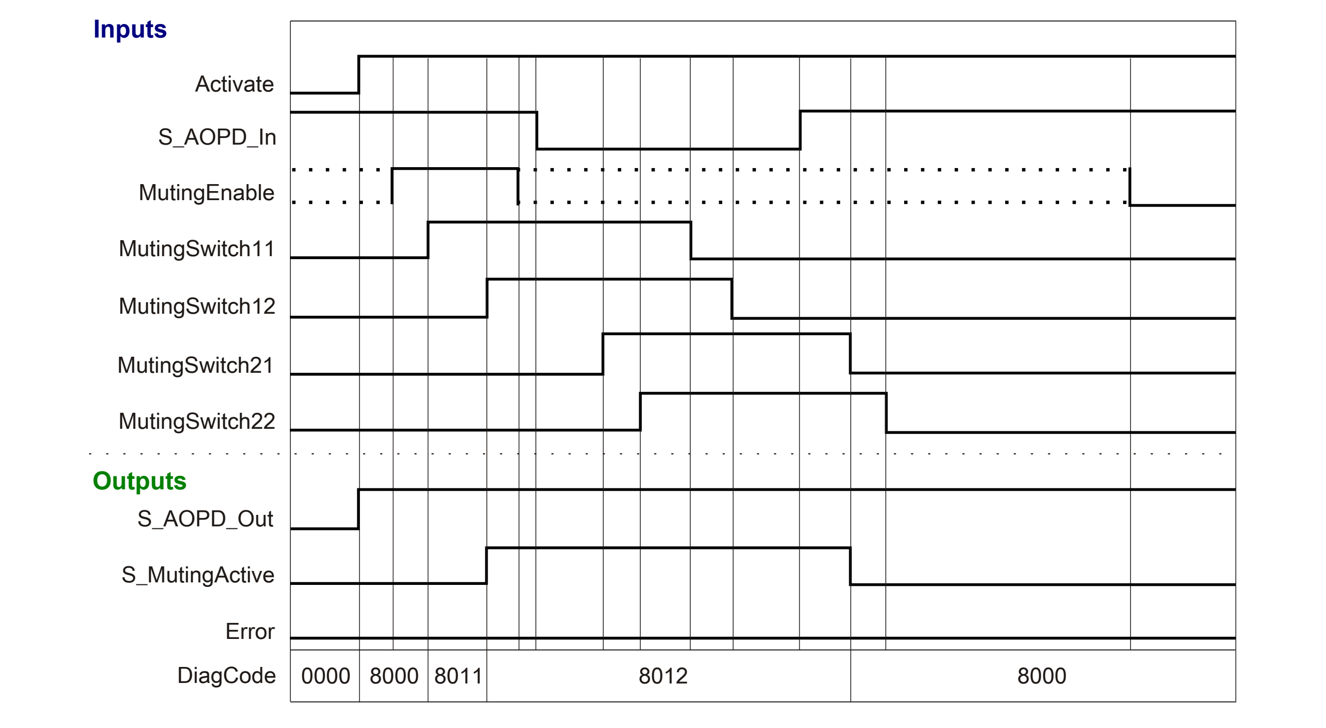


Figure 479: "SF_MutingSeq": Signal sequence diagram

6.6.15.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement sequential muting with 4 sensors.

The following example describes function block connections in a muting application when controlling with the signals from 4 muting sensors (sequential muting, see section 6.6.15.7.2 "Sequential muting with 4 sensors").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.15.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "SM_S15".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlock

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signals on "MutingSwitch11", "MutingSwitch12", "MutingSwitch21" and "MutingSwitch22", a rising edge on input parameter "Reset" is required to enable enable output "S_AOPD_Out".

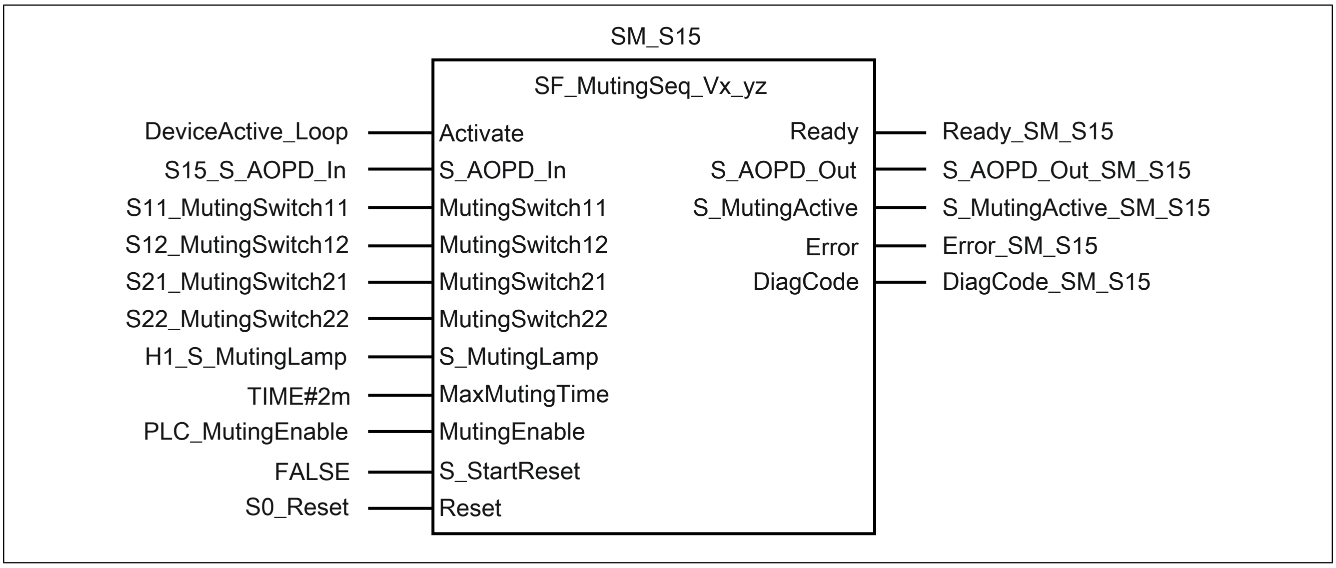


Figure 480: "SF_MutingSeq": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_Loop	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the states of the safe and non-safe input and output devices to which the muting sensors, light curtain and muting lamp involved in the safety function are connected.
S15_S_AOPD_In	SAFEBOOL	In this example, this signal comes from a 2-channel light curtain.
S11_MutingSwitch11	BOOL	Signal input of muting sensor 1
S12_MutingSwitch12	BOOL	Signal input of muting sensor 2
S21_MutingSwitch21	BOOL	Signal input of muting sensor 3
S22_MutingSwitch22	BOOL	Signal input of muting sensor 4
H1_S_MutingLamp	SAFEBOOL	Feedback signal of the muting lamp
TIME#2m on "MaxMutingTime"	TIME	Specification of the maximum time for the complete muting process
PLC_MutingEnable	BOOL	Startup specification for the muting process
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 672: "SF_MutingSeq": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_SM_S15	BOOL	Function block active / Function block inactive. Message for further external processing.
S_AOPD_Out_SM_S15	SAFEBOOL	Release signal. This output is used for further processing in the program on the safety controller.
S_MutingActive_SM_S15	SAFEBOOL	Muting active. This output is output via a safe output device. This output is used to control the muting lamp.
Error_SM_S15	BOOL	Error message from function block for further external processing
DiagCode_SM_S15	WORD	Diagnostic message from function block for further external processing

Table 673: "SF_MutingSeq": Outputs connected to output parameters

6.6.15.7.2 Sequential muting with 4 sensors

This example illustrates function block connections in a muting application when controlling with the signals from 4 muting sensors. The 4 muting sensors are arranged in series (see chapter 6.6.15.2 "Function").

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.15.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "SM_S15".

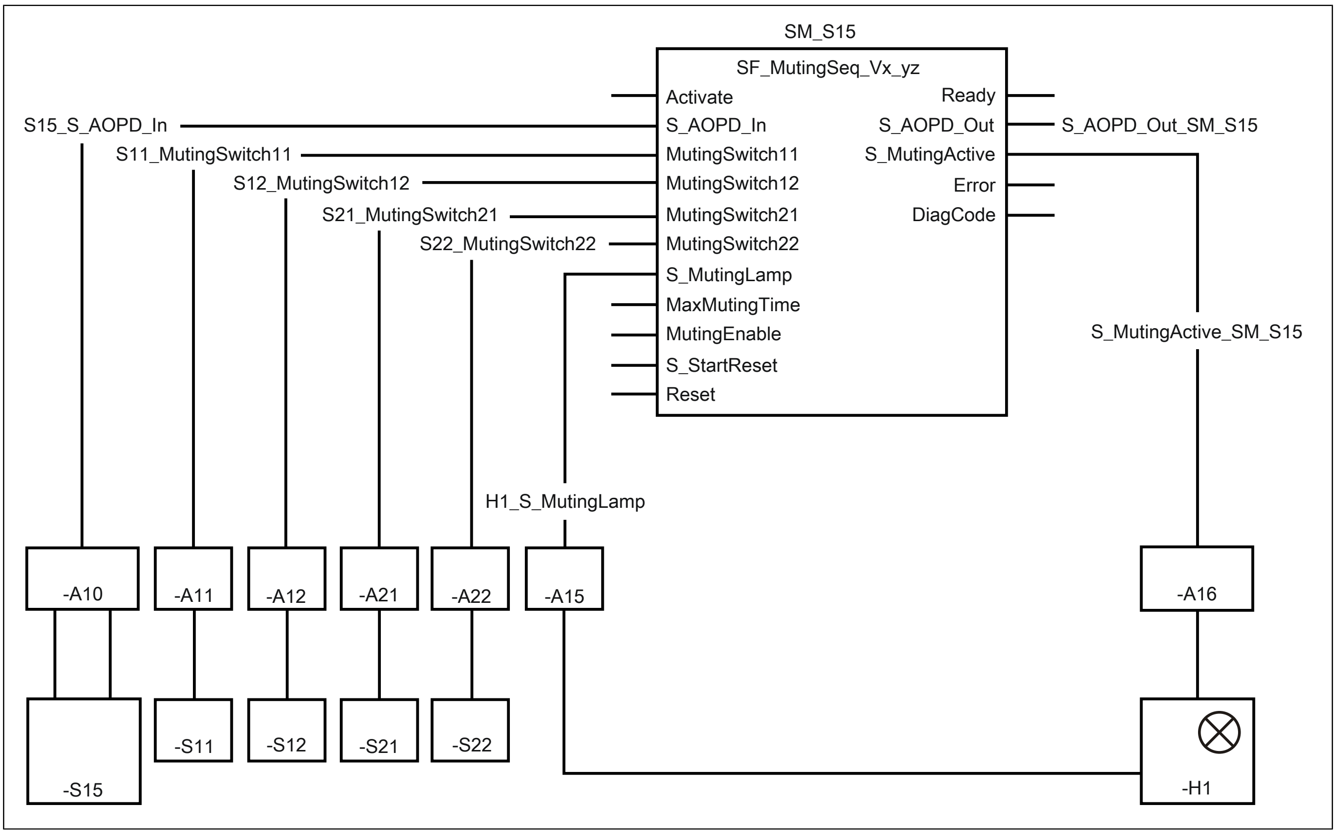


Figure 481: "SF_MutingSeq": Sequential muting with 4 sensors

List of equipment

-A10	2-channel input of a safe input device
-A11, -A12, -A21, -A22	1-channel inputs of the input devices
-A15	1-channel input of a safe input device
-A16	1-channel output of a safe output device
-S15	Light curtain, 2-channel
-S11, -S12	Muting sensors arranged in front of the light curtain, see Fig. 474
-S21, -S22	Muting sensors arranged behind the light curtain, see Fig. 474
-H1	Muting lamp monitored by logic

Connected inputs and outputs

S15_S_AOPD_In	Input on "S_AOPD_In"
S11_MutingSwitch11	Input on "MutingSwitch11"
S12_MutingSwitch12	Input on "MutingSwitch12"
S21_MutingSwitch21	Input on "MutingSwitch21"
S22_MutingSwitch22	Input on "MutingSwitch22"
H1_S_MutingLamp	Input on "S_MutingLamp"
S_AOPD_Out_SM_S15	Output on "S_AOPD_Out"
S_MutingActive_SM_S15	Output on "S_MutingActive"

Description

In this example:

- Protective equipment "-S15" (light curtain) is connected to an input of safe device "-A10" via terminals over 2 channels.
- The resulting signal of the 2-channel protective equipment "-S15" from an input of safe device "-A10" is connected to input "S15_S_AOPD_In". This input is connected to function block input parameter "S_AOPD_In" for further processing.
- Muting sensor "-S11" is connected to an input of device "-A11" via terminals over 1 channel.
- The resulting signal of muting sensor "-S11" from an input of device "-A11" is connected to input "S11_MutingSwitch11". This input is connected to function block input parameter "MutingSwitch11" for further processing.
- Muting sensor "-S12" is connected to an input of device "-A12" via terminals over 1 channel.
- The resulting signal of muting sensor "-S12" from an input of device "-A12" is connected to input "S12_MutingSwitch12". This input is connected to function block input parameter "MutingSwitch12" for further processing.
- Muting sensor "-S21" is connected to an input of device "-A21" via terminals over 1 channel.
- The resulting signal of muting sensor "-S21" from an input of device "-A21" is connected to input "S21_MutingSwitch21". This input is connected to function block input parameter "MutingSwitch21" for further processing.
- Muting sensor "-S22" is connected to an input of device "-A22" via terminals over 1 channel.
- The resulting signal of muting sensor "-S22" from an input of device "-A22" is connected to input "S22_MutingSwitch22". This input is connected to function block input parameter "MutingSwitch22" for further processing.
- Feedback signal of muting lamp "-H1" is connected to an input of safe device "-A15" via terminals over 1 channel.
- The resulting signal of muting lamp "-H1" from an input of safe device "-A15" is connected to input "H1_S_MutingLamp". This input is connected to function block input parameter "S_MutingLamp" for further processing.
- Output parameter "S_MutingActive" is connected to output "S_MutingActive_SM_S15".
- Output "S_MutingActive_SM_S15" controls a 1-channel output of safe device "-A16". The muting lamp is connected to this output of safe device "-A16" via terminals over 1 channel.
- Output "S_AOPD_Out_SM_S15" connected to output parameter "S_AOPD_Out" controls the stop request to the application.

6.6.15.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 61496-1	Muting appendix	**Muting process:** Note that the following only describes the direction of material flow from muting sensor "MS_11" to muting sensor "MS_22". The function block also supports the reverse direction of material flow from muting sensor MS_22" to muting sensor "MS_11". The functional sequence is identical. **Safeguarding the danger zone:** If the function block does not detect the active muting process on its inputs "MutingSwitch11" and "MutingSwitch12", then signal FALSE on "S_AOPD_In" results in a safe state of output parameter "S_AOPD_Out" (FALSE). **Enabling the muting process:** The state of the signals on "MutingSwitch11" and "MutingSwitch12" switches from FALSE to TRUE. The specified time on input parameter "MaxMutingTime" is started. **Active muting process:** While the muting process is active, signal FALSE on "S_AOPD_In" does not result in a safe state of output parameter "S_AOPD_Out". The active muting process must be completed within the time value specified on "MaxMutingTime". Otherwise, enable output "S_AOPD_Out" takes on the safe state (FALSE). **Completing the muting process:** "MutingSwitch11", "MutingSwitch12" and "MutingSwitch21" are set from TRUE to FALSE. The not-yet-expired maximum time for the complete muting process ("MaxMutingTime") is stopped. **Invalid muting sequences:** Invalid states for the muting process on "MutingSwitch11" and/or "MutingSwitch12" and/or "MutingSwitch21" and/or "MutingSwitch22" result in "S_AOPD_Out" taking on the safe state (FALSE). **Start interlocks:** After the function block is enabled, the function block optionally supports a start interlock depending on the specification on "S_StartReset". A start interlock is active after the function block has detected an invalid muting sequence or other error. **State of the muting process:** The function block outputs the state of the muting process on "S_MutingActive".
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_StartReset" = FALSE) • Enabling the function block ("S_StartReset" = FALSE) The function block supports a start interlock after the following: • Invalid muting sequence • Error detection by the function block • The safety function is no longer being requested If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal and/or the signal on "S_StartReset".
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (release signal "S_AOPD_Out") executes a category 0 stop.

Table 674: "SF_MutingSeq": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.16 SF_OutControl

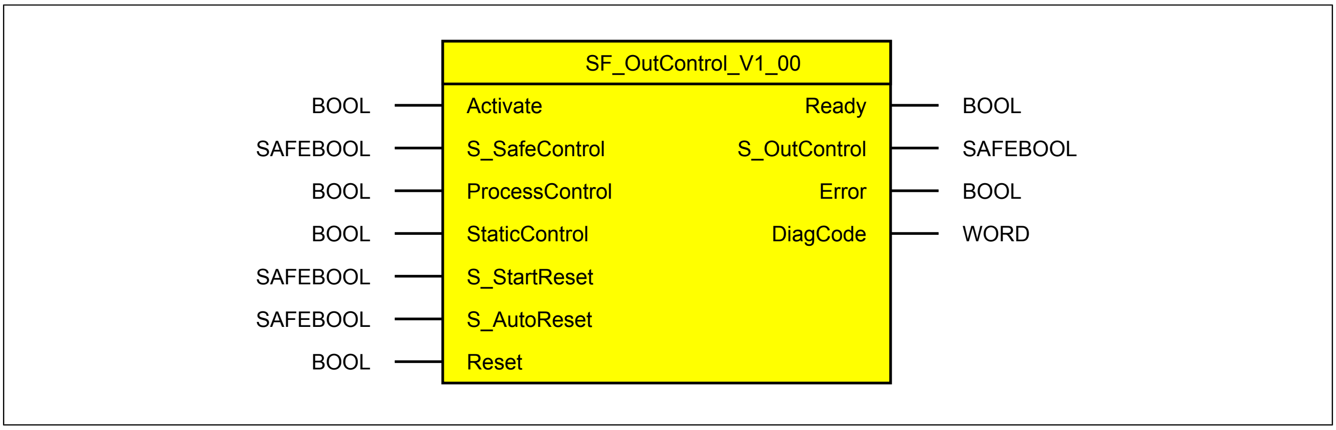


Figure 482: Function block "SF_OutControl"

6.6.16.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_SafeControl	SAFEBOOL	Variable	State	FALSE	Signal input. Signal from the safety controller
ProcessControl	BOOL	Variable/ Constant	State/ Edge	FALSE	Signal input. Signal from the standard controller
StaticControl	BOOL	Constant	State	FALSE	Optional specification for an additional operational stop when the safety function is triggered and/or the function block is disabled
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after signal TRUE returns to "S_SafeControl"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 675: "SF_OutControl": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_OutControl	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 676: "SF_OutControl": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 677: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.16.2 Function

Function block "SF_OutControl" controls the output of a safe device using its enable output "S_OutControl".

The safe output parameter is controlled based on a signal from the standard controller (operational start/stop) and a safe signal (monitoring of a safety function).

The safe signal typically comes from an upstream safety function (e.g. emergency switch-off).

6.6.16.2.1 Optional additional operational stop on triggered safety function

Depending on the specification on input parameter "StaticControl", an operational stop ("ProcessControl" = FALSE) in the standard controller is optionally required when the upstream safety function is triggered and/or the function block is disabled. For this, the signal from the standard controller is monitored by the function block (input parameter "ProcessControl"). An operational start ("ProcessControl" = TRUE) is only possible with this option if the function block is enabled and a non-triggered safety function is detected.

Note that the function block can directly control a downstream safe output in the application in the event of outputs that cannot be read back. For outputs that can be read back, you must connect function block enable output "S_OutControl" to the safe output in the application using function block "SF_EDM".

> **Danger!**
>
> **Inputs "StaticControl", "S_StartReset" and "S_AutoReset" are only permitted to indicate state TRUE if there are assurances in place that no hazardous situation can arise when the safety controller / application is started.**

6.6.16.2.2 Safety function no longer requested

Within the safe control system, the function block optionally (see start interlock) ensures that the signal on the enable output is not set to TRUE solely because the safety function is no longer being requested. Additional manual intervention on input parameter "Reset" is required for this (see start interlock).

6.6.16.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after the signal returns to the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.16.2.4 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> **Danger!**
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.16.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.16.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.16.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.16.3.3 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

In this case, the status of the release signal depends on signal input "S_SafeControl" (safety function active / not active) and the specification of the optional start interlock after the triggering of a safety function is detected (signal TRUE on "S_SafeControl").

6.6.16.3.4 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.16.3.5 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.16.4 Input parameters

6.6.16.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.16.4.2 S_SafeControl

General function

- Signal input. Input for the signal from the safety controller.

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to a signal from the safety controller that specifies whether a triggered safety function is present (typical connection: emergency switch-off function). Input parameter "S_SafeControl" is then controlled via this signal.**

> ## Information:
> **Connect this input parameter with the output signal of a safe function block, such as "SF_EmergencyStop".**

Function description

The signal connected to input parameter "S_SafeControl" is processed by the function block.

The signal input processes the state of an upstream safety function.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream safety function was not triggered.

After a signal change from FALSE to TRUE on input parameter "S_SafeControl", the function block optionally supports a start interlock ("S_AutoReset" = FALSE). An active start interlock is indicated accordingly on output parameter "DiagCode". While the start interlock is active, enable output "S_OutControl" remains set to FALSE. The start interlock is reset with a rising edge of "Reset" ("Reset": FALSE → TRUE). The reset changes enable output "S_OutControl" from FALSE to TRUE.

FALSE

The upstream safety function has been triggered, the wiring for the upstream safety function is interrupted or the safe device connected to the upstream safety function is shut down or defective.

Enable output "S_OutControl" is set to FALSE; output parameter "DiagCode" is set accordingly.

6.6.16.4.3 ProcessControl

General function

- Signal input. Input for the signal from the standard controller.

Data type

- BOOL

Connection

- Variable or constant

> ## Information:
> **Connect this input parameter to a signal from the standard controller. Input parameter "ProcessControl" is then controlled via this signal.**

Function description

The signal connected to input parameter "ProcessControl" is processed by the function block.

The signal input processes the request by the process from the standard controller for setting the function block enable output to TRUE if the remaining input signal combination is valid for this.

The signal input controls the operational start/stop for setting an output of the safe device to TRUE under consideration of the state on "S_SafeControl" (upstream safety function).

Depending on the specification on "StaticControl", the signal input is state-controlled and/or edge-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

This input parameter processes the request by the process from the standard controller for setting enable output "S_OutControl" to TRUE.

FALSE

This input parameter processes the request by the process from the standard controller for setting enable output "S_OutControl" to FALSE.

6.6.16.4.4 StaticControl

General function

- Optional specification for an additional operational stop when the safety function is triggered and/or the function block is disabled

Data type

- BOOL

Connection

- Constant

Information:

Connect this input parameter to a constant in order to select the required function in your application.

Function description

This input parameter initiates an additional operational stop in the standard controller when a safety function is triggered and/or the function block is disabled in order to reduce the risk of unexpected startup. The function block evaluates the operational stop and operational start on "ProcessControl".

You are responsible for planning the behavior of stop functions according to the results of your risk analysis for the safety function.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

Danger!

Toggling signal levels on "ProcessControl"

Toggling signal levels on "ProcessControl" are interpreted by the function block as an operational stop and operational start. The toggling signal level therefore controls the safe output signal of the function block if the remaining input signal combination is valid for this.

TRUE

No additional operational stop on "ProcessControl" is required when the safety function is triggered and/or the function block is disabled.
A static TRUE signal on "ProcessControl" sets enable output "S_OutControl" to TRUE if the remaining input signal combination is valid for this.

FALSE

An additional operational stop on "ProcessControl" is required when the safety function is triggered and/or the function block is disabled.
A static TRUE signal on "ProcessControl" results in an error message after the function block is enabled and the safety request is revoked. Enable output "S_OutControl" is not set to TRUE.

"ProcessControl" must be set from FALSE to TRUE (operational start) after the function block is enabled and the safety request is revoked in order to set enable output "S_OutControl" to TRUE if the remaining input signal combination is valid for this.

6.6.16.4.5 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.**

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.**

6.6.16.4.6 S_AutoReset

General function

- Specification of the start interlock after signal TRUE returns to "S_SafeControl"

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after the signal has returned to safe input parameter "S_SafeControl" (return of the safety function).

TRUE

After signal TRUE returns to safe input parameter "S_SafeControl", the function block does not support start interlock.

No action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After signal TRUE returns to safe input parameter "S_SafeControl", the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.16.4.7 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.16.5 Output parameters

6.6.16.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.16.5.2 S_OutControl

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Connect this output parameter to a safe 1-channel or 2-channel output of a safe device or to another function block (e.g. "SF_EDM").**

Function description

The release signal is the safe signal of the connected safe function block for the process being controlled used to control an output of a safe device or another function block (e.g. "SF_EDM").

The release signal is controlled based on the state of the safe function block and start interlock.

In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_OutControl", this output is referred to as the "enable output".

TRUE

The safe output of the safe device is set to TRUE. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The function block did not detect a triggered safety function ("S_SafeControl" = TRUE).
- A start interlock is not active.
- An operational start is requested on "ProcessControl" (TRUE).
- The function block did not detect any faults.

FALSE

The safe output of the safe device is set to FALSE. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The function block detected a triggered safety function ("S_SafeControl" = FALSE).
- A start interlock is active.
- An operational stop is requested on "ProcessControl" (FALSE).
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_OutControl" is set to TRUE only if input "S_SafeControl" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

Input parameters		Action	Start interlock	Reset	Enable output
S_AutoReset	TRUE	After signal TRUE returns on the safe input, the start interlock is...	...inactive.	No action on "Reset" is required...	• ...to set enable output "S_OutControl" to TRUE on a valid input signal combination. • ...to end the start interlock.
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	
S_StartReset	TRUE	After the function block is enabled / cold restart of the safety controller, the start interlock is...	...inactive.	No action on "Reset" is required...	
	FALSE		...active.	"Reset" must be set from FALSE to TRUE...	

Table 678: "SF_OutControl": Input parameter "S_AutoReset" / "S_StartReset"

6.6.16.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.16.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.16.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. A triggered safety function is not present. An operational start from the standard controller is active on "ProcessControl" ("ProcessControl" = TRUE).	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral. • Check the configuration of the function block. • Check control of the operational start from the standard controller.
8001	The function block has been enabled. The function block's start interlock is active. "S_StartReset" indicates state FALSE.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The upstream safety function was triggered.	Revoke the triggered safety function.
8003	The triggered safety function was revoked. The function block's start interlock is active. "S_AutoReset" indicates state FALSE.	Execute a reset on the function block in order to end the active start interlock of the function block and to set the release signal to TRUE.
8010	A triggered safety function is not present. An operational stop from the standard controller is requested on "ProcessControl" ("ProcessControl" = FALSE).	Request an operational start on "ProcessControl" from the standard controller.
C001	The function block detected a static reset signal on "Reset". "S_StartReset" indicates state FALSE.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	The function block detected a static reset signal on "Reset". "S_AutoReset" indicates state FALSE.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	After the function block is enabled or after the safety request is revoked, the function block detected the request for an operational start from the standard controller. "StaticControl" indicates state FALSE.	• Revoke the request for the operational start from the standard controller. • Adjust the configuration of "StaticControl" if a static operational start from the standard controller is permitted in your safety function after the function block is enabled and the safety request is revoked. • Take the results of your risk analysis into account for this!
C111	The function block detected a simultaneous signal change on "Reset" and "ProcessControl".	• Ensure the independence of the inputs connected to "ProcessControl" and "Reset" during verification of your program on the safety controller. • Set "Reset" to state FALSE.
C211	The function block detected a simultaneous signal change on "Reset" and "ProcessControl".	• Ensure the independence of the inputs connected to "ProcessControl" and "Reset" during verification of your program on the safety controller. • Set "Reset" to state FALSE.

Table 679: "SF_OutControl": Diagnostic codes

6.6.16.6 Signal sequence diagrams of function block

Please note that not all temporary intermediate states are represented in the signal sequence diagrams. These diagrams only illustrate typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figures are specified in hexadecimal.

Signal sequence diagram 1

"StaticControl" = TRUE
"S_StartReset" = FALSE
"S_AutoReset" = FALSE

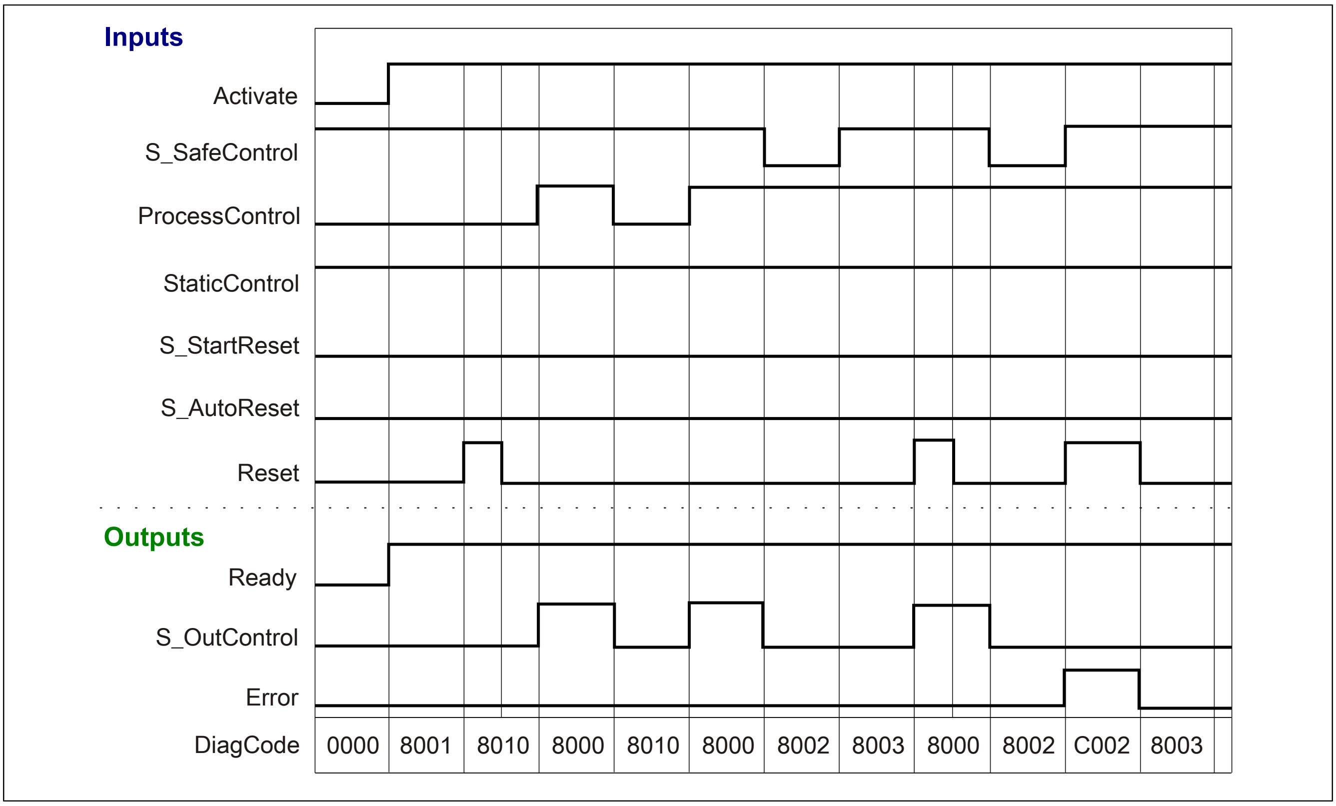


Figure 483: "SF_OutControl": Signal sequence diagram 1

Signal sequence diagram 2

"StaticControl" = FALSE
"S_StartReset" = TRUE
"S_AutoReset" = FALSE

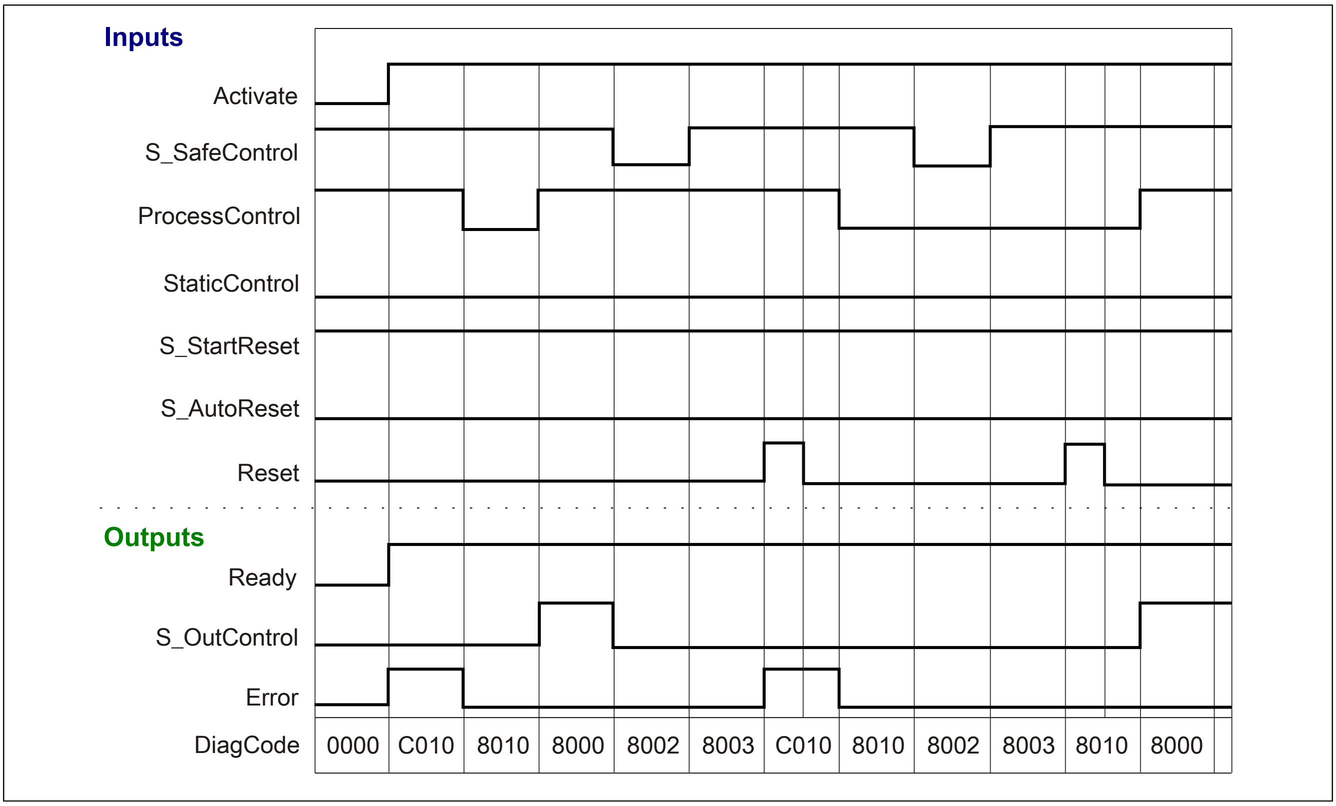


Figure 484: "SF_OutControl": Signal sequence diagram 2

6.6.16.7 Application examples

This chapter illustrates possible applications in which the function block can be used to implement enabling for a signal from the standard controller.

The following examples describe function block connections when the following applies after the safety request is revoked:

- No additional operational start is necessary to set enable output "S_OutControl" to TRUE ("StaticControl" = TRUE, see section 6.6.16.7.2 "Controlling a safe output without consideration of an additional operational start").
- An additional operational start is necessary to set enable output "S_OutControl" to TRUE ("StaticControl" = FALSE, see section 6.6.16.7.3 "Controlling a safe output with consideration of an additional operational start").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.16.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "OC_V1".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signal on "S_SafeControl", a rising edge on input parameter "Reset" is required in order for enable output "S_OutControl" to be enabled.

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the return of the safe input signal on "S_SafeControl". In addition to the safe input signal on "S_SafeControl", a rising edge on input parameter "Reset" is required in order for enable output "S_OutControl" to be enabled.

Input parameter "StaticControl" specifies the operating behavior of the function block when signal "S_SafeControl" returns if "ProcessControl" = TRUE. Input parameter "StaticControl" is connected to constant TRUE. Because of this, a rising edge on "ProcessControl" is not required to enable enable output "S_OutControl" after the return of the safe input signal on "S_SafeControl".

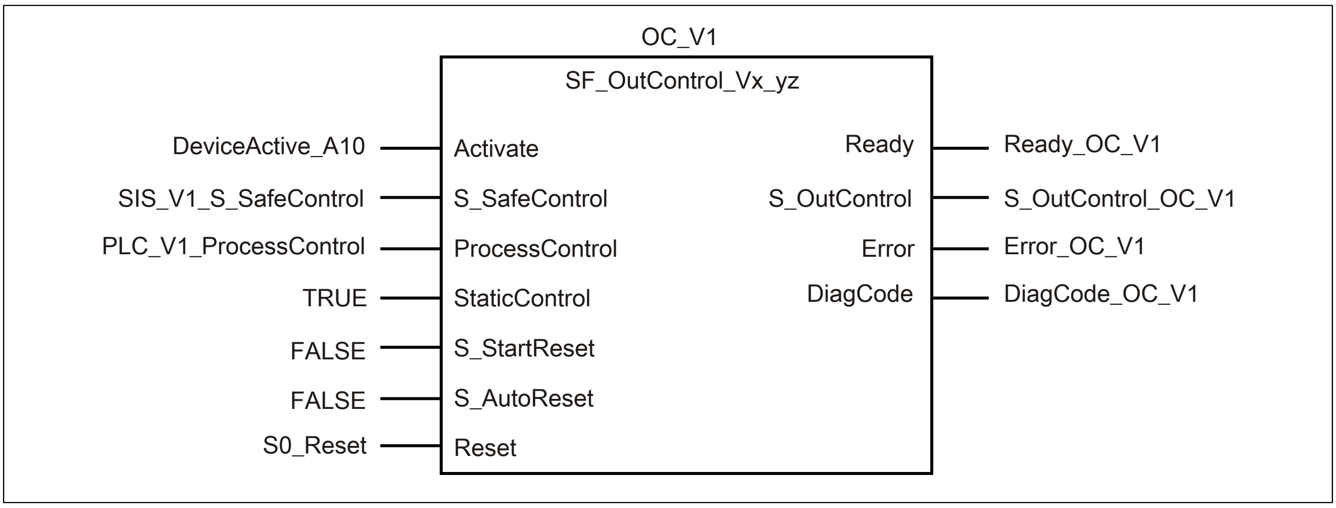


Figure 485: "SF_OutControl": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block.
SIS_V1_S_SafeControl	SAFEBOOL	Signal from the upstream safety function
PLC_V1_ProcessControl	BOOL	Signal of the operational start/stop from the standard controller
TRUE on "StaticControl"	BOOL	Sequence evaluation of signals "S_SafeControl" and "ProcessControl". Signal TRUE on "StaticControl" means that a rising edge on "ProcessControl" is not required to set enable output "S_OutControl" to TRUE after enabling input "S_SafeControl" if the remaining input signal combination is valid for this.
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after return of signal TRUE on the signal input
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 680: "SF_OutControl": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_OC_V1	BOOL	Function block active / Function block inactive. Message for further external processing.
S_OutControl_OC_V1	SAFEBOOL	Release signal. The release signal controls an output on a safe device.
Error_OC_V1	BOOL	Error message from function block for further external processing
DiagCode_OC_V1	WORD	Diagnostic message from function block for further external processing

Table 681: "SF_OutControl": Outputs connected to output parameters

6.6.16.7.2 Controlling a safe output without consideration of an additional operational start

This example illustrates function block connections when no additional operational start is necessary to set enable output "S_OutControl" to TRUE after revoking the safety request if "StaticControl" = TRUE.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.16.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "OC_V1".

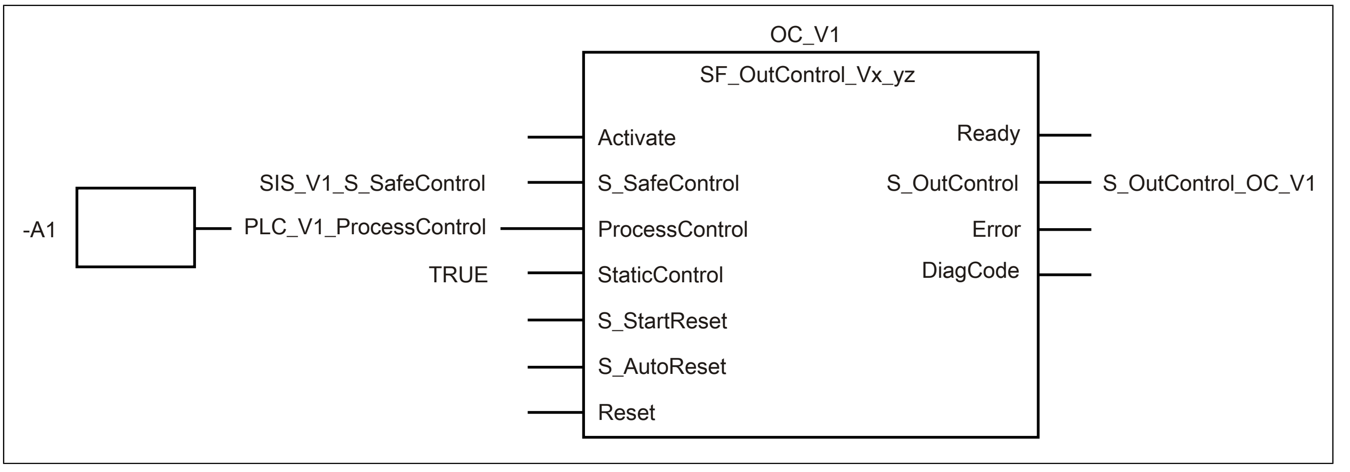


Figure 486: "SF_OutControl": Controlling a safe output without consideration of an additional operational start

List of equipment

-A1	Standard controller

Connected inputs and outputs

SIS_V1_S_SafeControl	Input on "S_SafeControl"
PLC_V1_ProcessControl	Input on "ProcessControl"
S_OutControl_OC_V1	Output on "S_OutControl"
TRUE	Constant on "StaticControl"

Description

In this example:

- The signal from standard controller "-A1" is connected to input "PLC_V1_ProcessControl".
- Input "PLC_V1_ProcessControl" is connected to input parameter "ProcessControl" for further processing.
- The signal from the safety controller is connected to input "SIS_V1_S_SafeControl".
- Input "SIS_V1_S_SafeControl" is connected to input parameter "S_SafeControl" for further processing.
- Output parameter "S_OutControl" is connected to output "S_OutControl_OC_V1".
- Output "S_OutControl_OC_V1" is used as an output signal for controlling a safe output. Outputs that cannot be read back can be directly controlled; outputs that can be read back must be controlled using function block "SF_EDM".

Behavior of the function block when "StaticControl" = TRUE

On a static TRUE on "ProcessControl" and switch from FALSE to TRUE on "S_SafeControl", a rising edge on "ProcessControl" is not required to set enable output "S_OutControl" to TRUE if the remaining input signal combination is valid for this.

6.6.16.7.3 Controlling a safe output with consideration of an additional operational start

This example illustrates function block connections when an additional operational start is necessary to set enable output "S_OutControl" to TRUE after revoking the safety request if "StaticControl" = FALSE.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.16.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "OC_V1".

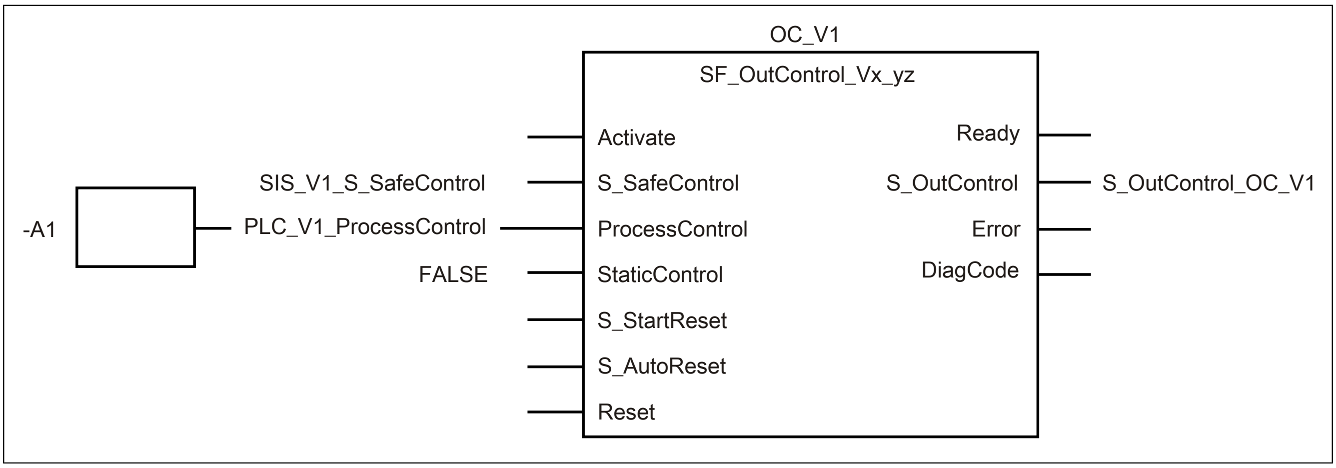


Figure 487: "SF_OutControl": Controlling a safe output with consideration of an additional operational start

List of equipment

-A1	Standard controller

Connected inputs and outputs

SIS_V1_S_SafeControl	Input on "S_SafeControl"
PLC_V1_ProcessControl	Input on "ProcessControl"
S_OutControl_OC_V1	Output on "S_OutControl"
FALSE	Constant on "StaticControl"

Description

In this example:

- The signal from standard controller "-A1" is connected to input "PLC_V1_ProcessControl".
- Input "PLC_V1_ProcessControl" is connected to input parameter "ProcessControl" for further processing.
- The signal from the safety controller is connected to input "SIS_V1_S_SafeControl".
- Input "SIS_V1_S_SafeControl" is connected to input parameter "S_SafeControl" for further processing.
- Output parameter "S_OutControl" is connected to output "S_OutControl_OC_V1".
- Output "S_OutControl_OC_V1" is used as an output signal for controlling a safe output. Outputs that cannot be read back can be directly controlled; outputs that can be read back must be controlled using function block "SF_EDM".

Behavior of the function block when "StaticControl" = FALSE

On a static TRUE on "ProcessControl" and switch from FALSE to TRUE on "S_SafeControl", a rising edge on "ProcessControl" is required to set enable output "S_OutControl" to TRUE if the remaining input signal combination is valid for this.

6.6.16.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 954-1	Stop function	The function block optionally supports the request for an additional operational stop during a stop function by triggering protective equipment or if the function block is disabled. A stop request by triggered protective equipment (signal on "S_SafeControl") and/or an operational stop (signal on "ProcessControl") sets "S_OutControl" to FALSE.
EN 954-1	Start and restart, fluctuations, loss and restoration of power sources	The function block optionally supports a start interlock after the following: • Cold restarting the safety controller • Enabling the function block • Return of the signal on "S_OutControl" The function block optionally supports the request for an additional operational stop during a stop function by triggering protective equipment or if the function block is disabled. A stop request by triggered protective equipment (signal on "S_SafeControl") and/or an operational stop (signal on "ProcessControl") sets "S_OutControl" to FALSE.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_AutoReset" = FALSE) • Enabling of the function block ("S_StartReset" = FALSE) • Signal change from FALSE to TRUE on "S_SafeControl" If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional operational start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 60204	Stop functions	This function block (release signal "S_OutControl") executes a category 0 stop.
EN 60204	Start	Release signal "S_OutControl" is only set to TRUE if the combination of input signals is valid for this.
EN 60204	Interlock / Sequential starting	The function block optionally supports the request for an additional operational stop during a stop function by triggering protective equipment or if the function block is disabled. A stop request by triggered protective equipment (signal on "S_SafeControl") and/or an operational stop (signal on "ProcessControl") sets "S_OutControl" to FALSE.

Table 682: "SF_OutControl": Implementation of requirements from standards

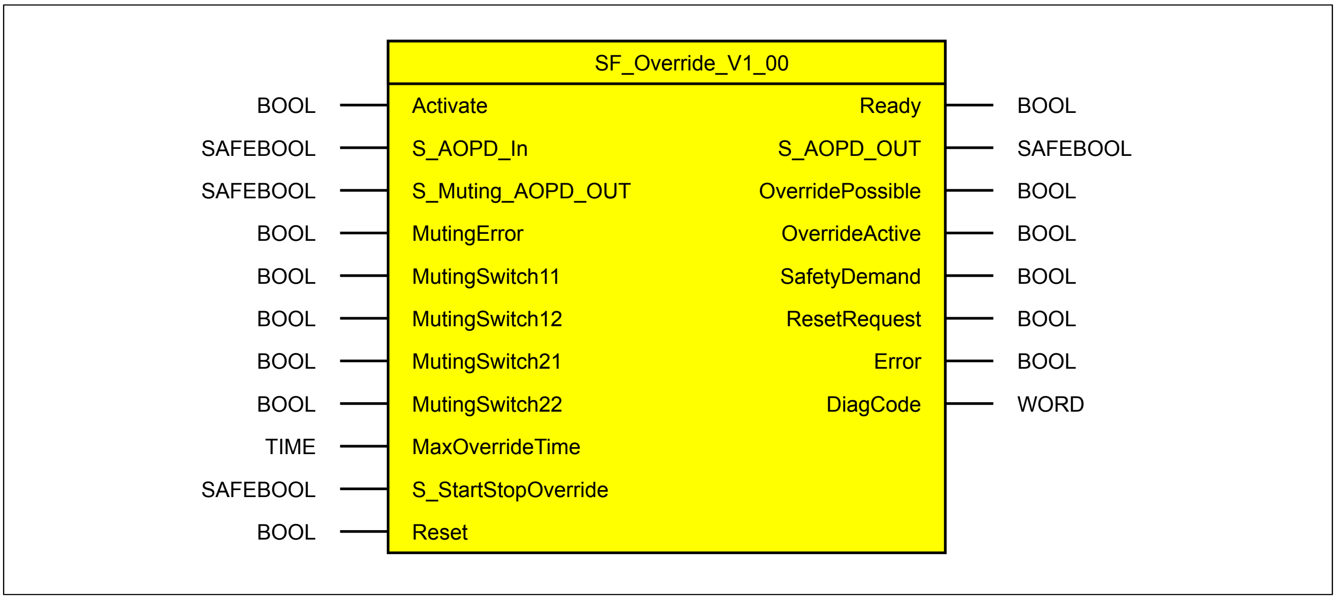
6.6.17 SF_Override

This function block makes it possible to free a transported product that is stuck in the muting area, for example due to an interrupted muting process.

Requirements for use

Function block supported in SafeDESIGNER V4.3.2 and later

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block ("Activate" = TRUE)
IN	S_AOPD_In	SAFEBOOL	Variable	FALSE	Signal input of the protective equipment (light curtain)
IN	S_Muting_AOPD_OUT	SAFEBOOL	Variable	FALSE	Signal input for signal "S_AOPD_Out" of the upstream muting function block
IN	MutingError	BOOL	Variable	FALSE	Signal input for signal "Error" of the upstream muting function block
IN	MutingSwitch11	BOOL	Variable	FALSE	State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g. "MutingSwitch11" for "SF_MutingPar")
IN	MutingSwitch12	BOOL	Variable	FALSE	State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g. "MutingSwitch12" for "SF_MutingPar")
IN	MutingSwitch21	BOOL	Variable	FALSE	State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g. "MutingSwitch21" for "SF_MutingPar"). This is not available when using muting function blocks that have 2 sensors (e.g. "SF_MutingPar_2Sensor").
IN	MutingSwitch22	BOOL	Variable	FALSE	State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g. "MutingSwitch22" for "SF_MutingPar"). This is not available when using muting function blocks that have 2 sensors (e.g. "SF_MutingPar_2Sensor").
IN	MaxOverrideTime	TIME	Constant	#0s	Specification of the maximum time for the entire override process
IN	S_StartStopOverride	SAFEBOOL	Variable	FALSE	Signal for starting and stopping the override function
IN	Reset	BOOL	Variable	FALSE	Resets error messages if the cause of error no longer exists
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled
OUT	S_AOPD_OUT	SAFEBOOL	Variable	FALSE	Function block release signal
OUT	OverridePossible	BOOL	Variable	FALSE	Indicates whether an override function is possible
OUT	OverrideActive	BOOL	Variable	FALSE	State of the override process
OUT	SafetyDemand	BOOL	Variable	FALSE	Indicates that the safety function is being requested
OUT	ResetRequest	BOOL	Variable	FALSE	Indicates that a reset is required on the function block
OUT	Error	BOOL	Variable	FALSE	Function block error message
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.6.17.1 Function description

Function block "SF_Override" implements the override function. This makes it possible to remove blockage that occurs in the safety zone during the muting process.

If the override function is enabled, a stop request for the protective equipment has no effect.

To be used properly, this function block must only be used in combination with upstream muting function blocks. These are responsible for the muting process. If an error occurs, the conveyor stops. Function block "SF_Override" is used to advance the product being transported out of the muting area. It is important to note that the muting process is canceled and the conveyor belt can be moved even though a muting error is present.

The results of output parameters "Error" and "DiagCode" from the upstream muting function block are not transferred to output parameters "Error" and "DiagCode" of function block "SF_Override".

Input parameter "MutingError" of function block "SF_Override" obtains this data from output parameter "Error" of the upstream muting function block.

The function block sets the override signal (output parameter "S_AOPD_OUT" of function block "SF_Override") to TRUE under the following conditions:

- The upstream muting function block returns an error.
- A static TRUE signal is present on input parameter "S_StartStopOverride" after a rising edge.
- The protective equipment (e.g. light curtain) and/or at least 1 muting sensor is damped.

The function block sets the override signal to FALSE under the following conditions:

- Signal FALSE is present on input parameter "S_StartStopOverride".
- The maximum time for the override process (input parameter "MaxOverrideTime") was exceeded.
- The protective equipment (e.g. light curtain) is not damped and all muting sensors are working properly.

Connection example

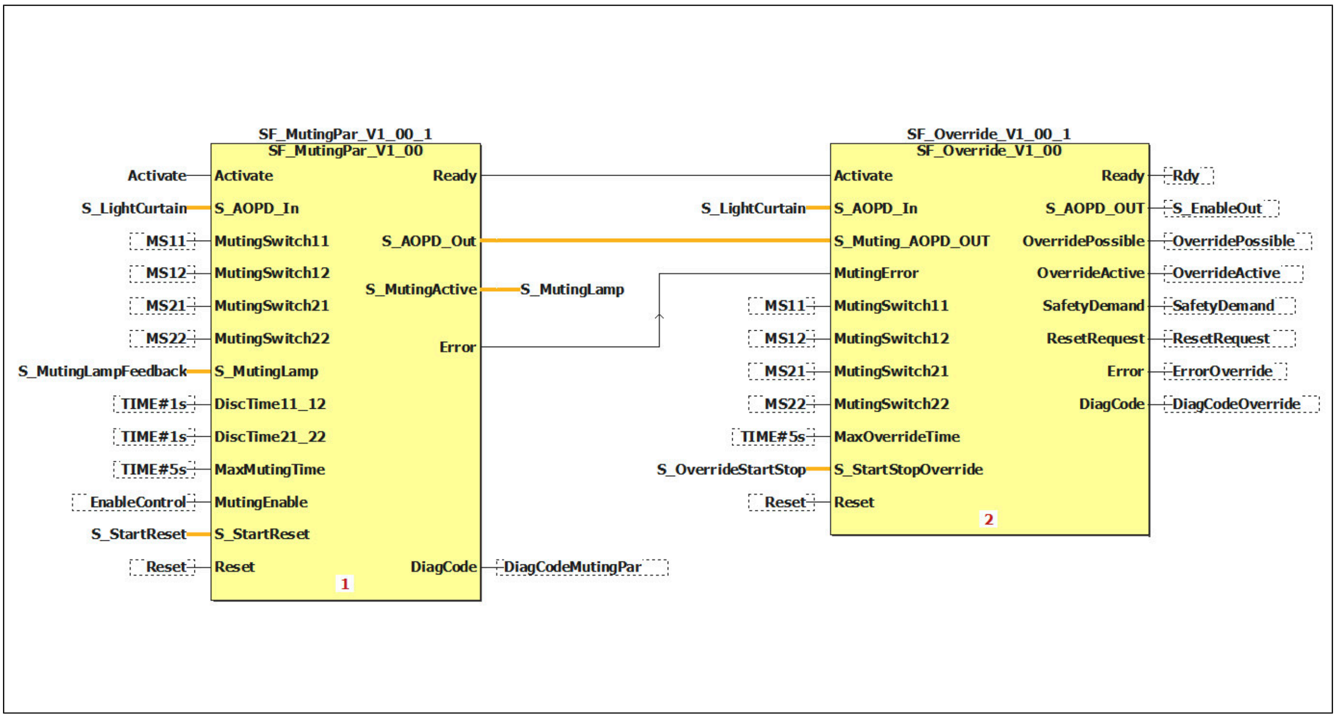


Figure 488: Example: Connecting "SF_MutingPar" with 4 sensors

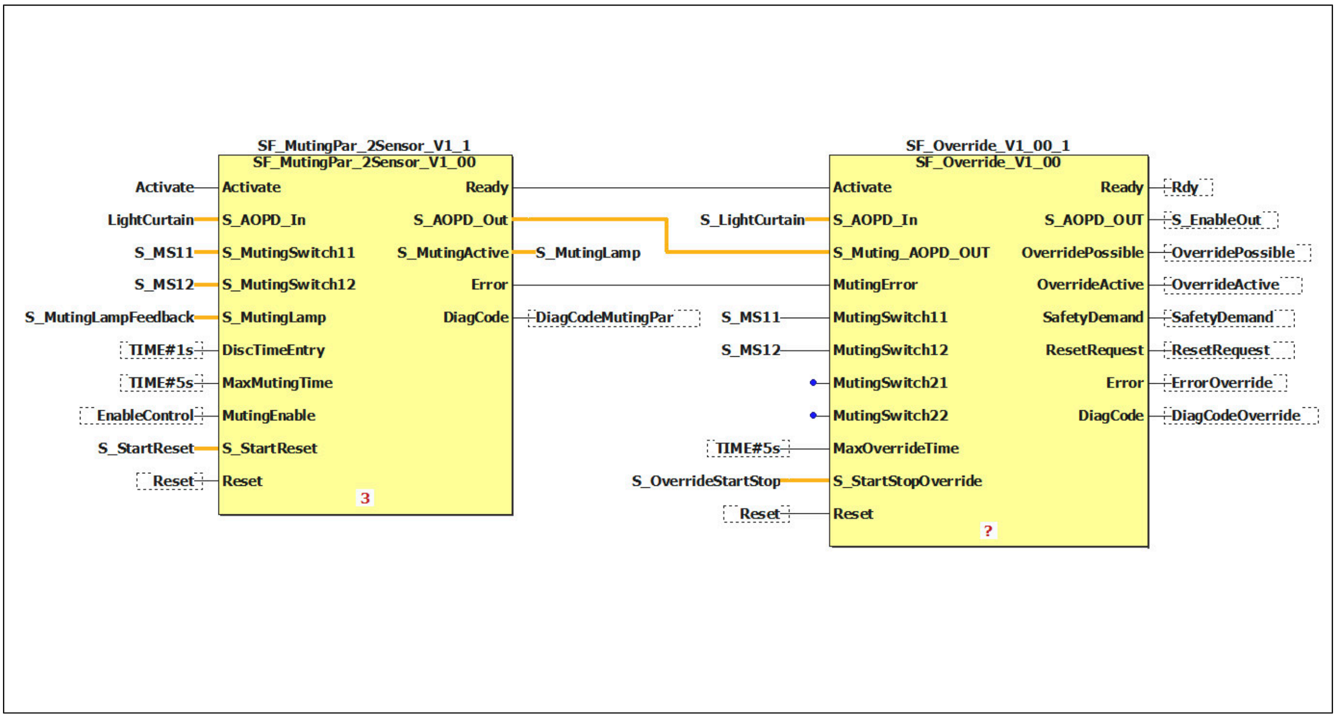


Figure 489: Example: Connecting "SF_MutingPar_2Sensor" with 2 sensors

6.6.17.2 Input parameters

Description of the function block input parameters.

6.6.17.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.17.2.2 S_AOPD_In

General function

- Signal input of the protective equipment (light curtain)

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
> **Connect this input parameter to the signal of a safe device that is connected to the protective equipment (e.g. light curtain) in the muting application over 1 or 2 channels. Input parameter "S_AOPD_In" is then controlled using this signal.**

Function description

The function block evaluates the state of the connected protective equipment using the signal connected on input parameter "S_AOPD_In".

Regardless of whether the protective equipment is connected to the safe device over 1 or 2 channels, "S_AOPD_In" is only connected to one signal.

If protective equipment is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent", while monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_AOPD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected protective equipment is not damped.

FALSE

The connected protective equipment is damped.

If muting is active, "S_AOPD_Out" does not initiate the safe state.
If muting is inactive, "S_AOPD_Out" initiates the safe state.

6.6.17.2.3 S_Muting_AOPD_OUT

General function

- Signal input for signal "S_AOPD_Out" of the upstream muting function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The function block evaluates the state of the connected muting function block (output parameter "S_AOPD_Out") via the signal connected on input parameter "S_Muting_AOPD_OUT".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream muting function block (output parameter "S_AOPD_Out") returns signal TRUE.

FALSE

The upstream muting function block (output parameter "S_AOPD_Out") returns signal FALSE.

6.6.17.2.4 MutingError

General function

- Signal input for signal "Error" of the upstream muting function block

Data type

- BOOL

Connection

- Variable

Function description

The function block evaluates the error state of the connected muting function block (output parameter "Error") via the signal connected on input parameter "MutingError".

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The upstream muting function block (output parameter "Error") returns an error.

FALSE

The upstream muting function block (output parameter "Error") does not return an error.

6.6.17.2.5 MutingSwitch11

General function

- State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g."MutingSwitch11" for "SF_MutingPar")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_11" in the muting application. Input parameter "MutingSwitch11" is then controlled using this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch11".

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.17.2.6 MutingSwitch12

General function

- State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g."MutingSwitch12" for "SF_MutingPar")

Data type

- BOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a device that is connected to muting sensor "MS_12" in the muting application. Input parameter "MutingSwitch12" is then controlled using this signal.

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch12".

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.17.2.7 MutingSwitch21

General function

- State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g."MutingSwitch21" for "SF_MutingPar")

Data type

- BOOL

Connection

- Variable

> ## Information:
> **This input parameter is not available when using muting function blocks that have 2 sensors (e.g."SF_MutingPar_2Sensor").**

> ## Information:
> **Connect this input parameter to the signal of a device that is connected to muting sensor "MS_21" in the muting application. Input parameter "MutingSwitch21" is then controlled using this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch21".

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.17.2.8 MutingSwitch22

General function

- State of the muting sensor signal connected to the input parameter of the upstream muting function block (e.g."MutingSwitch22" for "SF_MutingPar")

Data type

- BOOL

Connection

- Variable

> **Information:**
>
> **This input parameter is not available when using muting function blocks that have 2 sensors (e.g."SF_MutingPar_2Sensor").**

> **Information:**
>
> **Connect this input parameter to the signal of a device that is connected to muting sensor "MS_22" in the muting application. Input parameter "MutingSwitch22" is then controlled using this signal.**

Function description

The function block evaluates the state of the connected muting sensor via the signal connected on input parameter "MutingSwitch22".

TRUE

The connected muting sensor is damped.

FALSE

The connected muting sensor is not damped.

6.6.17.2.9 MaxOverrideTime

General function

- Specification of the maximum time for the entire override process

Data type

- TIME

Connection

- Constant

Function description

This input parameter specifies the maximum time for the entire override process. This time starts if the starting conditions for the override process are met.

The time is stopped if all muting sensors are no longer damped.

You must define and validate the time value for input parameter "MaxOverrideTime" based on your application and risk analysis.

Range of values: 0 to 10 minutes

6.6.17.2.10 S_StartStopOverride

General function

- Signal for starting and stopping the override function

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter enables the override function. A rising edge on this input parameter is required to start the override function. Input parameter "MaxOverrideTime" also begins counting.

TRUE

The override process starts if all conditions for the override function are met. The time for "MaxOverrideTime" is also started simultaneously.

FALSE

The override process is stopped. The time for "MaxOverrideTime" continues to run until the muting process is completed.

6.6.17.2.11 Reset

General function

- Input parameter for resetting error messages if the error has been corrected

Data type

- BOOL

Depending on the safety requirements, you must use data type SAFEBOOL or BOOL for the connection. With a SAFEBOOL connection, you avoid unexpected startups that may result from errors in the standard (non-safe) system.
You can reduce the risk of unexpected startup with further measures such as an additional function stop.

Connection

- Variable

Function description

The function block internally monitors edge transitions for this input parameter. The function is only executed on a rising edge of input parameter "Reset". A continuing static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or to reset errors detected by the function block once the cause of error is no longer present.

6.6.17.3 Output parameters

Description of the function block output parameters.

6.6.17.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.17.3.2 S_AOPD_OUT

General function

- Function block release signal

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this output parameter to the safety application in such a way that the safety application takes on and maintains the safe state if signal FALSE is output.

Function description

The release signal returns the state of the muting process or override signal.

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- And: The safeguarded area of the protective equipment (light curtain beams) is not damped.
- Or: The muting process is active.
- Or: The override process is active.
- And: The function block did not detect an error.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

Possible causes:

- The function block is not enabled ("Activate" = FALSE).
- And: The safeguarded area of the protective equipment is damped.
- And: The muting process is not active.
- Or: The override process is not active.
- Or: The function block detected an error.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_AOPD_OUT" is only set to TRUE if input "S_AOPD_In" indicates state TRUE and a reset has been carried out (no start interlock active).

6.6.17.3.3 OverridePossible

General function

- Indicates whether an override function is possible

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether an override function is possible.

TRUE

An override function is possible.

FALSE

An override function is not possible.

6.6.17.3.4 OverrideActive

General function

- State of the override process

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether an override process is active.

TRUE

The override process is active.

FALSE

The override process is not active.

6.6.17.3.5 SafetyDemand

General function

- Indicates that the safety function is being requested

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending safety function request. You must perform a corresponding action to exit this state (see table in section "Status numbers").

TRUE

The enabled function block detected a request for the safety function. During the active override process, "S_StartStopOverride" was changed from TRUE to FALSE. "MaxOverrideTime" continues running even on signal TRUE.

FALSE

The function block is not enabled, or the enabled function block did not detect a safety function request.

6.6.17.3.6 ResetRequest

General function

- Indicates that a reset is required on the function block

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates the need to perform acknowledgment on input parameter "Reset" (see table in section "Status numbers").

TRUE

The enabled function block detected an error and acknowledgment is required (rising edge on "Reset").

FALSE

The function block is not enabled, or the enabled function block has not detected an error.

Output parameter "DiagCode" indicates the state.

6.6.17.3.7 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

To exit an error state ("Error" = TRUE), you must set input parameter "Reset" to FALSE if there is a static TRUE signal on "Reset".
In other error states (see table in section "Status numbers"), you must switch input parameter "Reset" from FALSE to TRUE.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.17.3.8 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.17.4 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.17.4.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by the function block or compiler. This is not always possible in the event of connection errors, however.

It is not possible for the function block to check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed. A static TRUE signal on input parameter "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters should have been connected but were not.

It is therefore important to note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.17.4.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.17.4.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.17.4.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

Danger!

You are responsible for the functional testing of protective equipment.
You must therefore validate the protective equipment!

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.17.5 Status numbers

Errors

Code (hex)	Description	Corrective measures
C011	The function block detected a static TRUE signal on "Reset".	Check the control device that controls input parameter "Reset" and the corresponding wiring.
C410	The maximum time for the override process was exceeded.	Check the protective equipment and corresponding parameters.

Table 683: "SF_Override": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The override process is active and time monitoring is running.	No corrective measures are required.
8002	Safety request. The field being monitored was interrupted, the muting function and/or override function is not active and time monitoring for "MaxOverrideTime" is reset.	Check the protective equipment (light curtain).
8012	The override function is not possible. The upstream function block returns an error, the protective equipment (e.g. light curtain) is not damped and the muting sensors are not damped.	Check the upstream function blocks.
8022	An override function is possible. The upstream function block returns an error and the protective equipment (e.g. light curtain) and/or at least 1 muting sensor is damped.	Check the protective equipment (light curtain) and sensors. Enable "S_StartStopOverride".
8100	The protective equipment is not damped. "S_AOPD_Out" of the upstream function block indicates signal TRUE.	No corrective measures are required.
8832	The override process is interrupted. The request signal was set to FALSE during the override process. Time monitoring continues to run.	Check the override process.

Table 684: "SF_Override": Diagnostic codes

6.6.17.6 Signal sequence diagram of the function block

Note that not all temporary intermediate states are shown in the signal sequence diagram. This diagram only shows typical combinations of input signals. Other signal combinations are possible.

"DiagCode" values in the following image are specified in hexadecimal format.

Signal sequence diagram

Startup, normal operation

Output parameters "SafetyDemand" and "ResetRequest" are not shown in the signal sequence diagram.

This diagram shows the functionality of function block "SF_Override" when combined with sequential muting ("SF_MutingSeq"). This is evident at the transition of the muting inputs in state "8000" and results from the movement of the object in the muted area.

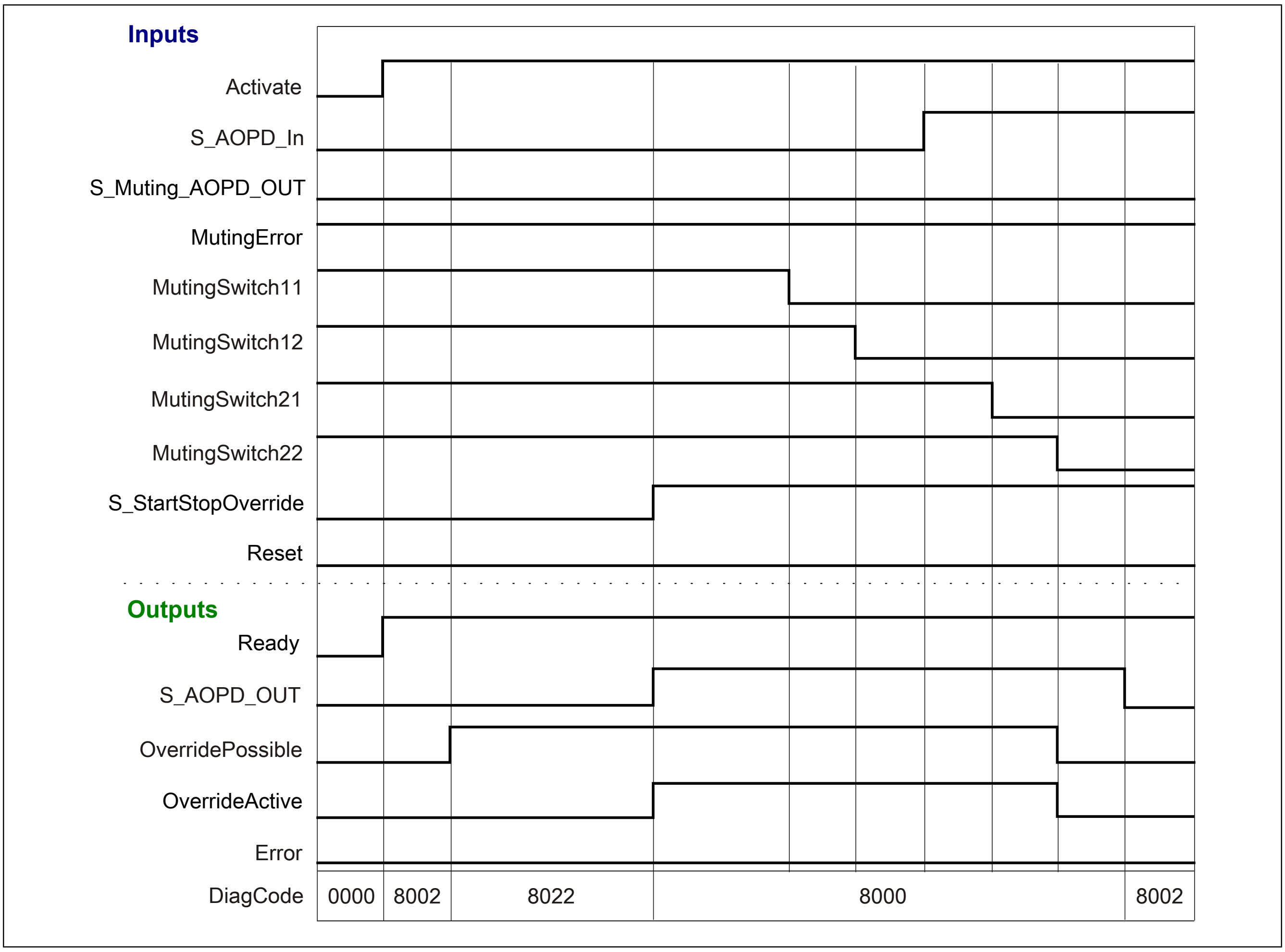


Figure 490: "SF_Override": Signal sequence diagram

6.6.18 SF_SafetyRequest

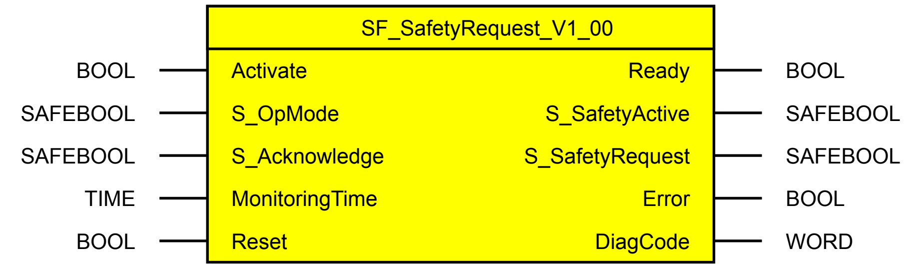


Figure 491: Function block "SF_SafetyRequest"

6.6.18.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_OpMode	SAFEBOOL	Variable	State	FALSE	Signal input. Request to the connected safe peripheral to execute a safety function
S_Acknowledge	SAFEBOOL	Variable	State	FALSE	Feedback signal from the connected safe peripheral
MonitoringTime	TIME	Constant	State	#0ms	Specification for the monitoring time between the request for and feedback from a safety function
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 685: "SF_SafetyRequest": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_SafetyActive	SAFEBOOL	Variable	State	FALSE	Status message from the connected safe peripheral
S_SafetyRequest	SAFEBOOL	Variable	State	FALSE	Request to the connected safe peripheral to execute a safety function
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 686: "SF_SafetyRequest": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 687: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.18.2 Function

Function block "SF_SafetyRequest" is used to support a safety function request function in an application (e.g. safe stop, safely reduced speed).

This function block serves as an interface between the safety application / safety controller and the connected safe peripheral (e.g. safe drive). The safety function for the safe peripheral connected to the function block can be requested by the function block from the program on the safety controller. The function block monitors the response to the request for a safety function via the feedback signal of the safe peripheral (input parameter "S_Acknowledge").

The data exchange for requesting a safe operating mode and feedback about the current operating mode is handled by an I/O coupling between the safety application and safe peripheral. The I/O signals are connected to the function block via inputs and outputs.

The function block outputs the state of the connected safe peripheral via a binary input parameter for further processing in the safety application.

Danger!

Note that the function block DOES NOT execute the safety function of the connected safe peripheral. The safe peripheral executes the safety function itself independently of the function block. The function block only requests the safety function and confirms that it has received feedback about the active safe state from the safe peripheral.

Use suitable measures to ensure that no hazard can arise from the safe peripheral when the safe peripheral executes the safety function!

The function block receives via input "S_OpMode" the request from the upstream safety application that the connected safe peripheral should execute or not execute a safety function. The function block forwards this request to the connected safe peripheral via output "S_SafetyRequest".

- When the connected safe peripheral receives the request to execute a safe operating mode:
  Signal FALSE on "S_OpMode" sets "S_SafetyRequest" to FALSE when the function block is active. If the function block receives feedback (signal TRUE) on input "S_Acknowledge" from the safe peripheral within the specified time (input parameter "MonitoringTime") that the peripheral is executing the safety function, then the function block generates confirmation (signal TRUE) on output parameter "S_SafetyActive". Otherwise, the function block outputs an error message on output parameter "DiagCode".

- When the connected safe peripheral receives the request to not execute a safe operating mode:
  Signal TRUE on "S_OpMode" sets "S_SafetyRequest" to TRUE and "S_SafetyActive" to FALSE when the function block is active.

6.6.18.2.1 Start interlock

A start interlock is active after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

You can reset a start interlock that was enabled after a detected error with a rising edge on "Reset".

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.18.2.2 Start interlock after cold restart of safety controller

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.18.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.18.3.1 Specified time frame on "MonitoringTime" not dimensioned correctly

If the feedback signal (input parameter "S_Acknowledge") of the connected safe peripheral is not present after the safety function is requested within this time frame, the function block detects an error. The function block does not detect a time frame that is too large as an error.

Possible causes:

- Programming error (user error)
- Time value calculated incorrectly (user error)

6.6.18.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.18.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.18.3.4 Invalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block.

6.6.18.3.5 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.18.3.6 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.18.4 Input parameters

6.6.18.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to support a start interlock after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports a start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.18.4.2 S_OpMode

General function

- Signal input. Request to the connected safe peripheral to execute a safety function.

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to the upstream safety application (e.g. protective equipment, mode selector switch or emergency switch-off control device) over 1 or 2 channels. Input parameter "S_OpMode" is then controlled via this signal.

Function description

The signal connected to input parameter "S_OpMode" is processed by the function block.

The signal input processes the state of the upstream safety application.

Regardless whether the operating element is connected to the safe device over 1 or 2 channels, "S_OpMode" is only connected to one signal.

If an operating element is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_OpMode". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_OpMode". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The safety function of the connected safe peripheral is not requested.

FALSE

The safety function of the connected safe peripheral is requested.

6.6.18.4.3 S_Acknowledge

General function

- Feedback signal from the connected safe peripheral

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to the upstream safety application (e.g. protective equipment, mode selector switch or emergency switch-off control device) over 1 or 2 channels. Input parameter "S_Acknowledge" is then controlled via this signal.

Function description

The signal connected to input parameter "S_Acknowledge" is processed by the function block.

Regardless whether the operating element is connected to the safe device over 1 or 2 channels, "S_Acknowledge" is only connected to one signal.

If an operating element is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_Acknowledge". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_Acknowledge". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The connected safe peripheral is reporting execution of the safety function.

FALSE

No feedback from the connected safe peripheral about execution of the safety function is available.

6.6.18.4.4 MonitoringTime

General function

- Specification for the monitoring time between the request for and feedback from a safety function

Data type

- TIME

Connection

- Constant

Function description

Specify the time interval on this input parameter after which feedback from the connected safe peripheral about execution of the safety function must be available. If this time interval is exceeded, the function block will detect an error.

You must define and validate the time value for input parameter "MonitoringTime" based on your application and risk analysis.

6.6.18.4.5 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting manual reset of a start interlock

Data type

- BOOL

> **Danger!**
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.18.5 Output parameters

6.6.18.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.18.5.2 S_SafetyActive

General function

- Status message from the connected safe peripheral

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this output parameter typically to function block "SF_EnableSwitch" to initiate the requested safe operating mode (e.g. safely reduced speed).

Function description

This output parameter indicates the state of the connected safe peripheral.

TRUE

The safe peripheral connected to the function block via the safe input device reports the safe state (e.g. safe stop, safely reduced speed, safely reduced torque). The safe peripheral controls this safe state itself independently of the function block.

FALSE

The safe peripheral connected to the function block via the safe input device does not report the safe state.

6.6.18.5.3 S_SafetyRequest

General function

- Request to the connected safe peripheral to execute a safety function

Data type

- SAFEBOOL

Connection

- Variable

> **Information:**
>
> **Connect this output parameter to a safe 1-channel or 2-channel output of a safe device to control the downstream safe peripheral.**

Function description

This output parameter controls the request to the connected safe peripheral to execute a safety function.

Regardless whether the downstream safe peripheral is connected to the safe device over 1 or 2 channels, the outputs of the safe device are controlled by only one signal from "S_SafetyRequest".

If the downstream safe peripheral is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device.

TRUE

Execution of the safety function is requested.

FALSE

Execution of the safety function is not requested.

6.6.18.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.18.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.18.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set "S_SafetyActive" to FALSE. The safe state is requested by signal FALSE on "S_OpMode". "S_SafetyRequest" outputs this request. To forward the request for the safe state to the peripheral, "S_SafetyRequest" must be connected to the peripheral. The connected peripheral feedback signal connected on "S_Acknowledge" was set to TRUE by the peripheral.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	The function block has been enabled. The function block's start interlock is active.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The safe state of the connected peripheral is not requested on "S_OpMode". The signal of the peripheral connected on "S_Acknowledge" is not reporting the safe state.	Intended event: • No corrective measures are required. Unintended event: • Set input parameter "S_OpMode" to FALSE to initiate the safe operating mode of the peripheral.
8003	The safe state is requested by signal FALSE on "S_OpMode". "S_SafetyRequest" outputs this request. To forward the request for the safe state to the peripheral, "S_SafetyRequest" must be connected to the peripheral. The connected peripheral feedback signal connected on "S_Acknowledge" was not yet set to TRUE by the peripheral. The function block monitors the time frame (value on "MonitoringTime") between the request for the safe state and feedback of the safe state. If this time is exceeded, the function block treats this an error.	Intended event: • No corrective measures are required.
8005	After acknowledgment of an error message, "S_OpMode" = FALSE.	Set "S_OpMode" to TRUE.
8012	The safe state of the connected peripheral is not requested. The connected peripheral feedback signal connected on "S_Acknowledge" is reporting the safe state, however.	• Check the safety functions. • Check the safety program. • Check the safe peripheral.
C002	The value on "S_Acknowledge" changed after the safe state was reported by the connected peripheral on "S_Acknowledge".	• Check the safety functions. • Check the safety program. • Check the safe peripheral.
C003	The safe state of the connected peripheral was requested on "S_OpMode". The connected peripheral feedback signal connected on "S_Acknowledge" did not respond within the specified time on "MonitoringTime".	• Check the safety functions. • Check the safety program. • Check the safe peripheral.
C004	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C005	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C006	The function block detected a static reset signal on "Reset".	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.

Table 688: "SF_SafetyRequest": Diagnostic codes

6.6.18.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

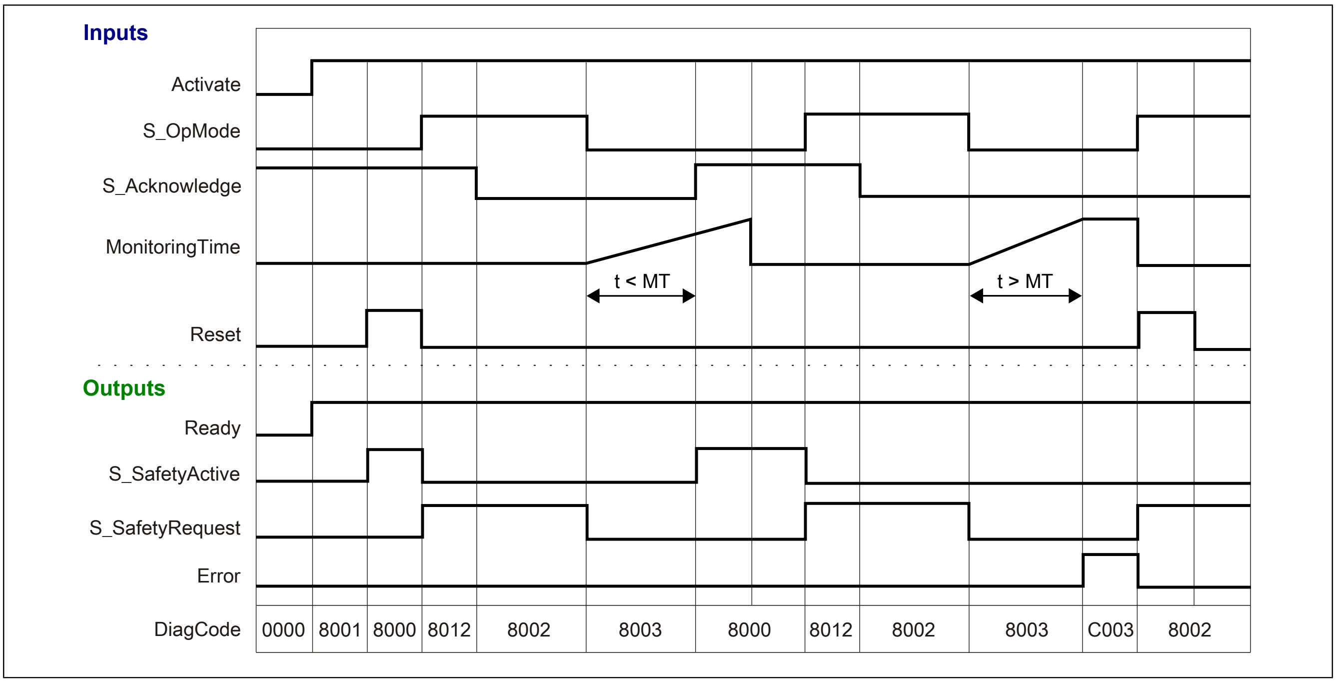


Figure 492: "SF_SafetyRequest": Signal sequence diagram

Key:

t	Time value that specifies whether feedback of the safe state is present
MT	Specified time value on input parameter "MonitoringTime" for feedback of the safe state of the connected peripheral

6.6.18.7 Application example

This chapter illustrates a possible application in which the function block can be used to control the connected safe peripheral to request the execution of a safety function. The function block sends the request to a safe drive to execute a safe operating mode.

The following example illustrates the connection of the function block when controlling using the signal from a feedback output of a safe drive connected over 2 channels (see section 6.6.18.7.2 "Safe drive, connected over 2 channels").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.18.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "SR_M5".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

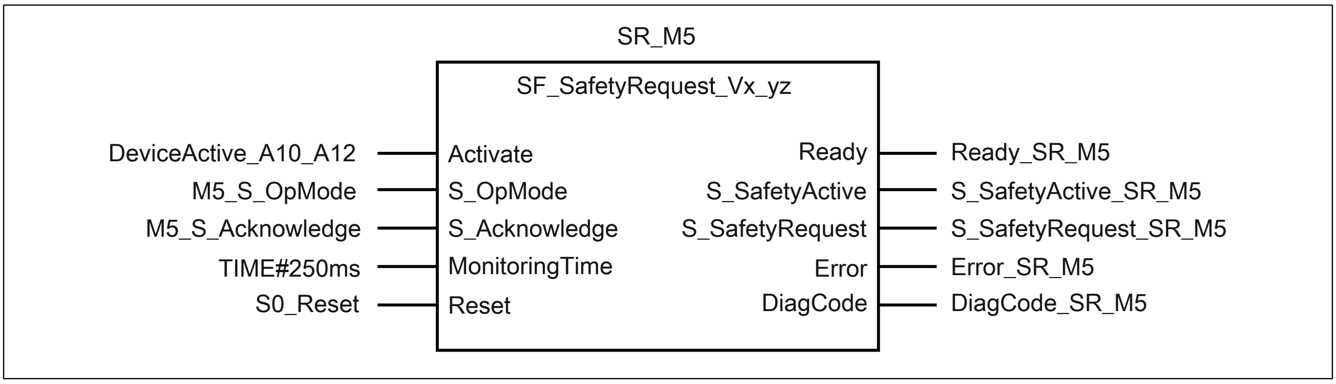


Figure 493: "SF_SafetyRequest": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input/output devices to which the safe drive is connected.
M5_S_OpMode	SAFEBOOL	Request for the safe operating mode. This signal comes from the program on the safety controller. The signal requests enabling of the safe operating mode.
M5_S_Acknowledge	SAFEBOOL	Feedback from the safe drive that the safe operating mode is being executed. This input is connected to a signal of the safe drive that specifies whether the safe drive is executing the safe operating mode.
TIME#250ms on "MonitoringTime"	TIME	Specification of the maximum response time of the safety application and safe drive for switching from the standard operating mode to the safe operating mode and reporting this.
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 689: "SF_SafetyRequest": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_SR_M5	BOOL	Function block active / Function block inactive. Message for further external processing.
S_SafetyActive_SR_M5	SAFEBOOL	Status message for enabling signals from the standard controller and safety controller. These signals from the standard controller and safety controller control the monitored safe outputs in the safe operating mode. You implement the stop function by making corresponding connections in the safe program. Since this is only an example, these connections are not shown here.
S_SafetyRequest_SR_M5	SAFEBOOL	Request to the connected safe peripheral to execute a safety function
Error_SR_M5	BOOL	Error message from function block for further external processing
DiagCode_SR_M5	WORD	Diagnostic message from function block for further external processing

Table 690: "SF_SafetyRequest": Outputs connected to output parameters

6.6.18.7.2 Safe drive, connected over 2 channels

This example illustrates the connection of the function block when controlling using the signal from a feedback output of a safe drive connected over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.18.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "SR_M5".

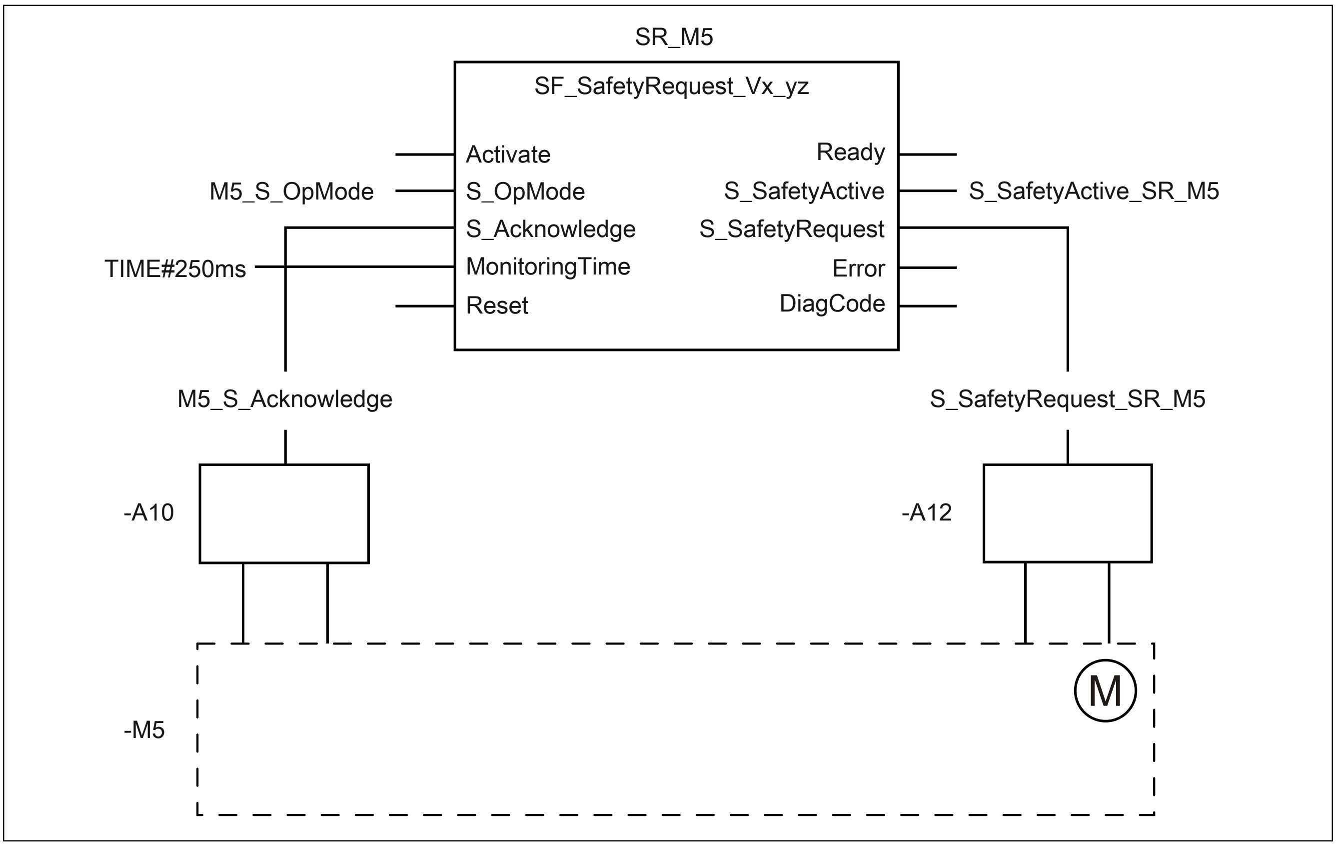


Figure 494: "SF_SafetyRequest": Safe drive, connected over 2 channels

List of equipment

-M5	Safe drive
-A10	2-channel safe input of a safe input device
-A12	2-channel safe output of a safe output device

Connected inputs and outputs

M5_S_OpMode	Input on "S_OpMode"
M5_S_Acknowledge	Input on "S_Acknowledge"
S_SafetyActive_SR_M5	Output on "S_SafetyActive"
S_SafetyRequest_SR_M5	Output on "S_SafetyRequest"

Description

In this example:

- Input "M5_S_OpMode" from the program on the safety controller is connected to function block input parameter "S_OpMode" to request the safe operating mode of the connected safe drive.
- The feedback signal of the safe drive (this signal confirms active safe operation) from safe input device "-A10" is connected to input "M5_S_Acknowledge". This input is connected to function block input parameter "S_Acknowledge" for further processing.
- The maximum response time for requesting a safe operating mode is specified on input parameter "MonitoringTime" using a constant.
- Output "S_SafetyRequest_SR_M5" is connected to function block output parameter "S_SafetyRequest". Output "S_SafetyRequest_SR_M5" is connected via safe output device "-A12" to a safe input of the safe drive that is controlling the corresponding safety request in the drive. This output is used to request the safe operating mode.
- Output parameter "S_SafetyActive" is connected to output "S_SafetyActive_SR_M5".
- Output "S_SafetyActive_SR_M5" is used as a status message for enabling signals from the standard controller and safety controller. These signals from the standard controller and safety controller control the monitored safe outputs.

6.6.18.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block supports a start interlock after the following: • Cold restarting the safety controller • Enabling the function block If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Category B to 4	1- or 2-channel switching must be designed based on the category.
EN 60204	Stop functions	This function block (output signal "S_SafetyActive") executes a category 0 stop.

Table 691: "SF_SafetyRequest": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.19 SF_TestableSafetySensor

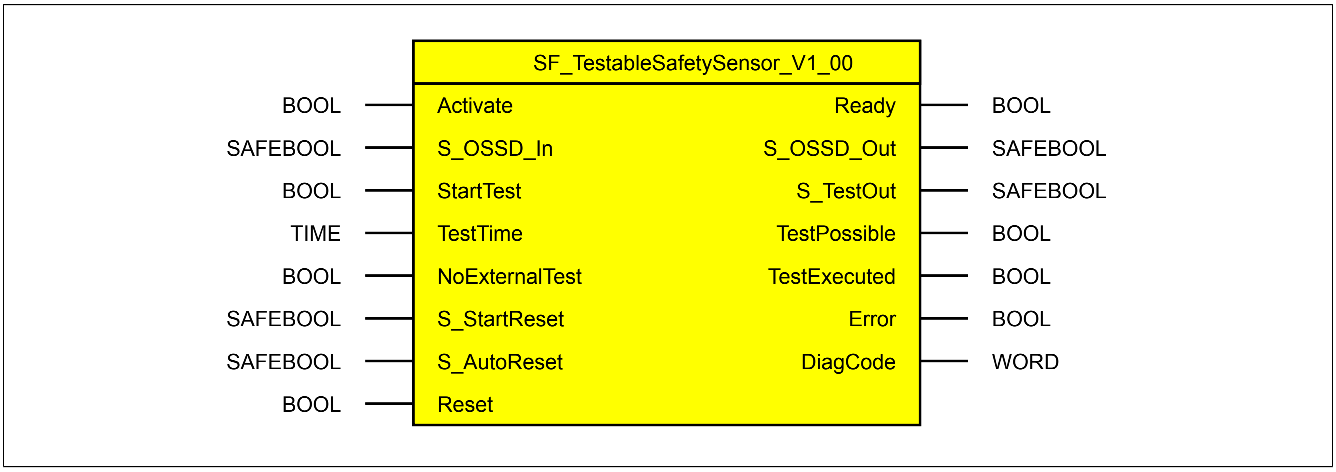


Figure 495: Function block "SF_TestableSafetySensor"

6.6.19.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_OSSD_In	SAFEBOOL	Variable	State	FALSE	Signal input for the state of the connected safety sensor
StartTest	BOOL	Variable	Edge	FALSE	Starting signal for the sensor test
TestTime	TIME	Constant	State	#10ms	Specification of the maximum response time of the sensor signal on "S_OSSD_In" during the sensor test
NoExternalTest	BOOL	Constant	State	FALSE	Specification for an external manual sensor test
S_StartReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller
S_AutoReset	SAFEBOOL	Variable/ Constant	State	FALSE	Specification of the start interlock after a signal change from FALSE to TRUE on "S_OSSD_In"
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 692: "SF_TestableSafetySensor": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_OSSD_Out	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
S_TestOut	SAFEBOOL	Variable	State	TRUE	Control of the sensor test input
TestPossible	BOOL	Variable	State	FALSE	Feedback: Automatic sensor test possible / not possible
TestExecuted	BOOL	Variable	Edge	FALSE	Message: Automatic sensor test executed / not executed / active / faulty
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 693: "SF_TestableSafetySensor": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)
TIME	Time	32	Time

Table 694: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.19.2 Function

Function block "SF_TestableSafetySensor" is used to test the functionality of optical/electronic protective equipment – "safety sensor" for short in the following (e.g. of a light curtain) – if the protective equipment (type 2 ESPE) supports the test function.

In addition, the function block monitors the state of the safety sensor and indicates it on function block enable output "S_OSSD_Out". If signal TRUE of the safety sensor returns after the safety function is triggered, the function block optionally supports a start interlock on the enable output. Another optional start interlock is enabled when the function block is enabled and on a cold restart of the safety controller.

If a piece of optical/electronic protective equipment connected to the function block does not behave in the application according to the defined testing algorithm, the function block sets its enable output "S_OSSD_Out" to the safe state (FALSE). If the connected optical/electronic protective equipment behaves according to the defined testing algorithm during the test procedure, the signal on enable output "S_OSSD_Out" remains TRUE.

Some of the things detected by this function block during the test function:

- Loss of sensitive properties of the optical/electronic protective equipment
- If the specified response time of the protective equipment is exceeded
- Irregular static TRUE signal of protective equipment on "S_OSSD_In"

Type 2 electro-sensitive protective equipment (EPSE) have the ability to be tested periodically for dangerous faults (e.g. loss of sensor capabilities, specified response time exceeded). The test signal simulates enabling of the safety sensor. When the test algorithm is executed, the sensor is shut down for maximum 150 ms. The test is meant to verify that each light beam of the sensor is working according to specifications. Set up separate/external functions to ensure that testing is performed at suitable intervals. Only enable inputs "S_StartReset" and "S_AutoReset" if you have ensured that no hazardous situation can occur when the safety controller is started.

Information:

You are responsible for controlling "StartTest" and therefore performing testing periodically.

Evaluate output parameter "TestPossible" before starting the test. You are responsible for performing periodic testing according to the results of your risk analysis by setting "StartTest" from FALSE to TRUE and evaluating the result on the function block.

6.6.19.2.1 Signal TRUE after triggering the safety function

If the triggered safety function is set back to TRUE in the application, the function block optionally (see start interlock) ensures within the safety control system that the release signal is not set to TRUE solely by this unlocking. Additional manual intervention on input parameter "Reset" is required for this (see start interlock).

6.6.19.2.2 Test mode sequence

Note that 2 time-delayed timers run through specified time "TestTime" during the test mode.

Test mode consists of the following sequences:

1. Test start
   - "StartTest" switches from FALSE to TRUE.
   - Monitoring time 1 is started.
   - "S_TestOut" switches from TRUE to FALSE.
2. Test procedure
   - "S_OSSD_In" switches from TRUE to FALSE.
   - "S_TestOut" switches from FALSE to TRUE.
   - Monitoring time 1 is stopped.
   - Monitoring time 2 is started.
3. Test end
   - "S_OSSD_In" switches from FALSE to TRUE.
   - Monitoring time 2 is stopped.

6.6.19.2.3 Start interlock (optional)

For optional support of the start interlock, this feature must be specified on the corresponding input parameters ("S_StartReset" / "S_AutoReset").

A start interlock is active after a signal change from FALSE to TRUE on the safe input and/or after a cold restart of the safety controller and/or after the function block is enabled. If start interlock is active, the safety-related output signal is in a safe state.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.19.2.4 Start interlock after cold restart of safety controller (optional)

Support for a start interlock must be defined accordingly on input parameter "S_StartReset" after the function block has been enabled.

After a cold restart of the safety controller, the function block supports a defined startup or restart of the application within the safe control system (see start interlock). This is achieved by the function block controlling the release signal as needed.

The start interlock only becomes inactive after manual intervention (rising edge) on input parameter "Reset" if the input signal combination is permitted for this.

> ## Danger!
>
> **Start interlocks are only permitted to be disabled if guarantees are in place that no hazardous situation can result from a machine/system startup or if the start interlocks are implemented at another location or with other methods.**

6.6.19.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.19.3.1 Error detection

The following conditions result in an error message on the function block:

- Specified testing time exceeded during the sensor test
- Response of sensor signal not detected during the sensor test
- Invalid static TRUE signals on "Reset"
- Invalid value on "TestTime"

6.6.19.3.2 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.19.3.3 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.19.3.4 Unvalid static signals when cold restarting the safety controller

A static TRUE signal on input parameter "Reset" during a cold restart of the safety controller causes an error message on the function block if the start interlock is specified after the function block is enabled ("S_StartReset" = FALSE).

If this start interlock is not specified for a cold restart of the safety controller, then the state of "Reset" is irrelevant.

6.6.19.3.5 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.6.19.3.6 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.19.4 Input parameters

6.6.19.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off. This connection allows the function block to optionally support a start interlock (as long as input parameter "S_StartReset" exists) after the device is enabled if the states of the safe devices involved in the safety function are connected to "Activate".
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

After the function block is enabled, it supports an optional start interlock. The start interlock is reset by a rising edge on "Reset". An active start interlock is indicated by a corresponding diagnostic code.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

Information:

Output parameter "S_TestOut" is set to TRUE.

All other binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.19.4.2 S_OSSD_In

General function

- Signal input for the state of the connected safety sensor

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to the safety sensor over 1 or 2 channels. Input parameter "S_OSSD_In" is then controlled via this signal.

Function description

The signal connected to input parameter "S_OSSD_In" is processed by the function block.

The signal input processes the state of the connected safety sensor.

Regardless whether the safety sensor is connected to the safe device over 1 or 2 channels, "S_OSSD_In" is only connected to one signal.

If a safety sensor is wired to the safe device over 2 channels, then the dual-channel redundancy is monitored by the safe device. This device passes on a signal to "S_OSSD_In". This signal is evaluated by the function block.

Alternatively, monitoring of the equivalence of 2 safe signals can be implemented with function block "SF_Equivalent"; monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent" or "SF_Equivalent") passes on a signal to "S_OSSD_In". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The safety sensor is in normal mode.
A test of the safety sensor is not requested and/or not active.

After a signal change from FALSE to TRUE on input parameter "S_OSSD_In", the function block optionally supports a start interlock outside of test mode ("S_OSSD_Out" = FALSE).

The start interlock is reset on a rising edge of "Reset" ("S_OSSD_Out" = FALSE → TRUE). An active start interlock is indicated accordingly via output parameter "DiagCode".

FALSE

The test of the safety sensor or safety request is active. The wiring to the safety sensor is interrupted or the safe device connected to the safety sensor is shut down or defective.

Output parameter "S_OSSD_Out" is set to FALSE; output parameter "DiagCode" is set accordingly.

6.6.19.4.3 StartTest

General function

- Starting signal for the sensor test

Data type

- BOOL

Connection

- Variable

Function description

This signal input starts the test of a connected safety sensor. The edges of the input parameter are monitored internally by the function block. Only a rising edge results in execution of the sensor test. An additional static TRUE signal following a rising edge does not trigger the function again.

> ## Information:
>
> **You are responsible for controlling "StartTest" and therefore performing testing periodically.**
>
> **Evaluate output parameter "TestPossible" before starting the test. You are responsible for performing periodic testing according to the results of your risk analysis by setting "StartTest" from FALSE to TRUE and evaluating the result on the function block.**

6.6.19.4.4 TestTime

General function

- Specification of the maximum response time of the sensor signal on "S_OSSD_In" during the sensor test

Data type

- TIME

Connection

- Constant

Function description

Use this time value to set the maximum response time of the sensor signal on "S_OSSD_In".

If "StartTest" switches from FALSE to TRUE, the first time monitoring begins with the value specified on "TestTime". In this state, "S_OSSD_In" must switch from TRUE to FALSE within this time value. The second time monitoring begins when "S_OSSD_In" switches from TRUE to FALSE. "S_OSSD_In" must be set back to TRUE within the value specified on "TestTime".
If one of these states is not achieved within the monitored times, "S_OSSD_Out" is set to FALSE. The function block generates and outputs an error message.

> ## Danger!
>
> **The value for "TestTime" depends on the safety application. Because of this, an automatic test cycle results in an extension of the total response time of the safety function. Determine the value for "TestTime" based on your risk analysis.**

> ## Danger!
>
> **Only specify on "TestTime" the value determined in your risk analysis.**
>
> **Larger values are impermissible since they result in the loss of safety functionality.**
>
> **Smaller values result in the loss of availability.**

Range of values: 0 to 150 ms

6.6.19.4.5 NoExternalTest

General function

- Specification for an external manual sensor test

Data type

- BOOL

Connection

- Constant

Function description

Following a faulty sensor test, ensure the error-free operation of the safety sensor!

Depending on the results of your risk analysis, this may require manual functional testing of the safety sensor independent of the function block. In this manual sensor test, you consciously interrupt the protective equipment and then consciously revoke the interruption.

The function block optionally provides support to you for this manual sensor test, which is performed independently of the function block. This support includes the function block checking whether a sensor signal on "S_OSSD_In" is set from TRUE to FALSE and back to TRUE again. After this signal sequence, another automatic test must be requested via "StartTest". Evaluate the results of this automatic test again.

TRUE

A manual test of the safety sensor performed independently of the function block is not supported by the function block.

An automatic sensor test can be performed without "S_OSSD_In" detecting a signal change following a faulty automatic test. Note that "TestPossible" must indicate state TRUE in order to perform the automatic sensor test.

FALSE

A manual test of the safety sensor performed independently of the function block is supported by the function block.

An automatic sensor test can only be performed if the function block detects a signal change on "S_OSSD_In" following a faulty automatic test.

6.6.19.4.6 S_StartReset

General function

- Specification of the start interlock after the function block is enabled and/or a cold restart of the safety controller

Data type

- SAFEBOOL

Connection

- Variable or constant

Information:

If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).

If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.

Function description

This input parameter specifies the startup behavior of the function block following its enabling and/or a cold restart of the safety controller.

TRUE

After being enabled, the function block does not support start interlock.

After the function block is enabled, no action is required on "Reset" to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

Automatic startup

Note that an unintended TRUE signal results in unexpected startup. "S_StartReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur when starting the safety controller or if startup is prevented by other measures.

FALSE

After being enabled, the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set the safe output parameters to TRUE if the input signal combination is valid for this.

Danger!

If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.

The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur if the safety controller starts.

6.6.19.4.7 S_AutoReset

General function

- Specification of the start interlock after a signal change from FALSE to TRUE on "S_OSSD_In"

Data type

- SAFEBOOL

Connection

- Variable or constant

> ## Information:
>
> **If using a variable to dynamically control this input parameter, you must validate the startup behavior for each state (FALSE/TRUE).**
>
> **If a static value is specified with a constant, the startup behavior only has to be validated for the specified value.**

Function description

This input parameter specifies the operating behavior of the function block after a signal change from FALSE to TRUE on safe input parameter "S_OSSD_In" (safety sensor is in normal mode).

TRUE

After a signal change from FALSE to TRUE on safe input parameter "S_OSSD_In", the function block does not support start interlock.

No action is required on "Reset" to set safe output parameter "S_OSSD_Out" to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **Automatic startup**
>
> **Note that an unintended TRUE signal results in unexpected startup. "S_AutoReset" is therefore only permitted to indicate value TRUE if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested or if startup is prevented by other measures.**

FALSE

After a signal change from FALSE to TRUE on safe input parameter "S_OSSD_In", the function block supports start interlock.

"Reset" must be changed from FALSE to TRUE in order to set safe output parameter "S_OSSD_Out" to TRUE if the input signal combination is valid for this.

> ## Danger!
>
> **If applicable standards require a start interlock for the safety function, then you must implement this start interlock within the safety function. The required start interlock is either supported by the function block or you must take responsibility for implementing the required start interlock outside of this function block.**
>
> **The start interlock is only permitted to be disabled if there are assurances that a hazardous situation cannot occur after the safety function is no longer requested.**

6.6.19.4.8 Reset

General function

- Input parameter for resetting error messages once the error has been corrected
  or
- Input parameter for supporting a manual reset mechanism if start interlock has been specified with input parameter "S_StartReset" and/or "S_AutoReset".

Data type

- BOOL

> ## Danger!
>
> **Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.**
>
> **Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.**
>
> **Further preventive measures such as additional function stops can be used to further reduce unexpected startup.**

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.6.19.5 Output parameters

6.6.19.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.19.5.2 S_OSSD_Out

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected safety sensor for the process being controlled. The release signal is controlled based on the state of the safety sensor and start interlocks. The release signal remains TRUE during the sensor test if no errors occur during the sensor test.

In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_OSSD_Out", this output is referred to as the "enable output".

Release signal "S_OSSD_Out" can be used for subsequent process control.

> ## Danger!
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- The sensor is not requesting a safety function and a test error of the sensor is not present ("S_OSSD_In" = TRUE).
- A start interlock is not active.
- The function block did not detect any faults.

FALSE

The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- The sensor is requesting a safety function and/or a test error of the sensor is present ("S_OSSD_In" = FALSE).
- A start interlock is active.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

Enable output "S_OSSD_Out" is set to TRUE only if input "S_OSSD_In" has state TRUE and a reset has been carried out (start interlock not active).

The following table explains this behavior in detail.

<table>
<tr><th colspan="2">Input parameters</th><th>Action</th><th>Start interlock</th><th>Reset</th><th>Enable output</th></tr>
<tr><td rowspan="2">S_AutoReset</td><td>TRUE</td><td rowspan="2">After a signal change from FALSE to TRUE on the safe input, the start interlock is...</td><td>...inactive.</td><td>No action on "Reset" is required...</td><td rowspan="4">• ...to set enable output "S_OSSD_Out" to TRUE on a valid input signal combination.
• ...to end the start interlock.</td></tr>
<tr><td>FALSE</td><td>...active.</td><td>"Reset" must be set from FALSE to TRUE...</td></tr>
<tr><td rowspan="2">S_StartReset</td><td>TRUE</td><td rowspan="2">After the function block is enabled / cold restart of the safety controller, the start interlock is...</td><td>...inactive.</td><td>No action on "Reset" is required...</td></tr>
<tr><td>FALSE</td><td>...active.</td><td>"Reset" must be set from FALSE to TRUE...</td></tr>
</table>

Table 695: "SF_TestableSafetySensor": Input parameter "S_AutoReset" / "S_StartReset"

6.6.19.5.3 S_TestOut

General function

- Control of the sensor test input

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter sets the test input of the type 2 safety sensor in order to perform an automatic test.

TRUE

A sensor test is not requested.

FALSE

A sensor test is requested.

6.6.19.5.4 TestPossible

General function

- Feedback to the process controlling the sensor test
- Automatic sensor test is possible / not possible

Data type

- BOOL

Connection

- Variable

Function description

The state of this output parameter indicates to the process whether an automatic sensor test is possible or not. Only with the feedback that the test can be executed can the test be started with a signal change from FALSE to TRUE on "StartTest". Otherwise, the state change on "StartTest" has no effect.

TRUE

An automatic sensor test is possible.

FALSE

An automatic sensor test is not possible.

6.6.19.5.5 TestExecuted

General function

- Message: Automatic sensor test executed / not executed / active / faulty

Data type

- BOOL

Connection

- Variable

Function description

A rising edge on "TestExecuted" indicates that the automatic sensor test completed without errors. Safe functionality of the safety sensor is only ensured if the automatic test has been performed.

TRUE

The automatic sensor test was performed successfully.

FALSE

The automatic sensor test has not yet been performed, is still active or was completed with errors.

> **Danger!**
>
> **Note that the safe functionality of the safety sensor is not ensured. Perform periodic testing of the safety sensor with "StartTest".**

6.6.19.5.6 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.6.19.5.7 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.19.5.8 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral. Note that it is not possible to start the test routine.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE. A safety request is not active. The test of the protective equipment was completed without errors. Note that another test of the connected protective equipment is possible in this state.	• No corrective measures are required. • To repeat the test of the protective equipment, a signal change from FALSE to TRUE on "StartTest" is required. **DANGER!** **You are solely responsible for planning and performing the automatic test routine at regular intervals according to the results of your risk analysis.**
8001	The function block has been enabled. The function block's start interlock is active. If "S_StartReset" = TRUE, this state is completed after one cycle of the safety controller without a reset having been executed. Note that it is not possible to start the test routine.	Execute a reset on the function block in order to end the active start interlock of the function block.
8002	The function block detected a safety request by the connected protective equipment when the function block was enabled or after completion of the manual test of the protective equipment. The detection area of the protective equipment is interrupted. Note that it is not possible to start the test routine.	Revoke the safety request of the connected protective equipment by revoking the interruption of the detection area of the connected protective equipment.
8003	A safety request of the connected protective equipment has been revoked again. The function block's start interlock is active. If "S_AutoReset" = TRUE, this state is completed after one cycle of the safety controller without a reset having been executed. Note that it is not possible to start the test routine.	Execute a reset on the function block in order to end the active start interlock of the function block.
8004	The test routine for an external manual test of the connected protective equipment is active. Note that it is not possible to start the test routine.	• Consciously trigger a safety request on the connected protective equipment by interrupting the detection area of the protective equipment. • When the detection area is interrupted, clear the detection area and trigger a new safety request.
8005	The test routine for an external manual test of the connected protective equipment is active. The interruption of the detection area of the protective equipment was detected by the function block. The detection area is still being interrupted. Note that it is not possible to start the test routine.	Revoke the conscious safety request by revoking the interruption of the detection area of the connected protective equipment.
8006	The test routine for an external manual test of the connected protective equipment was completed successfully. Note that it is not possible to start the test routine.	Reset the function block.
8010	The function block has not detected a status event or error in order to set the enable output to FALSE. A safety request is not active. The test of the protective equipment was not performed. **DANGER!** **The safe function of the connected protective equipment is not guaranteed.** **Perform a test of the protective equipment to ensure safe operation of your application.**	• Start the test of the protective equipment with a signal change from FALSE to TRUE on "S_StartReset". **DANGER!** **You are solely responsible for planning and performing the automatic test routine at regular intervals according to the results of your risk analysis.**
8012	The function block detected a safety request by the connected protective equipment after an automatic test of the connected protective equipment without errors. The detection area of the protective equipment is interrupted. Note that it is not possible to start the test routine.	Revoke the safety request of the connected protective equipment by revoking the interruption of the detection area of the connected protective equipment.
8013	A safety request of the connected protective equipment has been revoked again. The function block's start interlock is active. If "S_AutoReset" = TRUE, this state is completed after one cycle of the safety controller without a reset having been executed. Note that it is not possible to start the test routine.	Execute a reset on the function block in order to end the active start interlock of the function block.
8020	The automatic test of the connected protective equipment is started via "StartTest". The test routine for this automatic test is therefore active. Function block test output "S_TestOut" is set to FALSE to start the test of the connected protective equipment. This state is time-monitored. The permissible dwell time of this state is limited by the specified value on "TestTime". Exceeding this value will abort the test routine and result in an error message on the function block. Note that it is not possible to start the test routine. **DANGER!** **Take the specified time frame on "TestTime" into account in your risk analysis.**	• No corrective measures are required. • The connected protective equipment must take on state FALSE in order to continue controlling the test routine. This state is evaluated by the function block on "S_OSSD_In".

Table 696: "SF_TestableSafetySensor": Diagnostic codes

Code (hex)	Description	Corrective measures
8030	The test routine for the automatic test of the connected protective equipment is active. Function block test output "S_TestOut" is set to TRUE to end the test. This state is time-monitored. The permissible dwell time of this state is limited by the specified value on "TestTime". Exceeding this value will abort the test routine and result in an error message on the function block. Note that it is not possible to start the test routine. **DANGER!** **Take the specified time frame on "TestTime" into account in your risk analysis.**	• No corrective measures are required. • The connected protective equipment must take on state TRUE in order to end the test routine. This state is evaluated by the function block on "S_OSSD_In".
C000	The specified time value on "TestTime" is invalid. The valid range of values is between 0 and 150 ms. Note that it is not possible to start the test routine.	• Specify a value between 0 and 150 ms. • Recompile your project in the safe programming environment and transfer it to the safety controller. • Then perform a cold restart on the safety controller. **DANGER!** **Only specify on "TestTime" the value determined in your risk analysis.** **Larger values are impermissible since they result in the loss of safety functionality.** **Smaller values result in the loss of availability.**
C001	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C002	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C003	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C004	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C005	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C006	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C007	The function block detected a static TRUE signal on "Reset". Note that it is not possible to start the test routine.	• Check the reset control device and its wiring. • If necessary, correct the error. • Set the function block parameter "Reset" to FALSE to correct the error.
C010	The function block detected a timeout while performing the automatic test of the connected protective equipment. The signal of the protective equipment being evaluated by "S_OSSD_In" was not set to FALSE during the test. Note that it is not possible to start the test routine.	• Check the connected protective equipment. • If necessary, correct the error. • Reset the function block. **DANGER!** **Repeat the automatic test of the protective equipment and reevaluate the results of the repeated test.**
C020	The function block detected a timeout while performing the automatic test of the connected protective equipment. The signal of the protective equipment being evaluated by "S_OSSD_In" was not set back to TRUE during the test. Note that it is not possible to start the test routine.	• Check the connected protective equipment. • If necessary, correct the error. • Reset the function block. **DANGER!** **Repeat the automatic test of the protective equipment and reevaluate the results of the repeated test.**

Table 696: "SF_TestableSafetySensor": Diagnostic codes

6.6.19.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

"S_StartReset" = FALSE
"S_AutoReset" = FALSE

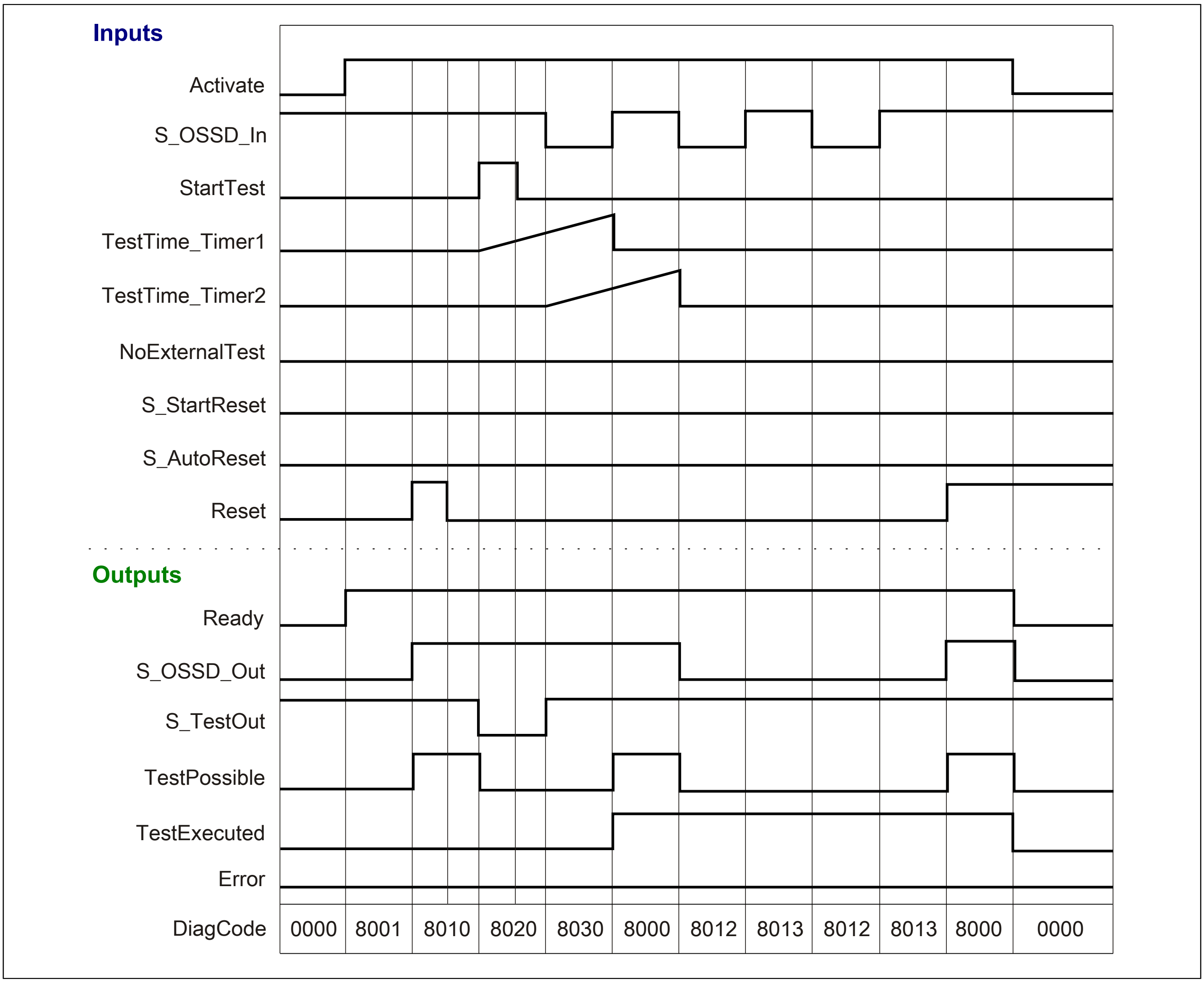


Figure 496: "SF_TestableSafetySensor": Signal sequence diagram

6.6.19.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement a test function for safety sensors during operation.

The following example illustrates the connection of the function block when controlling using the signal from a light curtain connected over 1 channel (see section 6.6.19.7.2 "Light curtain, connected over 1 channel").

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.19.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "TSS_S8".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

Start interlocks

Input parameter "S_StartReset" defines the startup behavior of the function block when enabled. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after the function block is enabled. In addition to the safe input signal on "S_OSSD_In", a rising edge on input parameter "Reset" is required in order for enable output "S_OSSD_Out" to be enabled.

Input parameter "S_AutoReset" defines the operational behavior of the function block. This input parameter is connected to the FALSE constant. Because of this, the start interlock is active after a change of the safe input signal on "S_OSSD_In" (FALSE → TRUE). In addition to the safe input signal on "S_OSSD_In", a rising edge on input parameter "Reset" is required in order for enable output "S_OSSD_Out" to be enabled.

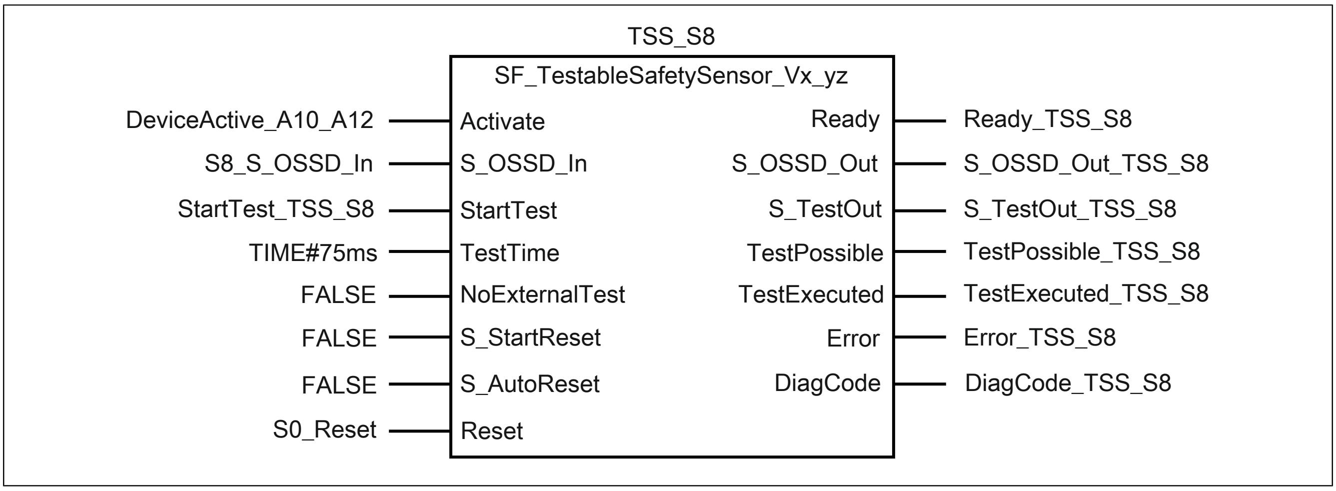


Figure 497: "SF_TestableSafetySensor": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10_A12	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In this example, the signal represents the state of the safe input device to which the light curtain is connected.
S8_S_OSSD_In	SAFEBOOL	Signal of the safety sensor. The signal comes from a 1-channel or 2-channel input of a safe input device. The evaluation of dual-channel redundancy does not take place in the function block.
StartTest_TSS_S8	BOOL	Signal for requesting the test function. This signal comes from the standard controller and is controlled according to the results of the risk analysis to start the periodic sensor test.
TIME#75ms on "TestTime"	TIME	Time value according to the risk analysis that was carried out for this example
FALSE on "NoExternalTest"	BOOL	Specification for an external manual sensor test
FALSE on "S_StartReset"	SAFEBOOL	Specification for the start interlock after cold restarting the safety controller / enabling the function block.
FALSE on "S_AutoReset"	SAFEBOOL	Specification for the start interlock after enabling signal "S_OSSD_In"
S0_Reset	BOOL	External control of "Reset". Resets error messages when the cause of the error no longer exists.

Table 697: "SF_TestableSafetySensor": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_TSS_S8	BOOL	Function block active / Function block inactive. Message for further external processing.
S_OSSD_Out_TSS_S8	SAFEBOOL	Release signal. The release signal can be used for subsequent process control.
S_TestOut_TSS_S8	SAFEBOOL	Test signal. This output is output via a safe output device and starts the test procedure for the sensor.
TestPossible_TSS_S8	BOOL	TRUE: Testing the protective equipment is possible.
TestExecuted_TSS_S8	BOOL	TRUE: The performed test was successful.
Error_TSS_S8	BOOL	Error message from function block for further external processing
DiagCode_TSS_S8	WORD	Diagnostic message from function block for further external processing

Table 698: "SF_TestableSafetySensor": Outputs connected to output parameters

6.6.19.7.2 Light curtain, connected over 1 channel

This example illustrates the connection of the function block when controlling using the signal from a light curtain connected over 1 channel.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.19.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "TSS_S8".

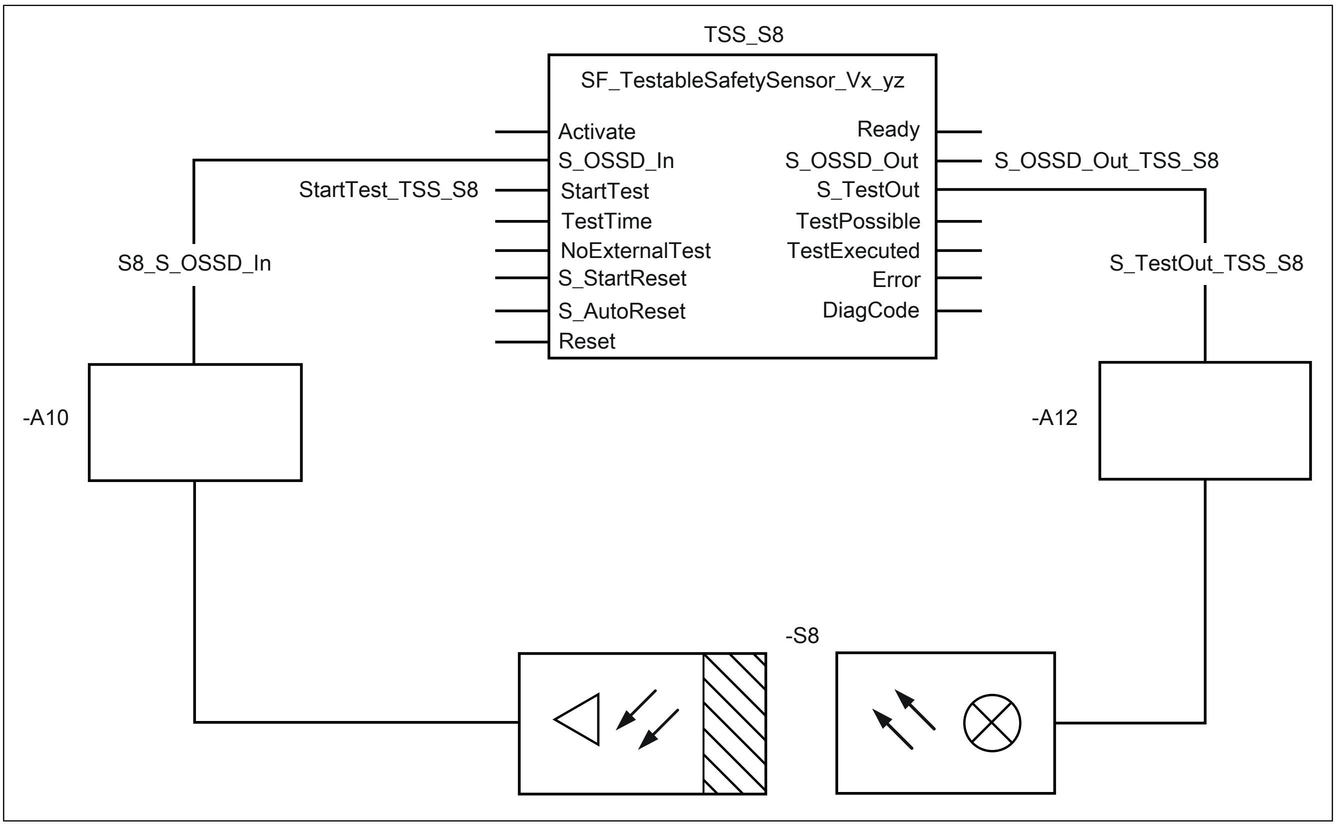


Figure 498: "SF_TestableSafetySensor": Light curtain, connected over 1 channel

List of equipment

-S8	Optical/Electronic protective equipment, 1-channel, with test input
-A10	1-channel input of a safe input device
-A12	1-channel output of a safe output device

Connected inputs and outputs

S8_S_OSSD_In	Input on "S_OSSD_In"
StartTest_TSS_S8	Input on "StartTest"
S_OSSD_Out_TSS_S8	Output on "S_OSSD_Out"
S_TestOut_TSS_S8	Output on "S_TestOut"

Please note that, depending on your application, other combinations of safe devices can be used in place of the safe devices shown here.

Description

In this example:

- The input signal of safe input device "-A10" is connected to input "S8_S_OSSD_In".
- Input "S8_S_OSSD_In" is connected to function block input parameter "S_OSSD_In" for further processing.
- The test procedure is initiated via non-safe input "StartTest_TSS_S8".
- Output parameter "S_TestOut" is connected to output "S_TestOut_TSS_S8".
- Output "S_TestOut_TSS_S8" is connected to the test input of light curtain "-S8" via the safe output device. The signal is used to execute the test procedure.
- Output parameter "S_OSSD_Out" is connected to output "S_OSSD_Out_TSS_S8".
- Output "S_OSSD_Out_TSS_S8" is used as a release signal to control the process in consideration of other safety functions.

6.6.19.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
IEC 61496-1	Special requirements for type 2 EPSE	The function block performs a plausibility test of the value on "TestTime". A value greater than 150 ms is not permitted. Test sequences "Test start" and "Test procedure" are each not permitted to take longer than the specified value on "TestTime". The function block optionally supports start interlocks after a cold restart of the safety controller and after the safety function returns. After a fault in the periodic test, the function block always implements a start interlock. The function block supports the periodic test. The test input of the safety sensor is set by the function block, and the state of the safety sensor is checked. The state of the safety sensor ("S_OSSD_In") must behave in accordance with the defined test sequences in order for the test to be completed without errors. Sequences "Test start" and "Test procedure" are monitored against the specified time on "TestTime". The specified time value on "TestTime" is restarted for each of these sequences. The following test sequences are time-monitored by the function block: 1) Test start • "StartTest" switches from FALSE to TRUE. • Monitoring time 1 is started. • "S_TestOut" switches from TRUE to FALSE. 2) Test procedure • "S_OSSD_In" switches from TRUE to FALSE. • "S_TestOut" switches from FALSE to TRUE. • Monitoring time 1 is stopped. • Monitoring time 2 is started. 3) Test end • "S_OSSD_In" switches from FALSE to TRUE. • Monitoring time 2 is stopped. The function block enters the safe state if the result of the periodic test is faulty. It does this by setting output parameter "S_OSSD_Out" to FALSE. It remains in the safe state since the start interlock is active.
EN 954-1	Manual reset device	Input parameter "Reset" supports the function of the manual reset device.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	The function block optionally supports a start interlock after the following: • Cold restart of the safety controller ("S_AutoReset" = FALSE) • Enabling the function block ("S_StartReset" = FALSE) • A TRUE signal after the safety function is triggered If "Activate" does not reflect the state of the safe devices, you will have to implement this function in a different way. You are responsible for planning and implementing the startup behavior in accordance with your risk analysis. In order to prevent unintended startup, an additional function start after the safety function is reset may be required depending on the results of the risk analysis and based on the signal path of the reset signal.
EN 954-1	Category B to 2	1- or 2-channel switching must be designed based on the category.

Table 699: "SF_TestableSafetySensor": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.20 SF_TwoHandControlTypeII

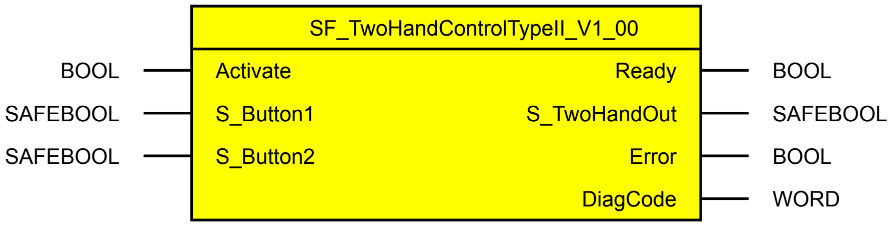


Figure 499: Function block "SF_TwoHandControlTypeII"

6.6.20.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_Button1	SAFEBOOL	Variable	State	FALSE	Input for button 1 on the two-hand control device
S_Button2	SAFEBOOL	Variable	State	FALSE	Input for button 2 on the two-hand control device

Table 700: "SF_TwoHandControlTypeII": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1)]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_TwoHandOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 701: "SF_TwoHandControlTypeII": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 702: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.20.2 Function

Function block "SF_TwoHandControlTypeII" is used to support a type II two-hand control device function in an application.

This function block is a safety-relevant function block for monitoring a type II two-hand control device.

This function block evaluates the state of 2 buttons on a two-hand control device. One of the buttons controls input parameter "S_Button1"; the other button controls input parameter "S_Button2". The function block output signal outputs the results of the evaluation.

6.6.20.2.1 Input parameter states

The state to which function block enable output "S_TwoHandOut" is set depends on the state of the input parameters.

- **When enabling the function block:**
  When the function block is enabled, both input parameters "S_Button1" and "S_Button2" must indicate state FALSE. If at least one of these input parameters indicates state TRUE, the function block will detect an error. In this case, the function block outputs an error message on output parameter "DiagCode".
- **To generate signal TRUE on the enable output:**
  If the function block is enabled ("Activate" = TRUE) and does not detect an error, the function block enable output is set to TRUE if input parameters "S_Button1" and "S_Button2" switch to TRUE successively or simultaneously.
- **To generate signal FALSE on the enable output:**
  The function block enable output is set to FALSE if one or both input parameters "S_Button1" and "S_Button2" switch from TRUE to FALSE.
- **To regenerate signal TRUE on the enable output:**
  The enable output can only be set again to TRUE if both input parameters "S_Button1" and "S_Button2" initially indicate state FALSE. To achieve this, both input parameters "S_Button1" and "S_Button2" must switch to state TRUE successively or simultaneously.

6.6.20.2.2 Connecting the buttons on the two-hand control device

The buttons on the two-hand control device must be connected to the safe control system being used over 1 or 2 channels.

Applications with a two-hand control device with 2 buttons (1-channel each)

If you are using a two-hand control device with 2 buttons (1-channel each), connect the signals to a safe input device individually. One of these signals will be connected to function block input parameter "S_Button1". The second signal must be connected to the second function block input parameter "S_Button2".

Danger!

A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.

The following components of the safety chain must be taken into account in each case:

- **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
- **The connection and wiring between the safe input device and the button on the two-hand control device**
- **The entire two-hand control device**

Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Applications with a two-hand control device with 2 buttons (2-channel each)

If you are using a two-hand control device with 2 buttons (2-channel each), you must connect the 4 signals to one or more safe input devices individually.

Outside of the function block, you must check each of the signals of a button for dual-channel redundancy (e.g. via safe devices or other function blocks such as "SF_Antivalent"). This test must produce a resulting signal for each of the 2 buttons. Connect the resulting signal of a button with input parameter "S_Button1". Connect the resulting signal from the test for the other button to input parameter "S_Button2".

6.6.20.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.20.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.20.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on input parameters cause the signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Manipulation of a 2-hand button (user error)
- Faulty 2-hand connection / 2-hand button (hardware fault)
- Faulty operation (user error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function
- Use of buttons with 2 separate contacts (normally closed contact and normally open contact) to eliminate changeover contact faults

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.20.3.3 Impermissible TRUE signals of the buttons on the two-hand control device

Impermissible TRUE signals of the buttons on the two-hand control device that exist on the function block input parameters after the function block is enabled result in an error message. The function block remains in the safe state ("S_TwoHandOut" = FALSE).

6.6.20.3.4 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.20.4 Input parameters

6.6.20.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

When enabling the function block, it checks the status of signal inputs "S_Button1" and "S_Button2". Both signal inputs must be FALSE when the function block is enabled. If this is not the case, the function block will detect an error.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.20.4.2 S_Button1

General function

- Input for button 1 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to button 1 of a safe two-hand control device over 1 or 2 channels. Input parameter "S_Button1" is then controlled using this signal.

Danger!

A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.

The following components of the safety chain must be taken into account in each case:

- **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
- **The connection and wiring between the safe input device and the button on the two-hand control device**
- **The entire two-hand control device**

Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Function description

The signal connected to input parameter "S_Button1" is processed by the function block.

Signal input 1 processes the state of the signal from button 1 (channel 1) of the safe two-hand control device connected over 1 or 2 channels. This signal must indicate state TRUE in order to set enable output "S_TwoHandOut" to TRUE under consideration of the state on "S_Button2".

Regardless whether button 1 on the two-hand control device is connected to the safe device over 1 or 2 channels, "S_Button1" is only connected to one signal.

If button 1 of the two-hand control device is wired to the safe device over 2 channels, then the dual-channel redundancy (line control) and antivalence are monitored by the safe device. This device passes on a signal to "S_Button1". This signal is evaluated by the function block.

Alternatively, monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent") passes on a signal to "S_Button1". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The two-hand control device (button 1) is actuated.

FALSE

The two-hand control device (button 1) is not actuated, the wiring for the two-hand control device is interrupted or the safe input device connected to the two-hand control device is shut down or defective.

6.6.20.4.3 S_Button2

General function

- Input for button 2 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

> ## Information:
>
> **Connect this input parameter to the signal of a safe device that is connected to button 2 of a safe two-hand control device over 1 or 2 channels. Input parameter "S_Button2" is then controlled using this signal.**

> ## Danger!
>
> **A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.**
>
> **The following components of the safety chain must be taken into account in each case:**
>
> - **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
> - **The connection and wiring between the safe input device and the button on the two-hand control device**
> - **The entire two-hand control device**
>
> **Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.**

Function description

The signal connected to input parameter "S_Button2" is processed by the function block.

Signal input 2 processes the state of the signal from button 2 (channel 2) of the safe two-hand control device connected over 1 or 2 channels. This signal must indicate state TRUE in order to set enable output "S_TwoHandOut" to TRUE under consideration of the state on "S_Button1".

Regardless whether button 2 on the two-hand control device is connected to the safe device over 1 or 2 channels, "S_Button2" is only connected to one signal.

If button 2 of the two-hand control device is wired to the safe device over 2 channels, then the dual-channel redundancy (line control) and antivalence are monitored by the safe device. This device passes on a signal to "S_Button2". This signal is evaluated by the function block.

Alternatively, monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent") passes on a signal to "S_Button2". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The two-hand control device (button 2) is actuated.

FALSE

The two-hand control device (button 2) is not actuated, the wiring for the two-hand control device is interrupted or the safe input device connected to the two-hand control device is shut down or defective.

6.6.20.5 Output parameters

6.6.20.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.20.5.2 S_TwoHandOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected two-hand control device for the process being controlled. The release signal is controlled depending on the state of the two-hand control device.

In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_TwoHandOut", this output is referred to as the "enable output".

Release signal "S_TwoHandOut" can be used for subsequent process control.

> ## Danger!
>
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The two-hand control device is actuated. The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- "S_Button1" and "S_Button2" were symmetrically set from FALSE to TRUE successively or simultaneously.
- "S_Button1" and "S_Button2" indicate state TRUE after the signal change from FALSE to TRUE.
- The function block did not detect any faults.

FALSE

The two-hand control device is not actuated. The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- "S_Button1" = FALSE and/or "S_Button2" = FALSE
- "S_Button1" and "S_Button2" were not symmetrically set from FALSE to TRUE. "S_Button1" or "S_Button2" had state TRUE, and the other input parameter switched from FALSE to TRUE.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

6.6.20.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

Danger!

It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

Information:

In order to exit an error state ("Error" = TRUE), "S_Button1" must be FALSE and "S_Button2" must be FALSE.

6.6.20.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.20.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	Initialization of the function block after it has been enabled. This state is completed after one cycle of the safety controller.	• No corrective measures are required.
8004	Signal FALSE is indicated on inputs "S_Button1" and "S_Button2".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8005	Signal TRUE is indicated on "S_Button1" to set the function block enable output to TRUE. Signal FALSE is indicated on "S_Button2".	According to the requirements of your application: • Set the signal on "S_Button2" to TRUE in addition to signal TRUE on "S_Button1" to set the function block enable output to TRUE. or • Set the signal on "S_Button1" to FALSE to cancel the request for generating signal TRUE on the function block enable output.
8006	Signal TRUE is indicated on "S_Button2" to set the function block enable output to TRUE. Signal FALSE is indicated on "S_Button1".	According to the requirements of your application: • Set the signal on "S_Button1" to TRUE in addition to signal TRUE on "S_Button2" to set the function block enable output to TRUE. or • Set the signal on "S_Button2" to FALSE to cancel the request for generating signal TRUE on the function block enable output.
8007	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, the signal on "S_Button2" switched to FALSE.	Set the signal on "S_Button1" to FALSE.
8008	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, the signal on "S_Button1" switched to FALSE.	Set the signal on "S_Button2" to FALSE.
8009	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, one of the signals on the signal inputs switched to FALSE. The signal on "S_Button1" and the signal on "S_Button2" then indicated state TRUE again.	Set the signal on "S_Button1" and the signal on "S_Button2" to FALSE.
8019	After one of the signals on "S_Button1" and "S_Button2" indicated state TRUE, one of the signals switched from TRUE to FALSE and the other signal switched from FALSE to TRUE.	Set the TRUE signal on one of the signal inputs to FALSE.
C001	When enabling the function block, signal TRUE was indicated on "S_Button1".	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C002	When enabling the function block, signal TRUE was indicated on "S_Button2".	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C003	When enabling the function block, signal TRUE was indicated on "S_Button1" and "S_Button2".	Set the signal on "S_Button1" and the signal on "S_Button2" to FALSE.

Table 703: "SF_TwoHandControlTypeII": Diagnostic codes

6.6.20.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

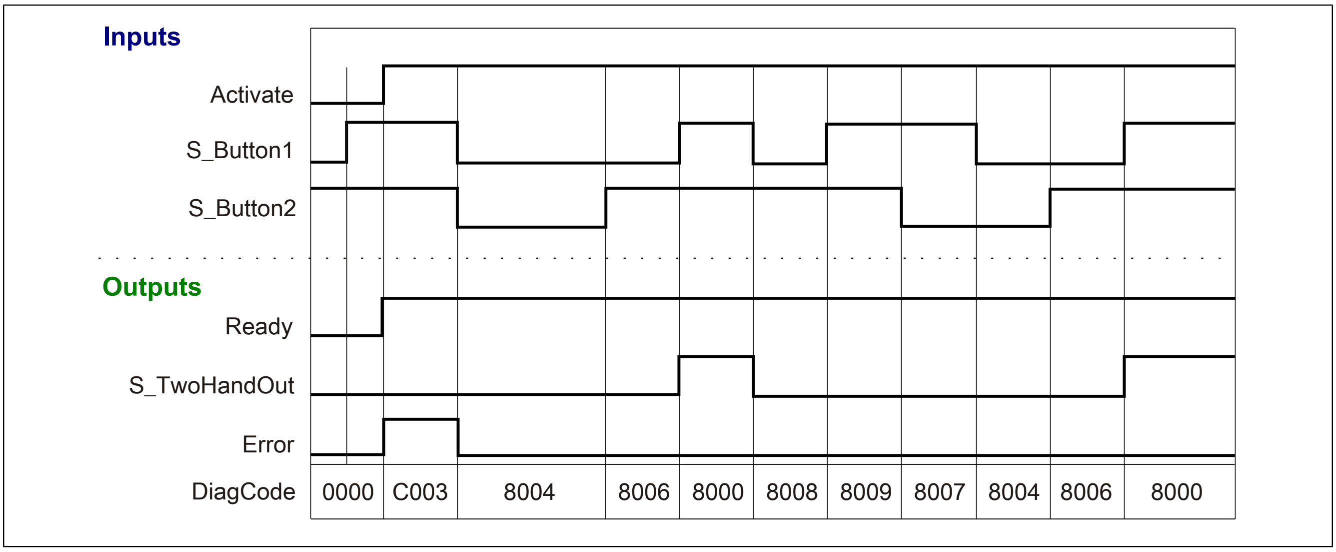


Figure 500: "SF_TwoHandControlTypeII": Signal sequence diagram

6.6.20.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement a two-hand control device.

Type II two-hand control device

The example in section 6.6.20.7.2 "Type II two-hand control device" illustrates the connection of the function block when controlling using each of the signals from 2 buttons of a two-hand control device connected over 2 channels.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

Danger!

Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.
For additional information, please see the documentation for the safe input/output device being used.

6.6.20.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "THC_S5_S6".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

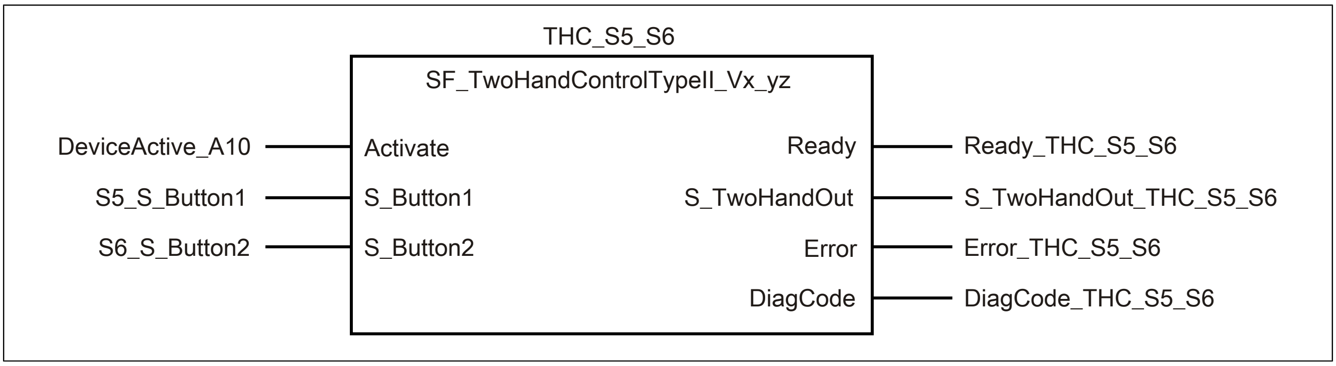


Figure 501: "SF_TwoHandControlTypeII": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In the example, this input is connected to a signal that indicates the state of the safe devices involved in the safety function. This prevents a safety function from being reported even if the safe devices are inactive.
S5_S_Button1	SAFEBOOL	Signal of the first button from a safe input device. The signal comes from a 2-channel operating element. The evaluation of dual-channel redundancy does not take place in the function block.
S6_S_Button2	SAFEBOOL	Signal of the second button from a safe input device. The signal comes from a 2-channel operating element. The evaluation of dual-channel redundancy does not take place in the function block.

Table 704: "SF_TwoHandControlTypeII": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_THC_S5_S6	BOOL	Function block active / Function block inactive. Message for further external processing.
S_TwoHandOut_THC_S5_S6	SAFEBOOL	Release signal. This signal is used for further processing in the safety application on the safety controller.
Error_THC_S5_S6	BOOL	Error message from function block for further external processing
DiagCode_THC_S5_S6	WORD	Diagnostic message from function block for further external processing

Table 705: "SF_TwoHandControlTypeII": Outputs connected to output parameters

6.6.20.7.2 Type II two-hand control device

This example illustrates the connection of the function block when controlling using each of the signals from 2 buttons (two-hand control device, 2-channel antivalent) antivalently connected over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.20.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "THC_S5_S6".

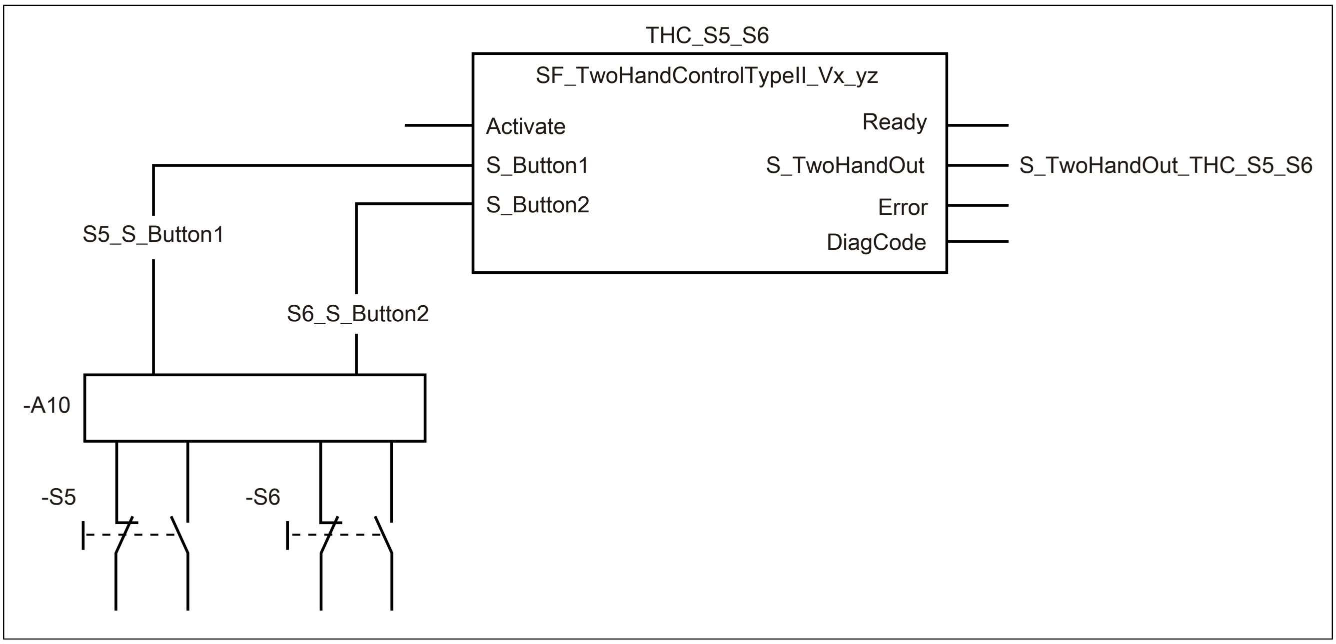


Figure 502: "SF_TwoHandControlTypeII": Type II two-hand control device

List of equipment

-S5	Two-hand button (2-channel, antivalent)
-S6	Two-hand button (2-channel, antivalent)
-A10	2-channel safe inputs of a safe input device (antivalent) with line control and antivalent monitoring

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S5_S_Button1	Input on "S_Button1"
S6_S_Button2	Input on "S_Button2"
S_TwoHandOut_THC_S5_S6	Output on "S_TwoHandOut"

Description

In this example:

- The resulting signal (checked for dual-channel redundancy and antivalence) of button "-S5" from the 2-channel input of safe input device "-A10" is connected to input "S5_S_Button1".
- Input "S5_S_Button1" is connected to function block input parameter "S_Button1" for further processing.
- The resulting signal (checked for dual-channel redundancy and antivalence) of button "-S6" from the 2-channel input of safe input device "-A10" is connected to input "S6_S_Button2".
- Input "S6_S_Button2" is connected to function block input parameter "S_Button2" for further processing.
- Output parameter "S_TwoHandOut" is connected to output "S_TwoHandOut_THC_S5_S6".
- Output "S_TwoHandOut_THC_S5_S6" is used as a release signal to control the process in consideration of other safety functions.

6.6.20.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 574	Use of both hands (simultaneous actuation)	The function block sets it release signal to TRUE only if the signals on "S_Button1" and "S_Button2" switch from FALSE to TRUE simultaneously or successively.
EN 574	Relationship between input signals and output signals / Termination of the output signal	The function block release signal set to TRUE is set to FALSE if "S_Button1" and/or "S_Button2" switches to FALSE.
EN 574	Regeneration of the output signal	If the signals on "S_Button1" and "S_Button2" switch from FALSE to TRUE, the function block only sets its release signal to TRUE if "S_Button1" and "S_Button2" initially indicate state FALSE after the release signal is cut off (release signal = FALSE).
EN 574	Use of category 3	The implementation of the two-hand control device must comply with the requirements of safety category 3 in accordance with EN 954-1.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	If "S_Button1" and/or "S_Button2" indicate state TRUE when the function block is enabled, then the function block detects this as a fault. If this error message occurs, it is not possible set the release signal to TRUE. To reset this fault, "S_Button1" and "S_Button2" must indicate state FALSE.
EN 954-1	Category 3	You are responsible for implementing the requirements of category 3 for the two-hand control device. A 1-channel configuration of the button is only permitted if the safety chain of the connected button on the two-hand control device complies with EN 954-1 requirements up to category 3 over 1 channel. The following components of the safety chain must be taken into account in each case: • The safe device (input device) of the safe control system, including the device configuration (line control), used in the two-hand control device • The connection and wiring between the safe input device and the button on the two-hand control device • The entire two-hand control device Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Table 706: "SF_TwoHandControlTypeII": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.21 SF_TwoHandControlTypeIII

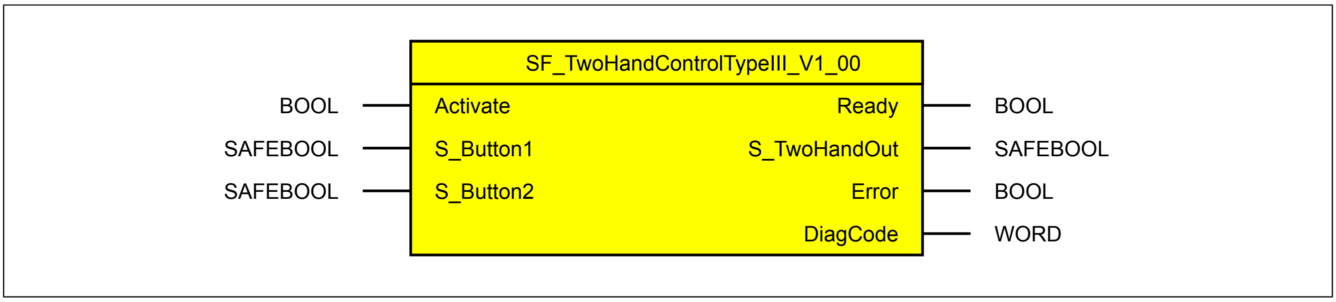


Figure 503: Function block "SF_TwoHandControlTypeIII"

6.6.21.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Activate	BOOL	Variable/ Constant	State	FALSE	Enables the function block ("Activate" = TRUE)
S_Button1	SAFEBOOL	Variable	State	FALSE	Input for button 1 on the two-hand control device
S_Button2	SAFEBOOL	Variable	State	FALSE	Input for button 2 on the two-hand control device

Table 707: "SF_TwoHandControlTypeIII": Overview of input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
Ready	BOOL	Variable	State	FALSE	Indicates that the function block is enabled
S_TwoHandOut	SAFEBOOL	Variable	State	FALSE	Release signal of the function block
Error	BOOL	Variable	State	FALSE	Function block error message
DiagCode	WORD	Variable	State	16#0000	Function block diagnostic message

Table 708: "SF_TwoHandControlTypeIII": Overview of output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

Type	Description	Size in bits	Format option
BOOL	Bit	1	Boolean
WORD	Word	16	Binary number, hexadecimal number, unsigned decimal number
SAFEBOOL	Bit	1	Boolean (signal source: safe device)

Table 709: Formats of data types used

It is possible to connect a safe signal with a non-safe input parameter. To do so, you need to use a function block for type conversion.

6.6.21.2 Function

Function block "SF_TwoHandControlTypeIII" is used to support a type III two-hand control device function in an application.

This function block is a safety-relevant function block for monitoring a type III two-hand control device.

This function block evaluates the state of 2 buttons on a two-hand control device. One of the buttons controls input parameter "S_Button1"; the other button controls input parameter "S_Button2". The function block output signal outputs the results of the evaluation.

The function block uses an internal timer to monitor whether safe input "S_Button2" switches from FALSE to TRUE within 500 ms after safe input "S_Button1" previously switched from FALSE to TRUE.

6.6.21.2.1 Input parameter states

The state to which function block enable output "S_TwoHandOut" is set depends on the state of the input parameters.

- **When enabling the function block:**
  When the function block is enabled, both input parameters "S_Button1" and "S_Button2" must indicate state FALSE. If at least one of these input parameters indicates state TRUE, the function block will detect an error. In this case, the function block outputs an error message on output parameter "DiagCode".
- **To generate signal TRUE on the enable output:**
  If the function block is enabled ("Activate" = TRUE) and does not detect an error, the function block enable output is set to TRUE if input parameters "S_Button1" and "S_Button2" switch from FALSE to TRUE successively or simultaneously. The function block monitors whether the signal change on both input parameters takes place synchronously within 500 ms. The function block detects an asynchronous signal change as a fault.
- **To generate signal FALSE on the enable output:**
  The function block enable output is set to FALSE if one or both input parameters "S_Button1" and "S_Button2" switch from TRUE to FALSE. In addition, the enable output remains set to FALSE if one of the signals on "S_Button1" or "S_Button2" switches to TRUE and the other signal did not switch to TRUE within the synchronization monitoring time.
- **To regenerate signal TRUE on the enable output:**
  The enable output can only be set again to TRUE if both input parameters "S_Button1" and "S_Button2" initially indicate state FALSE. To achieve this, both input parameters "S_Button1" and "S_Button2" must switch to state TRUE successively or simultaneously. The function block monitors whether the signal change on both input parameters takes place synchronously within 500 ms. The function block detects an asynchronous signal change as a fault.

6.6.21.2.2 Connecting the buttons on the two-hand control device

The buttons on the two-hand control device must be connected to the safe control system being used over 1 or 2 channels.

Applications with a two-hand control device with 2 buttons (1-channel each)

If you are using a two-hand control device with 2 buttons (1-channel each), connect the signals to a safe input device individually. One of these signals will be connected to function block input parameter "S_Button1". The second signal must be connected to the second function block input parameter "S_Button2".

Danger!

A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.

The following components of the safety chain must be taken into account in each case:

- **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
- **The connection and wiring between the safe input device and the button on the two-hand control device**
- **The entire two-hand control device**

Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Applications with a two-hand control device with 2 buttons (2-channel each)

If you are using a two-hand control device with 2 buttons (2-channel each), you must connect the 4 signals to one or more safe input devices individually.

Outside of the function block, you must check each of the signals of a button for dual-channel redundancy (e.g. via safe devices or other function blocks such as "SF_Antivalent"). This test must produce a resulting signal for each of the 2 buttons. Connect the resulting signal of a button with input parameter "S_Button1". Connect the resulting signal from the test for the other button to input parameter "S_Button2".

6.6.21.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.6.21.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.6.21.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on input parameters cause the signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Manipulation of a 2-hand button (user error)
- Faulty 2-hand connection / 2-hand button (hardware fault)
- Faulty operation (user error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function
- Use of buttons with 2 separate contacts (normally closed contact and normally open contact) to eliminate changeover contact faults

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.6.21.3.3 Impermissible TRUE signals of the buttons on the two-hand control device

Impermissible TRUE signals of the buttons on the two-hand control device that exist on the function block input parameters after the function block is enabled result in an error message. The function block remains in the safe state ("S_TwoHandOut" = FALSE).

The function block detects an asynchronous signal change from FALSE to TRUE on "S_Button1" and "S_Button2" within 500 ms as a fault.

6.6.21.3.4 Machine/System startup without performing functional testing of safety equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> ## Danger!
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.6.21.4 Input parameters

6.6.21.4.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

When enabling the function block, it checks the status of signal inputs "S_Button1" and "S_Button2". Both signal inputs must be FALSE when the function block is enabled. If this is not the case, the function block will detect an error.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.6.21.4.2 S_Button1

General function

- Input for button 1 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to button 1 of a safe two-hand control device over 1 or 2 channels. Input parameter "S_Button1" is then controlled using this signal.

Danger!

A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.

The following components of the safety chain must be taken into account in each case:

- **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
- **The connection and wiring between the safe input device and the button on the two-hand control device**
- **The entire two-hand control device**

Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Function description

The signal connected to input parameter "S_Button1" is processed by the function block.

Signal input 1 processes the state of the signal from button 1 (channel 1) of the safe two-hand control device connected over 1 or 2 channels. This signal must indicate state TRUE in order to set enable output "S_TwoHandOut" to TRUE under consideration of the state on "S_Button2".

Regardless whether button 1 on the two-hand control device is connected to the safe device over 1 or 2 channels, "S_Button1" is only connected to one signal.

If button 1 of the two-hand control device is wired to the safe device over 2 channels, then the dual-channel redundancy (line control) and antivalence are monitored by the safe device. This device passes on a signal to "S_Button1". This signal is evaluated by the function block.

Alternatively, monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent") passes on a signal to "S_Button1". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The two-hand control device (button 1) is actuated.

FALSE

The two-hand control device (button 1) is not actuated, the wiring for the two-hand control device is interrupted or the safe input device connected to the two-hand control device is shut down or defective.

6.6.21.4.3 S_Button2

General function

- Input for button 2 on the two-hand control device

Data type

- SAFEBOOL

Connection

- Variable

Information:

Connect this input parameter to the signal of a safe device that is connected to button 2 of a safe two-hand control device over 1 or 2 channels. Input parameter "S_Button2" is then controlled using this signal.

Danger!

A 1-channel configuration of the button for CAT 3 is only permitted if the safety chain of the connected button on the two-hand control device complies with safety-related requirements up to CAT 3 over 1 channel.

The following components of the safety chain must be taken into account in each case:

- **The safe input device of the safe control system, including the device configuration (line control), used in the two-hand control device**
- **The connection and wiring between the safe input device and the button on the two-hand control device**
- **The entire two-hand control device**

Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.

Function description

The signal connected to input parameter "S_Button2" is processed by the function block.

Signal input 2 processes the state of the signal from button 2 (channel 2) of the safe two-hand control device connected over 1 or 2 channels. This signal must indicate state TRUE in order to set enable output "S_TwoHandOut" to TRUE under consideration of the state on "S_Button1".

Regardless whether button 2 on the two-hand control device is connected to the safe device over 1 or 2 channels, "S_Button2" is only connected to one signal.

If button 2 of the two-hand control device is wired to the safe device over 2 channels, then the dual-channel redundancy (line control) and antivalence are monitored by the safe device. This device passes on a signal to "S_Button2". This signal is evaluated by the function block.

Alternatively, monitoring of the antivalence of 2 safe signals can be implemented with function block "SF_Antivalent". In this case, the function block ("SF_Antivalent") passes on a signal to "S_Button2". This signal is evaluated by the function block.

The signal input is state-controlled. These states only lead to the following connection results if the function block is enabled ("Activate" = TRUE).

TRUE

The two-hand control device (button 2) is actuated.

FALSE

The two-hand control device (button 2) is not actuated, the wiring for the two-hand control device is interrupted or the safe input device connected to the two-hand control device is shut down or defective.

6.6.21.5 Output parameters

6.6.21.5.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.6.21.5.2 S_TwoHandOut

General function

- Release signal of the function block

Data type

- SAFEBOOL

Connection

- Variable

Function description

The release signal is the safe enable signal of the connected two-hand control device for the process being controlled. The release signal is controlled depending on the state of the two-hand control device.

In addition, the release signal controls the request for the stop function.

Since the release signal is present on output "S_TwoHandOut", this output is referred to as the "enable output".

Release signal "S_TwoHandOut" can be used for subsequent process control.

> **Danger!**
>
> **The release signal is only permitted to control the process directly if it does not impair the safety function.**
> **The entire safety function chain must be validated, including the startup behavior of the process being controlled.**

TRUE

The two-hand control device is actuated. The process to be controlled is enabled. The request for the stop function is not active.

The following conditions must be met for this:

- The function block is enabled ("Activate" = TRUE).
- "S_Button1" and "S_Button2" were synchronously and symmetrically set from FALSE to TRUE successively or simultaneously within 500 ms.
- "S_Button1" and "S_Button2" indicate state TRUE after the signal change from FALSE to TRUE.
- The function block did not detect any faults.

FALSE

The two-hand control device is not actuated. The process to be controlled is not enabled. The request for the stop function is active.

This may be due to one of the following reasons:

- The function block is not enabled ("Activate" = FALSE).
- "S_Button1" = FALSE and/or "S_Button2" = FALSE
- "S_Button1" and "S_Button2" were not symmetrically set from FALSE to TRUE, or not within 500 ms. "S_Button1" or "S_Button2" had state TRUE, and the other input parameter switched from FALSE to TRUE.
- The function block detected a fault.

The risk of unexpected startup and/or manipulation can be reduced by combining a stop request from the safety application and an operational stop from the standard application.

6.6.21.5.3 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

> ## Information:
>
> **In order to exit an error state ("Error" = TRUE), "S_Button1" must be FALSE and "S_Button2" must be FALSE.**

6.6.21.5.4 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.6.21.5.5 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block has not detected a status event or error in order to set the enable output to FALSE.	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8001	Initialization of the function block after it has been enabled. This state is completed after one cycle of the safety controller.	• No corrective measures are required.
8004	Signal FALSE is indicated on inputs "S_Button1" and "S_Button2".	Intended event: • No corrective measures are required. Unintended event: • Check the safety functions. • Check the safety program. • Check the safe peripheral.
8005	Signal TRUE is indicated on "S_Button1" to set the function block enable output to TRUE. Signal FALSE is indicated on "S_Button2".	According to the requirements of your application: • Set the signal on "S_Button2" to TRUE in addition to signal TRUE on "S_Button1" to set the function block enable output to TRUE. or • Set the signal on "S_Button1" to FALSE to cancel the request for generating signal TRUE on the function block enable output.
8006	Signal TRUE is indicated on "S_Button2" to set the function block enable output to TRUE. Signal FALSE is indicated on "S_Button1".	According to the requirements of your application: • Set the signal on "S_Button1" to TRUE in addition to signal TRUE on "S_Button2" to set the function block enable output to TRUE. or • Set the signal on "S_Button2" to FALSE to cancel the request for generating signal TRUE on the function block enable output.
8007	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, the signal on "S_Button2" switched to FALSE.	Set the signal on "S_Button1" to FALSE.
8008	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, the signal on "S_Button1" switched to FALSE.	Set the signal on "S_Button2" to FALSE.
8009	After the signal on "S_Button1" and the signal on "S_Button2" indicate state TRUE and the enable output has been set to state TRUE, one of the signals on the signal inputs switched to FALSE. The signal on "S_Button1" and the signal on "S_Button2" then indicated state TRUE again.	Set the signal on "S_Button1" and the signal on "S_Button2" to FALSE.
8019	After one of the signals on "S_Button1" and "S_Button2" indicated state TRUE, one of the signals switched from TRUE to FALSE and the other signal switched from FALSE to TRUE.	Set the TRUE signal on one of the signal inputs to FALSE.
C001	When enabling the function block, signal TRUE was indicated on "S_Button1".	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C002	When enabling the function block, signal TRUE was indicated on "S_Button2".	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C003	When enabling the function block, signal TRUE was indicated on "S_Button1" and "S_Button2".	Set the signal on "S_Button1" and the signal on "S_Button2" to FALSE.
C004	Synchronization fault on "S_Button2". When the timer for monitoring synchronization expired, "S_Button1" = TRUE and "S_Button2" = FALSE.	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C005	Synchronization fault on "S_Button1". When the timer for monitoring synchronization expired, "S_Button1" = FALSE and "S_Button2" = TRUE.	Set the TRUE signals indicated on "S_Button1" and/or "S_Button2" to FALSE.
C006	Synchronization error on "S_Button1" and "S_Button2" When the timer for monitoring synchronization expired, "S_Button1" = TRUE and "S_Button2" = TRUE. One of the signals on "S_Button1" or "S_Button2" switched from FALSE to TRUE within the synchronization monitoring time.	Set the signal on "S_Button1" and the signal on "S_Button2" to FALSE.

Table 710: "SF_TwoHandControlTypeIII": Diagnostic codes

6.6.21.6 Signal sequence diagram of the function block

Please note that not all temporary intermediate states are represented in the signal sequence diagram. This diagram only illustrates typical combinations of input signals. Other signal combinations are possible.

The "DiagCode" values in the following figure are specified in hexadecimal.

Signal sequence diagram

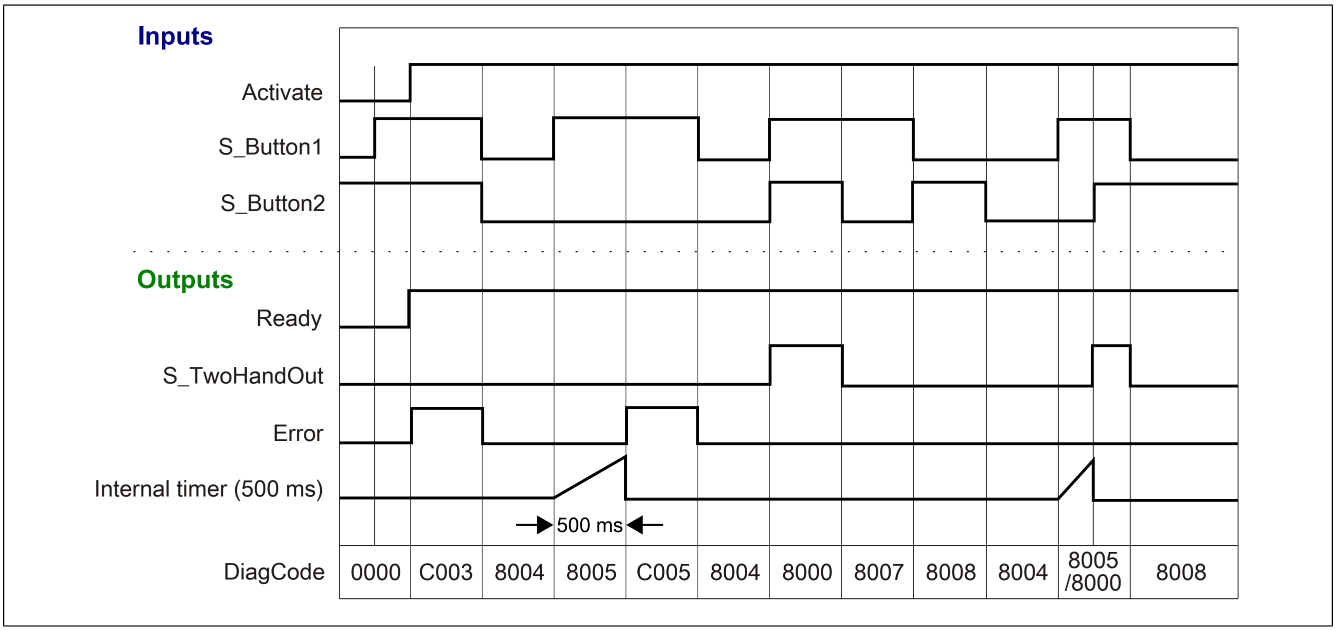


Figure 504: "SF_TwoHandControlTypeIII": Signal sequence diagram

6.6.21.7 Application example

This chapter illustrates a possible application in which the function block can be used to implement a two-hand control device.

Type III two-hand control device

The example in section 6.6.21.7.2 "Type III two-hand control device" illustrates the connection of the function block when controlling using each of the signals from 2 buttons of a two-hand control device connected over 2 channels.

The use of this function block in a real-world application is only permitted after performing a risk analysis.

A direct circuit diagram on a safe input/output device has been avoided on purpose in order to simplify as much as possible the implementation of the application example into a real application.

CAT, PL and SIL values have also been omitted since these classifications always depend on the application where the function block is being used.

> ## Danger!
>
> **Using the function block by itself is not sufficient to execute the safety function in accordance with the CAT, PL or SIL determined from the risk analysis. Additional measures in connection with the safe input/output device being used are required to achieve the safety function. This includes the corresponding wiring and configuration of the inputs and outputs as well as measures for excluding undetectable faults, for example.**
> **For additional information, please see the documentation for the safe input/output device being used.**

6.6.21.7.1 Example: Calling the block and connecting inputs/outputs

The following figure illustrates an example call of this function block.

Instantiation

An instance (call) of the function block is shown here with the name "THC_S5_S6".

The instance name can be shown in the diagnostic tools.

When implementing the example, connect each relevant input and output parameter to an input or output.

The two tables that follow describe the example connections in the following figure.

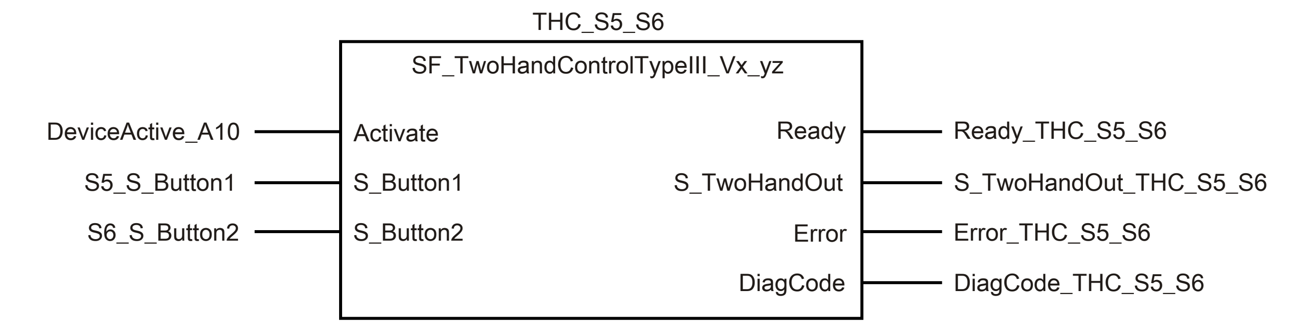


Figure 505: "SF_TwoHandControlTypeIII": Calling the function block and connected inputs/outputs

Name/Literal	Type	Description
DeviceActive_A10	BOOL	Local control of "Activate" from the safety application on the safety controller. Enables the function block. In the example, this input is connected to a signal that indicates the state of the safe devices involved in the safety function. This prevents a safety function from being reported even if the safe devices are inactive.
S5_S_Button1	SAFEBOOL	Signal of the first button from a safe input device. The signal comes from a 2-channel operating element. The evaluation of dual-channel redundancy does not take place in the function block.
S6_S_Button2	SAFEBOOL	Signal of the second button from a safe input device. The signal comes from a 2-channel operating element. The evaluation of dual-channel redundancy does not take place in the function block.

Table 711: "SF_TwoHandControlTypeIII": Inputs connected to input parameters

Name/Literal	Type	Description
Ready_THC_S5_S6	BOOL	Function block active / Function block inactive. Message for further external processing.
S_TwoHandOut_THC_S5_S6	SAFEBOOL	Release signal. This signal is used for further processing in the safety application on the safety controller.
Error_THC_S5_S6	BOOL	Error message from function block for further external processing
DiagCode_THC_S5_S6	WORD	Diagnostic message from function block for further external processing

Table 712: "SF_TwoHandControlTypeIII": Outputs connected to output parameters

6.6.21.7.2 Type III two-hand control device

This example illustrates the connection of the function block when controlling using each of the signals from 2 buttons (two-hand control device, 2-channel antivalent) antivalently connected over 2 channels.

Please note that the complete wiring of the function block is not shown in this chapter. This can be found in section 6.6.21.7.1 "Example: Calling the block and connecting inputs/outputs".

An instance of the function block is shown here with the name "THC_S5_S6".

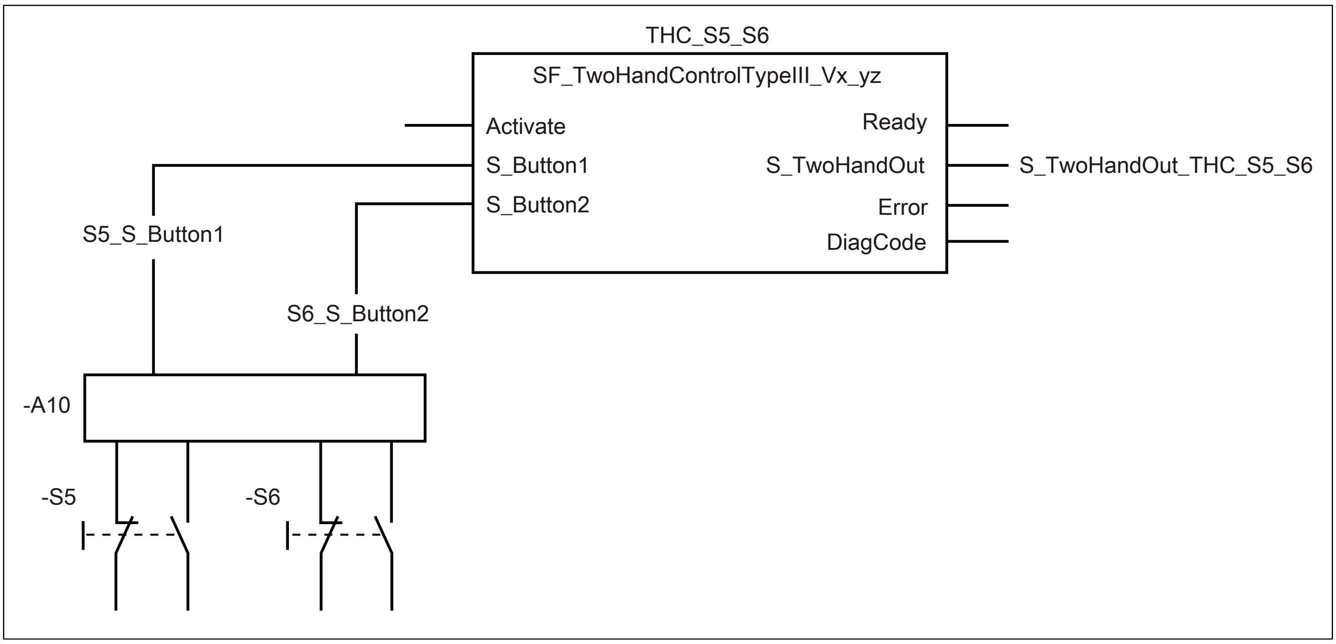


Figure 506: "SF_TwoHandControlTypeIII": Type III two-hand control device

List of equipment

-S5	Two-hand button (2-channel, antivalent)
-S6	Two-hand button (2-channel, antivalent)
-A10	2-channel safe inputs of a safe input device (antivalent) with line control and antivalent monitoring

Please note that, depending on your application, other combinations of safe devices can be used in place of a single safe device.

Connected inputs and outputs

S5_S_Button1	Input on "S_Button1"
S6_S_Button2	Input on "S_Button2"
S_TwoHandOut_THC_S5_S6	Output on "S_TwoHandOut"

Description

In this example:

- The resulting signal (checked for dual-channel redundancy and antivalence) of button "-S5" from the 2-channel input of safe input device "-A10" is connected to input "S5_S_Button1".
- Input "S5_S_Button1" is connected to function block input parameter "S_Button1" for further processing.
- The resulting signal (checked for dual-channel redundancy and antivalence) of button "-S6" from the 2-channel input of safe input device "-A10" is connected to input "S6_S_Button2".
- Input "S6_S_Button2" is connected to function block input parameter "S_Button2" for further processing.
- Output parameter "S_TwoHandOut" is connected to output "S_TwoHandOut_THC_S5_S6".
- Output "S_TwoHandOut_THC_S5_S6" is used as a release signal to control the process in consideration of other safety functions.

6.6.21.8 Implementation of safety requirements from applicable standards

This function block was developed in accordance with the safety requirements listed in this chapter (from applicable standards).

All additional requirements from these standards must be taken into account when implementing the safety function.

The following table lists which requirements from the standards are met by the function block as well as any measures that you will need to implement to fulfill these requirements.

Standard	Chapter	Implementation
EN 574	Use of both hands (simultaneous actuation)	The function block sets it release signal to TRUE only if the signals on "S_Button1" and "S_Button2" switch from FALSE to TRUE simultaneously or successively.
EN 574	Relationship between input signals and output signals / Termination of the output signal	The function block release signal set to TRUE is set to FALSE if "S_Button1" and/or "S_Button2" switches to FALSE.
EN 574	Regeneration of the output signal	If the signals on "S_Button1" and "S_Button2" switch from FALSE to TRUE, the function block only sets its release signal to TRUE if "S_Button1" and "S_Button2" initially indicate state FALSE after the release signal is cut off (release signal = FALSE).
EN 574	Synchronous actuation	The function block monitors the synchronous signal change from FALSE to TRUE on "S_Button1" and "S_Button2". If one of the signals is not set (or delayed) from FALSE to TRUE after the other signal already completed this switch, the function block treats this behavior as a fault. It is not possible to set the function block release signal to TRUE in this error state.
EN 574	Use of categories 1, 3 and 4	The implementation of the two-hand control device in your application must comply with the EN 954-1 requirements to which the two-hand control devices (type IIIA, IIIB or IIIC) are described in EN 574.
EN ISO 12100-2	Startup after power failure / Spontaneous restart	If "S_Button1" and/or "S_Button2" indicate state TRUE when the function block is enabled, then the function block detects this as a fault. If this error message occurs, it is not possible set the release signal to TRUE. To reset this fault, "S_Button1" and "S_Button2" must indicate state FALSE.
EN 954-1	Category 3, 1-channel	You are responsible for implementing the requirements of category 3 for the two-hand control device. A 1-channel configuration of the button is only permitted if the safety chain of the connected button on the two-hand control device complies with EN 954-1 requirements up to category 3 over 1 channel. The following components of the safety chain must be taken into account in each case: • The safe device (input device) of the safe control system, including the device configuration (line control), used in the two-hand control device • The connection and wiring between the safe input device and the button on the two-hand control device • The entire two-hand control device Additional necessary measures can be found in the device-specific user documentation for the devices being used in the safety chain.
EN 954-1	Category 1, 3 or 4	Implement the required categories of the two-hand control device you are using depending on its type (IIIA, IIIB or IIIC) in accordance with EN 574.

Table 713: "SF_TwoHandControlTypeIII": Implementation of requirements from standards

Danger!

The function block does not monitor for possible dual-channel redundancy (line control). It is your responsibility to implement this monitoring in the safety control system outside of the function block.

6.6.22 Version history

Version	Date	Comment
1.20	November 2017	• Function block "SF_EDM": Updated information for "S_EDM1" and "S_EDM2". • Added new function block "SF_Override".
1.10	April 2017	New edition • Changeover to new style sheet. • Editorial revisions.

Table 714: Version history

6.7 ProfiSafe_SF

This library makes it possible to send and receive PROFIsafe data.
The following components are required to use PROFIsafe:

- PROFIsafe gateway - Represents the hardware components
- Function block "SF_ProfiSafeFB_24" - Makes possible secure communication between a B&R F-Device and F-Host

6.7.1 System requirements

Library "ProfiSafe_SF" is part of SafeDESIGNER and only permitted to be used there.

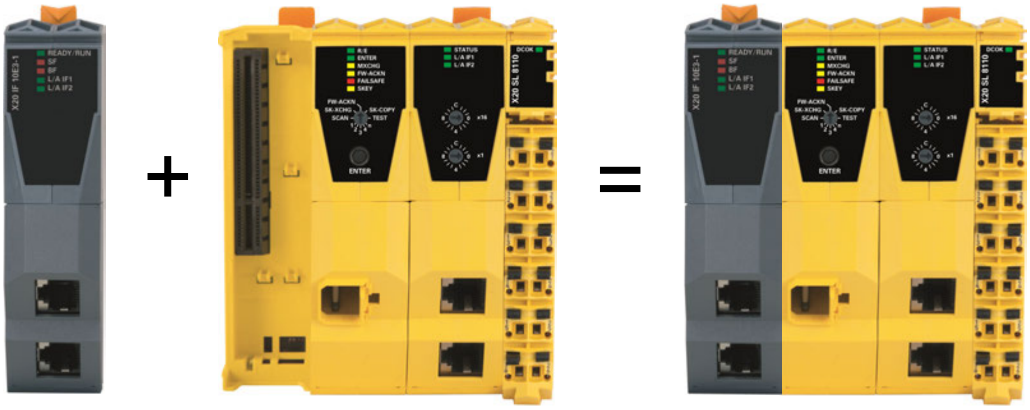
The following requirements apply in order to use library "ProfiSafe_SF":

- SafeDESIGNER: 4.2.3 or later
- Automation Studio: 4.2.5 or later
- Automation Runtime: 4.25 or later
- SafeLOGIC: Safety Release 1.10 or later
- SafeLOGIC-X: Currently not supported
- Software license for using library "ProfiSafe_SF"
- PROFIsafe stack: 2.4

6.7.2 PROFIsafe gateway

"PROFIsafe gateway" refers to the combination of safety controller and standard PROFINET interface module:

- SafeLOGIC X20SL8110
- PROFINET interface module X20IF10E3-1



X20IF10E3-1 X20SL8110 PROFIsafe gateway

Figure 507: PROFIsafe gateway: Term definitions

This chapter describes how the PROFIsafe gateway is used as a safety device for safety applications (subsequently referred to as "F-Device").

It also explains how the F-Device can be used in connection with PROFINET I/O and a safety controller for safety applications (subsequently referred to as "F-Host").

6.7.2.1 Required hardware and software

Hardware

- SafeLOGIC X20SL8110
- PROFINET interface module X20IF10E3-1
- B&R controller with POWERLINK interface
- F-Host

Software

- Automation Studio: 4.2.5 or later
- Automation Runtime: 4.25 or later
- SafeDESIGNER: 4.2.3 or later
- Firmware upgrade for X20SL8110: 323 or later
- Firmware upgrade for X20IF10E3-1: 44 or later
- GSDML file: 2.32 or later

Special information

X20SL8110/X20IF10E3-1 is a device that can only be operated in conjunction with a B&R system. Standalone operation was not intended. Accordingly, PROFINET must consider the B&R system together with the gateway as a closed system. Since there is no absence from feedback here, but in particular it is not desired, the fault event - disconnecting the POWERLINK connection with subsequent reset of the device - is equivalent to a serious hardware error.

6.7.2.2 Description of function

The PROFIsafe gateway is a product that supports the PROFIsafe protocol. It can be used very flexibly and is suitable for satisfying a wide range of requirements.

Some of these include:

- Transferring safe cyclic data
- Setting failsafe states
- Issuing acknowledgment requests
- Handling CRC errors
- Diagnostic messages from failsafe parameters

Data structure

The cyclic data exchanged between the F-Host and F-Device is structured as follows:

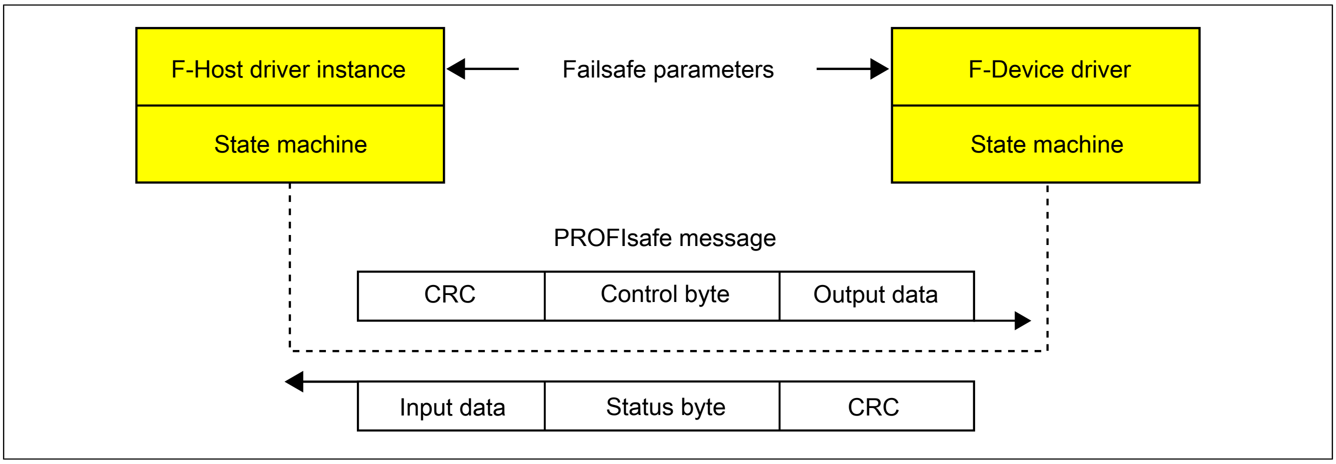


Figure 508: PROFIsafe: Data exchange

Data	Size
Input data (F-Device → F-Host)	4 bytes + 1 integer
Output data (F-Host → F-Device)	4 bytes + 1 integer
CRC	3 bytes
Control byte (F-Host → F-Device)	1 bytes
Status byte (F-Device → F-Host)	1 bytes

Information:

Function block "SF_ProfiSafeFB_24", available in SafeDESIGNER, processes only the payload data (input and output data). For more details, see section "SF_ProfiSafeFB_24" on page 2097.

Direct access to the control byte and status byte is not permitted!

6.7.2.3 Commissioning

Information:

Before commissioning, the required notes regarding PROFINET IO must be observed. These can be found in the PROFINET guidelines for commissioning.

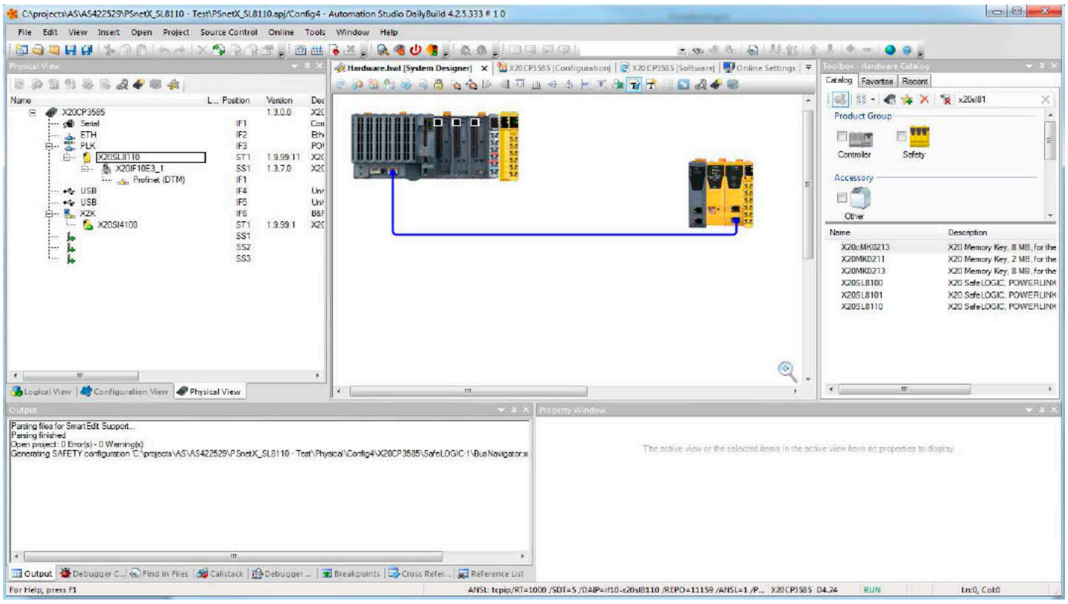
Proceed as follows to use the functionality of PROFIsafe:

- Create the F-Host application.
  For details, see the device documentation for the F-Host.
- Configure the F-Host application.
  In order to use a PROFIsafe module in an F-Host application, the module must be contained in the hardware catalog for the hardware configuration. If this is not the case, the module's associated GSDML file must be installed.

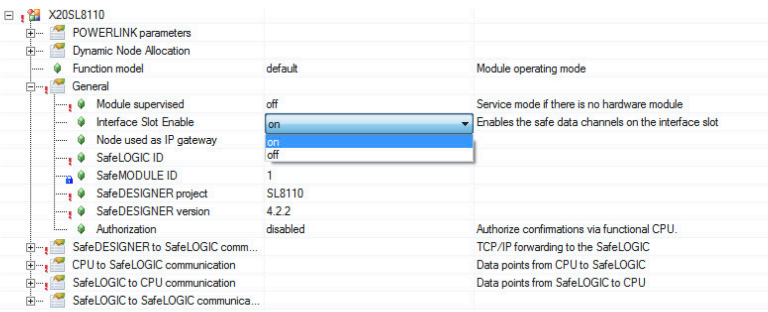
  ## Information:

  The GSDML file provides the information required for PROFINET IO devices as well as for device identification. During startup, this information is checked by the PROFINET IO controller. In the event of error, a corresponding error message is output.

- Arrange, select and add the PROFIsafe device in the PROFINET IO system to slot 1 from hardware catalog "PS_4BYTE_1INT_IO_Safe24" using drag-and-drop.
- Configure the parameters for the PROFIsafe gateway.
- Create and configure the Automation Studio project.
  - This project contains the hardware required for PROFIsafe.

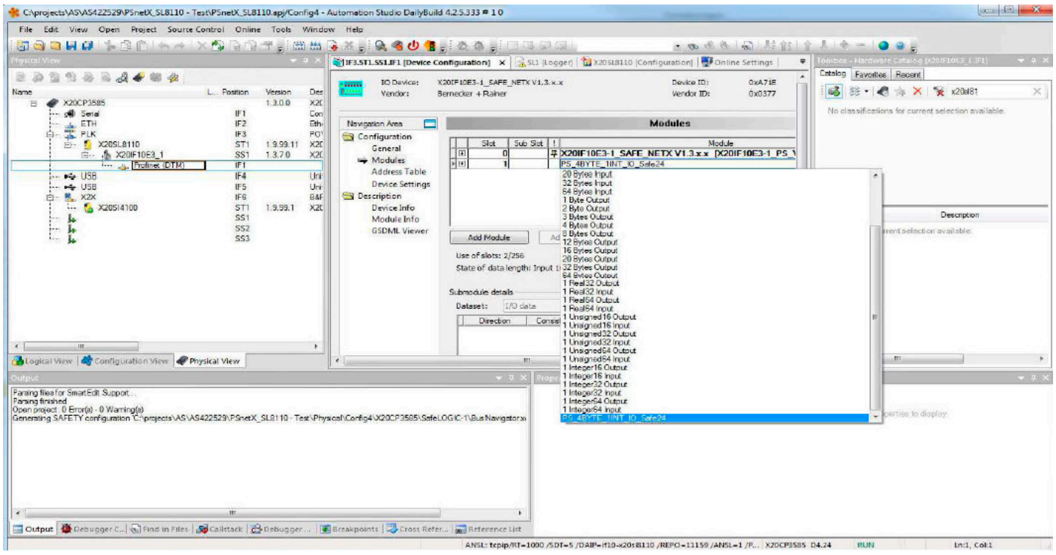


  - The following settings are relevant for SafeLOGIC controller X20SL8110:
    - Interface Slot Enable: On (default value = On)
    - SafeDESIGNER: 4.2.3 or later



  - It is important to make sure that the PROFINET interface module is connected to the X20SL8110.

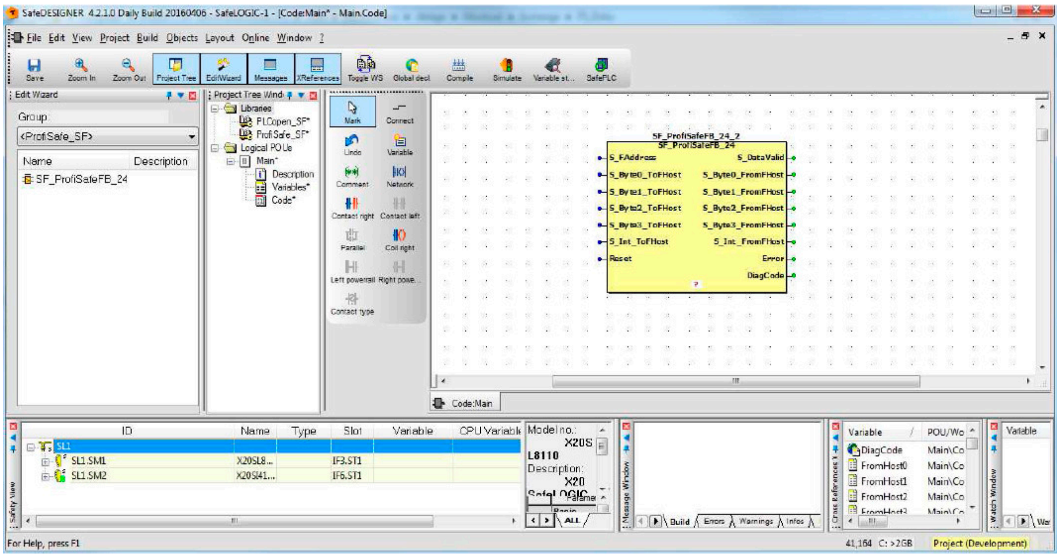
- Configure the PROFIsafe module.
  To do so, select option "Profinet (DTM)" under module X20IF10E3-1 and launch "Device configuration". A list is shown for slot 1. Select entry "PS_4Byte_1INT_IO_Safe24" for the PROFIsafe gateway.



Information:

If no PROFIsafe module exists in the list, use "Tools → Update DTM catalog" to update the DTM catalog.

- Create the SafeDESIGNER application.
  - Start SafeDESIGNER.
  - Add function block "SF_ProfiSafeFB_24".



Information:

Specific notes about function block "SF_ProfiSafeFB_24":

As soon as the application has been transferred to the SafeLOGIC controller and valid communication with the F-Host has been detected, the function block switches to mode "Run" (diagnostic code 0x8000 or 0x8810).

6.7.2.4 Configuration

> ## Information:
> **The PROFIsafe gateway does not have any specific "iParameters".**

Failsafe parameters

Parameter	Type	Description
F_SIL	Bit range	This parameter describes the safety class of the failsafe PROFIsafe gateway. The parameter value is set to SIL 3 and cannot be modified.
F_CRC_Length	Bit range	This parameter specifies the length of the CRC test value and cannot be modified.
F_Par_Version	Bit range	This parameter identifies the supported PROFIsafe mode and cannot be modified.
F_Source_Add	Unsigned 16	This parameter returns the address for identifying the source and cannot be modified.
F_Dest_Add	Unsigned 16	This parameter returns the address for identifying the target. The value can be chosen from within the permitted range. • Permitted values: 1 to 65534 The value configured in the F-Host application must be configured in function block "SF_ProfiSafeFB_24" in SafeDESIGNER.
F_WD_Time	Unsigned 16	This parameter defines the monitoring time (in ms). The F-Host must receive a valid, current safety telegram within the monitoring time. If this does not take place, the F-Device switches to the safe state. The monitoring time should generally be chosen high enough so that the telegram propagation times required for communication are tolerated. It still must be ensured that the error reaction function is executed quickly enough in the event of error, however. • Recommended values: 100 to 10,000 ms
F-Par_CRC	Unsigned 16	This parameter describes the checksum of all of the device's failsafe parameters. Parameter "F-Par_CRC" ensures the transfer of the failsafe parameters and cannot be modified. The calculation takes place externally in the F-Host's engineering tool.

Table 715: PROFIsafe: Overview of failsafe parameters

6.7.2.5 Error description and solution

If function block "SF_ProfiSafeFB_24" detects an error in mode "Run", it switches to the safe state (diagnostic code 0x8810) and the data on the output is set to "0".

The safe state is enabled in the following cases:

- On system start. Acknowledgment by the user is required.
- On hardware error. This includes disconnected cables, power failure on the F-Host, etc.
- On error involving safe parameters. This includes the incorrect configuration of "F_Source_Add", "F_Dest_Add", "F_SIL", "F_WD_Time", etc.
- On an error involving safety-oriented communication between the F-Host and F-Device.

The safe state is left when the input and output data are again provided with valid values. In order to replace the "0" values with valid process values, acknowledgment by the user is required after an error occurs.

Frequently occurring error scenarios

1. An incorrect value was predefined for "F_Dest_Add" in SafeDESIGNER. The value must correspond to the value that was defined for the F-Host under "PS_4BYTE_1INT_IO_Safe24".
   In the event of error, a corresponding error message is output on the F-Host.
2. Function block "SF_ProfiSafeFB_24" remains in state "INIT" (0x8001) after receiving and checking the safe parameters.
   PROFIsafe data exchange is not started, and the output data is set to "0".

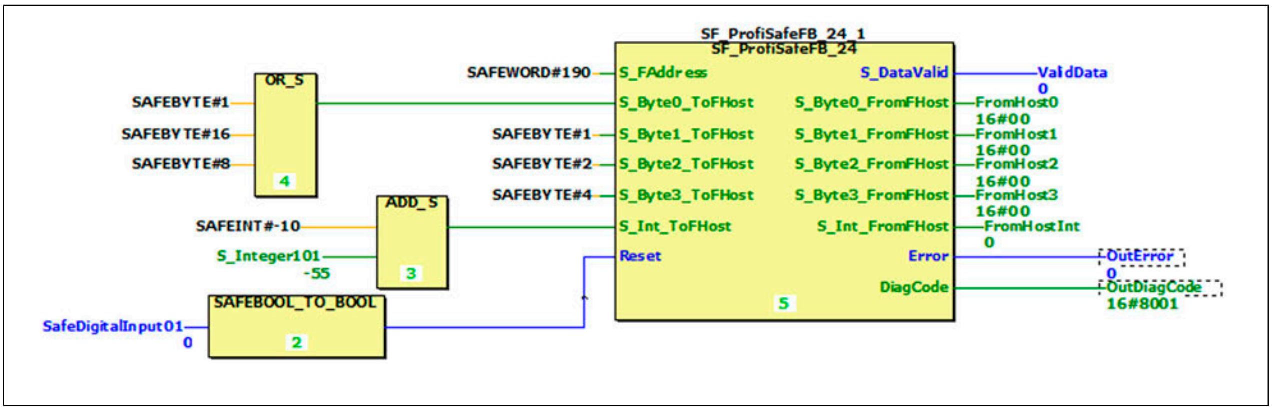


Figure 509: Example of hardware fault

3. The PROFIsafe protocol detects an error on configured parameter "F_WD_Time" and switches to the safe state (0x8810). Output parameter "S_DataValid" is set to FALSE.
   To leave this error state, the value on "F_WD_Time" must be modified.

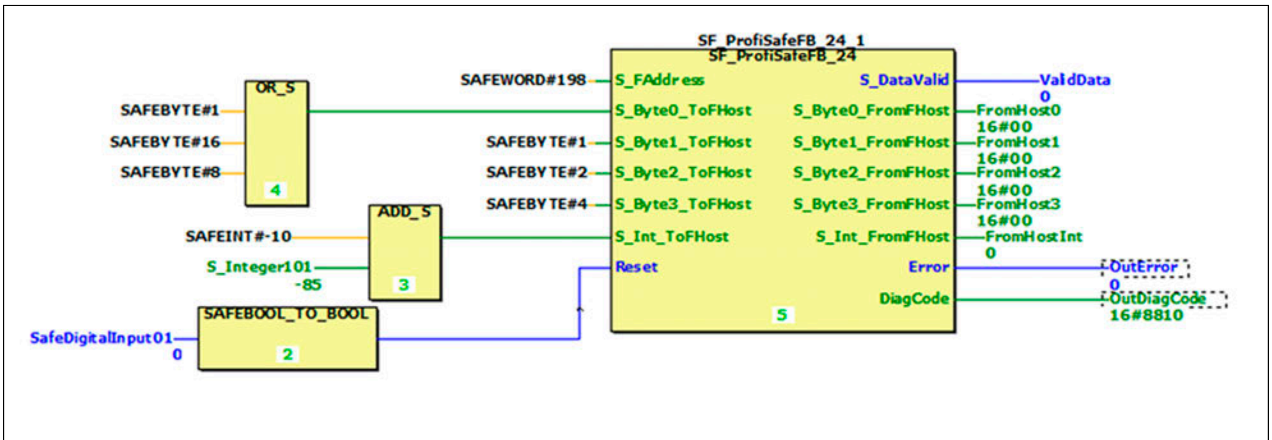


Figure 510: Example for faulty monitoring time ("F_WD_Time")

6.7.2.6 X20IF10E3-1 - LED status indicators

LED	Color	Cause	Error	S_DataValid	DiagCode	Solution
READY/RUN	Green	F-Device not synchronized with F-Host	FALSE	FALSE	0x8810	Acknowledgment of the safety application by the user is required.
SF	Off					
BF	Off					
L/A IF1/2	Green			TRUE	0x8000	-
READY/RUN	Green	F-Host not connected or failed	FALSE	FALSE	0x8001 0x8810	Checking the F-Host's power supply and cable connection is required.
SF	Off					
BF	Red					
L/A IF1/2	Off					
READY/RUN	Green	Error in safe parameters (e.g. "F_Dest_Add", "F_SIL", "F_WD_Time" 0 value)	FALSE	FALSE	0x8001 0x8002	The I/O device diagnostic buffer makes it possible to detect incorrectly configured safe parameters (see section "Configuration" on page 2093). The diagnostic alarm on the I/O controller can be triggered by a configuration error. Correct the configuration error.
SF	Red					
BF	Off					
L/A IF1/2	Green					
READY/RUN	Green	Error in the safety-related communication between the F-Host and F-Device (e.g. open circuit)	FALSE	FALSE	0x8810	Checking the configured monitoring time ("F_WD_Time") is required.
SF	Off					
BF	Off					
L/A IF1/2	Green					
READY/RUN	Green	"PSD_HARD_FAIL" is active (e.g. due to inconsistency, communication error)	TRUE	FALSE	0xC010	A reset or restart of the F-Device is required.
SF	Off					
BF	Off					
L/A IF1/2	Green					
READY/RUN	Green	F-Host not found	FALSE	FALSE	0x8001	Checking the F-Host's power supply, cable connection, device name and IP address is required.
SF	Off					
BF	Blinking					
L/A IF1/2	Green					

Table 716: X20IF10E3-1 - LED status indicators

Information:

Checking the F-Host's diagnostic data as well as the diagnostic data from the Automation Studio application (Logger) is required.

6.7.3 Term definitions

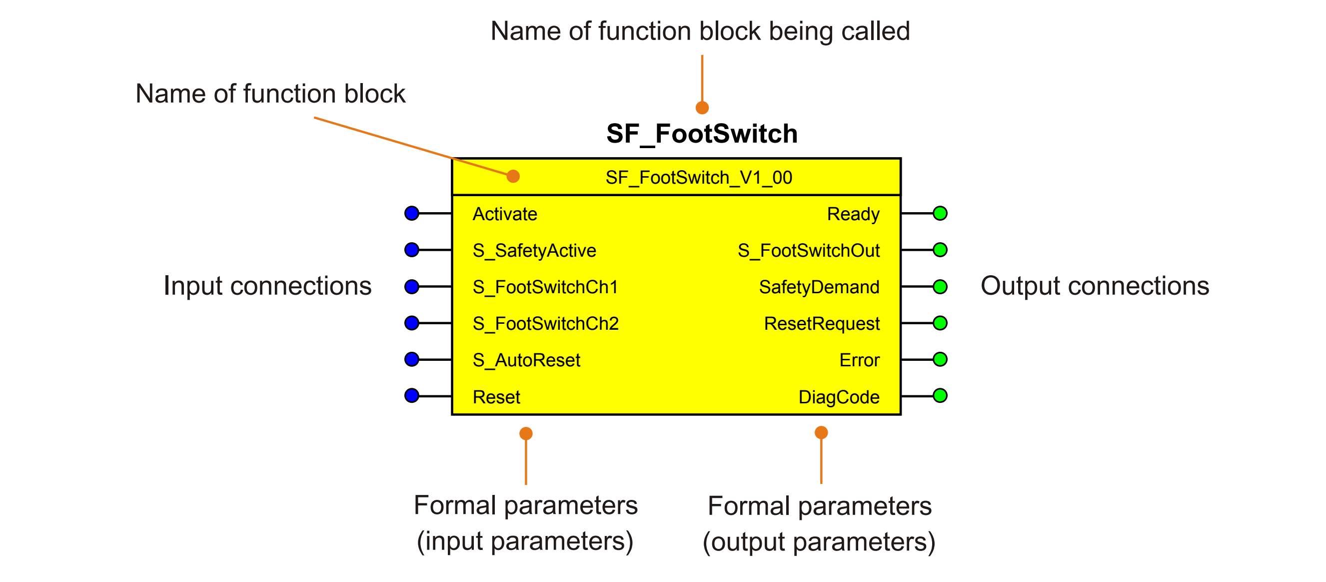


Figure 511: Components of a function block

When calling a function block, the inputs supply the input parameters with the current values of the variables or constants.
The output parameters supply the outputs with the associated values.

Inputs and outputs do not need to have the same name as the associated formal parameters, but they must be of the same data type. A difference in data type between formal parameters and inputs/outputs is reported as an error after compilation.

The name of a function block is composed of the function itself (e.g. "SF_FootSwitch", SF = safety function) and its version (Vx_yz). The format used to represent the version number in this document, Vx_yz, is a placeholder. For the actual version, see the function block being used.

6.7.3.1 Extended constants

Constants can be used in a safety application. These have the property that their values cannot be modified during runtime. For flexible solutions, however, it may be necessary to safely adjust parameters during runtime. "Safe commissioning options" are available for this purpose.
Extended constant means that both the conventional constants as well as the safe options are permitted to be connected on the input parameter.
Connecting an extended constant with a safe variable is technically not latched. However, all parameters specified as extended constants are not permitted to change during runtime; otherwise, the function block will issue an error message at runtime.

Possible applications of extended constants:

- Connecting with a SafeDESIGNER constant, e.g. SAFEINT#315
- Connecting with a SafeDESIGNER-global constant
- Connecting with a safe commissioning option

6.7.4 SF_ProfiSafeFB_24

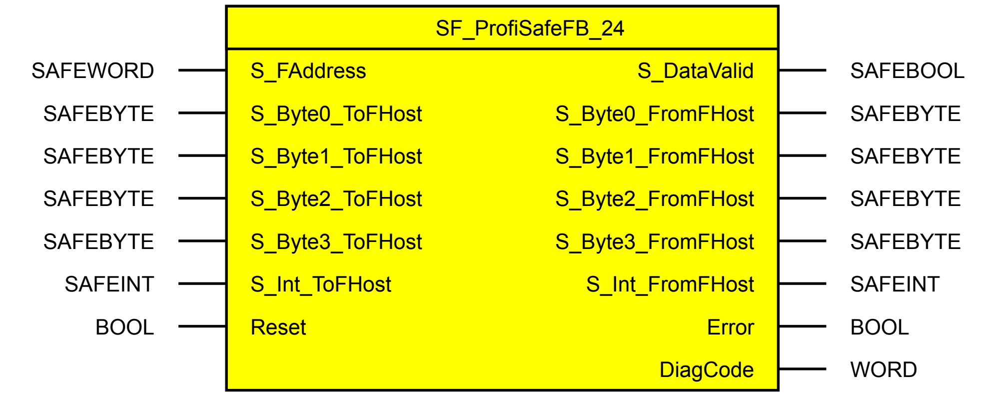


Figure 512: Function block "SF_ProfiSafeFB_24"

6.7.4.1 Formal parameters of the function block

In the following, the term "variable" refers to both a variable as well as a graphic connection.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
S_FAddress	SAFEWORD	Extended constant	Status	16#0000	Address of the F-Device
S_ByteX_ToFHost (X = 0 to 3)	SAFEBYTE	Variable	Status	0	Bytes that are transmitted from the F-Device to the F-Host
S_Int_ToFHost	SAFEINT	Variable	Status	0	Integer value that is transmitted from the F-Device to the F-Host
Reset	BOOL	Variable	Edge	FALSE	Resets error messages when the cause of the error no longer exists

Table 717: Overview of the input parameters

1) Evaluation of the input parameter signals in the function block. You must control the signals accordingly.

Name	Type	Connection	Signal type [1]	Initial value	Description / General function
S_DataValid	SAFEBOOL	Variable	Status	FALSE	Indicates valid output data
S_ByteX_FromFHost (X = 0 to 3)	SAFEBYTE	Variable	Status	0	Bytes that are transmitted from the F-Host to the F-Device
S_Int_FromFHost	SAFEINT	Variable	Status	0	Integer value that is transmitted from the F-Host to the F-Device
Error	BOOL	Variable	Status	FALSE	Function block error message
DiagCode	WORD	Variable	Status	16#0000	Function block diagnostic message

Table 718: Overview of the output parameters

1) Output of the output parameter signals. You must evaluate and/or further process the signals accordingly.

6.7.4.2 Function

Function block "SF_ProfiSafeFB_24" makes possible safe communication between the B&R F-Device and an F-Host. The function block's input data represents the data that is received by the F-Host. The data transmitted by the F-Host is located on the function block's output.

4 channels of data type BYTE and 1 channel of data type INT are available to the user in each direction.

4 SafeLOGIC cycles are always required to execute the function block. This must be taken into account when defining the PROFIsafe timeout parameters.

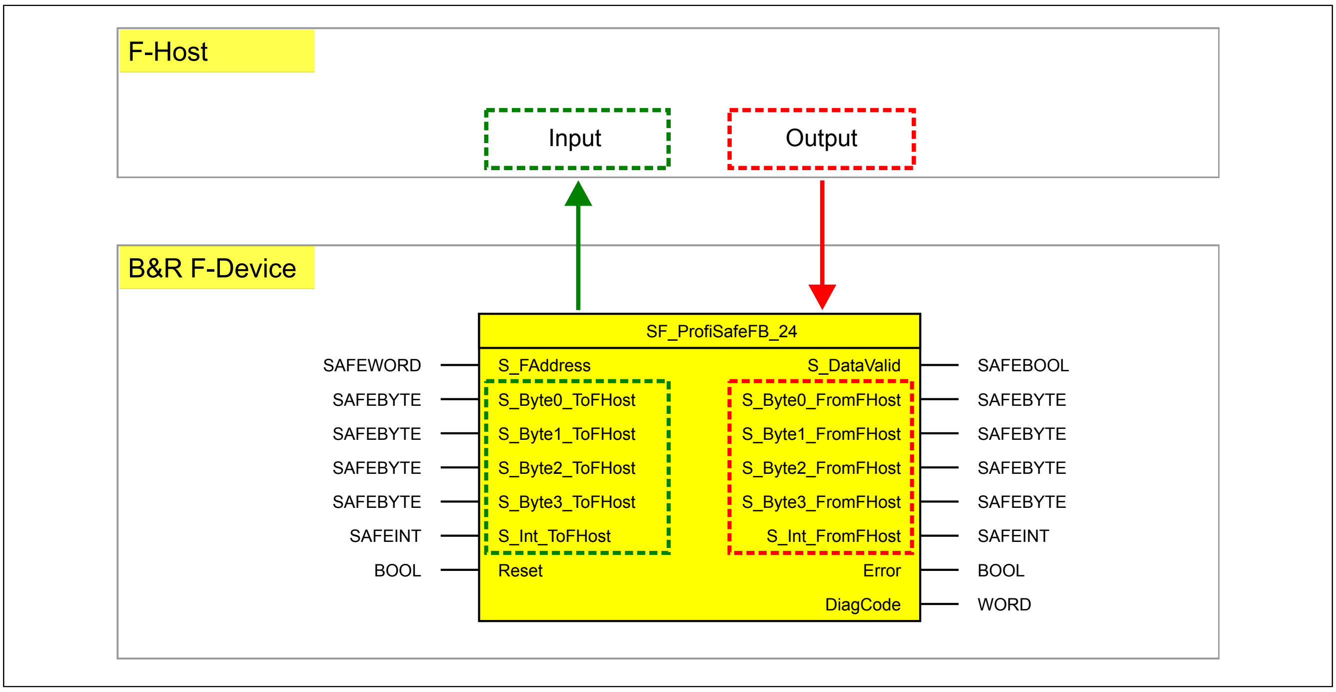


Figure 513: Data exchange between F-Host and F-Device

6.7.4.3 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> ## Danger!
>
> **Always validate the entire safety function!**

6.7.4.3.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants within the valid range are in fact incorrect for the safety function being executed. A static TRUE signal on input "Reset" is detected by the function block and reported as a fault, however.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> ## Danger!
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.7.4.3.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.7.4.3.3 Simultaneous edge change

To reduce the risk of unexpected startup, make sure that you only connect input parameter "Reset" to the signal of a manual reset device. This signal is based on your risk analysis.

6.7.4.3.4 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.7.4.4 Input parameters

6.7.4.4.1 S_FAddress

General function

- Address of the F-Device

Data type

- SAFEWORD

Connection

- Extended constant

Function description

The value predefined on input "S_FAddress" corresponds to the address of the F-Device.

6.7.4.4.2 S_ByteX_ToFHost (X = 0 to 3)

General function

- Bytes that are transmitted from the F-Device to the F-Host

Data type

- SAFEBYTE

Connection

- Variable

Function description

These inputs involve the data of type SAFEBYTE that is sent to the F-Host. 4 individual safe inputs are available to the user for this.

6.7.4.4.3 S_Int_ToFHost

General function

- Integer value that is transmitted from the F-Device to the F-Host

Data type

- SAFEINT

Connection

- Variable

Function description

This input involves the data of type SAFEINT that is sent from the F-Device to the F-Host. 1 safe input is available to the user for this.

6.7.4.4.4 Reset

General function

- Input parameter for resetting error messages once the error has been corrected

Data type

- BOOL

Danger!

Depending on the safety requirements, the data type SAFEBOOL or BOOL must be connected. You will determine which data type is appropriate in the course of your risk analysis.

Depending on the results of the risk analysis, it may be necessary to connect the SAFEBOOL data type. This prevents unexpected startup caused by errors in the standard system. This connection can be made using the "SAFEBOOL_TO_BOOL" data type converter.

Further preventive measures such as additional function stops can be used to further reduce unexpected startup.

Connection

- Variable

Function description

The edges of this input parameter are monitored internally by the function block. The function is only executed on a rising edge of input parameter "Reset". An additional static TRUE signal following a rising edge does not trigger the function again.

The function block detects a static TRUE signal on input parameter "Reset" as an error in states where a rising edge on "Reset" is required. Set "Reset" to FALSE to exit the error state.

This input parameter is used to acknowledge a start interlock or reset errors detected by the function block once the source of the error has been corrected.

6.7.4.5 Output parameters

6.7.4.5.1 S_DataValid

General function

- Indicates valid output data

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output indicates the validity of the data received by the B&R F-Device.

TRUE

The data pending on the output is valid.

FALSE

The data pending on the output is invalid. There is a problem receiving data. Output parameter "DiagCode" returns additional information.

6.7.4.5.2 S_ByteX_FromFHost (X = 0 to 3)

General function

- Bytes that are transmitted from the F-Host to the F-Device

Data type

- SAFEBYTE

Connection

- Variable

Function description

These outputs involve the data of type SAFEBYTE that is sent from the F-Host to the F-Device. 4 individual safe outputs are available to the user for this.

6.7.4.5.3 S_Int_FromFHost

General function

- Integer value that is transmitted from the F-Host to the F-Device

Data type

- SAFEINT

Connection

- Variable

Function description

This output involve the data of type SAFEINT that is sent from the F-Host to the F-Device. 1 safe output is available to the user for this.

6.7.4.5.4 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

To leave an error state ("Error" = TRUE), input parameter "Reset" must be set to FALSE if there is a static TRUE signal on "Reset".
In other error states (see the "Diagnostic code" table), you must set input parameter "Reset" from FALSE to TRUE.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.7.4.5.5 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter is used to output specific diagnostic and status messages from the function block.

The function block indicates the presence of an error message via output parameter "Error".

Additional details can be found in the "Diagnostic codes" table.

Diagnostic code

The diagnostic code is specified as a WORD data type. The following table specifies each value in hexadecimal form.

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.7.4.5.6 Overview of diagnostic codes

Code (hex)	Description	Corrective measures
8000	The data on the output is valid ("S_DataValid" = TRUE).	• No corrective measures are required.
8001	The function block is initialized and waiting for valid data.	• Transmit valid data.
8002	Parameter inconsistency. Invalid data was received. This occurs on CRC errors, incorrect data length or if the source or destination address is incorrect, for example.	• Check the configuration and parameter settings.
8810	PROFINET communication is aborted.	• Check whether a connection exists between the devices. • Check the configuration and parameter settings.
C010	General transfer or parameter error.	• Check the configuration and parameter settings.

Table 719: "SF_ProfiSafeFB_24": Diagnostic codes

6.7.5 Version history

Version	Date	Comment
1.31	May 2019	• Added chapter 6.7.3.1 "Extended constants". • Input parameter "S_FAddress": Updated connection.
1.30	June 2018	• Chapter 6.7.2.1 "Required hardware and software": Added special information.
1.20	March 2018	• Chapter 6.7.2.4 "Configuration": Removed "F_Block_ID" and "F_iPar_CRC". • Editorial changes.
1.11	February 2017	• Updated chapter 6.7.1 "System requirements". • Chapter 6.7.2.3 "Commissioning": Updated parameter name "Interface Slot Enable". • Chapter 6.7.4.1 "Formal parameters of the function block": Updated description of "S_ByteX_ToFHost" and "S_ByteX_FromFHost".
1.10	December 2016	First edition

Table 720: Version history

6.8 SafeOPTION_SF

A description of the SafeOPTION functions in provided in Automation Help (see section MpSafetyX).

6.8.1 Overview

Overview of function blocks in library "SafeOPTION_SF":

Function block	Description
SF_SafeOPTION_MN	This function block is responsible for the communication between the SafeLOGIC controller and standard controller on the SafeDESIGNER side.
SF_SafeOPTION_BOOL	This function block makes it possible to use safe machine options of data type BOOL.
SF_SafeOPTION_INT	This function block makes it possible to use safe machine options of data type INT.

6.8.2 System requirements

Library "SafeOPTION_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements apply in order to use library "SafeOPTION_SF":

- mapp upgrade: 1.50 or later
- SafeDESIGNER: 4.2.3 or later
- SafeLOGIC: HW-Upgrade 1.10.1 or later
- Safety Release: 1.10 or later

6.9 Table_SF

This library evaluates table types.

6.9.1 General information

This chapter lists the system requirements for using library "Table_SF". Major changes compared to the previous version are listed in the version history.

6.9.1.1 System requirements

Library "Table_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements must be met in order to use library "Table_SF":

- SafeDESIGNER: 2.90 to 3.0 or 4.1 or later
- Automation Studio: 3.0.90 or later
- SafeLOGIC: Safety Release 1.5 or later
- SafeLOGIC-X: Currently not supported

6.9.1.2 Version history

Version	Date	Comment
1.51	February 2019	• Chapter 6.9.2.1.1 "Table data": Updated description. • Chapter 6.9.2.1.2 "Table format A": Updated description.
1.50	September 2017	• Restructuring: Placed descriptions of input and output parameters used by multiple function blocks in a central location. Shortened descriptions for individual input and output parameters. • Added table format E. • Added new function blocks: – SF_TableTypeE_Read – SF_TableTypeE_Write – SF_TableTypeE_Reset
1.40	July 2017	• Created separate section for available table formats. • Added new function block: – SF_TableTypeB_Read • Editorial changes
1.30	February 2016	First edition

Table 721: Version history

6.9.2 Technical information

This chapter contains information about tables and table formats as well as generally valid descriptions of the input and outputs parameters for the function blocks being used.

6.9.2.1 Tables

List of the table formats contained in this library and supported function blocks sorted in descending order starting with the version from which they are available.

Supported starting with SafeDESIGNER	Name	Short description	Supported by
4.3.2	Table format E	Array[X][Y] with configurable number of entries.	SF_TableTypeE_Read SF_TableTypeE_Reset SF_TableTypeE_Write
4.3.2	Table format B	Lookup table - A value is located at unique position X/Y/Z.	SF_TableTypeB_Read
2.90	Table format A	Lookup table - A value is located at position X/Y.	SF_TableTypeA

6.9.2.1.1 Table data

Table data can be edited via SafeDESIGNER. It uses the program for editing CSV files for this.

> **Information:**
>
> **To edit a table object, right-click on it in SafeDESIGNER and select "Edit".**

File structure

The file is divided into a header and data section. This structure must be maintained at all times; otherwise, the file can no longer be processed correctly. A CRC value, timestamp and the name of the Windows user who locked the file are entered in the file automatically.

The user specifies the row or column where the data begins and ends. Additional information (notes/comments) can also be included as long as it does not overlap with the data section (row/column).

The file size is calculated from the number of table entries and their data type.

Header section

- TableID:
  - "TableID" is updated by the system automatically when a table is edited or imported.
  - Permissible values: 1 to 99
  - Depends on the table object being used in SafeDESIGNER
- TableFormat: Specifies the type of table
- Data: Specifies the row or column where the data begins and ends (defined by the user)
- Additional parameters valid for the entire table (defined by the user)

Data section

The data section is determined by the table type or table format being used.

6.9.2.1.2 Table format A

Function description

This table format makes it possible to define a lookup table with tolerance parameters.

Header section

- MaxToleranceX:
    - Maximum tolerance range for the x-position
    - Checked against input parameter "S_X_Tolerance". The value on input parameter "S_X_Tolerance" is not permitted to exceed this value.
- MaxToleranceY:
    - Maximum tolerance range for the y-position
    - Checked against input parameter "S_Y_Tolerance". The value on input parameter "S_Y_Tolerance" is not permitted to exceed this value.

Data section

- X-position
- Y-position
- Return value (column "Byte code")

	A	B	C	D	E
1	TableID	1			
2	TableFormat	A			
3	Data	C	7	E	17
4	MaxToleranceX	10			
5	MaxToleranceY	10			
6	Koordinate-X	Koordinate-Y	X-Position	Y-Position	Bytecode für Z
7	0	0	2300	1088	3
8	0	1	2300	2692	3
9	0	2	2300	2928	3
10	0	3	2300	4892	3
11	0	4	2300	5132	3
12	0	5	2300	7092	3
13	0	6	2300	7330	3
14	0	0	2359	1088	3
15	0	1	2359	2692	3
16	0	2	2359	2928	3
17	0	3	2359	4892	3
18					
19	649386282				
20	MaxMustermann	24.04.2012 15:05			

Figure 514: Table format A - Example

Information:

Positions must be sorted in ascending order.

- **Multiple (ascending) y-positions can be defined for one x-position.**
- **Multiple (ascending) x-positions can be defined for one y-position.**

6.9.2.1.3 Table format B

Function description

This table format makes it possible to define a lookup table with 3 search indexes (x, y, z).

Header section

There are no user-specific parameters for this table format.

Data section

- X: Value for the x-position
- Y: Value for the y-position
- Z: Value for the z-position
- Value: Value at the x/y/z-position
- Values are of data type SAFEDINT.

The following example illustrates the data section for this table type. It shows the data section highlighted in yellow. The gray cells are there to provide support for the user and not permitted to be included when defining the section.

Example: The data section is located between cells A5 and D15. Value 1088 is located at position X=0 / Y=2 / Z=1; value 2928 is located at position X=1 / Y=0 / Z=1.

	A	B	C	D	E
1	TableID	1			
2	TableFormat	B			
3	Data	A	5	D	15
4	X	Y	Z	Value	
5	0	0	0	1088	
6	0	0	1	2692	
7	0	0	2	2928	
8	0	1	0	4892	
9	0	1	1	5132	
10	0	1	2	7092	
11	0	2	0	7330	
12	0	2	1	1088	
13	0	2	2	2692	
14	1	0	1	2928	
15	1	0	2	4892	
16					
17	649386282				
18	MaxMustermann	24.04.2012 15:05			

Figure 515: Table format B - Example

Information:

Positions must be sorted in ascending order. The combination of x-, y- and z-values must be unique.

- **Multiple (ascending) y- and z-positions can be defined for one x-position.**
- **Multiple (ascending) z-positions can be defined for one x- and y-position.**

6.9.2.1.4 Table format E

Function description

This table format makes it possible to define a two-dimensional array.

Header section

The size of the array is defined by "DimensionX" and "DimensionY". A check is carried out as to whether this data covers the defined data section ("Data"). If this is not the case, SafeDESIGNER returns an error.

- DimensionX (rows): Defines the number of required x-values.
- DimensionY (columns): Defines the number of required y-values.

Data section

Values of data type SAFEDINT are located in the data section. The size of the section must match the numbers for "DimensionX" and "DimensionY". If this is not the case, SafeDESIGNER returns an error when "locking" the table.

The following example illustrates the data section for this table type. It shows the data section highlighted in yellow. The gray cells are there to provide support for the user. They represent the x-index and y-index of the array and are not permitted to be included when defining the section.

Example: The data section is located between B8 and F11 with 4 x-values and 5 y-values. Value 1200 is located at position Array[0][0] – cell B8; value 2 is located at position Array[3][4] – cell F11.

	A	B	C	D	E	F
1	TableID	1				
2	TableFormat	E				
3	Data	B	8	F	11	
4	DimensionX	4				
5	DimensionY	5				
6						
7		Y0	Y1	Y2	Y3	Y4
8	X0	1200	4	4456	12	789
9	X1	1300	745	9	113	1000
10	X2	2100	6	789	112	561
11	X3	10	0	0	456	2
12						
13	649386282					
14	MaxMustermann	24.12.2016 15:05				

Figure 516: Table format E - Example

Information:

The x- and y-values must be unique and sorted in ascending order.

The entire range of the array must be filled with values. Gaps are not permitted.

6.9.2.2 General input parameters

This section describes general input parameters that are implemented in multiple function blocks in this library.

6.9.2.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter enables the function block.

- If you are switching safe devices on or off, connect "Activate" to a variable that indicates the state (switched on or off) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is switched off.
- If "Activate" indicates state TRUE when a cold restart of the safety controller is performed, then the function block will behave exactly the same during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/peripherals on the function block do not result in an invalid diagnostic message.
- You also have the possibility of connecting "Activate" to a constant (TRUE) in order to enable the function block. In this case, the function block interprets signal FALSE from an inactive safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing on the function block interface that can be used to determine whether a triggered safety function or an inactive safe device is the cause of the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnostic information from the function block for error messages from safe devices and/or disabled safe devices according to your own diagnostic concept, connect "Activate" to a signal that indicates the state of the safe devices involved in the safety function supported by the function block. Form this signal using only safe devices whose I/O signals are connected to the function block via input parameters. In this way, you will prevent safety functions triggered by inactive safe devices from being reported. This measure is only used to control the diagnostic information from inactive safe devices in a defined manner.

6.9.2.2.2 S_TableID

General function

- Assigns a table object to the function block

Data type

- TableID

Connection

- Constant

Function description

This input parameter assigns a table object to the function block.

Connect the corresponding table object to this input parameter in SafeDESIGNER using drag-and-drop.

> ## Information:
>
> **The same "TableID" can be used in multiple table function blocks of the same table type in the safety application.**

6.9.2.3 General output parameters

This section describes general output parameters that are implemented in multiple function blocks in this library.

6.9.2.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.9.2.3.2 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending function block error message. Each error message results in the safe output parameters being set to FALSE or 0 and remaining in this state.

It is your responsibility to ensure that all necessary corrective measures are initiated after an error occurs since subsequent errors can result in a hazard!

TRUE

The enabled function block detected an error.

The safe output parameters for handling information in bit form are set to FALSE.
The safe output parameters for handling information in numerical form are set to 0.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.9.2.3.3 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.9.3 Function blocks

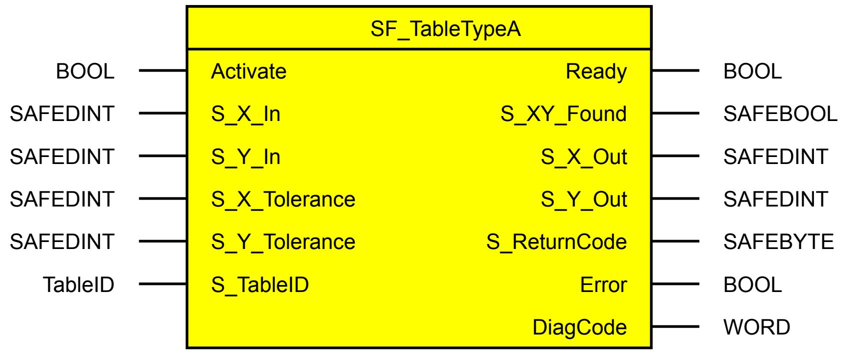
List of the function blocks contained in this library sorted in descending order starting with the version from which they are available.

Supported starting with SafeDESIGNER	Name	Short description
4.3.2	SF_TableTypeE_Read	This function block reads the data from the table of type E.
4.3.2	SF_TableTypeE_Reset	This function block resets the data from the table of type E to its initial values.
4.3.2	SF_TableTypeE_Write	This function block modifies at runtime the data that is stored in volatile memory from the table of type E.
4.3.2	SF_TableTypeB_Read	This function block reads the data from the table of type B.
2.90	SF_TableTypeA	This function block evaluates x- and y-coordinates and returns a corresponding value for each.

6.9.3.1 SF_TableTypeA

This function block evaluates x- and y-coordinates and returns a corresponding value for each.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_X_In	SAFEDINT	Variable	0	Input parameter for the x-position
IN	S_Y_In	SAFEDINT	Variable	0	Input parameter for the y-position
IN	S_X_Tolerance	SAFEDINT	Variable/ Constant	0	Input parameter for the tolerance of the x-position. This value is not permitted to be greater than the value defined for "MaxToleranceX" in the table.
IN	S_Y_Tolerance	SAFEDINT	Variable/ Constant	0	Input parameter for the tolerance of the y-position. This value is not permitted to be greater than the value defined for "MaxToleranceY" in the table.
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_XY_Found	SAFEBOOL	Variable	FALSE	Status information indicating whether an entry was found for the position (± tolerance). TRUE: The current x- and y-position were found in the table.
OUT	S_X_Out	SAFEDINT	Variable	0	Indicates the found x-position
OUT	S_Y_Out	SAFEDINT	Variable	0	Indicates the found y-position
OUT	S_ReturnCode	SAFEBYTE	Variable	0	Return value for the current position
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.9.3.1.1 Function description

Function block "SF_TableTypeA" evaluates x- and y-coordinates and returns a corresponding value for each.

This function block takes the position (input parameter "S_X_In" / "S_Y_In") with the specified tolerance range (input parameter "S_X_Tolerance" / "S_Y_Tolerance") and checks it against the entries in the table object (see Fig. 514 "Table format A - Example", column "X-position" / "Y-position"). If a match is found (Position ± Tolerance), then the return value defined for this position (see Fig. 514 "Table format A - Example", column "Byte code") is returned on output parameter "S_ReturnCode".

This match is indicated by state TRUE on output parameter "S_XY_Found". In addition, the values defined in the table for "X-position" and "Y-position" are made available on output parameters "S_X_Out" and "S_Y_Out".

If no match is found for the current coordinates ("S_XY_Found" = FALSE), then output parameters "S_X_Out" and "S_Y_Out" are set to 0.

Information:

In addition to checking the position, the specified tolerance values on input parameters "S_X_Tolerance" and "S_Y_Tolerance" are compared with the maximum permissible tolerance from the table (column "MaxToleranceX" / "MaxToleranceY").

Information:

The function block uses table format A.

General procedure

- Select the table object in the Safety View and edit the table data.
- Lock and check the table data.
- Use the function block in the safety application.
- Link the table object to the function block.

6.9.3.1.2 Fault avoidance

Danger!

Validation

In addition to validating the safety function implemented with the function block, it is also necessary during validation to verify that the table data is correct.

6.9.3.1.3 Status and error information

Additional details about output parameters that provide status and error information.

6.9.3.1.3.1 Status numbers

Information:

In the event of any error, state "FailSafe" is triggered on the safety controller. For corresponding error codes, see the Safety Logger.

Errors

Code (hex)	Description	Corrective measures
C025	The function block is reporting an error. State "FailSafe" is triggered on the safety controller. For additional information, see the Safety Logger.	Check the messages in the Safety Logger.

Table 722: "SF_TableTypeA": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block finds an entry for the current coordinates (± tolerance) and returns the corresponding return value on output parameter "S_ReturnCode".	No corrective measures are required.
8002	The function block does not find an entry for the current coordinates (± tolerance).	Change the coordinates to find a valid entry.

Table 723: "SF_TableTypeA": Diagnostic codes

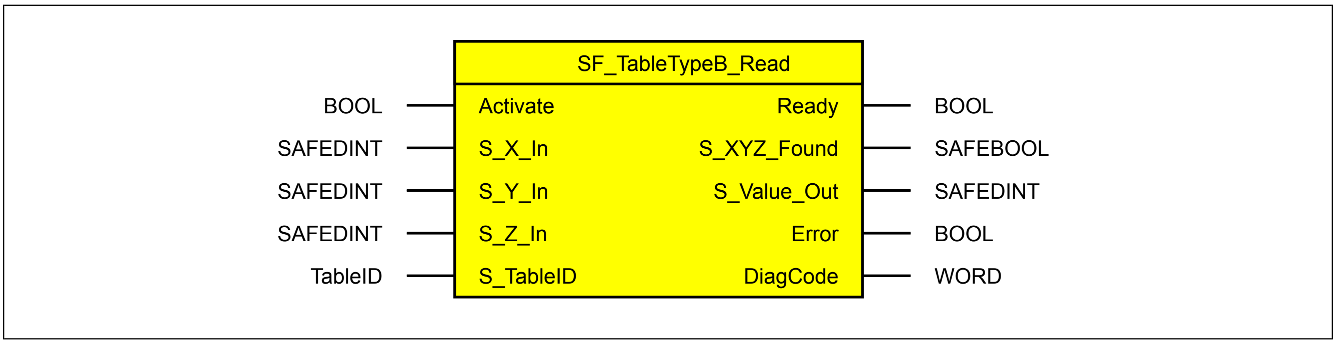
6.9.3.2 SF_TableTypeB_Read

This function block reads the data from the lookup table of type B (see "Table format B" on page 2115).

Requirements for use

Function block supported in SafeDESIGNER V4.3.2 and later

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_X_In	SAFEDINT	Variable	0	Input parameter for the x-position
IN	S_Y_In	SAFEDINT	Variable	0	Input parameter for the y-position
IN	S_Z_In	SAFEDINT	Variable	0	Input parameter for the z-position
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_XYZ_Found	SAFEBOOL	Variable	FALSE	Status information indicating whether an entry for the positions was found. TRUE: The current position was found in the table.
OUT	S_Value_Out	SAFEDINT	Variable	0	Output value of the current positions
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.9.3.2.1 Function description

Function block "SF_TableTypeB_Read" reads the data from the lookup table of type B.

The required table object is defined by input parameter "S_TableID". This object forms the basis upon which the values passed to the function block are searched for (input parameter "S_X_In" / "S_Y_In" / "S_Z_In").
If a value is found at the specified positions, it is written to output parameter "S_Value_Out" and output parameter "S_XYZ_Found" is set to TRUE.
If the specified positions are not found, output parameter "S_XYZ_Found" is set to FALSE and output parameter "S_Value_Out" is set to 0.

6.9.3.2.2 Fault avoidance

Danger!

Validation

In addition to validating the safety function implemented with the function block, it is also necessary during validation to verify that the table data is correct.

6.9.3.2.3 Status and error information

Additional details about output parameters that provide status and error information.

6.9.3.2.3.1 Status numbers

Information:

In the event of any error, state "FailSafe" is triggered on the safety controller. For corresponding error codes, see the Safety Logger.

Errors

Code (hex)	Description	Corrective measures
C025	The function block is reporting an error. State "FailSafe" is triggered on the safety controller. For additional information, see the Safety Logger.	Check the messages in the Safety Logger.

Table 724: "SF_TableTypeB_Read": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8001	The function block found an entry for the current coordinates.	No corrective measures are required.
8002	The function block did not find an entry for the current coordinates.	Change the coordinates to find a valid entry.

Table 725: "SF_TableTypeB_Read": Diagnostic codes

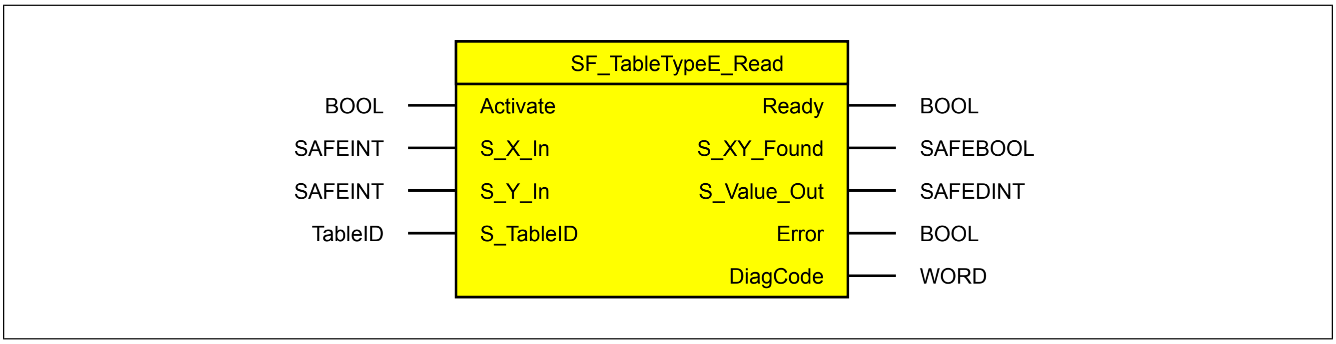
6.9.3.3 SF_TableTypeE_Read

This function block reads the data from the table of type E (see "Table format E" on page 2116).

Requirements for use

Function block supported in SafeDESIGNER V4.3.2 and later

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_X_In	SAFEINT	Variable	0	Input parameter for the x-position
IN	S_Y_In	SAFEINT	Variable	0	Input parameter for the y-position
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_XY_Found	SAFEBOOL	Variable	FALSE	Status information indicating whether an entry for the positions was found. TRUE: The current position was found in the table.
OUT	S_Value_Out	SAFEDINT	Variable	0	Output value of the current positions
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.9.3.3.1 Function description

Function block "SF_TableTypeE_Read" reads the data from the table of type E.

The required table object is defined by input parameter "S_TableID". This object forms the basis upon which the values passed to the function block are searched for (input parameter "S_X_In" / "S_Y_In").
If a value is found at the specified positions, it is written to output parameter "S_Value_Out" and output parameter "S_XY_Found" is set to TRUE.
If the specified positions are not found, output parameter "S_XY_Found" is set to FALSE and output parameter "S_Value_Out" is set to 0.

6.9.3.3.2 Fault avoidance

Danger!

Validation

In addition to validating the safety function implemented with the function block, it is also necessary during validation to verify that the table data is correct.

6.9.3.3.3 Status and error information

Additional details about output parameters that provide status and error information.

6.9.3.3.3.1 Status numbers

Information:

In the event of any error, state "FailSafe" is triggered on the safety controller. For corresponding error codes, see the Safety Logger.

Errors

Code (hex)	Description	Corrective measures
C025	The function block is reporting an error. State "FailSafe" is triggered on the safety controller. For additional information, see the Safety Logger.	Check the messages in the Safety Logger.

Table 726: "SF_TableTypeE_Read": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8001	The function block found an entry for the current coordinates.	No corrective measures are required.
8002	The function block did not find an entry for the current coordinates.	Change the coordinates to find a valid entry.

Table 727: "SF_TableTypeE_Read": Diagnostic codes

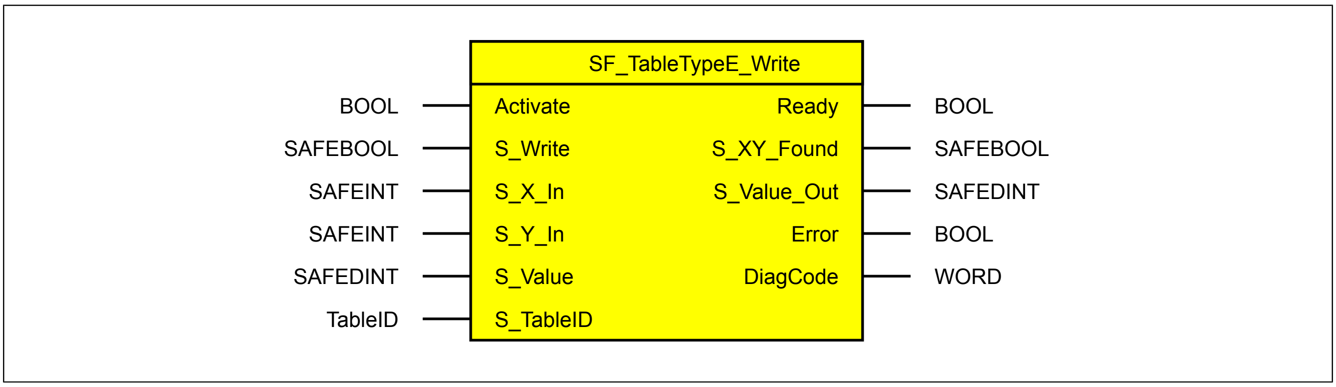
6.9.3.4 SF_TableTypeE_Write

This function block modifies at runtime the data that is stored in volatile memory from the table of type E (see "Table format E" on page 2116).

Requirements for use

Function block supported in SafeDESIGNER V4.3.2 and later

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_Write	SAFEBOOL	Variable	FALSE	Input parameter for executing the write procedure of "S_Value" to "S_X_In" and "S_Y_In". TRUE: The write request is executed cyclically (not edge-controlled).
IN	S_X_In	SAFEINT	Variable	0	Input parameter for the x-position
IN	S_Y_In	SAFEINT	Variable	0	Input parameter for the y-position
IN	S_Value	SAFEDINT	Variable	0	Predefined value for selected positions "S_X_In" and "S_Y_In"
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_XY_Found	SAFEBOOL	Variable	FALSE	Status information indicating whether an entry for the positions was found. TRUE: The current position was found in the table.
OUT	S_Value_Out	SAFEDINT	Variable	0	Output value of the current positions
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.9.3.4.1 Function description

Function block "SF_TableTypeE_Write" modifies at runtime the data that is stored in volatile memory from the table of type E.
The desired table object is defined by input parameter "S_TableID".
As long as signal TRUE is present on "S_Write", value "S_Value" at the existing positions ("S_X_In" and "S_Y_In") is replaced. No new values (input parameter "S_X_In" / "S_Y_In") can be added or entries deleted.
In order for the data to be changed, the system automatically copies the table to memory that can be modified at runtime.
If a value is found at the specified positions, it is written to output parameter "S_Value_Out" and output parameter "S_XY_Found" is set to TRUE.
If the specified positions are not found, output parameter "S_XY_Found" is set to FALSE and output parameter "S_Value_Out" is set to 0. A write request is executed cyclically (not edge-controlled).

> **Information:**
>
> **The changes made are not saved permanently.**

6.9.3.4.2 Fault avoidance

Danger!

Validation

In addition to validating the safety function implemented with the function block, it is also necessary during validation to verify that the table data is correct.

6.9.3.4.3 Status and error information

Additional details about output parameters that provide status and error information.

6.9.3.4.3.1 Status numbers

Information:

In the event of any error, state "FailSafe" is triggered on the safety controller. For corresponding error codes, see the Safety Logger.

Errors

Code (hex)	Description	Corrective measures
C025	The function block is reporting an error. State "FailSafe" is triggered on the safety controller. For additional information, see the Safety Logger.	Check the messages in the Safety Logger.

Table 728: "SF_TableTypeE_Write": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block is in state "Active" and no write request is pending ("S_Write" = FALSE).	No corrective measures are required.
8001	The function block found an entry for the current coordinates.	No corrective measures are required.
8002	The function block did not find an entry for the current coordinates.	Change the coordinates to find a valid entry.

Table 729: "SF_TableTypeE_Write": Diagnostic codes

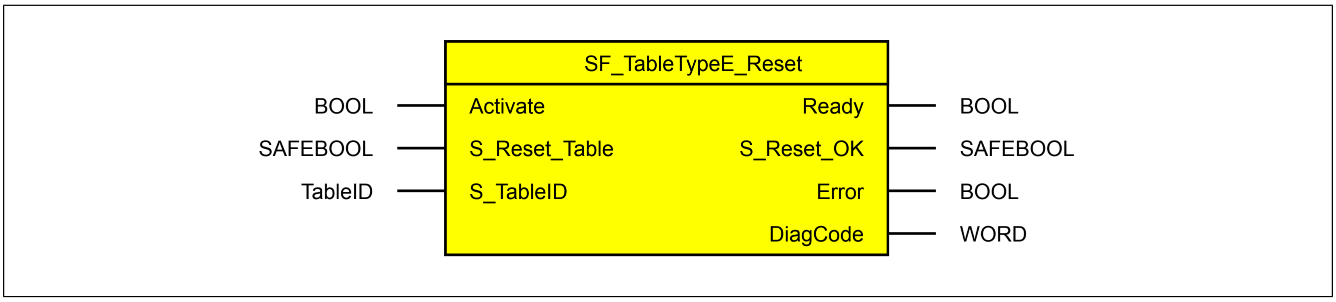
6.9.3.5 SF_TableTypeE_Reset

This function block resets the data from the table of type E (see "Table format E" on page 2116) to its initial values.

Requirements for use

Function block supported in SafeDESIGNER V4.3.2 and later

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block. TRUE: The function block is enabled.
IN	S_Reset_Table	SAFEBOOL	Variable	FALSE	Resets the table data. TRUE: The reset request is executed cyclically (not edge-controlled).
IN	S_TableID	TableID	Constant	0	Assigns a table object to the function block
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled. TRUE: The function block is enabled.
OUT	S_Reset_OK	SAFEBOOL	Variable	FALSE	Indicates that the table data was reset successfully. TRUE: The table data was reset successfully.
OUT	Error	BOOL	Variable	FALSE	Function block error message. TRUE: An error is present.
OUT	DiagCode	WORD	Variable	16#0000	Function block diagnostic message

6.9.3.5.1 Function description

Function block "SF_TableTypeE_Reset" resets the data from the table of type E to its initial values.
The desired table object is defined by input parameter "S_TableID".
Changed values that were written using function block "SF_TableTypeE_Write" are overwritten with their initial values. The reset request is executed cyclically (not edge-controlled).

6.9.3.5.2 Fault avoidance

> **Danger!**
>
> **Validation**
>
> **In addition to validating the safety function implemented with the function block, it is also necessary during validation to verify that the table data is correct.**

6.9.3.5.3 Status and error information

Additional details about output parameters that provide status and error information.

6.9.3.5.3.1 Status numbers

> **Information:**
>
> **In the event of any error, state "FailSafe" is triggered on the safety controller. For corresponding error codes, see the Safety Logger.**

Errors

Code (hex)	Description	Corrective measures
C025	The function block is reporting an error. State "FailSafe" is triggered on the safety controller. For additional information, see the Safety Logger.	Check the messages in the Safety Logger.

Table 730: "SF_TableTypeE_Reset": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not enabled. If "Activate" is connected to a variable that indicates the state of the connected safe device (active, inactive or peripheral fault detected), then the safe device is not active or has detected a fault in the connected peripheral.	• Enable the function block by setting "Activate" to TRUE. • Enable the safe device if "Activate" is connected to a variable that indicates the state of a connected safe device (active, inactive or peripheral fault detected), or correct the fault in the peripheral according to the device description.
8000	The function block is in state "Active" and no reset request is pending ("S_Reset_Table" = FALSE).	No corrective measures are required.
8001	A reset request is pending ("S_Reset_Table" = TRUE).	No corrective measures are required.

Table 731: "SF_TableTypeE_Reset": Diagnostic codes

6.10 Utilities_SF

This library stores application data in a remanent data area.

6.10.1 System requirements

Library "Utilities_SF" is part of SafeDESIGNER and only permitted to be used there.

The following requirements apply in order to use library "Utilities_SF":

- SafeDESIGNER: 3.1 or later
- Automation Studio: 4.0 or later
- SafeLOGIC: Safety Release 1.7 or later
- SafeLOGIC-X: Conversion blocks with Safety Release 1.7 and later, remanent data currently not supported
- License for using remanent data

6.10.2 Version history

Version	Date	Comment
1.21	May 2019	• Chapter 6.10.4.4 "Conversion function blocks": Updated description of "(SAFE)DINT_TO_(SAFE)WORDS".
1.20	May 2017	• Added new function blocks: – SF_GET_SYSTEMTIME – SAFEWORDS_TO_SAFEDINT / WORDS_TO_DINT – SAFEDINT_TO_SAFEWORDS / DINT_TO_WORDS • Editorial changes
1.10	February 2016	First edition

Table 732: Version history

6.10.3 Term definitions

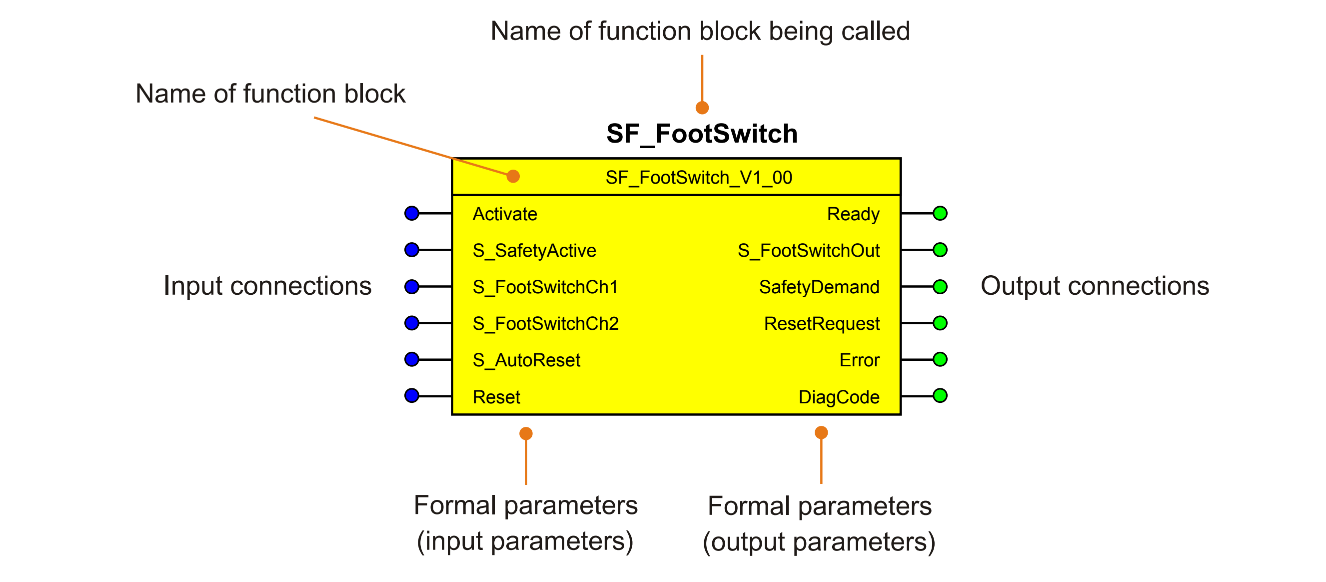


Figure 517: Components of a function block

When calling a function block, the inputs supply the input parameters with the current values of the variables or constants.
The output parameters supply the outputs with the associated values.

Inputs and outputs do not need to have the same name as the associated formal parameters, but they must be of the same data type. A difference in data type between formal parameters and inputs/outputs is reported as an error after compilation.

The name of a function block is composed of the function itself (e.g. "SF_FootSwitch", SF = safety function) and its version (Vx_yz). The format used to represent the version number in this document, Vx_yz, is a placeholder. For the actual version, see the function block being used.

6.10.4 Function blocks

List of the function blocks in this library sorted in descending order according to the version from which they are available.

Supported starting with SafeDESIGNER	Name	Short description
4.3.1	SF_GET_SYSTEMTIME	This function block provides the safe system time since the last restart of the safety controller.
4.3.1	SAFEWORDS_TO_SAFEDINT / WORDS_TO_DINT	This function block converts 2 (SAFE)WORD input variables into a (SAFE)DINT output variable.
4.3.1	SAFEDINT_TO_SAFEWORDS / DINT_TO_WORDS	This function block splits a (SAFE)DINT input variable into its low word (output parameter "(S_)OUT_WL") and high word (output parameter "(S_)OUT_WH").
3.1.0	SF_RemanentData_SAFEDINT	This function block saves application data in SAFEDINT format to a remanent data area.
3.1.0	SF_RemanentData_SAFEDWORD	This function block saves application data in SAFEDWORD format to a remanent data area.
3.1.0	SAFEINT_TO_SAFEDINT / INT_TO_DINT	This function block converts a (SAFE)INT input variable into a (SAFE)DINT output variable.
3.1.0	SAFEDINT_TO_SAFEINT / DINT_TO_INT	This function block converts a (SAFE)DINT input variable into a (SAFE)INT output variable.
3.1.0	SAFEDINT_TO_SAFETIME / DINT_TO_TIME	This function block converts a (SAFE)DINT input variable into a (SAFE)TIME output variable.
3.1.0	SAFETIME_TO_SAFEDINT / TIME_TO_DINT	This function block converts a (SAFE)TIME input variable into a (SAFE)DINT output variable.
3.1.0	SAFETIME_TO_SAFEDWORD / TIME_TO_DWORD	This function block converts a (SAFE)TIME input variable into a (SAFE)DWORD output variable.
3.1.0	SAFEDWORD_TO_SAFETIME / DWORD_TO_TIME	This function block converts a (SAFE)DWORD input variable into a (SAFE)TIME output variable.

6.10.4.1 SF_GET_SYSTEMTIME

This function block provides the safe system time since the last restart of the safety controller.

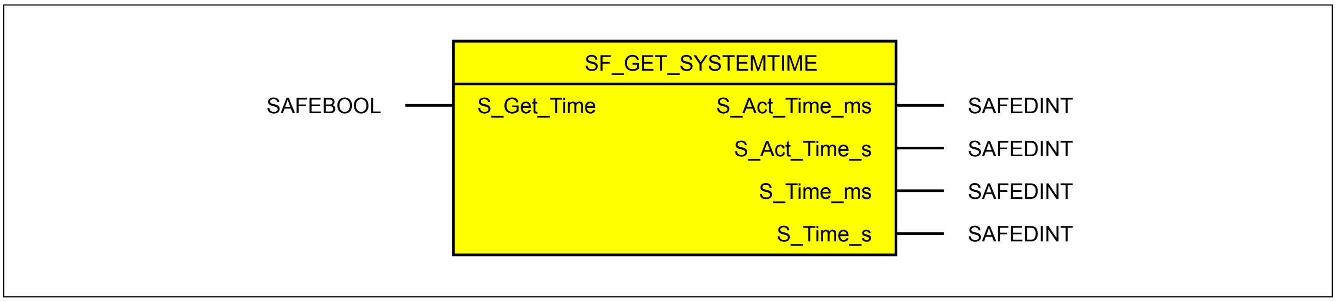
Requirements for use

Function block supported with SafeDESIGNER version: 4.3.1 and later

Danger!

When using the function block, the precision of the time must be taken into account. This amounts to up to 2.5% of the measured value.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	S_Get_Time	SAFEBOOL	Variable	FALSE	Requests an update of the triggered output parameters. TRUE: Updates the system time of the triggered output parameters
OUT	S_Act_Time_ms	SAFEDINT	Variable	0 ms	Outputs the continuous system time in ms (valid range: 0 to 1000 ms)
OUT	S_Act_Time_s	SAFEDINT	Variable	0 s	Outputs the continuous system time in s
OUT	S_Time_ms	SAFEDINT	Variable	0 ms	Outputs the triggered system time in ms (valid range: 0 to 1000 ms)
OUT	S_Time_s	SAFEDINT	Variable	0 s	Outputs the triggered system time in s

6.10.4.1.1 Function description

Function block "SF_GET_SYSTEMTIME" provides the safe system time since the last restart of the safety controller. The time is expressed in milliseconds since this is the same range as the cycle time of the safety controller. The function block allows for continuous operation over several years. To prevent the continuous time from overflowing, the time value is split into milliseconds and seconds. When output parameter "S_Act_Time_ms" reaches a value of 1000 ms, the counter for this output parameter starts again at 0 and output parameter "S_Act_Time_s" is increased by 1.
Input parameter "S_Get_Time" serves as the trigger. A rising edge returns the value at the time of the trigger on the two output parameters ("S_Time_ms" and "S_Time_s").

6.10.4.1.2 Input parameters

Description of the function block input parameters.

6.10.4.1.2.1 S_Get_Time

General function

- Requests an update of the triggered output parameters.

Data type

- SAFEBOOL

Connection

- Variable

Function description

Input parameter "S_Get_Time" is used to request an update of the triggered output parameters ("S_Time_ms" and "S_Time_s").

TRUE

An update of the triggered output parameters is requested.

FALSE

An update of the triggered output parameters is not requested.

6.10.4.1.3 Output parameters

Description of the function block output parameters.

6.10.4.1.3.1 S_Act_Time_ms

General function

- Outputs the continuous system time in ms (valid range: 0 to 1000 ms)

Data type

- SAFEDINT

Connection

- Variable

Function description

This output parameter specifies the continuous system time in milliseconds. When it reaches the maximum value of 1000 ms, it starts again at 0 and output parameter "S_Act_Time_s" is increased by 1.

- Valid range: 0 to 1000 ms

6.10.4.1.3.2 S_Act_Time_s

General function

- Outputs the continuous system time in s

Data type

- SAFEDINT

Connection

- Variable

Function description

This output parameter specifies the continuous system time in seconds.

6.10.4.1.3.3 S_Time_ms

General function

- Outputs the triggered system time in ms (valid range: 0 to 1000 ms)

Data type

- SAFEDINT

Connection

- Variable

Function description

This output parameter specifies the triggered system time in milliseconds.

- Valid range: 0 to 1000 ms

6.10.4.1.3.4 S_Time_s

General function

- Outputs the triggered system time in s

Data type

- SAFEDINT

Connection

- Variable

Function description

This output parameter specifies the triggered system time in seconds.

6.10.4.1.4 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.10.4.1.4.1 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> **Danger!**
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.10.4.1.4.2 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

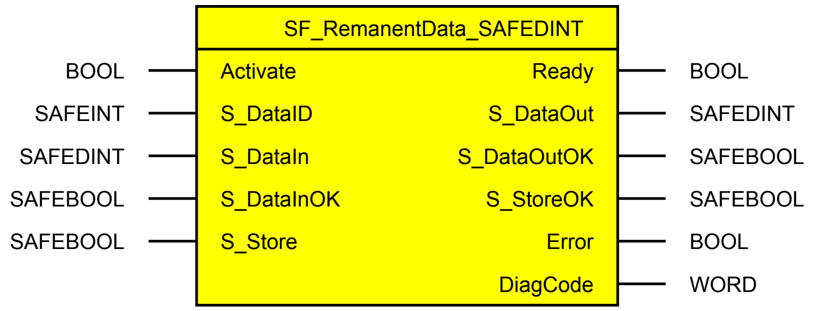
The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.10.4.2 SF_RemanentData_SAFEDINT

This function block saves application data in SAFEDINT format to a remanent data area.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block ("Activate" = TRUE)
IN	S_DataID	SAFEINT	Constant	0	ID to which the value on input parameter "S_DataIn" should be saved
IN	S_DataIn	SAFEDINT	Variable	0	Current value that should be saved
IN	S_DataInOK	SAFEBOOL	Variable	FALSE	Indicates whether the value on input parameter "S_DataIn" is valid ("S_DataInOK" = TRUE)
IN	S_Store	SAFEBOOL	Variable	FALSE	Trigger for initiating a write procedure
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled
OUT	S_DataOut	SAFEDINT	Variable	0	Currently read value from the remanent data area
OUT	S_DataOutOK	SAFEBOOL	Variable	FALSE	Indicates whether the read value is valid ("S_DataOutOK" = TRUE)
OUT	S_StoreOK	SAFEBOOL	Variable	FALSE	Indicates whether the last write procedure was completed successfully ("S_StoreOK" = TRUE)
OUT	Error	BOOL	Variable	FALSE	Error message from function block
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.10.4.2.1 Function description

Function block "SF_RemanentData_SAFEDINT" saves application data in SAFEDINT format in a remanent data area.

After startup, changing from "Debug" to "Safe" and enabling the function block, the value referenced by input parameter "S_DataID" is read from the remanent data area. The read-back value is provided on function block output parameter "S_DataOut".

Up to 32 values can be saved. Function block input parameter "S_DataID" defines which value should be written or read.

The value on input parameter "S_DataIn" must be valid in order to start a write procedure. Here, input parameter "S_DataInOK" must return signal TRUE.

A write procedure is triggered by a rising edge on input parameter "S_Store". The write procedure can take multiple safety controller cycles. During this time, no new write procedures are permitted to be triggered on the same function block (via a rising edge). If an additional write procedure is triggered on a different function block, it is deferred until the current write procedure is completed.

Output parameter "S_DataOutOK" indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid ("S_DataOutOK" = TRUE).

Output parameter "S_StoreOK" indicates that the last requested write procedure was completed successfully ("S_StoreOK" = TRUE).

6.10.4.2.2 Input parameters

Description of the function block input parameters.

6.10.4.2.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When shutting down or starting up safe devices, connect "Activate" to a variable that indicates the state (shut down or started up) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- If "Activate" is set to TRUE when performing a cold restart of the safety controller, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/ peripherals on the function block do not result in a faulty diagnostic message.
- It is also possible to connect "Activate" to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnosis from the function block in your diagnostic concept for error messages from safe devices and/or disabled safe devices, connect "Activate" to a signal that represents the state of the safe devices involved in the safety function supported by the function block. Create this signal only for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.10.4.2.2.2 S_DataID

General function

- ID to which the value on input parameter "S_DataIn" should be saved

Data type

- SAFEINT

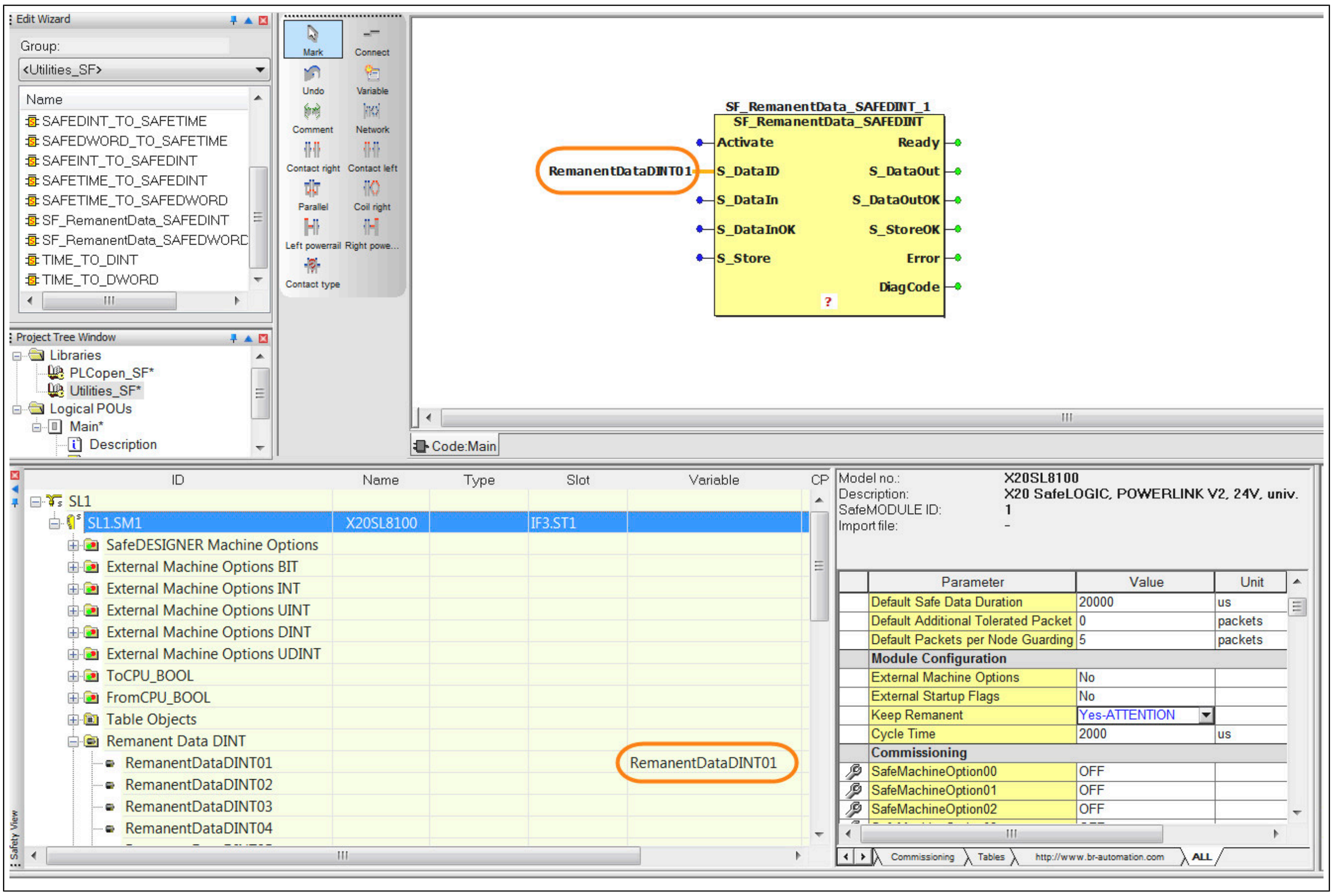
Connection

- Constant

Function description

Up to 32 values can be saved. Function block input parameter "S_DataID" defines which value should be written or read.

32 defined channels ("RemanentDataDINT") are available in the Safety View. Use the drag-and-drop functionality in SafeDESIGNER to connect the corresponding ID to input parameter "S_DataID".



6.10.4.2.2.3 S_DataIn

General function

- Current value that should be saved

Data type

- SAFEDINT

Connection

- Variable

Function description

The variable value on input parameter "S_DataIn" corresponds to the value that should be saved to the remanent data area.

6.10.4.2.2.4 S_DataInOK

General function

- Indicates whether the value on input parameter "S_DataIn" is valid ("S_DataInOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter indicates whether the value on input parameter "S_DataIn" is valid. If the value is not valid, a write procedure cannot be started.

TRUE

The value on input parameter "S_DataIn" is valid.

FALSE

The value on input parameter "S_DataIn" is invalid.

6.10.4.2.2.5 S_Store

General function

- Trigger for initiating a write procedure

Data type

- SAFEBOOL

Connection

- Variable

Function description

A write procedure is triggered by a rising edge on input parameter "S_Store". The value on input parameter "S_DataIn" is saved to the remanent data area.

The write procedure can take multiple safety controller cycles. During this time, no new write procedures are permitted to be triggered on the same function block (via a rising edge).

> **Information:**
>
> **A static TRUE signal on input parameter "S_Store" is not permitted.**

TRUE

The write procedure is triggered.

FALSE

The write procedure is not triggered.

6.10.4.2.3 Output parameters

Description of the function block output parameters.

6.10.4.2.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.10.4.2.3.2 S_DataOut

General function

- Currently read value from the remanent data area

Data type

- SAFEDINT

Connection

- Variable

Function description

This output parameter provides the value stored in the remanent data area.

6.10.4.2.3.3 S_DataOutOK

General function

- Indicates whether the read value is valid ("S_DataOutOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid.

TRUE

The read value is valid.

FALSE

The read value is invalid.

6.10.4.2.3.4 S_StoreOK

General function

- Indicates whether the last write procedure was completed successfully ("S_StoreOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates that the last requested write procedure was completed successfully.

TRUE

The last requested write procedure was completed successfully.

FALSE

The last requested write procedure was not completed or not completed successfully.

6.10.4.2.3.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.10.4.2.3.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.10.4.2.4 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.10.4.2.4.1 Multiple use of the same ID

Using the same ID multiple times for different function blocks is detected as an error during compilation. This causes an error message to be output in SafeDESIGNER.

> **Information:**
>
> **Using an ID multiple times causes an error message to be output in SafeDESIGNER during compilation.**

Possible cause:

- The same ID is used multiple times in the application program (programming error, user error).

6.10.4.2.4.2 Value on input parameter "S_DataIn" unstable during the write procedure

The value on input parameter "S_DataIn" is latched on a rising edge of input parameter "S_Store". If the value on input parameter "S_DataIn" changes during the write procedure, this change is detected by the function block. In this case, the write procedure is still indicated as being successful ("S_StoreOK" = TRUE).

Possible cause:

- The value on input parameter "S_DataIn" is not stable over an extended period of time (user error).

6.10.4.2.4.3 No TRUE signal on input parameter "S_DataInOK"

If signal TRUE is not on input parameter "S_DataInOK" when a write procedure is triggered, the function block outputs an error (see section "Status numbers").

During the write procedure, the signal on input parameter "S_DataInOK" can change to FALSE. In this case, the write procedure is still indicated as being successful ("S_StoreOK" = TRUE).

Possible causes:

- Incorrect variables connected on input parameters in the application program (user error)
- Incorrect device configuration (user error)
- Wiring error (user error)
- Open circuit in cables (user error, wiring error)

6.10.4.2.4.4 Static TRUE signal on input parameter "S_Store"

If there is a static TRUE signal on input parameter "S_Store" at the end of a write procedure, no additional write procedures can be triggered. The completed write procedure is indicated as being successful ("S_StoreOK" = TRUE). The function block outputs an appropriate diagnostic code (see section "Status numbers") to indicate this to the user. Additional write procedures are not possible. Input parameter "S_Store" must be set to FALSE before additional write procedures can take place.

Information:

If there is a static TRUE signal on input parameter "S_Store", a corresponding diagnostic code is output (see section "Status numbers").

Possible causes:

- Unintentional specification of a static TRUE signal on input parameter "S_Store" in the application program (user error)
- Wiring error (user error)

6.10.4.2.4.5 Toggling the signal on input parameter "S_Store"

If a rising edge on input parameter "S_Store" triggers additional write procedures while a write procedure is already in progress, this situation is detected. In this case, the value of input parameter "S_DataIn" that is latched at the beginning of the first write procedure is used, and the write procedure is indicated as being unsuccessful ("S_StoreOK" = FALSE). The function block outputs an appropriate diagnostic code (see section "Status numbers") to indicate this to the user.

Possible cause:

- Multiple write procedures are requested before the current write procedure has been completed.

6.10.4.2.4.6 Incorrect evaluation of output parameters "S_DataOutOK" and "S_StoreOK"

Problems and errors during the write procedure are detected by the function block. Appropriate evaluation must be carried out using output parameters "S_DataOutOK" and "S_StoreOK".
Only a TRUE signal on output parameter "S_DataOutOK" indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid.
Only a TRUE signal on output parameter "S_StoreOK" indicates that the last requested write procedure was completed successfully.

Danger!

Use output parameters "S_DataOutOK" and "S_StoreOK" to verify the validity of the data and to evaluate a write command.

Possible causes:

- Multiple write procedures are requested before the current write procedure has been completed.
- A value has not yet been written ("S_DataOutOK" = FALSE).
- A power failure occurs during a write procedure.

6.10.4.2.4.7 Power failure during a write procedure

A power failure that occurs during a write procedure can result in inconsistent data. In this case, you must verify the returned value.
In addition, it is not possible to guarantee that the value on input parameter "S_DataIn" was saved on output parameter "S_DataOut".

Danger!

If a power failure occurs during a write procedure, the consistency of data cannot be guaranteed. You must check the returned values in all cases.

Possible cause:

- Power failure during a write procedure

6.10.4.2.4.8 Inconsistent data when using multiple function blocks as a group

Since a write and read procedure can take several cycles and only one function block at a time writes data, you must verify the consistency of the data in this case. You must evaluate function block output parameters "S_DataOutOK" and "S_StoreOK".

Danger!

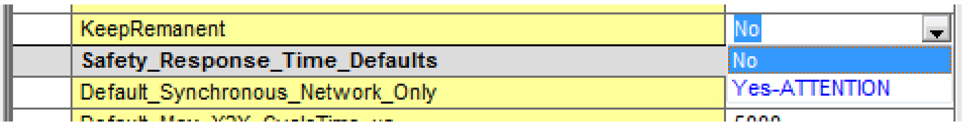
Incorrect use of the function blocks as a group can result in inconsistent data. You must always verify all output parameters of the involved function blocks in all cases.

Possible cause:

- Not all output parameters of the function blocks involved are being used.

6.10.4.2.4.9 Remanent data with project download

The safety controller automatically deletes saved data during a project download (parameter "KeepRemanent = No"). The default value for this parameter is "No" (do not save data). This function can be disabled using parameter "KeepRemanent" with setting "Yes-ATTENTION" so that the data is retained after a project download.



Danger!

When data is saved after a project download, you must verify that it still has the same significance in the application program.

6.10.4.2.4.10 Replacing the SafeKEY

Remanent data is stored on the SafeKEY. If the SafeKEY is inserted into a different safety controller, you must verify that the data and its significance is still correct.

Danger!

When replacing a SafeKEY, you must verify that the values of the remanent data are correct for the new safety controller.

6.10.4.2.4.11 Deleting remanent data

Remanent data is deleted using commands "SafeKEY FORMAT" and "CLEAR DATA" on the safety controller.

> ## Information:
>
> **Deleting remanent data makes this data invalid. Verify function block output parameter "S_DataOutOK".**

6.10.4.2.4.12 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> ## Danger!
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.10.4.2.4.13 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.10.4.2.4.14 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

> **Danger!**
>
> **You are responsible for the functional testing of protective equipment.**
> **You must therefore validate the protective equipment!**

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.10.4.2.5 Status numbers

Error

Code (hex)	Description	Corrective measures
C010	The specified ID is outside the permissible range (1 to 32).	Check the constant on input parameter "S_DataID". Use the predefined channels from the Safety View.
C020	When a write procedure is triggered, signal FALSE is on input parameter "S_DataInOK". The write procedure is not carried out.	Be sure that signal TRUE is on input parameter "S_DataInOK" when triggering a write procedure.

Table 733: "SF_RemanentData_SAFEDINT / SF_RemanentData_SAFEDWORD": Error codes

Status information

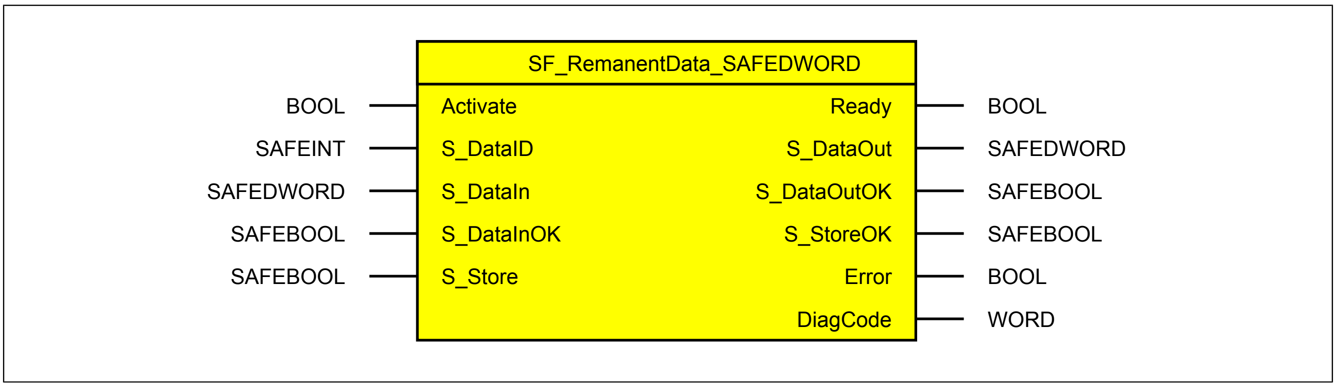
Code (hex)	Description	Corrective measures
0000	The function block is not active.	Enable the function block by setting input parameter "Activate" to TRUE.
8000	There are no errors or no actions are being executed on the function block. The write procedure was completed successfully.	No corrective measures are required.
8010	The value on input parameter "S_DataIn" was not stored since it is the same as the value that is already stored.	No corrective measures are required.
8020	The write procedure was completed successfully. Signal TRUE is still on input parameter "S_Store".	To be able to carry out an additional write procedure, you must reset input parameter "S_Store" to FALSE.
8812	A value has not yet been written to the specified ID in the remanent data area.	Save a value to the specified ID.
8814	The value for the specified ID was read successfully from the remanent data area.	No corrective measures are required.
8816	The write procedure to the remanent data area is active.	No corrective measures are required.
8818	The value will be written in the next write cycle since a different function block is currently writing to the remanent data area.	No corrective measures are required.
8822	A new write procedure was triggered during the active write procedure via a rising edge on input parameter "S_Store". The write procedure could not be completed successfully.	To save the current value, a new write procedure must be triggered.

Table 734: "SF_RemanentData_SAFEDINT / SF_RemanentData_SAFEDWORD": Diagnostic codes

6.10.4.3 SF_RemanentData_SAFEDWORD

This function block saves application data in SAFEDWORD format to a remanent data area.

Function block



Interface

I/O	Name	Data type	Connection	Initial value	Description
IN	Activate	BOOL	Variable/ Constant	FALSE	Enables the function block ("Activate" = TRUE)
IN	S_DataID	SAFEINT	Constant	0	ID to which the value on input parameter "S_DataIn" should be saved
IN	S_DataIn	SAFEDWORD	Variable	0	Current value that should be saved
IN	S_DataInOK	SAFEBOOL	Variable	FALSE	Indicates whether the value on input parameter "S_DataIn" is valid ("S_DataInOK" = TRUE)
IN	S_Store	SAFEBOOL	Variable	FALSE	Trigger for initiating a write procedure
OUT	Ready	BOOL	Variable	FALSE	Indicates that the function block is enabled
OUT	S_DataOut	SAFEDWORD	Variable	0	Currently read value from the remanent data area
OUT	S_DataOutOK	SAFEBOOL	Variable	FALSE	Indicates whether the read value is valid ("S_DataOutOK" = TRUE)
OUT	S_StoreOK	SAFEBOOL	Variable	FALSE	Indicates whether the last write procedure was completed successfully ("S_StoreOK" = TRUE)
OUT	Error	BOOL	Variable	FALSE	Error message from function block
OUT	DiagCode	WORD	Variable	16#0000	Diagnostic message from function block

6.10.4.3.1 Function description

Function block "SF_RemanentData_SAFEDWORD" saves application data in SAFEDWORD format to a remanent data area.

After startup, changing from "Debug" to "Safe" and enabling the function block, the value referenced by input parameter "S_DataID" is read from the remanent data area. The read-back value is provided on function block output parameter "S_DataOut".

Up to 32 values can be saved. Function block input parameter "S_DataID" defines which value should be written or read.

The value on input parameter "S_DataIn" must be valid in order to start a write procedure. Here, input parameter "S_DataInOK" must return signal TRUE.

A write procedure is triggered by a rising edge on input parameter "S_Store". The write procedure can take multiple safety controller cycles. During this time, no new write procedures are permitted to be triggered on the same function block (via a rising edge). If an additional write procedure is triggered on a different function block, it is deferred until the current write procedure is completed.

Output parameter "S_DataOutOK" indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid ("S_DataOutOK" = TRUE).

Output parameter "S_StoreOK" indicates that the last requested write procedure was completed successfully ("S_StoreOK" = TRUE).

Information:

In SafeDESIGNER, data type (SAFE)DWORD corresponds to data type (SAFE)UDINT.

6.10.4.3.2 Input parameters

Description of the function block input parameters.

6.10.4.3.2.1 Activate

General function

- Enables the function block ("Activate" = TRUE)

This input parameter is used to enable the function block.

- When shutting down or starting up safe devices, connect "Activate" to a variable that indicates the state (shut down or started up) of the corresponding safe devices. This ensures that the function block does not output a triggered safety function as diagnostic information when a device is cut off.
- If "Activate" is set to TRUE when performing a cold restart of the safety controller, the function block will behave in exactly the same way during the cold restart of the safety controller as when the function block is enabled.
- In the event of error messages from safe devices, set "Activate" to FALSE so that defective hardware/ peripherals on the function block do not result in a faulty diagnostic message.
- It is also possible to connect "Activate" to a constant (TRUE) in order to enable the function block. In this situation, the function block interprets a FALSE signal from a disabled safe device on a safety-related input parameter as a triggered safety function and outputs a corresponding diagnostic message. During this process, there is nothing at the function block interface to distinguish whether a triggered safety function or a disabled safe device caused the diagnostic message.

Data type

- BOOL

Connection

- Variable or constant

Function description

The enabling of the function block is state-controlled.

TRUE

The function block is enabled.

FALSE

The function block is not enabled.

All binary output parameters are set to FALSE.
Output parameter "DiagCode" is set to WORD#16#0000.

If you would like to control the diagnosis from the function block in your diagnostic concept for error messages from safe devices and/or disabled safe devices, connect "Activate" to a signal that represents the state of the safe devices involved in the safety function supported by the function block. Create this signal only for safe devices whose I/O signals are connected to the function block via input parameters. This prevents triggered safety functions from being reported by disabled safe devices. This measure is only used to control diagnostics in the event of inactive safe devices.

6.10.4.3.2.2 S_DataID

General function

- ID to which the value on input parameter "S_DataIn" should be saved

Data type

- SAFEINT

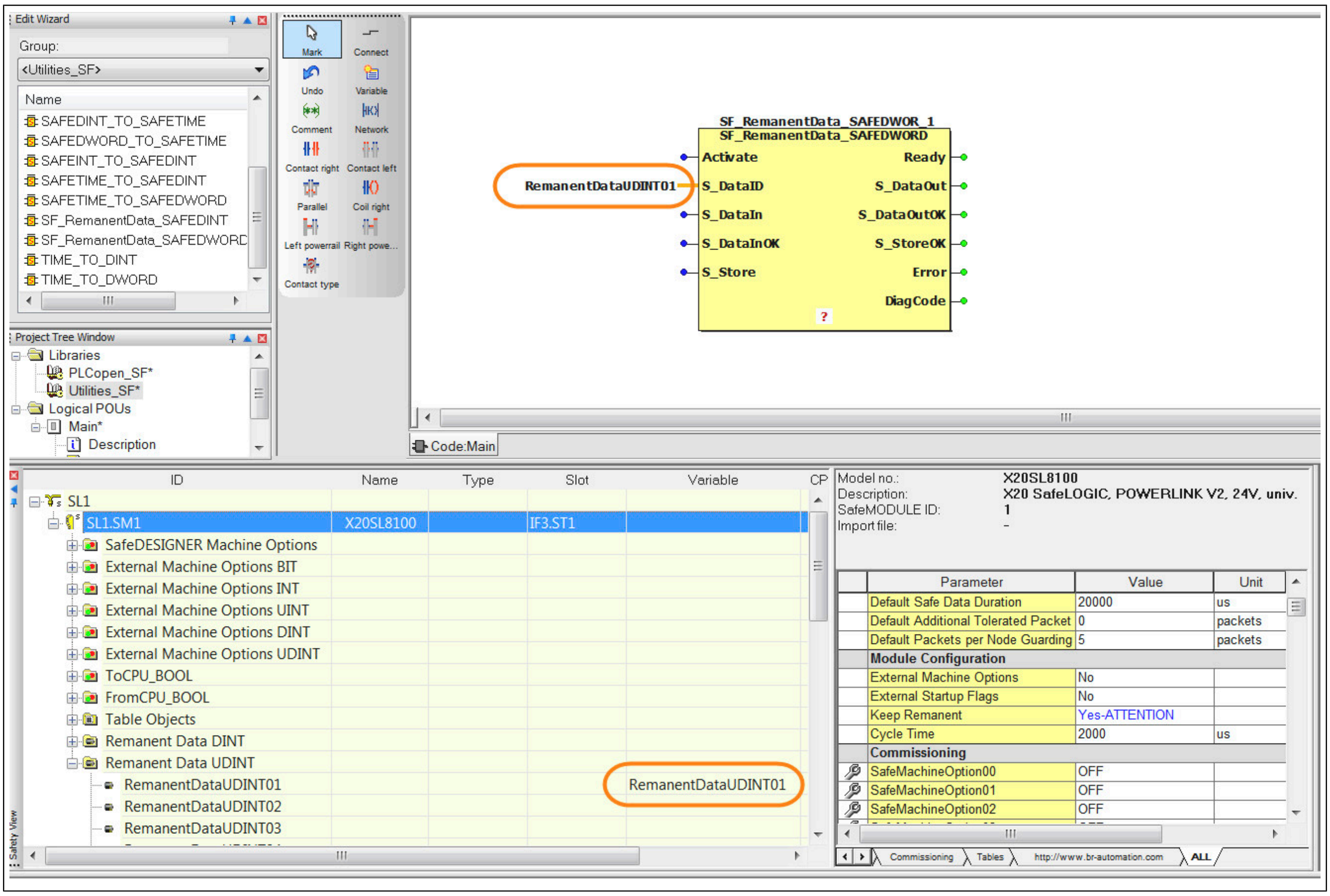
Connection

- Constant

Function description

Up to 32 values can be saved. Function block input parameter "S_DataID" defines which value should be written or read.

32 defined channels ("RemanentDataUDINT") are available in the Safety View window. Use the drag-and-drop functionality in SafeDESIGNER to connect the corresponding ID to input parameter "S_DataID".



6.10.4.3.2.3 S_DataIn

General function

- Current value that should be saved

Data type

- SAFEDWORD

Connection

- Variable

Function description

The variable value on input parameter "S_DataIn" corresponds to the value that should be saved to the remanent data area.

6.10.4.3.2.4 S_DataInOK

General function

- Indicates whether the value on input parameter "S_DataIn" is valid ("S_DataInOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This input parameter indicates whether the value on input parameter "S_DataIn" is valid. If the value is not valid, a write procedure cannot be started.

TRUE

The value on input parameter "S_DataIn" is valid.

FALSE

The value on input parameter "S_DataIn" is invalid.

6.10.4.3.2.5 S_Store

General function

- Trigger for initiating a write procedure

Data type

- SAFEBOOL

Connection

- Variable

Function description

A write procedure is triggered by a rising edge on input parameter "S_Store". The value on input parameter "S_DataIn" is saved to the remanent data area.

The write procedure can take multiple safety controller cycles. During this time, no new write procedures are permitted to be triggered on the same function block (via a rising edge).

> **Information:**
>
> **A static TRUE signal on input parameter "S_Store" is not permitted.**

TRUE

The write procedure is triggered.

FALSE

The write procedure is not triggered.

6.10.4.3.3 Output parameters

Description of the function block output parameters.

6.10.4.3.3.1 Ready

General function

- Indicates that the function block is enabled

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates whether the function block is enabled or not.

TRUE

The function block is enabled ("Activate" = TRUE). The output parameters indicate the current state of the safety function.

FALSE

The function block is not enabled ("Activate" = FALSE). The function block output parameters are set to FALSE.

6.10.4.3.3.2 S_DataOut

General function

- Currently read value from the remanent data area

Data type

- SAFEDWORD

Connection

- Variable

Function description

This output parameter provides the value stored in the remanent data area.

6.10.4.3.3.3 S_DataOutOK

General function

- Indicates whether the read value is valid ("S_DataOutOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid.

TRUE

The read value is valid.

FALSE

The read value is invalid.

6.10.4.3.3.4 S_StoreOK

General function

- Indicates whether the last write procedure was completed successfully ("S_StoreOK" = TRUE)

Data type

- SAFEBOOL

Connection

- Variable

Function description

This output parameter indicates that the last requested write procedure was completed successfully.

TRUE

The last requested write procedure was completed successfully.

FALSE

The last requested write procedure was not completed or not completed successfully.

6.10.4.3.3.5 Error

General function

- Function block error message

Data type

- BOOL

Connection

- Variable

Function description

This output parameter indicates a pending error message from the function block. Each error message causes the safe output parameters to be set to and remain at FALSE or 0.

> ## Danger!
>
> **It is your responsibility to ensure that all necessary repair measures are initiated after an error occurs since subsequent errors can result in a hazard!**

TRUE

The enabled function block has detected an error.

The safe output parameters are set to FALSE for bit information.
The safe output parameters are set to 0 for numeric information.

Output parameter "DiagCode" indicates the error code.

FALSE

The function block is not enabled, or the enabled function block has not detected an error. Output parameter "DiagCode" indicates the state.

6.10.4.3.3.6 DiagCode

General function

- Function block diagnostic message

Data type

- WORD

Connection

- Variable

Function description

This output parameter outputs specific function block diagnostic and status messages.

The function block indicates a pending error message via output parameter "Error".

For more details, see the tables in section "Status numbers".

Diagnostic code

The diagnostic code is specified as data type WORD. The hexadecimal value for each is specified in the tables (see section "Status numbers").

In the event of status messages (0xxxhex, 8xxxhex), output parameter "Error" is set to FALSE by the function block.

All other messages are error messages (Cxxxhex), i.e. "Error" = TRUE.

6.10.4.3.4 Fault avoidance

The following note regarding validation applies to all errors listed in this chapter.

> **Danger!**
>
> **Always validate the entire safety function!**

6.10.4.3.4.1 Multiple use of the same ID

Using the same ID multiple times for different function blocks is detected as an error during compilation. This causes an error message to be output in SafeDESIGNER.

> **Information:**
>
> **Using an ID multiple times causes an error message to be output in SafeDESIGNER during compilation.**

Possible cause:

- The same ID is used multiple times in the application program (programming error, user error).

6.10.4.3.4.2 Value on input parameter "S_DataIn" unstable during the write procedure

The value on input parameter "S_DataIn" is latched on a rising edge of input parameter "S_Store". If the value on input parameter "S_DataIn" changes during the write procedure, this change is detected by the function block. In this case, the write procedure is still indicated as being successful ("S_StoreOK" = TRUE).

Possible cause:

- The value on input parameter "S_DataIn" is not stable over an extended period of time (user error).

6.10.4.3.4.3 No TRUE signal on input parameter "S_DataInOK"

If signal TRUE is not on input parameter "S_DataInOK" when a write procedure is triggered, the function block outputs an error (see section "Status numbers").

During the write procedure, the signal on input parameter "S_DataInOK" can change to FALSE. In this case, the write procedure is still indicated as being successful ("S_StoreOK" = TRUE).

Possible causes:

- Incorrect variables connected on input parameters in the application program (user error)
- Incorrect device configuration (user error)
- Wiring error (user error)
- Open circuit in cables (user error, wiring error)

6.10.4.3.4.4 Static TRUE signal on input parameter "S_Store"

If there is a static TRUE signal on input parameter "S_Store" at the end of a write procedure, no additional write procedures can be triggered. The completed write procedure is indicated as being successful ("S_StoreOK" = TRUE). The function block outputs an appropriate diagnostic code (see section "Status numbers") to indicate this to the user. Additional write procedures are not possible. Input parameter "S_Store" must be set to FALSE before additional write procedures can take place.

> **Information:**
>
> **If there is a static TRUE signal on input parameter "S_Store", a corresponding diagnostic code is output (see section "Status numbers").**

Possible causes:

- Unintentional specification of a static TRUE signal on input parameter "S_Store" in the application program (user error)
- Wiring error (user error)

6.10.4.3.4.5 Toggling the signal on input parameter "S_Store"

If a rising edge on input parameter "S_Store" triggers additional write procedures while a write procedure is already in progress, this situation is detected. In this case, the value of input parameter "S_DataIn" that is latched at the beginning of the first write procedure is used, and the write procedure is indicated as being unsuccessful ("S_StoreOK" = FALSE). The function block outputs an appropriate diagnostic code (see section "Status numbers") to indicate this to the user.

Possible cause:

- Multiple write procedures are requested before the current write procedure has been completed.

6.10.4.3.4.6 Incorrect evaluation of output parameters "S_DataOutOK" and "S_StoreOK"

Problems and errors during the write procedure are detected by the function block. Appropriate evaluation must be carried out using output parameters "S_DataOutOK" and "S_StoreOK".
Only a TRUE signal on output parameter "S_DataOutOK" indicates that the saved value (referenced via input parameter "S_DataID") was read successfully and is valid.
Only a TRUE signal on output parameter "S_StoreOK" indicates that the last requested write procedure was completed successfully.

> **Danger!**
>
> **Use output parameters "S_DataOutOK" and "S_StoreOK" to verify the validity of the data and to evaluate a write command.**

Possible causes:

- Multiple write procedures are requested before the current write procedure has been completed.
- A value has not yet been written ("S_DataOutOK" = FALSE).
- A power failure occurs during a write procedure.

6.10.4.3.4.7 Power failure during a write procedure

A power failure that occurs during a write procedure can result in inconsistent data. In this case, you must verify the returned value.
In addition, it is not possible to guarantee that the value on input parameter "S_DataIn" was saved on output parameter "S_DataOut".

> **Danger!**
>
> **If a power failure occurs during a write procedure, the consistency of data cannot be guaranteed. You must check the returned values in all cases.**

Possible cause:

- Power failure during a write procedure

6.10.4.3.4.8 Inconsistent data when using multiple function blocks as a group

Since a write and read procedure can take several cycles and only one function block at a time writes data, you must verify the consistency of the data in this case. You must evaluate function block output parameters "S_DataOutOK" and "S_StoreOK".

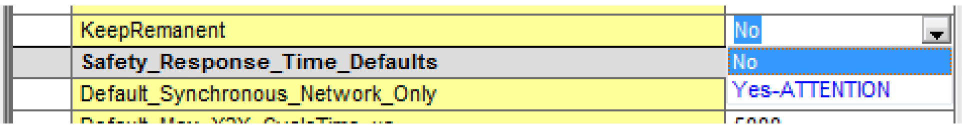
> **Danger!**
>
> **Incorrect use of the function blocks as a group can result in inconsistent data. You must always verify all output parameters of the involved function blocks in all cases.**

Possible cause:

- Not all output parameters of the function blocks involved are being used.

6.10.4.3.4.9 Remanent data with project download

The safety controller automatically deletes saved data during a project download (parameter "KeepRemanent = No"). The default value for this parameter is "No" (do not save data). This function can be disabled using parameter "KeepRemanent" with setting "Yes-ATTENTION" so that the data is retained after a project download.



> **Danger!**
>
> **When data is saved after a project download, you must verify that it still has the same significance in the application program.**

6.10.4.3.4.10 Replacing the SafeKEY

Remanent data is stored on the SafeKEY. If the SafeKEY is inserted into a different safety controller, you must verify that the data and its significance is still correct.

> **Danger!**
>
> **When replacing a SafeKEY, you must verify that the values of the remanent data are correct for the new safety controller.**

6.10.4.3.4.11 Deleting remanent data

Remanent data is deleted using commands "SafeKEY FORMAT" and "CLEAR DATA" on the safety controller.

> ## Information:
>
> **Deleting remanent data makes this data invalid. Verify function block output parameter "S_DataOutOK".**

6.10.4.3.4.12 Plausibility errors

Plausibility errors (limit values, data types, variables/constants) that occur when using the function block are detected and reported by either the function block or compiler. This is not always possible in the event of connection errors, however.

The function block cannot check whether:

- Input parameter values or constants are within their valid range but incorrect for the safety function being executed.
- Input parameters have been connected incorrectly.
- Input/Output parameters were erroneously not connected.

Therefore, note the following:

> ## Danger!
>
> **You are responsible for the parameter connections and therefore the implementation of the safety function (sub-application)!**
> **Check the connections when validating the sub-application!**

6.10.4.3.4.13 Sporadically changing/toggling signal levels or impermissible signals

Sporadically changing or toggling signal levels on edge-controlled input parameters cause the function block to interpret this signal as an edge, which results in an unintended corresponding action being triggered in the function block if fault avoidance measures are not taken.

Sporadically changing or toggling signal levels on state-controlled input parameters cause this signal to trigger an unintended corresponding action if fault avoidance measures are not taken.

Impermissible signals on input parameters can cause an unexpected initial movement, non-execution of a requested action or an error message.

Possible causes of these signals:

- Programming error in the application (user error)
- Cross fault, short circuit or open circuit (user error, wiring error)
- Error on the standard controller

To avoid this, the following measures can be taken depending on the safety function:

- Using signals from safe devices
- Implementing additional measures for preventing a hazard if using a signal from the standard controller (e.g. executing an additional function start after resetting a triggered safety function or correcting an error)
- Suitable wiring when using non-safe signals from the standard controller
- Verifying the source code in the application and final validation of the safety function

The specified measures can also be combined to reliably avoid errors.

Note that a signal change detected on a state-controlled input parameter is output as a diagnostic code.

6.10.4.3.4.14 Machine/System startup without performing functional testing of protective equipment

Faulty protective equipment is only detected following functional testing. Functional testing is not supported by this function block. If additional measures are not taken, faulty protective equipment can result in errors.

Danger!

You are responsible for the functional testing of protective equipment.
You must therefore validate the protective equipment!

Possible causes of faulty protective equipment:

- Defective devices (hardware fault)
- Cross fault, short circuit or open circuit (user error, wiring error)

6.10.4.3.5 Status numbers

Error

Code (hex)	Description	Corrective measures
C010	The specified ID is outside the permissible range (1 to 32).	Check the constant on input parameter "S_DataID". Use the predefined channels from the Safety View.
C020	When a write procedure is triggered, signal FALSE is on input parameter "S_DataInOK". The write procedure is not carried out.	Be sure that signal TRUE is on input parameter "S_DataInOK" when triggering a write procedure.

Table 735: "SF_RemanentData_SAFEDINT / SF_RemanentData_SAFEDWORD": Error codes

Status information

Code (hex)	Description	Corrective measures
0000	The function block is not active.	Enable the function block by setting input parameter "Activate" to TRUE.
8000	There are no errors or no actions are being executed on the function block. The write procedure was completed successfully.	No corrective measures are required.
8010	The value on input parameter "S_DataIn" was not stored since it is the same as the value that is already stored.	No corrective measures are required.
8020	The write procedure was completed successfully. Signal TRUE is still on input parameter "S_Store".	To be able to carry out an additional write procedure, you must reset input parameter "S_Store" to FALSE.
8812	A value has not yet been written to the specified ID in the remanent data area.	Save a value to the specified ID.
8814	The value for the specified ID was read successfully from the remanent data area.	No corrective measures are required.
8816	The write procedure to the remanent data area is active.	No corrective measures are required.
8818	The value will be written in the next write cycle since a different function block is currently writing to the remanent data area.	No corrective measures are required.
8822	A new write procedure was triggered during the active write procedure via a rising edge on input parameter "S_Store". The write procedure could not be completed successfully.	To save the current value, a new write procedure must be triggered.

Table 736: "SF_RemanentData_SAFEDINT / SF_RemanentData_SAFEDWORD": Diagnostic codes

6.10.4.4 Conversion function blocks

SafeDESIGNER provides function blocks for converting signals.

The name of a function block indicates which data type must be connected to the input parameter and which must be connected to the output parameter.
Example: A DINT variable must be connected to the input parameter of DINT_TO_INT. The output parameter requires data type INT.

SAFEINT_TO_SAFEDINT / INT_TO_DINT

This function block converts a (SAFE)INT input variable into a (SAFE)DINT output variable.

SAFEDINT_TO_SAFEINT / DINT_TO_INT

This function block converts a (SAFE)DINT input variable into a (SAFE)INT output variable.

Information:

After an overflow/underflow (e.g. 16-bit overflow), an error is reported and the safety controller is stopped (status "Stop").

SAFEDINT_TO_SAFETIME / DINT_TO_TIME

This function block converts a (SAFE)DINT input variable into a (SAFE)TIME output variable. The time base here is milliseconds (ms).

Information:

After the permissible IEC 61131-3 limit for data type TIME is exceeded, an error is reported and the safety controller is stopped (status "Stop").

SAFETIME_TO_SAFEDINT / TIME_TO_DINT

This function block converts a (SAFE)TIME input variable into a (SAFE)DINT output variable. The time base here is milliseconds (ms).

SAFETIME_TO_SAFEDWORD / TIME_TO_DWORD

This function block converts a (SAFE)TIME input variable into a (SAFE)DWORD output variable. The time base here is milliseconds (ms).

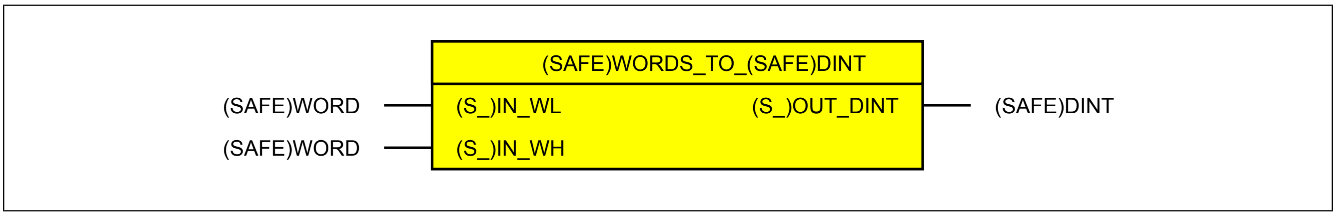
SAFEDWORD_TO_SAFETIME / DWORD_TO_TIME

This function block converts a (SAFE)DWORD input variable into a (SAFE)TIME output variable. The time base here is milliseconds (ms).

Information:

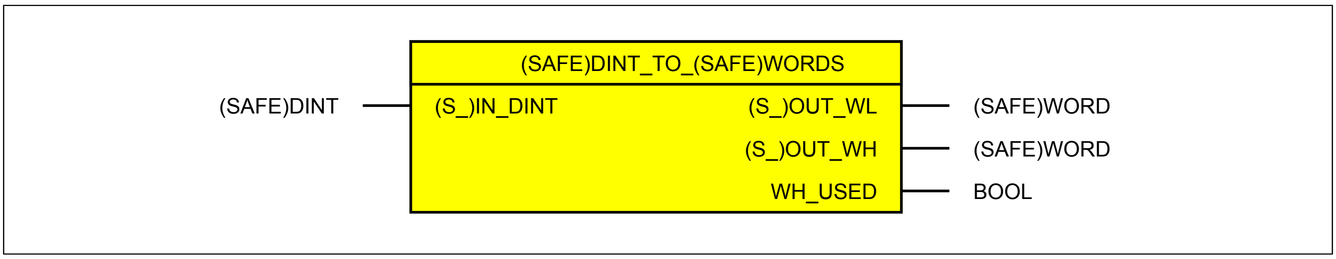
After the permissible IEC 61131-3 limit for data type TIME is exceeded, an error is reported and the safety controller is stopped (status "Stop").

SAFEWORDS_TO_SAFEDINT / WORDS_TO_DINT



This function block converts 2 (SAFE)WORD input variables into a (SAFE)DINT output variable. DINT is a 4-byte data type. Data type WORD is 2 bytes in size whether signed or unsigned. If the most significant bit on the input parameter for the high word "(S_)IN_WH" has value 1, then the negative data range is represented.

SAFEDINT_TO_SAFEWORDS / DINT_TO_WORDS



This function block splits a (SAFE)DINT input variable into its low word (output parameter "(S_)OUT_WL") and high word (output parameter "(S_)OUT_WH"). Signal TRUE on output parameter "WH_USED" indicates that output parameter "(S_)OUT_WH" is being used. Data type WORD is 2 bytes in size whether signed or unsigned. DINT is a 4-byte data type.

To display negative values, the function block internally processes the two's complement and returns the result for the high and low words for the corresponding output parameter.

Example: (S_)IN_DINT = -2 returns (S_)OUT_WH = 0xFFFF and (S_)OUT_WL = 0xFFFE

The correct treatment of negative values must be checked by the user.

Appendix A Abbreviations

Abbreviations appear throughout the user's manual, for example in technical data tables or descriptions of pinouts.

A.A Overview

Abbreviation	Stands for	Description
1oo2	1 out of 2	Code for the safety architecture. 1oo2 describes an architecture consisting of 2 channels where each of the two channels can carry out the safety function independently.
ESPE	Non-contact electro-sensitive protective equipment	Non-contact electro-sensitive protective equipment is protective equipment defined in accordance with EN 61496-1 that is based, for example, on one of the following physical principles: • Optical sensors • Ultrasound • Inductive sensors • Capacitive sensors • Infrared motion sensors
CCF	Common cause failure	Failure due to common cause: Failure caused by one or more outputs, which can result in the simultaneous failures in two or more isolated channels in a multi-channel system and a subsequent system failure.
DC	Diagnostic coverage	Diagnostic coverage: Partial reduction of the probability of dangerous hardware failures resulting from the use of automatic diagnostic tests
HFT	Hardware fault tolerance	Hardware fault tolerance: A hardware fault tolerance of N means that N+1-faults could lead to a loss of the safety function.
SRP/CS	Safety-related part of a control system	ISO 13849: Part of a controller that reacts to safety-related input signals and generates safety-related output signals.
FMEA	Failure mode and effects analysis	Analytical methods of reliability technology to detect potential weaknesses.
Cat.	Category	Category in accordance with EN 954-1 or EN ISO 13849
MTTFd	Mean time to dangerous failure	Average time before hazardous failure occurs
NC	Normally closed	A normally closed relay contact
	Not connected.	Used in pinout descriptions if a terminal or pin is not connected to a module
ND	Not defined	In technical data tables, this stands for a value that has not been defined, for example because a cable manufacturer does not provide certain technical data.
NO	Normally open	A normally open relay contact
OSSD	Output signal switching device	Output switching element of a safe output with test pulses
PFD	Probability of dangerous failure on demand	The probability of a dangerous failure of the safety function in an operating mode with a low request rate.
PFH	Probability of dangerous failure per hour	The probability of a dangerous failure of the safety function per hour in an operating mode with a high or continuous request rate.
PL	Performance level	Performance level in accordance with EN ISO 13849
POE	Program organization Unit	Program organization units are the language elements of a PLC program. They are small, independent software units containing program code.
POU	Program organization unit	See POU.
PT	Proof test interval	Recurring test interval for failure detection in a safety-related system that can restore the system to a "like new" condition or as close to it as possible from a practical standpoint.
SIL	Safety integrity level	Safety integrity level (in accordance with IEC 61508 or IEC 62061)
TBD	To be defined	Used in technical data tables when certain information is not yet available. The value will be provided later.
UDID	Unique Device ID	Unique device identification of an openSAFETY device

Table 737: Abbreviations in this user's manual

A.A.1 Terminology

Term	Description
Line Control	Line termination and line break monitoring. Clock outputs are used by the hardware for this.
POWERLINK MN	POWERLINK managing node

Table 738: Terms in this user's manual

8 B&R ID codes

Module ID codes are displayed in Automation Studio error messages, for example. The ID code and following tables can be used to determine the affected module and associated data sheet.

8.1 B&R ID codes sorted by model number

Model number	B&R ID code (hex.)	B&R ID code (dec.)	on page
X20SA4430	0xB8B5	47285	653
X20SC0402	0xE7F8	59384	465
X20SC0806	0xE75A	59226	465
X20SC0842	0xE7F9	59385	465
X20SC2212	0xBDA5	48549	510
X20SC2432	0xA7A4	42916	555
X20SD1207	0xCAC1	51905	713
X20SI2100	0x1F15	7957	372
X20SI4100	0x1DBD	7613	372
X20SI8110	0xE742	59202	372
X20SI9100	0xAEC8	44744	372
X20SL8100	0xDD61	56673	163
X20SL8101	0xE649	58953	163
X20SL8110	0xE64A	58954	163
X20SLX210	0xC5B0	50608	234
X20SLX402	0xE7EA	59370	299
X20SLX410	0xC5B2	50610	234
X20SLX806	0xE758	59224	299
X20SLX811	0xE757	59223	234
X20SLX842	0xE7EB	59371	299
X20SLX910	0xC5B1	50609	234
X20SO2110	0x1F16	7958	436
X20SO2120	0x2009	8201	436
X20SO2530	0xD205	53765	591
X20SO4110	0x1DBE	7614	436
X20SO4120	0x2007	8199	436
X20SO6300	0xB815	47125	405
X20SO6530	0xF22A	61994	591
X20SP1130	0x1DBF	7615	623
X20SRT402	0xE7EC	59372	741
X20SRT806	0xE759	59225	741
X20SRT842	0xE7F7	59383	741
X20ST4492	0xB419	46105	684
X20cSA4430	0xDD9F	56735	653
X20cSC2212	0xDD9D	56733	510
X20cSC2432	0xDD5D	56669	555
X20cSD1207	0xE1CB	57803	713
X20cSI4100	0xDD5A	56666	372
X20cSI9100	0xDD5B	56667	372
X20cSL8100	0xE287	57991	163
X20cSL8101	0xE926	59686	163
X20cSLX402	0xF210	61968	299
X20cSLX410	0xE288	57992	234
X20cSLX910	0xE4D1	58577	234
X20cSO2530	0xDD86	56710	591
X20cSO4110	0xDD84	56708	436
X20cSO4120	0xDD5C	56668	436
X20cSO6300	0xDD88	56712	405
X67SC4122.L12	0xA7A6	42918	895
X67SI8103	0xBB7C	47996	862

8.2 B&R ID codes sorted by ID code

Model number	B&R ID code (hex.)	B&R ID code (dec.)	on page
X20SI4100	0x1DBD	7613	372
X20SO4110	0x1DBE	7614	436
X20SP1130	0x1DBF	7615	623
X20SI2100	0x1F15	7957	372
X20SO2110	0x1F16	7958	436
X20SO4120	0x2007	8199	436
X20SO2120	0x2009	8201	436
X20SC2432	0xA7A4	42916	555
X67SC4122.L12	0xA7A6	42918	895
X20SI9100	0xAEC8	44744	372
X20ST4492	0xB419	46105	684
X20SO6300	0xB815	47125	405
X20SA4430	0xB8B5	47285	653
X67SI8103	0xBB7C	47996	862
X20SC2212	0xBDA5	48549	510
X20SLX210	0xC5B0	50608	234
X20SLX910	0xC5B1	50609	234
X20SLX410	0xC5B2	50610	234
X20SD1207	0xCAC1	51905	713
X20SO2530	0xD205	53765	591
X20cSI4100	0xDD5A	56666	372
X20cSI9100	0xDD5B	56667	372
X20cSO4120	0xDD5C	56668	436
X20cSC2432	0xDD5D	56669	555
X20SL8100	0xDD61	56673	163
X20cSO4110	0xDD84	56708	436
X20cSO2530	0xDD86	56710	591
X20cSO6300	0xDD88	56712	405
X20cSC2212	0xDD9D	56733	510
X20cSA4430	0xDD9F	56735	653
X20cSD1207	0xE1CB	57803	713
X20cSL8100	0xE287	57991	163
X20cSLX410	0xE288	57992	234
X20cSLX910	0xE4D1	58577	234
X20SL8101	0xE649	58953	163
X20SL8110	0xE64A	58954	163
X20SI8110	0xE742	59202	372
X20SLX811	0xE757	59223	234
X20SLX806	0xE758	59224	299
X20SRT806	0xE759	59225	741
X20SC0806	0xE75A	59226	465
X20SLX402	0xE7EA	59370	299
X20SLX842	0xE7EB	59371	299
X20SRT402	0xE7EC	59372	741
X20SRT842	0xE7F7	59383	741
X20SC0402	0xE7F8	59384	465
X20SC0842	0xE7F9	59385	465
X20cSL8101	0xE926	59686	163
X20cSLX402	0xF210	61968	299
X20SO6530	0xF22A	61994	591