X20SL80xx

1 Organization of safety notices

The safety notices in this manual are organized as follows:

Safety notice	Description
Danger!	Disregarding the safety regulations and guidelines can result in major damage to material, severe injury or death.
Information:	Important information for preventing errors.

Table 1: Organization of safety notices

2 Order data

Model number	Short description
	CPUs
X20SL8000	X20 SafeLOGIC, safety PLC standard, supports up to 20 safety nodes, exchangeable application memory: memory key, 1 POWERLINK V2 interface, controlled node, integrated 2x hub, incl. power supply module, X20TB52 terminal block, X20AC0SR1 X20 end plate right included, order memory key separately.
X20SL8001	X20 SafeLOGIC, safety PLC plus, supports up to 100 safety nodes, 32 machine options, POWERLINK safety gateway, exchangeable application memory: memory key, 1 POWERLINK V2 interface, controlled node, integrated 2x hub, incl. power supply module, X20TB52 terminal block, X20AC0SR1 X20 end plate right included, order memory key separately.
X20SL8010	X20 SafeLOGIC, safety PLC standard,SafeMC supports up to 20 safety nodes incl. SafeMC nodes, exchangeable application memory: memory key, 1 POWERLINK V2 interface, controlled node, integrated 2x hub, incl. power supply module, X20TB52 terminal block, X20AC0SR1 X20 end plate right included, order memory key separately.
X20SL8011	X20 SafeLOGIC, safety PLC plus, SafeMC supports up to 100 safety nodes incl. SafeMC nodes, 32 machine options, POWERLINK safety gateway, exchangeable application memory: memory key, 1 POWERLINK V2 interface, controlled node, integrated 2x hub, incl. power supply module, X20TB52 terminal block, X20AC0SR1 X20 end plate right included, order memory key separately.
	Mandatory accessories
	Accessories
X20MK0201	X20 memory key, 2 MB
X20MK0203	X20 memory key, 8 MB

Table 2: X20SL8000, X20SL8001, X20SL8010, X20SL8011 - Order data

3 Technical data

Product ID	X20SL8000	X20SL8001	X20SL8010	X20SL8011
Short description				
Interfaces	POWERLINK V2			
System module	CPU			
General information				
Cooling	Fan-free			
Status indicators	CPU function, POWERLINK, SafeKEY			
Diagnostics				
CPU function	Yes, with status LED			
POWERLINK	Yes, with status LED			
SafeKEY	Yes, with status LED			
Power consumption	5.1 W			
Certification types				
CE	Yes			
c-UL-us	Yes			
GOST-R	Yes			
IEC 61508	Yes			
IEC 62061	Yes			
EN 13849	Yes			
Functionality				
Number of supported safety nodes	Max. 20	Max. 100	Max. 20	Max. 100
Communication with each other	Communication only possible with a SafeL-OGIC SL8001 or SL8011	Free communication with max. 10 other SafeL-OGIC devices possible	Communication only possible with a SafeL-OGIC SL8001 or SL8011	Free communication with max. 10 other SafeL-OGIC devices possible
Supports machine options	No	Yes	No	Yes
Support of SafeMC (Safe Motion Control)	No		Yes	
Controllers				
Real-time clock	Nonvolatile memory, resolution 1 second			
Modular interface slots	None			
Processor	Intel XSCALE 266 MHz			
SafeKEY slot	1x			
Fastest task class cycle time	1 ms			
Fieldbus				
Type	POWERLINK V2			
Design	Internal 2x hub, 2x shielded RJ45 port			
Cable length	Max. 100 m between two stations (segment length)			
Transfer rate	100 Mbit/s			
Cycle time	Max. 20 ms			
Power supply				
Rated voltage	+24 V (-15% / +20%)			
Fuse	Integrated, cannot be replaced			
Reverse polarity protection	Yes			
Operating conditions				
Mounting orientation				
Horizontal	Yes			
Vertical	Yes			
Installation at altitudes above sea level				
0 to 2000 m	No derating			
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m			
EN 60529 protection	IP20			
Environmental conditions				
Temperature				
Operation				
Horizontal installation	0 to 55°C			
Vertical installation	0 to 45°C			
Storage	-25 to 70°C			
Transport	-25 to 70°C			
Relative humidity				
Operation	5 to 95%			
Storage	5 to 95%			
Transport	5 to 95%			
Mechanical characteristics				
Note	Order application memory (SafeKEY) separately X20 locking plate (right) included in delivery X20 terminal block, 12-pin, safety coded, included in delivery SafeKEY cover is included in delivery			
Dimensions				
Width	87.5 mm			
Height	99 mm			
Depth	75 mm			

Table 3: X20SL8000, X20SL8001, X20SL8010, X20SL8011 - Technical data

4 Safety characteristics

Criteria	Characteristic value
Category in accordance with EN ISO 13849	CAT 4
Maximum performance level in accordance with EN ISO 13849	PL e
Maximum safety integrity level in accordance with IEC 62061	SIL 3
Maximum safety integrity level in accordance with IEC 61508	SIL 3
PFH (probability of failure per hour)	< $1*10^{-10}$
PFD (probability of failure on demand)	< $1*10^{-5}$ at a proof test interval of 10 years < $2*10^{-5}$ at a proof test interval of 20 years
PT (proof test interval)	Max. 20 years
DC (diagnostic coverage)	>90%
MTTFd (mean time to failure - dangerous)	2500 years

Table 4: X20SL80xx - Safety characteristics

5 Control and connection elements

LEDs and buttons/switches are provided for operating the SafeLOGIC. With these elements,

- module replacement, including a test of the complete module configuration (7.1 "Module exchange" on page 17 section)
- firmware replacement (7.3 "Confirmation of firmware change" on page 19 section)
- SafeKEY replacement, including possible transfer of module configuration from the old SafeKEY (7.5 "SafeKEY" on page 19 section)
- and SafeLOGIC controller replacement (7.6 "Replacing a SafeLOGIC controller" on page 21 section)

can take place.

SafeLOGIC has the following operating and connection elements:

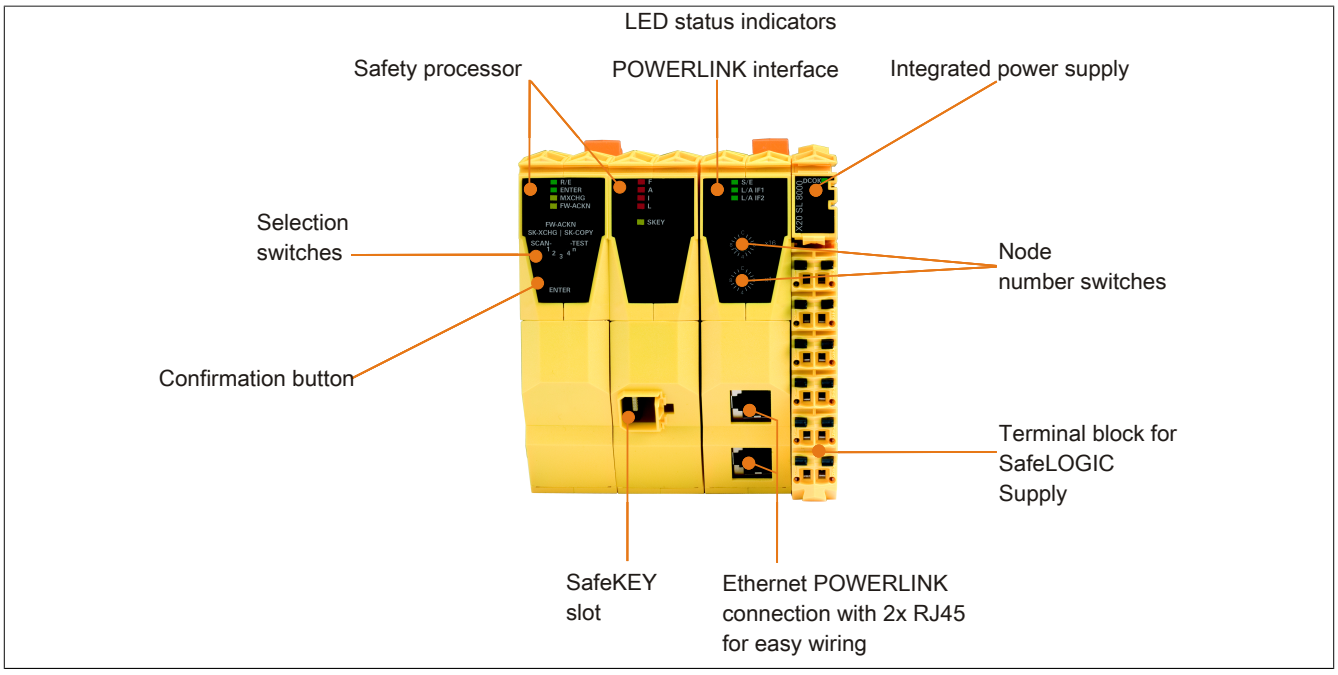
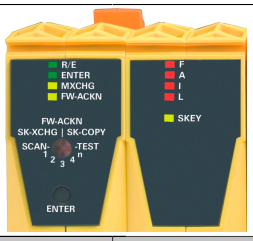


Image 1: X20SL80xx operating elements

The interface explained in the 8.1 "Remote control" on page 22 section can also be used to operate the SafeLOGIC controller using an operator panel.

5.1 Safety processor

5.1.1 Status LEDs for the safety processor



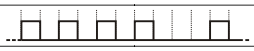
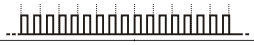
LED	Color	Status	Description
R/E		Off	Boot phase
	Green	On	Application found and executed
		Blinking	Application exists but is not being executed (In the download dialog box for the SafeDESIGNER, "Automatic Start" was not selected OR in the boot phase i.e. not all necessary safe modules on the network were configured correctly.)
	Orange	On	SafeDESIGNER in debug mode
		Blinking	SafeDESIGNER in debug mode, application stopped
		Blinking quickly	No application found on the SafeKEY
ENTER	Green	On	Authorization missing
		1x blinks 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 has been updated
		On	SafeKEY was replaced
ENTER MXCHG FW-ACKN	Green Orange Orange	Running sequence	Executing module scan or boot phase (beginning with Release 1.5 - Note: Check "STATUS" LED Status LEDs for the POWERLINK interface!).
FAIL	Red		The four "FAIL" LEDs indicate the boot status, and once the system is running they indicate the general fail-safe status of the entire module.
		F A I L x x x x x x x x **X** x **X** **X** x x x x x = illuminated **X** = Brightly illuminated	**Meaning** Boot phase, firmware loading, status when SafeKEY is missing Complete hardware test (max. duration approx. 5 s) Initialization and firmware startup Preoperational state Operational state Fail-safe status of the entire module
		Alternating blinking of "FI" and "AL"	SafeDESIGNER is in run - debug mode
SKEY	Orange	Off	No access to the SafeKEY
		Blinking	Access to the SafeKEY

Table 5: X20SL80xx safety processor - Status indicator

Danger!

Static lit "FAIL" LEDs indicate a defective module, which must be changed immediately.
It is your responsibility to ensure that all necessary measures for repair are initiated after an error occurs as successive errors can result in dangerous situations.

5.1.2 LED test

With help from the following sequence, the function of the LEDs can be tested:

- Place selection switch to TEST
- Press the confirmation button
- All the safety processor LEDs turn on (left and middle SafeLOGIC module) for the exact duration that the confirmation button is pressed
  On Release versions < 1.4, the "SKEY" LED will not be turned on during this test

5.1.3 Selection switch and confirmation button

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

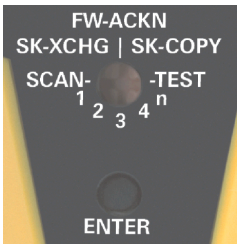


Image 2: X20SL80xx selection switch and confirmation button

Switch position	Functionality	Description
FW-ACKN	Firmware acknowledgment	Acknowledge firmware change on one or more modules[1)]
Unlabeled position between FW-ACKN and SK-XCHG	Format SafeKEY	Formatting SafeKEY (Release 1.4 and higher)[1)]
SK-XCHG	SafeKEY replacement	Acknowledge the SafeKEY replacement[1)]
SK-COPY	SafeKEY copy	Copy of the configuration data from the SafeKEY[1)]
SCAN	Scan	Perform module scan
TEST	Test	Perform LED test
1,2,3,4,n	Module replacement	Confirm the replacement of 1, 2, 3, 4 or more than 4 modules

Table 6: X20SL80xx - Confirmation modes

1) Triggers an automatic restart

Confirmation (all functions except for "Format SafeKEY")

The confirmation button must be pressed for 0.5 - 5 s to receive a confirmation. After 0.5 s, the "ENTER" LED (see 5.1.1 "Status LEDs for the safety processor" on page 4 section) is lit. After releasing the confirmation button, the "ENTER" LED remains illuminated for an extra 0.8 s. A correct entry is signaled in this sequence.

- 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 selection switch. If the user wants to confirm a module replacement for, e.g. one specific module, then the selection switch must be at position "1" (see 7.1.4 "Exchanging the individual module" on page 18 section). In this case, if a placement other than "1" is confirmed, 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 - 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. A correct entry is signaled in this sequence.

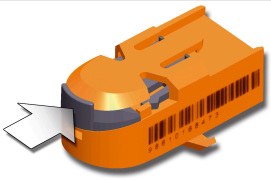
- If the confirmation button is released before 20 s, it has no effect.
- If the confirmation button is pressed for longer than 30 s, then the "ENTER" LED blinks for 5 s to display an error.

All data will be deleted (including password) - this is why we recommend going online with the SafeDESIGNER and assigning a new password.

5.2 Slot for application memory (SafeKEY)

Program memory (SafeKEY) to save the program, the parameters and the system configuration are required to operate the SafeLOGIC. From the X20 System accessories, memory key types X20MK0201 (2 MB) and X20MK0203 (8 MB) are available as a SafeKEY. The memory key is not included with the delivery of the SafeLOGIC, instead it must be ordered as an accessory.

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

 Image 3: SafeKEY unlocked	 Image 4: SafeKEY locked

Information:

Removing the SafeKEY during operation results in a restart of SafeLOGIC and a cutoff of all safety-related actuators.

Pulling the SafeKEY during operation can destroy the data on the SafeKEY.

Removing the SafeKEY during operation must be avoided.

5.3 POWERLINK interface

5.3.1 Status LEDs for the POWERLINK interface

Image	LED	Color	Status	Description
	STATUS[1]	Green / red		Status/Error LED. The statuses of the LEDs are described in the following section.
	L/A IFx	Green	On	A link to the remote station has been established.
			Blinking	A link to the remote station has been established. The LED blinks when Ethernet activity is present on the bus.

Table 7: X20SL80xx POWERLINK interface status indicators

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

5.3.2 LED STATUS

The STATUS LED is a green/red dual LED. The color green (status) is superimposed on the color red (error).

Red - error	Description
On	The POWERLINK interface has encountered an error (failed Ethernet frames, increased number of collisions on the network, etc.). Note: The LED blinks red several times immediately after startup. This is not an error.

Table 8: X20SL80xx POWERLINK interface status/error LED is red

Green - status	Description
Off	The POWERLINK interface is either not getting power, or it is NOT_ACTIVE. The POWERLINK interface waits in this state for about 5 seconds after a restart. Communication with the POWERLINK interface is not possible. If no POWERLINK communication is detected during these 5 s, the POWERLINK interface goes into the BASIC_ETHERNET state (flickering). If, however, POWERLINK communication is detected during this time, the POWERLINK interface goes directly into the PRE_OPERATIONAL_1 status (single flash).
Green flickering (approx. 10 Hz)	The POWERLINK interface did not recognize the POWERLINK communication. In this state you can communicate directly with the POWERLINK interface using UDP. If POWERLINK communication is detected while in this status, the POWERLINK interface goes into the PRE_OPERATIONAL_1 state (single flash).
Single flash (approx. 1 Hz)	The POWERLINK interface is in the PRE_OPERATIONAL_1 state. The CN (Controlled Node) waits until it receives an SoC frame and then switches to PRE_OPERATIONAL_2 status (double flash).
Double flash (approx. 1 Hz)	The POWERLINK interface is in the PRE_OPERATIONAL_2 state. In this status the POWERLINK interface is normally configured by the manager. After this, a command changes the status to READY_TO_OPERATE (triple flash). Note: If an incorrect node number is configured or the module is disabled in AS, for example, the system does not switch to the next status.
Triple flash (approx. 1 Hz)	The POWERLINK interface is READY_TO_OPERATE. The manager switches the status via command to OPERATIONAL.
On	The POWERLINK interface is in the OPERATIONAL state.
Blinking (approx. 2.5 Hz)	The POWERLINK interface is STOPPED. No output data is produced and no input data is received. Only the appropriate command from the manager can enter or leave this state.

Table 9: X20SL80xx POWERLINK interface status/error LED is green

5.3.3 POWERLINK station number

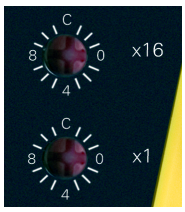


Image 5: X20SL80xx - POWERLINK station number switches

The station number for the POWERLINK station is set using the two number switches. Station numbers are permitted between $01 and $EF.

Switch position	Description
$00	Reserved, switch position is not permitted.
$01 - $EF	Station number of the POWERLINK station. Operation as controlled node.
$F0 - $FF	Reserved, switch position is not permitted.

Table 10: X20SL80xx - Station numbers - POWERLINK V2

5.3.4 RJ45 ports

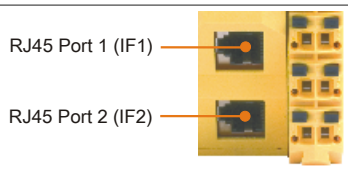


Image 6: X20SL80xx RJ45 ports

Pin	assignment
1	RXD
2	RXD\
3	TXD
4	Termination
5	Termination
6	TXD\
7	Termination
8	Termination

Table 11: X20SL80xx pin assignment for RJ45 port

RXD ... Receive data TXD ... Transmit data

5.4 SG support

SG3 / SGC

The SafeLOGIC is not supported at the moment on SG3 targets and SGC.

SG4

The SafeLOGIC controller comes with preinstalled firmware. Furthermore, the firmware version that matches the Safety Release will also be saved to the standard CPU when downloading the Automation Studio project.

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

When changing safety-related firmware on the SafeLOGIC controller, the measures listed in the section "Confirmation of firmware change" on page 19 must be taken.

5.5 Integrated power supply

A power supply is integrated for the SafeLOGIC.

5.5.1 Status LEDs for integrated power supply

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

Table 12: X20SL80xx status LEDs for integrated power supply

5.5.2 Pin assignments for the integrated power supply

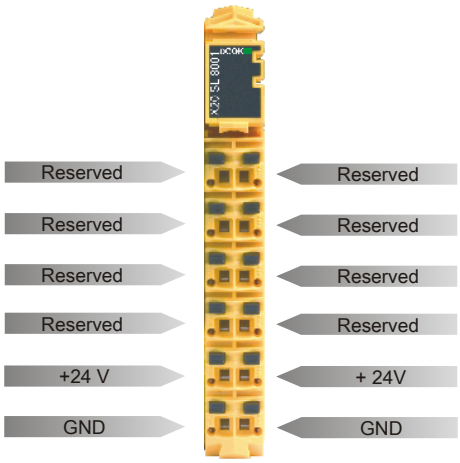


Image 7: SafeLOGIC pin assignments of the integrated power supply

5.5.3 Connection example

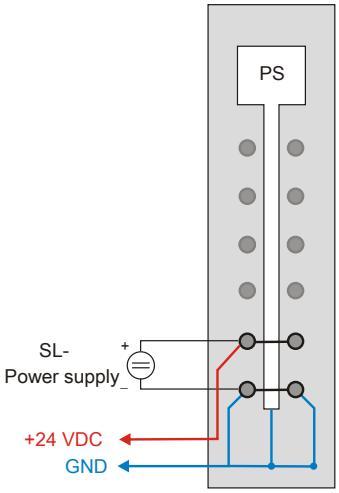


Image 8: SafeLOGIC connection example

6 Register description - X20SL80xx

6.1 Parameters in the I/O configuration

Group: POWERLINK parameters

Parameter	Description	Default value	Units
Mode	SafeLOGIC can only be operated as a "controlled node". A "management node (MN)" is not supported.	controlled node	-
Response timeout [us]	Response timeout for POWERLINK. • Permissible values: 1 - 30000	25	µs
Multiplexed station	Specifying the multiplexed station operating mode.	Off	-

Parameter value	Description
On	SafeLOGIC is operated as a multiplexed station.
Off	SafeLOGIC is not operated as a multiplexed station.

Table 13: I/O configuration parameters: POWERLINK parameters

Group: Function model

Parameter	Description	Default value	Units
Function model	This parameter is reserved for future function expansions.	Default	-

Table 14: I/O configuration parameters: Function model

Group: General

Parameter	Description	Default value	Unit
Module supervised	System behavior when a module is missing	On	-

Parameter value	Description
On	Missing module causes service mode to be activated.
Off	Missing module is ignored.

Node used as IP gateway	This parameter is reserved for future function expansions.	240	-
SafeLOGIC ID	For applications with multiple SafeLOGIC controllers, the parameter specifies the unique SafeLOGIC address. • Permissible values: 1 - 1024	Assigned automatically	-
SafeMODULE ID	Unique safety address for the module • Permissible values: 1	1	-
SafeDESIGNER project	Name of the safety project	Assigned automatically	-
Safe Runtime version (up to Release 1.3)	Reserved	-	-
SafeDESIGNER version	SafeDESIGNER version of the safe project for this SafeLOGIC controller.	Assigned automatically	-
Authorization	To activate the "Authorization" function, see Authorization.	disabled	-

Parameter value	Description
enabled	The "Authorization" function is active, the standard CPU can block acknowledgment actions from the SL controller.
disabled	The "Authorization" function is deactivated, the standard CPU has no effect on acknowledgment functions.

Table 15: I/O configuration parameters: General

Group: Communication from SafeDESIGNER to SafeLOGIC

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

When SPROXY is activated, the SafeLOGIC can be accessed via a TCP/IP port on the standard CPU, which uses the SafeDESIGNER setting "SL communikation via the CPU" (SafeDESIGNER V2.80 or higher).

Parameter	Description	Default value	Units
Activate SPROXY	Activates the SafeDESIGNER online connection	Off	-
Server Communication Port	TCP/IP port numbers used for accessing the SafeLOGIC. **Note:** If multiple SafeLOGICs are present in the project, then a different port number must be set for each one!	50000	-

Table 16: I/O configuration parameters: Communication from SafeDESIGNER to SafeLOGIC

Group: Communication from CPU to SafeLOGIC

Parameter	Description	Default value	Units
Number of BOOL channels	Number of BOOL channels from CPU to SafeLOGIC. • 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 CPU to SafeLOGIC. • 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 CPU to SafeLOGIC. • Permissible values: 0 - 30;	0	-
Number of UINT channels	Number of UINT channels from CPU to SafeLOGIC. • Permissible values: 0 - 30;	0	-
Number of DINT channels (Safety Release 1.4 and AR V3.08 required)	Number of DINT channels from CPU to SafeLOGIC. • Permissible values: 0-15;	0	-
Number of UDINT channels	Number of UDINT channels from CPU to SafeLOGIC. • Permissible values: 0 - 15;	0	-

Table 17: I/O configuration parameters: Communication from CPU to SafeLOGIC

Group: Communication from SafeLOGIC to CPU

Parameter	Description	Default value	Units
Number of BOOL channels	Number of BOOL channels from SafeLOGIC to 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 SafeLOGIC to 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 SafeLOGIC to CPU. • Permissible values: 0 - 30;	0	-
Number of UINT channels	Number of UINT channels from SafeLOGIC to CPU. • Permissible values: 0 - 30;	0	-
Number of DINT channels (Safety Release 1.4 and AR V3.08 required)	Number of DINT channels from SafeLOGIC to CPU. • Permissible values: 0-15;	0	-
Number of UDINT channels	Number of UDINT channels from SafeLOGIC to CPU. • Permissible values: 0 - 15;	0	-

Table 18: I/O configuration parameters: Communication from SafeLOGIC to CPU

Group: Communication from SafeLOGIC to SafeLOGIC

Parameter	Description	Default value	Units
Use as source SafeLOGIC	This parameter configures this SafeLOGIC as data source for another SafeLOGIC.		
	Parameter value / **Description** On: This SafeLOGIC is available as a data source for another SafeLOGIC device. Off: This SafeLOGIC is not available as a data source for other SafeLOGIC devices.		
Extended source SafeLOGIC communication (Safety Release 1.4 and AR V3.08 required)	Activates the possibility to configure the number of data points for SafeLOGIC-to-SafeLOGIC communication for connections on which this SafeLOGIC serves as data source for another SafeLOGIC.	Off	-
Connected SafeLOGIC modules[1)] SafeLOGIC ID of connection *1-10* (up to Safety Release 1.3)	This parameter configures SafeLOGIC to SafeLOGIC communication. An X20SL8001 is capable of communicating with 10 other SafeLOGIC devices, i.e. 10 communication links are available here. The SafeLOGIC ID for the SafeLOGIC device relevant for the respective communication link should be entered here as parameter value.	0	-
Group: Connected SafeLOGIC modules[1)] (Safety Release 1.4 or higher)			
Group: Connection *1-10*	Configuration of the maximum 10 SafeLOGIC devices to which this SafeLOGIC will establish a connection.		
SafeLOGIC ID of connection *1-10*	SafeLOGIC ID where the connection should be made	0	-
Group: Output channels (Safety Release 1.4 and AR V3.08 required)			
Number of BOOL channels	Number of channels with the respective data type	8	-
Number of INT channels		0	-
Number of UINT channels		0	-
Number of DINT channels		0	-
Number of UDINT channels		0	-
Group: Input channels (Safety Release 1.4 and AR V3.08 required)			
Number of BOOL channels	Number of channels with the respective data type	8	-
Number of INT channels		0	-
Number of UINT channels		0	-
Number of DINT channels		0	-
Number of UDINT channels		0	-

Table 19: I/O configuration parameters: Communication from SafeLOGIC to SafeLOGIC

1) only X20SL8001 and X20SL8011

6.2 SafeDESIGNER parameters

Group: Basic

Parameter	Description	Default value	Unit
Min_required_FW_Rev	This parameter is reserved for future function expansions.	Basic release	-
Cycle_Time_us	This parameter determines the cycle time of the SafeLOGIC controller. • Permissible values: 800 - 20000 µs The defined value is internally rounded up to the next whole number multiple of the POWERLINK cycle time.	2000	µs
Cycle_Time_max_us (beginning with Release 1.5)	Parameter for checking whether a maximum time between 2 cycles is exceeded. • Permissible values: 800 - 21000 µs IMPORTANT: This value should not be the same as the actual cycle time; network jitter must also be taken into account.	21000	µs
SSDO_Creation	This parameter defines the number of acyclic processing steps per SafeLOGIC cycle. This parameter can be used to optimize the system's boot behavior. The default value "Time dependent" ensures compatibility with Release 1.1.	Time dependent	-

Parameter value	Description
Time dependent	Depends on the SafeLOGIC cycle time (compatible to Release 1.1): • With cycle times <= 3 ms = 1_per_5_cycles • With cycle times > 3 ms = 1_per_cycle
1 every 5 cycles	One acyclic processing step is distributed over 5 SafeLOGIC cycles • Can lead to long boot times • Minimized communication overhead in each cycle
1 every cycle	One acyclic processing step per SafeLOGIC cycle • Average boot times • Average communication overhead in each cycle
5 every cycle	5 acyclic processing steps per SafeLOGIC cycle • Minimum boot times • Maximum communication overhead in each cycle

Node_Guarding_Timeout_s	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 SafeLOGIC. This parameter also defines how long it takes for the SafeLOGIC controller to detect a missing module. **Notes** • The shorter the time, the more data is asynchronous. • This setting is not critical to safety functionality. The time for safely turning off actuators is determined independently using the Worst_Case_Response_Time parameter.	60	s
ExternalMachineOptions (beginning with Release 1.4)	Activation for the external machine options	No	-

Parameter value	Description
Yes-CAUTION	External machine options are activated
No	External machine options are deactivated

ExternalStartupFlags (beginning with Release 1.4)	Activation for the external startup flags	No	-

Parameter value	Description
Yes-CAUTION	External startup flags are activated
No	External startup flags are deactivated

RemoteControlAllowed (beginning with Release 1.4)	Activates remote control of the SafeLOGIC controller	No	-

Parameter value	Description
Yes-CAUTION	Remote control of SafeLOGIC controller enabled
No	Remote control of SafeLOGIC controller blocked

Table 20: SafeDESIGNER parameters: Basic

Information:

The 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 the "Info" function. If the parameter "Cycle_Time_us" is smaller than or too close to the necessary processing time, a cycle time violation can occur.

Additional information about this can also be found in section 6.4 "SafeLOGIC info dialog box in SafeDESIGNER" on page 16.

Danger!

As long as one of the parameters "ExternalMachineOptions", "ExternalStartupFlags" or "RemoteControlAllowed" is set to "YES - Caution" (thereby enabling one of these functions to be used in the SafeDESIGNER), the corresponding notices in the section 8 "POWERLINK data interface" on page 22 must be taken into consideration. Failure to do so can result in hazardous situations caused by malfunctions.

Group: Safety_Response_Time_Defaults

Generally, the parameters for safe response time are configured the same for all stations involved in the application. This is why these parameters are configured in the SafeDESIGNER for the SafeLOGIC in the Safety_Response_Time_Defaults group.

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

Parameter	Description	Default value	Units
Synchronous_Network_Only	This parameter determines the synchronization properties of the underlying network. **Parameter value** / **Description** Yes: In order to calculate the safe 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 safe response time. • Permissible values: 200 - 30000 µs	5000	µs
Max_Powerlink_CycleTime_us	This parameter specifies the maximum POWERLINK cycle time used to calculate the safe response time. • Permissible values: 200 - 30000 µs	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 safe response time. A value of 0 means that a copy task was not included for the response time. • Permissible values: 0 - 30000 µs	5000	µs
Min_X2X_CycleTime_us	This parameter specifies the minimum X2X cycle time used to calculate the safe response time. • Permissible values: 200 - 30000 µs	200	µs
Min_Powerlink_CycleTime_us	This parameter specifies the minimum POWERLINK cycle time used to calculate the safe response time. • Permissible values: 200 - 30000 µs	200	µs
Min_CPU_CrossLinkTask_CycleTime_us	This parameter specifies the minimum cycle time for the copy task in the CPU used to calculate the safe response time. A value of 0 means that configurations without copy tasks were included for the response time. • Permissible values: 0 - 30000 µs	0	µs
Worst_Case_Response_Time_us	This parameter specifies the limit value for monitoring the safe response time. The value of the parameter can be found in the calculation tool for the safe response time. • Permissible values: 3000 - 500000 µs	50000	µs

Table 21: SafeDESIGNER parameters: Safety_Response_Time_Defaults

Group: Commissioning (only X20SL8001 and X20SL8011)

The parameter SafeMachineOption00 - SafeMachineOption31 makes it possible to activate or deactivate dedicated machine options during start-up.

Parameter	Description	Default value	Units
SafeMachineOptionXX	With this parameter, individual machine options can be enabled or disabled during commissioning.	OFF	-

Parameter value	Description
ON	Machine option XX is activated. The channel SafeMaschineOptionXX is constantly set to SAFETRUE.
OFF	Machine option XX is deactivated. The channel SafeMachineOptionXX is constantly set to SAFEFALSE.

Table 22: SafeDESIGNER parameters: Commissioning (only X20SL8001 and X20SL8011)

6.3 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	-	UINT	Module code
HardwareVariant	Read	-	UINT	Hardware variants
FirmwareVersion	Read	-	UINT	Module firmware version
UDID_low	Read	-	UDINT	UDID, lower 4 bytes
UDID_high	Read	-	UINT	UDID, upper 2 bytes
SafeModuleOK	-	Read	SAFEBOOL	Indicates if the safe communication channel is OK
BOOL1xx	Write	Read	BOOL	Communication channel - CPU to SafeLOGIC
BOOLext1xxx	Write	Read	BOOL	Communication channel - CPU to SafeLOGIC
INT1xx	Write	Read	INT	Communication channel - CPU to SafeLOGIC
UINT1xx	Write	Read	UINT	Communication channel - CPU to SafeLOGIC
UDINT1xx	Write	Read	UDINT	Communication channel - CPU to SafeLOGIC
BOOL0xx	Read	Write	BOOL	Communication channel - SafeLOGIC to CPU
BOOLext0xxx	Read	Write	BOOL	Communication channel - SafeLOGIC to CPU
INT0xx	Read	Write	INT	Communication channel - SafeLOGIC to CPU
UINT0xx	Read	Write	UINT	Communication channel - SafeLOGIC to CPU
UDINT0xx	Read	Write	UDINT	Communication channel - SafeLOGIC to CPU
SafeBOOLx	-	Write	SAFEBOOL	Communication channel - SafeLOGIC to SafeLOGIC
SafeMachineOptionxx[1)]	-	Read	SAFEBOOL	Internal channel for machine options

Table 23: Channel list

1) Only X20SL8001 and X20SL8011

Information:

Channels for SafeLOGIC to SafeLOGIC communication: see Display in SafeDESIGNER

6.4 SafeLOGIC info dialog box in SafeDESIGNER

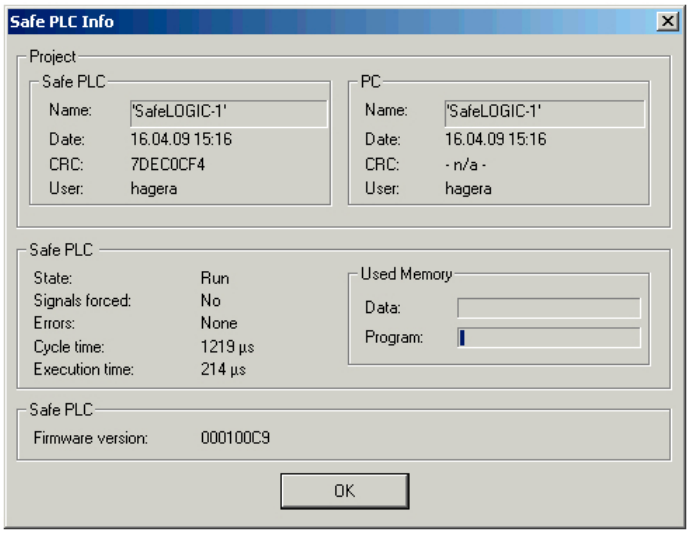


Image 9: SafeLOGIC info dialog box

Project	Data defined in the project	
Safe PLC	Project data that is saved on the SafeKEY used for the SafeLOGIC controller.	
	Name	Name of the project
	Date	Date of the last change
	CRC	CRC
	User	User that 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 compiled
	User	User that made the last change
Safe PLC	Status and information about the SafeLOGIC controller	
State	Run	The safety application is being executed.
	On	There is not a valid program on the SafeKEY used for the SafeLOGIC controller.
	Stop [Safe]	The SafeLOGIC controller is in Safe mode. A program is loaded, but is not being executed.
	Run [Safe]	The SafeLOGIC controller is in Safe mode. The program is being executed.
	Stop [Debug]	The SafeLOGIC controller is in Debug mode. The program is not being executed.
	Run [Debug]	The SafeLOGIC controller is in Debug mode. The program is being executed.
	Halt [Debug]	The SafeLOGIC controller is in Debug mode. The program has been halted (single cycle).
	No Execution	The SafeLOGIC controller is booting: ready for "Run", but still waiting for modules.
	TIMEOUT	Communication problem between SafeDESIGN-ER and the SafeLOGIC controller.
	Failure	The SafeLOGIC controller is in Fail SAFE mode.
Signals forced	No	No variables are forced
	Yes	Variables are forced
Errors	Information regarding existing error messages in the SafeDESIGNER message window.	
Cycle time	Cycle time that is actually required, maximum value since the last power up This value is only meaningful if Safe PLC state = Run	
Execution time	Actual application execution time; This value corresponds to the Safe PLC cycle time minus the system and communication overhead	
Used memory	Bar that shows the system resources that are being used	
	Data	Data memory for the safe application
	Program	Program memory for the safe application
Firmware version	Firmware version	

See the SafeDESIGNER online help for detailed information about the SafeLOGIC controller info dialog box in SafeDESIGNER.

7 Maintenance scenarios

A description of operating elements can be found in the 5 "Control and connection elements" on page 3 section.

7.1 Module exchange

SafeLOGIC recognizes, on its own, when safe modules have been exchanged. Following a module replacement, the entire system (SafeLOGIC, openSAFETY) automatically ensures that the module is operated again with the correct parameters and that incompatible modules are rejected. However, the following possible errors may remain after a module exchange:

- Mix-up of the terminals between several modules
- Wiring errors
- Mix-ups of SafeIO modules with each other

7.1.1 Mix-up of the terminals between several modules

To avoid mixing up the terminals between several modules, the user must test the safety function by performing a wiring test.

> **Danger!**
>
> **The user must make sure that the wiring test can detect a mix-up of the terminals.**
>
> **You must always validate the overall safety function.**

7.1.2 Wiring errors

A wiring error could 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.**
>
> **You must always validate the overall safety function.**

7.1.3 Mix-ups of SafeIO modules with each other

Errors in the functional application can cause SafeIO modules to get mixed up, which appears identical to a module exchange in SafeLOGIC. To prevent this error, the user must confirm the number of exchanged modules on the SafeLOGIC. This means that the number of modules exchanged by the user and the exchanges recognized by the system are linked and additional exchanges can be detected.

The SafeLOGIC informs the user via blink code on the MXCHG LED of the number of exchanged modules. One blink code represents up to 4 different modules. The blink code lasts for 4 s, and the LED is switched on for each module present. The MXCHG LED blinks continuously if there are more than 5 different modules.

The user must check if the number of exchanged modules recognized by the SafeLOGIC corresponds to the actual number of exchanged modules. If the values are the same, the user must confirm the number and execute a wiring test. The wiring test can focus on the exchanged modules.

If a difference should arise, the user must confirm the number of exchanges determined by SafeLOGIC and execute a comprehensive wire test for all modules.

> **Danger!**
>
> **You must always validate the overall safety function.**

7.1.4 Exchanging the individual module

In situations requiring just one module exchange (MXCHG LED signals a blink code for an exchanged module) where the wiring remains the same, the user can skip the wiring test, because in this case

- Mix-up of the terminals between several modules
- Wiring errors
- Mix-ups of SafeIO modules with each other

can be ruled out.

> **Danger!**
>
> **The wiring test can only be excluded, if, in the course of an individual module exchange, no additional changes are made (e.g. unplugging terminals, removing the wiring, etc.).**

7.1.5 Confirming a module exchange

The selection switch must be in one of the following positions to confirm the number of exchanged modules:

- 1 - one module exchanged
- 2 - 2 modules exchanged
- 3 - 3 modules exchanged
- 4 - 4 modules exchanged
- n - five or more modules exchanged.

The exchange can be confirmed and the accompanying wiring test can be focused on the exchanged modules when up to four modules are exchanged. When more than four modules are exchanged, a comprehensive wiring test must be performed for all modules.

Following confirmation of the module exchange, the SafeLOGIC immediately commences a module scan.

> **Danger!**
>
> **The user must make sure that the wiring test can detect a wiring error or mix-up of the terminals.**
>
> **You must always validate the overall safety function.**

7.2 Other errors in module configuration

The aforementioned differences are limited exclusively to module exchange. An error is signaled if a device is missing (except for when the device is defined as optional), has an incorrect HW code, or other problems are present on the module (e.g. incorrect parameters that may not be changed by the SafeLOGIC). In any of these cases, the "MXCHG" LED blinks constantly. This status is only indicated if there is no "module exchange" status and no firmware exchange. The status cannot be acknowledged.

> **Danger!**
>
> **It is your responsibility to ensure that all necessary measures for repair are initiated after an error occurs as successive errors can result in dangerous situations.**

7.3 Confirmation of firmware change

A firmware change is indicated by the blinking "FW-ACKN" LED. Selecting the "FW-ACKN" position confirms this status. A firmware exchange must always be concluded with a full function test.

Danger!

The function test can only be performed by someone who is familiar with the safety application and its functions and is trained in the procedure of exchanging firmware.

You must always validate the overall safety function.

Danger!

Only use firmware revisions that are listed in the "List of Module Versions" belonging to the TÜV certificate for B&R safety technology (see the B&R website under Service > General downloads).

7.4 Module scan execution

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 delay during a module replacement until the SafeLOGIC controller recognizes the new module, the user can also manually trigger this function. The result of the scan is described in the following sections:

- "Module exchange" on page 17
- "Other errors in module configuration" on page 18
- "Confirmation of firmware change" on page 19

The procedure itself is started with the selection switch in the "SCAN" position and signaled by a running light with the "ENTER", "MXCHG" and "FW-ACKN" LEDs. At the conclusion of the scan, the "ENTER" LED lights up for 0.8 s. After that, the results are signaled (e.g. three modules replaced).

7.5 SafeKEY

7.5.1 Removing a SafeKEY

Removing a SafeKEY always results in a BOOT status change (the LED letters "F", "I" and "L" light up) and a complete cutoff of the safe application.

Information:

Removing the SafeKEY during operation results in a restart of SafeLOGIC and a cutoff of all safety-related actuators.

Pulling the SafeKEY during operation can destroy the data on the SafeKEY.

Removing the SafeKEY during operation must be avoided.

The sequence "Creating a backup for the SafeKEY" is not affected by this.

7.5.2 SafeKEY exchange confirmation

A SafeKEY exchange is indicated by permanent illumination of the "FW-ACKN" LED and must be acknowledged with the confirmation sequence "SK-XCHG". Additionally, a complete function test is required.

Danger!

Exchanging a SafeKEY activates the safety application stored on the SafeKEY. Always check the project CRC and date the safety application project was saved on the SafeKEY.

Danger!

You must always validate the overall safety function.

7.5.3 Changing the application on the SafeLOGIC device by exchanging the SafeKEY

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 should be carried out.

- Set selection switch to "SK-COPY" position.
- Press the confirmation button - action acknowledged with the "ENTER" LED.
- The SafeKEY configuration data is saved on the SafeLOGIC device. The "SKEY" LED blinks with every access.
- The "FW-ACKN" LED will flash after the copying procedure. The prior SafeKEY can now be replaced by the SafeKEY with the new application. A maximum of 30 s is provided for this process. The "FW-ACKN" LED blink frequency increases after 20 seconds to signal the end of the exchange phase.
- The acknowledge key must be pressed again after the new SafeKEY has been inserted. The selection switch remains on the setting "SK-COPY".
- The internal, temporarily saved configuration data is saved on the new SafeKEY. Then a reset is triggered automatically and the data from the new SafeKEY is transferred.
- Following the reset, the SafeKEY exchange must be confirmed. To do this, move the selection switch to the setting "SK-XCHG".
- Press the confirmation button - action acknowledged with the "ENTER" LED.
- Execution of a complete function test.

Information:

If the new SafeKEY is not acknowledged after 30 seconds, the function ends, i.e. in case the function is triggered inadvertently, the copy function ends automatically after 30seconds. If no SafeKEY is inserted after 30 seconds, the SafeLOGIC switches to BOOT status (the letters LEDs "F", "I", and "L" illuminate).

Danger!

This procedure activates the safety application stored on the SafeKEY. Always check the project CRC and date the safety application project was saved on the SafeKEY.

Danger!

You must always validate the overall safety function.

Information:

This sequence can also be used to create a SafeKEY backup by using a second SafeKEY with an identical safety application. After executing the sequence, two identical SafeKEYs are available (backup copy).

7.6 Replacing a SafeLOGIC controller

Replacing a SafeLOGIC controller involves the same mechanisms as a normal module exchange. 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!**
>
> **You must always validate the overall safety function.**

7.7 Authorization

Functionality

- Confirming a module exchange
- Confirming a firmware exchange
- SafeKEY exchange confirmation
- Backing up the SafeKEY
- Replacing a SafeLOGIC controller

can be blocked by the functional CPU. This allows the actions to be made dependent on one application-specific user concept. This option is not possible from a safety perspective because the functions take place in the functional CPU.

The objects in Index 0x2402 that can be accessed via the POWERLINK library are available here.

Index:Subindex	Object description	Data type	Access	Values	Description
0x2402:0x00	Number of entries	USINT	R	0x22	Number of entries on this index
0x2402:0x01	EnableAutorization	UDINT	RW	"AENA", 0x41454E41	Activate the authorization
				"ADIS", 0x41444953	Deactivate the authorization
0x2402:0x04	EnableModuleExchange	UDINT	RW	"UDID", 0x554444944	Authorization to confirm module exchange is provided
				All other values	Authorization to confirm module exchange is not provided
0x2402:0x05	EnableFWMismatch	UDINT	RW	"FWAC“, 0x46574143	Authorization to confirm firmware updates is provided
				All other values	Authorization to confirm firmware updates is not provided
0x2402:0x06	EnableSKeyExchange	UDINT	RW	"SKEY", 0x534B4559	Authorization to confirm SafeKEY exchange is provided
				All other values	Authorization to confirm SafeKEY exchange is not given

User requests on the SafeLOGIC that are not authorized by the CPU are signaled with a steadily lit "ENTER" LED.

8 POWERLINK data interface

8.1 Remote control

Requirements

Parameter environment	Parameter	Value
Automation Studio: Properties dialog box "Change Runtime Versions"	Safety Release	>= 1.4
SafeDESIGNER: Parameters from the SafeLOGIC group, "Basic"	RemoteControlAllowed	YES-Caution

Danger!

- **In an FMEA, the user must examine how the function is applied and if there are any potential risks. In particular, any predictable misuse and typical application-specific sources of error must be taken into consideration In the FMEA. Potential risks must be minimized with additional measures. This function can only be enabled and used in the SafeDESIGNER once the determined residual risk has been estimated as low enough for the intended application.**
- **The program sections in the functional application that are involved in executing the function must meet the requirements specified in ISO 13849-1:2007, chapter 4.6.4 or IEC 62061, chapter 6.11.2. The program sections must be executed properly (i.e. in accordance with one of these standards) and documented accordingly.**
- **The functions can only be executed by people with proper authorization. Access to the respective visualization components must be limited to the authorized group of personnel.**
- **Local personnel must be informed when one of these functions is accessed. The user must implement suitable measures to ensure that remote access is not possible without notification to the local personnel.**
- **Proper functionality must be verified with a thorough function test. The test procedures and results must be documented. The test must be able to identify any data mismatches between the visualization application and the safety application. Proper functionality must again be verified in a thorough function test after changes have been made in Automation Runtime or after changes to the functional application.**

General information

In Safety Release 1.4 and higher, the confirmation sequences needed for the various maintenance scenarios can also be triggered remotely by the functional application. To make this possible, a POWERLINK object interface was implemented on the SafeLOGIC module that can be operated in Automation Studio by using the library"AsEPL".

Remote control interface

POWERLINK V2 object:

Index:Subindex	Object description	Data type	Access	Values	Description
0x2406:0x00	NumberOfEntries	USINT	R	0x02	Number of entries on this index
0x2406:0x01	RemoteRequest_OCT	See command structure	W	-	The command structure is written to this element
0x2406:0x02	RemoteResponse_OCT	See status structure	R	-	Return value from the SafeLOGIC after status query command

Table 24: SAF_RemoteControl_REC: Remote control interface

Command structure

A command structure must first be prepared and filled with values in order to send a command to the SL. This structure must be written to the the object "RemoteRequest_OCT" of the remote control interface using a POWERLINK write command.

The structure cannot contain any filler bytes and must look similar to this:

Element	Data type	Comment
Command	UINT	Remote control command, see Commands
Number	UINT	Consecutive command number, specified by the programmer, can be read back in the status structure
Data	UINT	Data for command. see Commands
Password	USINT[16]	MD5 hash code of the SafeKEY password
NewPassword	USINT[16]	MD5 hash code of the new SafeKEY password

Table 25: Command structure

Note:

The entry "NewPassword" can only be applied to and transferred together with the structure in the event of the command "PASSWORD_CHANGE".

Commands

Command	Description	Data	Comment
0x0100	ENTER	0x0020	Acknowledge more than 4 UDID mismatches
		0x0030	Acknowledge 4 UDID mismatches
		0x0040	Acknowledge 3 UDID mismatches
		0x0050	Acknowledge 2 UDID mismatches
		0x0060	Acknowledge 1 UDID mismatch
		0x0100	FW-ACKN, acknowledge a new firmware versions
		0x0200	SK-XCHG, acknowledge a SafeKEY exchange
		0x1000	TEST, start an LED test (5s)
		0x2000	SCAN, start a system scan
		0x3000	SK-COPY, copy SafeKEY
		0x4000	Resume copy after a new SafeKEY has been inserted
		0x5000	Change SafeKEY password
		0x6000	Formatting the SafeKEY
0x0200	STATUS_SL	0x0000	Query status of SL

Table 26: Commands

Note:

- **Password protection:**
  Commands will only be executed if the correct password is entered and remote control is activated.

  Exceptions

  - The command STATUS_SL also works without password and without activated remote control.
  - With the "Change SafeKEY password" command, the password is not checked if there is no valid data on the SafeKEY.
    Note: This makes it possible to reinstall a blank/formatted SafeKEY.

- **Locking:**
  Manual operation of the SL via selection switch and the ENTER key and remote control are locked in the firmware so that only one at a time can be active.
  If a manual command is being executed, then no commands can be made via remote control and vice versa.
  Note: The command SK-COPY can also be executed right on the SL after exchanging a SafeKEY (SK-COPY via remote control starts the copying procedure. SK-COPY can be used to acknowledge an exchanged SafeKEY on the SL).
- Only ONE command can be executed at a time. As long as one command is running, all other commands are rejected.

- **Response:**
  Every command automatically generates a response. To view the status of the ENTER commands, the command STATUS_SL must be sent and the status structure must be read.
- **Logging:**
  All commands except for STATUS_SL are logged in the safety logbook, regardless of whether or not they were able to be executed (e.g. rejected due to lack of authorization).

- The commands "Acknowledge x UDID mismatches" trigger a module scan after being executed, just like in manual operation.
- The commands SK-XCHG, SK-COPY and FW-ACKN cause the SL to restart, just like in manual operation. Note: Beginning with R1.5, restarting is delayed by 5 s so that the standard CPU has time to evaluate the command response.
- When multiple commands are combined, the module scan and/or SL restart are not performed until ALL commands have been executed.

Read status

After the SL status has been queried via the command "STATUS_SL", the values are then saved to the object "RemoteResponse_OCT". The values can be read using a POWERLINK read command and are arranged according to the following structure.

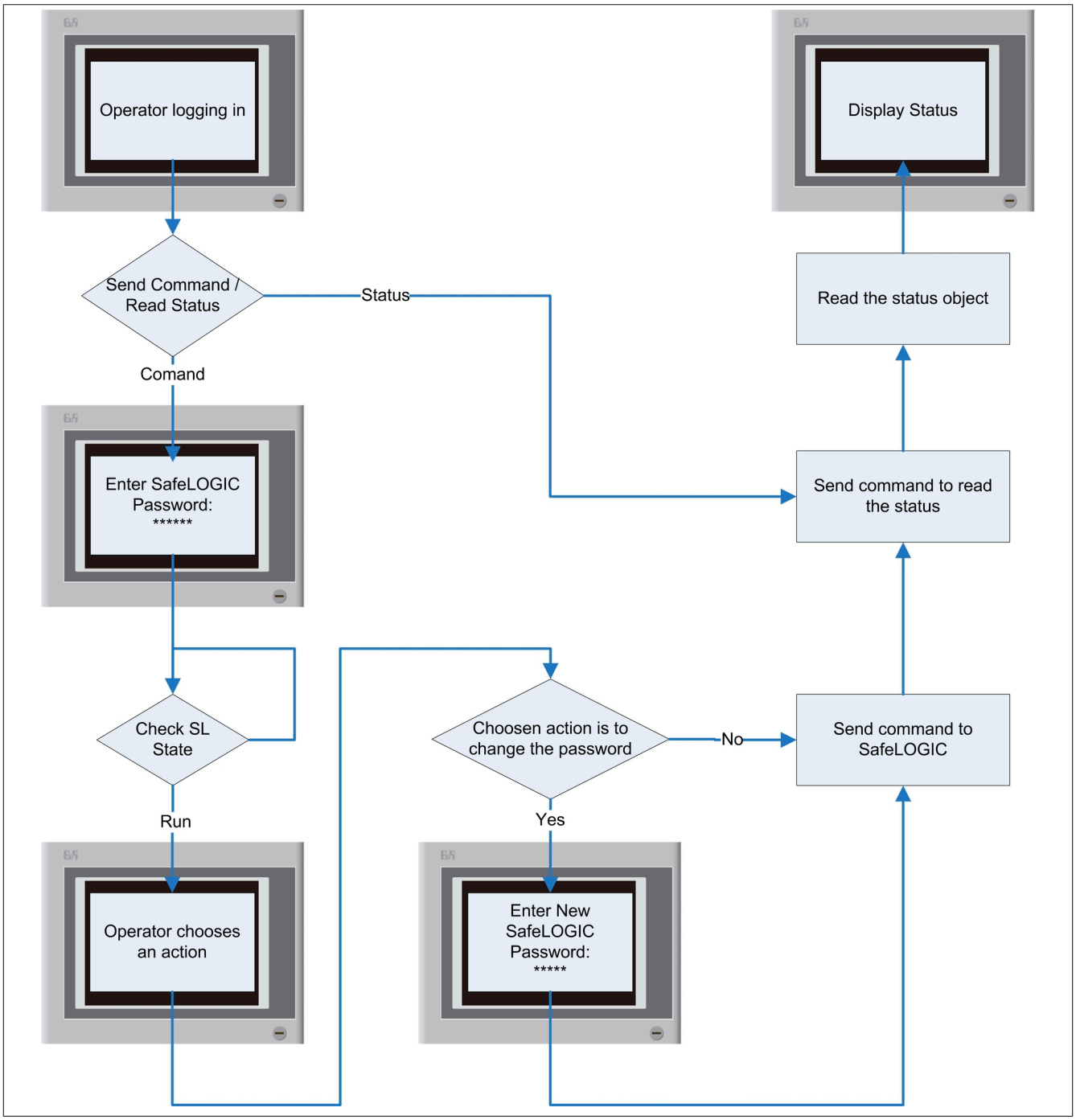
Element	Data type	Value	Comment
Command	UINT	-	Last received command
Number	UINT	-	Running number of last received command
Status	UINT	-	The status numbers correspond to the error numbers entered in the logbook
		0	The last received command was valid and is being executed
		1	Reserved: HMI command was executed successfully (Logger)
		2	Reserved: HMI command was executed with errors (Logger)
		3	Reserved: Remote command was executed successfully (Logger)
		4	Reserved: Remote command was executed with errors (Logger)
		5	The command is unknown
		6	The ENTER command in the Data field is unknown
		7	Remote control is not activated via the SafeDESIGNER
		8	Incorrect password
		9	Remote Control State Machine is not in IDLE (last command still being processed)
		10	Locked by HMI (command activated via rotary switch and ENTER button)
		11	SK_ACKN command not authorized
		12	FW_ACKN command not authorized
		13	SMX_ACKN to CMX_ACKN command not authorized
		14	SK_ACKN command cannot be executed, SafeKEY was not exchanged
		15	FW_ACKN command cannot be executed, different Firmware not found
		16	SMX_ACKN command cannot be executed, no modules or multiple modules were replaced
		17	DMX_ACKN command cannot be executed, fewer or more than two modules were replaced
		18	TMX_ACKN command cannot be executed, fewer or more than three modules were replaced
		19	QMX_ACKN command cannot be executed, fewer or more than four modules were replaced
		20	CMX_ACKN command cannot be executed, fewer or more than five modules were replaced
		21	SK_CONTINUE command cannot be executed, SK_COPY not started or time SK_CONTINUE expired
		22	ENTER command not possible, SK_ACKN required
		23	SK_FORMAT command is being processed, no other commands can be sent at this time
		24	SK_COPY command is being processed, no other commands can be sent at this time
		25	SK_ACKN command is being processed, no other commands can be sent at this time
		26	SMX_ACKN to CMX_ACKN command is being processed, no other commands can be sent at this time
		27	A SCAN is being executed, no other commands can be sent at this time
		28	Reserved: Remote status send failed (Logger)
		29	Incorrect length of the 0x5000 command (change password)
		30	Incorrect length of the command (for commands other than 0x5000)
State	UINT	-	State of the last ENTER command
		0	IDLE, waiting for next command
		1	ENTER command received
		2	Execute ENTER command
EnterData	UINT	-	Last received ENTER command that was correct and executed
EnterNumber	UINT	-	Running number of the last received ENTER command

Table 27: Status structure

Element	Data type	Value	Comment
EnterExecuteStatus	UINT	-	Status of the last received ENTER command, same value as usEnterData, valid value if eState = IDLE
		0x0000	Status at the beginning of execution, if state != IDLE.
			Status after execution with errors, if state = IDLE
SafeOSstate	USINT	-	Status of the safety application
SafeKeyChanged	USINT	0x01	SafeKEY has been exchanged, acknowledgment required
LedTestActive	USINT	0x01	LED test active
Scanning	USINT	0x01	Module scan active
openSAFETYstate	USINT	-	Status of openSAFETY stack
FailSafe	USINT	0x55	Module status OK - Status of the safe application: see SafeOSstate
		All other values	Module status Fail-Safe - Valid safe data is no longer being generated (**regardless of all other statuses**)
NumberOfMissingModules	UINT	0 - n	Number of missing modules
NumberOfUdidMismatches	UINT	0 - n	Number of mismatched modules
NumberOfDiffFirmware	UINT	0 - n	Number of modules with different firmware
SAddr[0..100]	UINT	0 - 1023	The safety address will be entered in this field for each SN, 0 = Module not present
MissingModules[0..15]	USINT	0x00 - 0xFF	128 bits each for displaying missing and mismatched modules, and modules with different firmware The safety address is entered in the field ausSAddr[0..100]. The respective status values are entered bitwise in these fields. Example: The address of the 9th module is entered in ausSAddr[8]. If this module is missing, then the 1st bit is set in ausMissingModules[1]
UdidMismatches[0..15]	USINT	0x00 - 0xFF	
DiffFirmware[0..15]	USINT	0x00 - 0xFF	

Table 27: Status structure

Remote control process



8.2 Machine option download

Requirements

Parameter environment	Parameter	Value
Automation Studio: Properties dialog box "Change Runtime Versions"	Safety Release	>= 1.4
SafeDESIGNER: Parameters from the SafeLOGIC group, "Basic"	ExternalMachineOptions	YES-Caution
SafeDESIGNER: Parameters from the SafeLOGIC group, "Basic"	ExternalStartupFlags	YES-Caution

Danger!

- **In an FMEA, the user must examine how the function is applied and if there are any potential risks. In particular, any predictable misuse and typical application-specific sources of error must be taken into consideration In the FMEA. Potential risks must be minimized with additional measures. This function can only be enabled and used in the SafeDESIGNER once the determined residual risk has been estimated as low enough for the intended application.**
- **The program sections in the functional application that are involved in executing the function must meet the requirements specified in ISO 13849-1:2007, chapter 4.6.4 or IEC 62061, chapter 6.11.2. The program sections must be executed properly (i.e. in accordance with one of these standards) and documented accordingly.**
- **The functions can only be executed by people with proper authorization. Access to the respective visualization components must be limited to the authorized group of personnel.**
- **Local personnel must be informed when one of these functions is accessed. The user must implement suitable measures to ensure that remote access is not possible without notification to the local personnel.**
- **The information used for the machine options cannot be changed, inverted or manipulated in any way in the functional application. These type of requirements (e.g. activating machine type A causes machine options 1, 2 and 3 to be activated) must be implemented in the safe application in SafeDESIGNER and not in the functional application.**
- **The sections of the program responsible for confirming the received configuration should be executed separately from the sections that transfer the configuration to the SafeLOGIC module. The visualization objects that are used must be arranged in such a way so that different pixel positions can be used for displaying the data on the screen.**
- **Proper functionality must be verified with a thorough function test. The test procedures and results must be documented. The test must be able to identify any data mismatches between the visualization application and the safety application. Proper functionality must again be verified in a thorough function test after changes have been made in Automation Runtime or after changes to the functional application. Make a list of potential dangers!**

General information

In Safety Release 1.4 and higher, machine configurations can be applied from the functional application. To make this possible, a POWERLINK object interface was implemented on the SafeLOGIC module that can be operated in Automation Studio by using the library"AsEPL".

This interface can be used to assign signals for external machine options, the startup behavior of the modules and UDIDs of the modules. If this type of structure is transferred to the SL, then the contained settings will be applied after restarting.

The settings can be made using the visualization application. The machine operator can change parameters in the visualization application and must check and acknowledge all of the settings after the download is complete.

Interface

POWERLINK V2 objects:

Index:Subinde	Object description	Data type	Access	Values	Description
0x2405:0x00	NumberOfEntries	USINT	R	0x08	Number of entries on this index
0x2405:0x01	Authorization_DOM	USINT[16]	W	-	Authorizes data transfer by writing the MD5 hash code of the SafeKEY password for this object
0x2405:0x02	FileStreamData_DOM	-	W	-	Data being transferred to the SL will be written to this object
0x2405:0x03	ParserStatus_U16	UINT	R	0	No errors during data transfer
				1	Wrong protocol version or error in the header
				2	File already open
				3	File invalid
				4	File too large
				5	Error while writing
				6	Error at the end of the stream
				7	Incorrect checksum
				8	Wrong UDID
				9	Wrong file size
				10	No write permission
0x2405:0x04	UnlockStatus_U16	UINT	R	0	No error occurred
				1	Error while obtaining file information
				2	Error while reading
				3	Write error
0x2405:0x05	Busy_BOOL	BOOL	R	FALSE	Idle data transfer or lock
				TRUE	Busy data transfer or lock
0x2405:0x06	Reboot_DOM	USINT[16]	W	-	Restarts the SL by writing the MD5 hash code of the password for this object
0x2405:0x07	ProjectKey_U64	LREAL	W	-	Releases the application by writing the unlock key on this object
0x2405:0x08	AutoCnfKey_U64	LREAL	W	-	Releases the machine options by writing the unlock key on this object
0x2405:0x09	ProjectID_U32	UDINT	R	-	Project CRC of the SafeDESIGNER project
0x2405:0x0A	AutoCnfID_U32	UDINT	R	-	Value "Timestamp of file" - see "Format"

Table 28: SAF_FileParser_REC: File interface

Format

In to specify these settings externally, a file must first be created that contains this information. This file can either be prepared on a PC and stored on the functional controller or created on the functional controller during operation. This can be done using the visualization application. The machine operator who selects or creates this file via the visualization application must check and acknowledge all of the settings after the download is complete.

Section	Name	Offset within the section	Byte	Meaning
Header	Amount	0	2	Number of sections described, typically 3
	Length	2	2	Length of header, typically 64
	Version	4	2	Version number for the header format (default 0x0400)
	offset	6	2	Position of the description for the 1st section, typically 8
	File time stamp	8	4	Unique time stamp for unique file identification
	Length of section 1	12	4	Length of the 1st section, depending on the number of exchanged modules
	Offset of section 1	16	4	Absolute position of the 1st section, typically 64
	Length of section 2	20	4	Length of the 2nd section, typically 76
	Offset of section 2	24	4	Absolute position of the 2nd section, depending on the length of the 1st section
	Length of section 3	28	4	Length of the 3rd section, typically 268
	Offset of section 3	32	4	Absolute position of the 3rd section, depending on the length of the 1st and 2nd section
	Reserved	36	24	Reserved
	CRC32	60	4	CRC32 of the header, amount until reserve[1)]
UDID list (Section 1)	Amount	0	2	Number of exchanged modules (<=101)
	Length	2	2	Length of an entry, typically 8
	Version	4	2	Version number for the format of the 1st section, typically 0x0300
	offset	6	2	Section offset of the first entry, typically 8
	SADR of the 1st safety node	8	2	Safety address of the 1st module
	UDID of the 1st safety node	10	2	UDID of the first module
	...			
	SADR of the "n"th safety node	8 + (n-1)*8	2	Safety address of the "n"th module
	UDID of the "n"th safety node	10 + (n-1)*8	6	UDID of the "n"th module
	CRC32	8 + n*8	4	CRC32 of the 1st section, amount until last UDID[1)]
Machine options (section 2)	Amount	0	2	Number of machine options, typically 512
	Length	2	2	Length of the data, typically 64
	Version	4	2	Version number for the format of the machine options, typically 0x100
	offset	6	2	Offset of the data, typically 8
	Data	8	64	512 bit machine options
	CRC32	72	4	CRC32 of the 2nd section, amount until data[1)]
Module flags (section 3)	Amount	0	2	Number of module flags
	Length	2	2	Length of the data
	Version	4	2	Version number for the format of the module flags, typically 0x100
	offset	6	2	Offset of the module flags, typically 8
	Optional flags	8	128	1024 bit for optional
	Startup flags	136	128	1024 bit for startup
	CRC32	264	4	CRC32 of the 3rd section, amount until startup flags[1)]
Total CRC	CRC32	0	4	CRC32 of the total file[1)]

1) CRC32 calculation, polynomial 0x1edc6f41, starting value 0

UDID list

In the UDID list the SL can be predefined to specify the safety address where each UDID can be found when booting. If this externally specified UDID matches the physical configuration, then an exchanged or newly added module no longer has to be acknowledged because the SL already "knows" the UDID. The UDID for a module can be read-out using I/O mapping in Automation Studio.

External machine options

The external machine options offer 512 variables that can be used in safe code. These variables can be assigned a value TRUE or FALSE in the machine options file. After this file has been transferred to the SL and the system restarted, the variables will be initialized with the specified value. The external machine options behave like constants.

Module flags

SafeDESIGNER allows the user to define for each module how the safe application should behave if that module cannot be found. This setting can also be specified externally via the "Machine option file". The user can choose for each safety address whether or not the "Optional Parameter" for the corresponding module should be configured to "optional", "Startup" or "No".

Structure

The machine option structure must be added to a higher-level structure before being downloaded.

Section	Name	Data type	Meaning
Header	Version	UINT	Version of the file container. The value 0x0100 must be entered
	Amount	UINT	Number of subsequent files. The value 0x0001 must be entered. One file should be transferred, the machine option structure
	UDID	USINT[6]	UDID of the SL to which the structure will be transferred
Machine option structure	File length	UDINT	File length of the machine option structure
	File name	USINT[13]	Name of the machine option structure. "AUTOCNF.BIN" must be entered here
	File	-	Machine option structure
Checksum	Checksum	UDINT	Additive checksum of the entire structure

Table 29: Download structure

Machine option download sequence

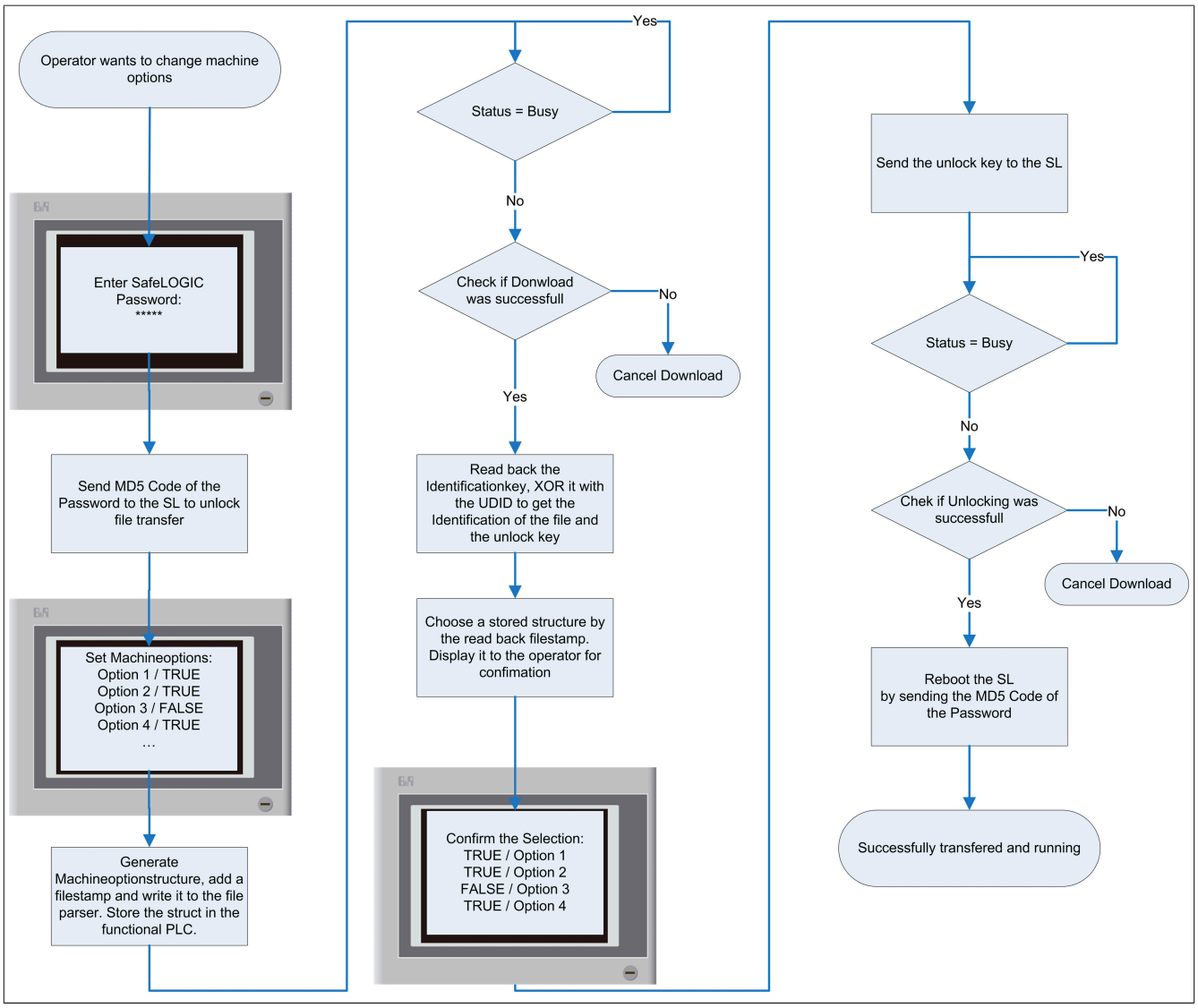


Image 10: Machine option download process diagram

A specified procedure must be followed in order to transfer the machine options from the functional controller to the SL. The download procedure must be initiated manually and intentionally by the machine operator. After being transferred, the data must then be verified. To do this, the operator must (after the download) be shown a list of all changes and settings that have been made. The defined parameters must be confirmed by the operator.

Proper downloading requires that the following steps be followed:

- Authorize download by writing the MD5 hash code of the SafeKEY password to the object "Authorization_DOM".
- Send download structure by writing it to the object "FileStreamData_DOM".
- Determine whether or not the transfer is complete by reading the object "Busy_BOOL".
- Query whether or not the data has been fully received by reading the object "ParserStatus_U16".
- Read the identification/key object. To do this, the object "AutoCnfKey_U64" must be read. This object must be linked with the UDID of the SL (by byte). This provides the user with the identification of the transferred file and the corresponding unlock key.

The UDID must be linked with the identification/key object XOR based on the following schema.

FileIdent[0] = EPLKey[0];

FileIdent[1] = EPLKey[1];

FileIdent[2] = EPLKey[2] ^ SL_UDID[0];

FileIdent[3] = EPLKey[3] ^ SL_UDID[1];

UnlockKey[0] = EPLKey[4] ^ SL_UDID[2];

UnlockKey[1] = EPLKey[5] ^ SL_UDID[3];

UnlockKey[2] = EPLKey[6] ^ SL_UDID[4];

UnlockKey[4] = EPLKey[7] ^ SL_UDID[5];

Result of the link:

Byte	Meaning
0	Identification, corresponds to the value of the element "File time stamp" in the machine option structure. This value must be used to show the operator the corresponding machine option structure.
1	
2	
3	
4	Unlock key for the machine option structure
5	
6	
7	

Table 30: Result UDID XOR link

- Writing the unlock key to the object "AutoCnfKey_U64". This requires writing the unlock key to the first 4 bytes.
- Determine whether or not decoding is complete by reading the object "Busy_BOOL".
- Query whether or not any errors occurred on the SL while decoding the data. To do this, the object "UnlockStatus_U16" must be read.
- Initiate SL reboot by writing the MD5 hash code of the SafeKEY password to the object "Reboot_DOM".

8.3 Application download

Requirements

Parameter environment	Parameter	Value
Automation Studio: Properties dialog box "Change Runtime Versions"	Safety Release	>= 1.4

Danger!

- **In an FMEA, the user must examine how the function is applied and if there are any potential risks. In particular, any predictable misuse and typical application-specific sources of error must be taken into consideration In the FMEA. Potential risks must be minimized with additional measures. This function can only be enabled and used in the SafeDESIGNER once the determined residual risk has been estimated as low enough for the intended application.**
- **The program sections in the functional application that are involved in executing the function must meet the requirements specified in ISO 13849-1:2007, chapter 4.6.4 or IEC 62061, chapter 6.11.2. The program sections must be executed properly (i.e. in accordance with one of these standards) and documented accordingly.**
- **The functions can only be executed by people with proper authorization. Access to the respective visualization components must be limited to the authorized group of personnel.**
- **Local personnel must be informed when one of these functions is accessed. The user must implement suitable measures to ensure that remote access is not possible without notification to the local personnel.**
- **The sections of the program responsible for confirming the received configuration should be executed separately from the sections that transfer the configuration to the SafeLOGIC module. The visualization objects that are used must be arranged in such a way so that different pixel positions can be used for displaying the data on the screen.**
- **Proper functionality must be verified with a thorough function test. The test procedures and results must be documented. The test must be able to identify any data mismatches between the visualization application and the safety application. Proper functionality must again be verified in a thorough function test after changes have been made in Automation Runtime or after changes to the functional application. Make a list of potential dangers! Make a list of potential dangers**

General information

In Safety Release 1.4 and higher, the safety-related application can be transferred from the functional application to the SafeKEY of the SafeLOGIC controller. To make this possible, a POWERLINK object interface was implemented on the SafeLOGIC controller that can be operated in Automation Studio using the "AsEPL" library.

This interface allows the user to transfer a container file with a predefined structure to the SafeLOGIC controller. If this type of structure is transferred to the SL, then the contained application will be applied after restarting.

Information:

To install a blank SafeKEY (e.g. new or formatted), a password must first be set (see "Change SafeKEY password" command in the "Commands" section.

File interface

POWERLINK V2 objects:

Index:Subinde	Object description	Data type	Access	Values	Description
0x2405:0x00	NumberOfEntries	USINT	R	0x08	Number of entries on this index
0x2405:0x01	Authorization_DOM	USINT[16]	W	-	Authorizes data transfer by writing the MD5 hash code of the SafeKEY password for this object
0x2405:0x02	FileStreamData_DOM	-	W	-	Data being transferred to the SL will be written to this object
0x2405:0x03	ParserStatus_U16	UINT	R	0	No errors during data transfer
				1	Wrong protocol version or error in the header
				2	File already open
				3	File invalid
				4	File too large
				5	Error while writing
				6	Error at the end of the stream
				7	Incorrect checksum
				8	Wrong UDID
				9	Wrong file size
				10	No write permission
0x2405:0x04	UnlockStatus_U16	UINT	R	0	No error occurred
				1	Error while obtaining file information
				2	Error while reading
				3	Write error
0x2405:0x05	Busy_BOOL	BOOL	R	FALSE	Idle data transfer or lock
				TRUE	Busy data transfer or lock
0x2405:0x06	Reboot_DOM	USINT[16]	W	-	Restarts the SL by writing the MD5 hash code of the password for this object
0x2405:0x07	ProjectKey_U64	LREAL	W	-	Releases the application by writing the unlock key on this object
0x2405:0x08	AutoCnfKey_U64	LREAL	W	-	Releases the machine options by writing the unlock key on this object
0x2405:0x09	ProjectID_U32	UDINT	R	-	Project CRC of the SafeDESIGNER project
0x2405:0x0A	AutoCnfID_U32	UDINT	R	-	Value "Timestamp of file" - see "Format"

Table 31: SAF_FileParser_REC: File interface

Download structure

The download file must contain all of the files intended for transfer to the SL. A safe application will be placed in the project directory after being compiled in SafeDESIGNER. Ten files are placed in the folder "*AS_PROJECT_PATH*\Physical*NAME_AS_CONFIGURATION*\PLC1*NAME_SD_PROJECT*\DLFiles". These files must be added to the download structure.

All files intended for transfer to the SL must be created in Little Endian format.

File format:

Section	Description	Data type	Meaning
Header	Version	UINT	Version of the file container. The value 0x0100 must be entered
	Amount	UINT	Number of following files. An application consists of ten files. The value 0x000A must be entered
	UDID	USINT[6]	UDID of the SL to which the file will be transferred
File 1	File length	UDINT	File length of the 1st file
	File name	USINT[13]	Name of the 1st file ("BUR_PARA.SAF", see section "Application files")
	File contents	-	Data from the 1st file in the DLFiles folder ("dlfile01.sos", see section "Application files")
...			
File 10	File length	UDINT	File length of the 10th file
	File name	USINT[13]	Name of the 10th file
	File contents	-	Data from the 10th file
Checksum	CRC32	UDINT	Additive checksum of the entire download file[1)]

Table 32: Download file

1) CRC32 calculation, polynomial 0x1edc6f41, starting value 0

Application files

Predefined names must be assigned in the download file for the files "dlfile01.sos" to "dlfile10.sos", so that they can be identified by the SL. These names are listed in the following table.

File name in the project directory	Name in the download file
dlfile01.sos	BuR_Para.saf
dlfile02.sos	CFooLibs.dll
dlfile03.sos	impldiag.zip
dlfile04.sos	sdevpara.saf
dlfile05.sos	Bootfile.pro
dlfile06.sos	ProjCRC.img
dlfile07.sos	SwapList.pr2
dlfile08.sos	Bootfile.pr2
dlfile09.sos	BusNvCRC.img
dlfile10.sos	SysFlags.dat

Table 33: File names - Application download

Download sequence

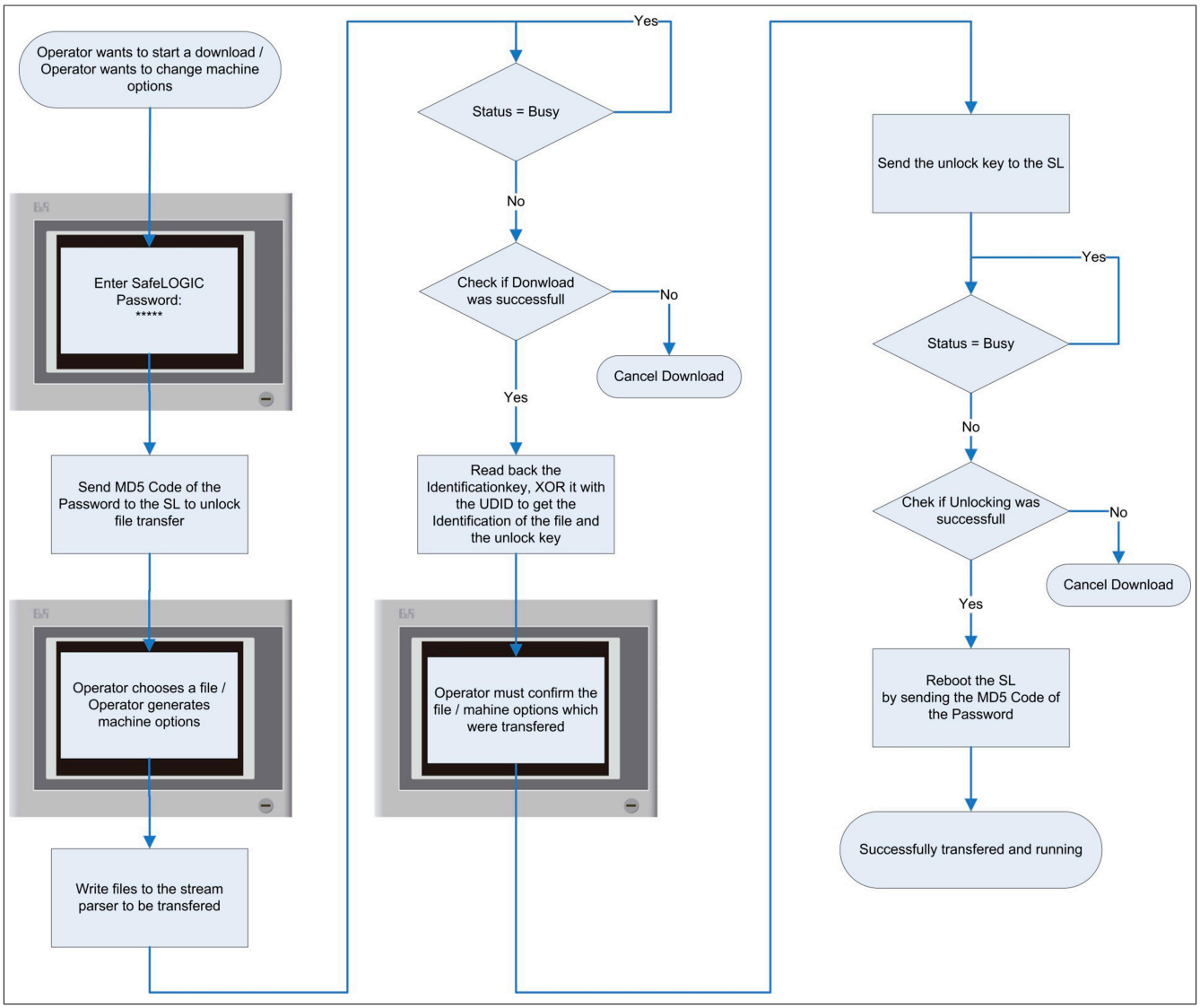


Image 11: Flow chart - File download

Information:

- **The operator must know the respective CRC before an application download. The return value, which contains the CRC, must show the operator which file has been transferred. The operator must then confirm that the correct file has been transferred.**
- **Machine option files must be identified by their timestamp after being downloaded. The operator must be shown all of the settings that were configured in this file. The operator must confirm that the settings are ok.**
- **The unlock key can only be transferred to the SL after the operator has confirmed the sent file.**

A specified procedure must be followed in order to transfer a safe application from the functional controller to the SL. The download procedure must be initiated manually and intentionally by the machine operator. After the data has been successfully transferred, their receipt on the SL must also be checked. The operator must be shown the checksum. The checksum must be confirmed by the operator. The SL must then be restarted in order to start the safe application that was transferred.

Proper downloading requires that the following steps be followed:

- Authorize download by writing the MD5 hash code of the SafeKEY password to the object "Authorization_DOM".
- Send download file by writing it to the object "FileStreamData_DOM".
- Determine whether or not the transfer is complete by reading the object "Busy_BOOL".
- Query whether or not the data has been fully received by reading the object "ParserStatus_U16".
- Read the identification/key object. This is done by reading the object "ProjectKey_U64". This object must be linked with the UDID of the SL (by byte). This provides the user with the identification of the transferred file and the corresponding unlock key.

The UDID must be linked with the identification/key object XOR based on the following schema.

FileIdent[0] = EPLKey[0];

FileIdent[1] = EPLKey[1];

FileIdent[2] = EPLKey[2] ^ SL_UDID[0];

FileIdent[3] = EPLKey[3] ^ SL_UDID[1];

UnlockKey[0] = EPLKey[4] ^ SL_UDID[2];

UnlockKey[1] = EPLKey[5] ^ SL_UDID[3];

UnlockKey[2] = EPLKey[6] ^ SL_UDID[4];

UnlockKey[4] = EPLKey[7] ^ SL_UDID[5];

Result of the link:

Byte	Meaning
0	Identification: In the case of an application download, the application's CRC shown like in SafeDESIGNER. In the case of a machine operation file, the value of the element "File time stamp". This value must show the operator which file has been transferred.
1	
2	
3	
4	Unlock key for the application / machine option file
5	
6	
7	

Table 34: Result of UDID identification/key object XOR link

- Writing the unlock key to the object "ProjectKey_U64". This requires writing the unlock key to the first 4 bytes.
- Determine whether or not decoding is complete by reading the object "Busy_BOOL".
- Query whether or not any errors occurred on the SL while decoding the data. This is done by reading the object "UnlockStatus_U16".
- Initiate SL reboot by writing the MD5 hash code of the SafeKEY password to the object "Reboot_DOM".

8.4 Extended status data

The following status data can be read via POWERLINK:

Index:Subindex	Object description	Data type	Access	Values	Description
0x2000:0x08	Project_CRC	UDINT	R	-	CRC of the SafeDESIGNER project
0x2000:0x09	Project_Time	DATE_AND_TIME	R	-	Timestamp
0x2000:0x0C	Project_Name	STRING (without zero termination)	R	-	Project name.
0x2000:0x0D	Project_Author	STRING (without zero termination)	R	-	Name of author
0x2000:0x0E	SafeOS_RUN_STATE	BOOL	R	0	SafeOS is not in RUN (identical to SafeOSstate=0x66)
				1	SafeOS is in RUN (identical to SafeOSstate==0x66)
0x2000:0x0F	BOOT_STATE	UDINT	R	General firmware boot status	
				0x00	Boot procedure not yet started
				0x01	Initialization started
				0x10	Cyclic hardware tests running
				0x11	openSAFETY stack running
				0x12	SafeOS running
0x2000:0x10	openSAFETYstate	UDINT	R	0	Preoperational state (all cyclic safe data is reset)
				1	Operational state
0x2000:0x11	SafeOSstate	UDINT	R	Status of the safety application, corresponds to the R/E LED on the SafeL-OGIC controller	
				0x00	Invalid (e.g. SafeKEY blank) Or boot not active (BOOT_STATE!=0x12)
				0x0F	ON (booting / internal initialization) or error (check logbook)
				0x33	Loading (booting / internal initialization)
				0x55	Stop [Safe]
				0x66	Run [Safe]
				0x99	Halt [Debug]
				0xAA	Stop [Debug]
				0xCC	Run [Debug]
				0xF0	No Execution

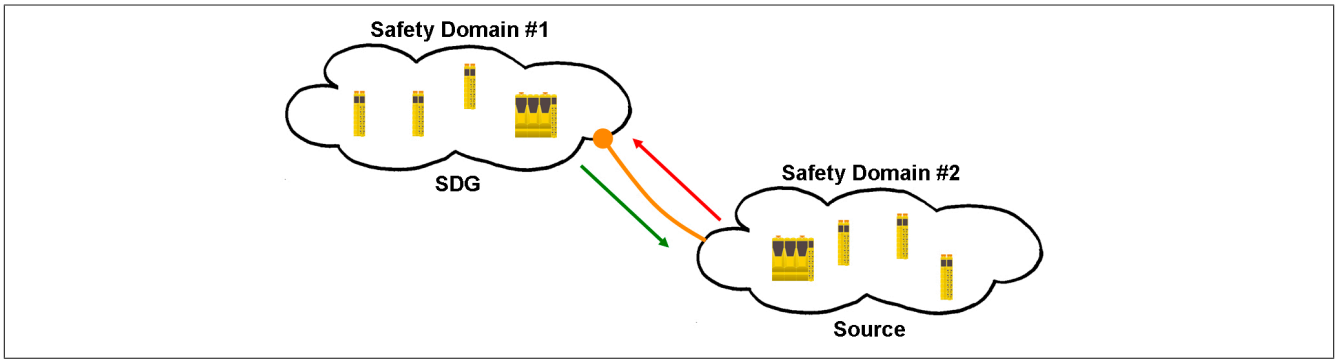
Table 35: System status data

Index:Subindex	Object description	Data type	Access	Values	Description
0x2001:0x05	openSAFETY_Instance	USINT	RW	-	Number of the openSAFETY instance which the statistic counter should read from.
				0	Safe I/O modules
				1-10	SDG connections to other SafeLOGIC controllers (see "Table 13: I/O configuration parameters: POWERLINK parameters" on page 10)
0x2001:0x06	Module_Index	USINT	RW	-	Index of the module whose statistics counter should be read.
				0-255	Safe I/O modules - these will be listed comprehensively starting at 0 (sorted by increasing SafeMODULE ID)
				0	For the SDG connection to another SafeLOGIC controller
0x2001:0x07	Statistics_Counter	UDINT	R	-	Statistic counter for the module defined with the sub-indices 05 and 06. The statistics counter is incremented each time the safe cyclic data connection is broken. **Notes** • The value is only available after the sub-indices 05 and 06 have been set • The value will be refreshed approximately every 30s

Table 36: Statistic counter for safe cyclic data connections

9 SafeLOGIC to SafeLOGIC communication

The safety system makes it possible to exchange safety-oriented information between two safety controllers (SafeLOGIC). This can be used for things like implementing a global E-stop across a machine network or when there is a dependency between the safety application 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.



Note:

The safety domain number is taken from the SafeLOGIC ID. In order to use this communication, SafeLOGIC IDs must be unique. This uniqueness should be taken into consideration from the very beginning.

To aid in this, a SafeLOGIC controller provides a Safety Domain Gateway (SDG) that can be used to connect additional SafeLOGIC controllers (Source). This gateway functionality ensures the communication between several different safety domains. The connection between Source SafeLOGIC controllers and SDG SafeLOGIC controller is displayed in the Source SafeLOGIC controller's project as an additional safety module with several communication channels. An SDG SL controller can by itself also be used as Source and connected to another SDG SL controller. This can be done to achieve a cascading of communication relationships.

A Source SL controller can also be connected several times to the same SDG SL controller. It is also 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.

Note:

A SDG SL controller is always a SafeLOGIC PLUS controller variant. Source SL controllers can be either standard SL controllers or PLUS controllers.

9.1 Possibilities

The system supports several different methods of communication. The type of communication to be used is specified using parameters in Automation Studio.

Fixed communication

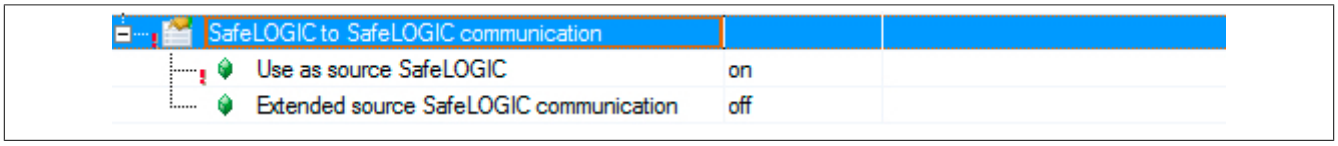
- 8 BOOL channels (1 byte) per communication direction
- One Source SL controller can only communicate with a SDG SL controller
- No constellation of any controller with any controller

Extended communication (Release 1.4 or higher and AS 3.0.90)

- 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 controller to any controller constellation possible

9.2 Configuration in Automation Studio

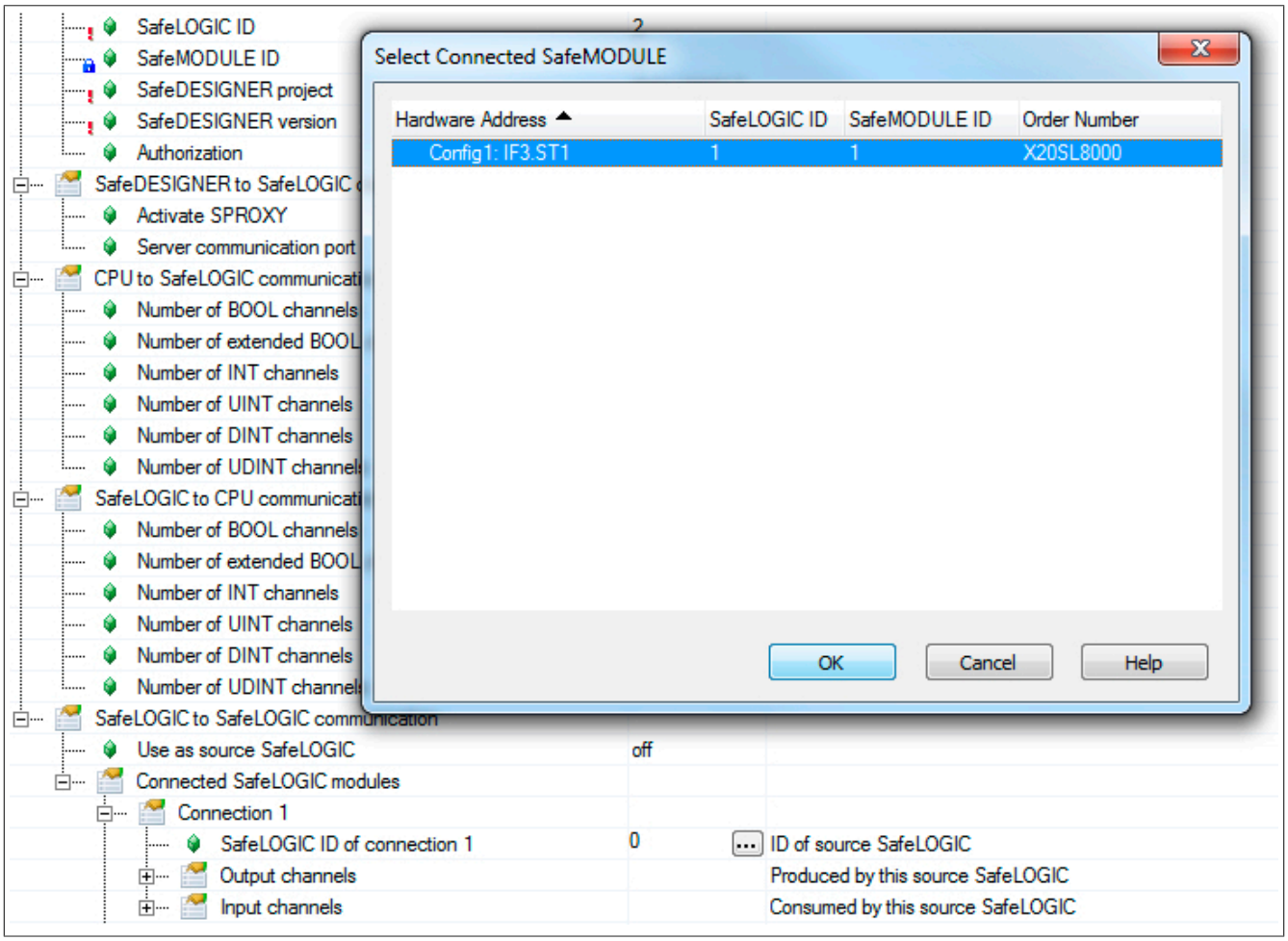
To use SafeLOGIC to SafeLOGIC communication, a SafeLOGIC controller first needs to be configured as a Source SL controller. This is done in the I/O configuration.



After the "Use as source SafeLOGIC" parameter has been selected, it's possible to define the type of SafeLOGIC to SafeLOGIC communication (fixed or extended). If the "Extended source SafeLOGIC communication" parameter is not enabled, then fixed communication is used.

> **Note:**
>
> **Changing the type of communication (fixed or extended) at a later time may result in channel overlap in SafeDESIGNER; the communication channels must therefore be reconnected.**

The Source SL controller is then connected to the SDG SL in the next step. This is done using the connection points in Automation Studio under the I/O configuration of a SafeLOGIC PLUS controller. Each SafeLOGIC ID (safety domain) is specified from the connection sections using the assistant.



The necessary communication channels must be defined under each connection. With fixed communication, they are limited to 8 BOOL channels for each direction.

Connected SafeLOGIC modules		
Connection 1		
SafeLOGIC ID of connection 1	1	ID of source SafeLOGIC
Output channels		Produced by this source SafeLOGIC
Number of BOOL channels	8	
Number of INT channels	0	
Number of UINT channels	0	
Number of DINT channels	0	
Number of UDINT channels	0	
Input channels		Consumed by this source SafeLOGIC
Number of BOOL channels	8	
Number of INT channels	0	
Number of UINT channels	0	
Number of DINT channels	0	
Number of UDINT channels	0	

9.3 Display in SafeDESIGNER

The communication channels are also shown in the SafeDESIGNER project for the respective SafeLOGIC controller (Source or SDG).

Caution!

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 doesn't agree, then the system issues a corresponding logger message in the Safety Logger and communication does not take place.

9.3.1 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 and 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 selected, it is possible to then configure the module's safety-related parameters (see section Connection parameters).

Fixed communication

Underneath the module are the input channels that are sent from the SDG SL controller to the Source SL controller as well as bit information regarding the status of the connection.

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					

Underneath the actual SL controller in the project are the output channels that are sent from the Source SL controller to the SDG SL controller in the "SafeLOGIC_SafeLOGIC" section.

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

Underneath the module are the input channels, the output channels and as bit information regarding the status of the connection.

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					

Safety View

Additional connection

An additional module underneath the same node is available with parameters and communication channels should the Source SL controller be connected an extra time to the same SDG SL controller.

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

An additional node for the safety domain as well as a module with parameters and communication channels is available if the Source SL controller should be connected to another SDG SL controller.

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

9.3.2 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 and 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

Note:

In the SDG SL controller's project, no connection parameters are available. These have to be configured in the Source SL controller's project.

Fixed communication

Underneath the module are the input channels, the output channels and as bit information regarding the status of the connection.

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

Underneath the module are the input channels, the output channels and as bit information regarding the status of the connection.

Safety View

SL2				SafeLOGIC ID 2
SL2.SM1.C1		IF3.ST2		X20SL8000
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				
SafeModuleOK				
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				

Additional connection

An additional module underneath the same node is available with the appropriate communication channels should the Source SL controller be connected an extra time to the SDG SL controller.

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
SL2.SM1.C2		IF3.ST2			X20SL8000

9.4 Connection parameters

Safety Release 1.4 or higher:

Cycle time parameters are also available for communication in order to define the worst case response 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).

> **Note:**
>
> **Since SafeLOGIC to SafeLOGIC communication is represented as an additional safety module, it is possible to configure the parameters for the connection 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 function expansions.	Basic release	-
Optional	The module can be optionally configured using this parameter. Optional modules do not have to be present, i.e. SafeLOGIC 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 absolutely necessary for the application. The module has to go into operational mode after start-up, and safe communication to the SafeLOGIC device must be properly established (SafeModuleOk= SAFETRUE). Processing of the safe application on the SafeLOGIC device is delayed after start-up until this state is achieved for all modules with "Optional = No". After start-up, module problems are indicated by a quickly blinking MXCHG LED on the SafeLOGIC device. An entry is also made in the logbook.
Yes	This module is not necessary for the application. The module is not taken into consideration during start-up, which means the safe 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 device. After start-up, module problems are NOT indicated by a quickly blinking MXCHG LED on the SafeLOGIC device. An entry is NOT made in the logbook.
Startup	This module is optional; the system determines how to proceed during start-up. If, during start-up, it's determined that the module is physically present (regardless of if it's in Operational mode or not), then the module behaves as if "Optional = No" is set. If, during start-up, it's determined that the module is not physically present, the module behaves as if "Optional = Yes" is set.

External_UDID	This parameter enables the option on the module of determining the expected UDID externally from the CPU.	No	-

Parameter value	Description
Yes-CAUTION	The UDID is determined by the CPU. SafeLOGIC must be restarted when the UDID is changed.
No	The UDID is determined by a teach-in procedure during startup.

Table 37: SafeDESIGNER parameters: Basic

Danger!

If the "External_UDID = Yes-CAUTION" function is used, incorrect specifications from the CPU can lead to safety-critical situations.
Perform an FMEA in order to detect and handle this situation properly using additional safety measures.

Group: Safety_Response_Time

Parameter	Description	Default value	Unit
Synchronous_Network_Only	This parameter determines the synchronization properties of the underlying network. **Parameter value** / **Description** Yes: In order to calculate the safe 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_SDG_Powerlink_CycleTime_us	This parameter specifies the maximum cycle time of the POWERLINK network where the other SafeLOGIC controller is being operated. • Permissible values: 200 - 30000 µs	5000	µs
Max_Powerlink_CycleTime_us	This parameter specifies the maximum POWERLINK cycle time used to calculate the safe response time. • Permissible values: 200 - 30000 µs	5000	µs
Max_CPU_CrossLinkTask_CycleTime_us	This parameter specifies the maximum cycle time for copying data between the two POWERLINK networks. A value of 0 means that both SafeLOGIC controllers are in the same POWERLINK network. • Permissible values: 0 - 3000000 µs	5000	µs
Min_SDG_Powerlink_CycleTime_us	This parameter specifies the maximum cycle time of the POWERLINK network where the other SafeLOGIC controller is being operated. • Permissible values: 200 - 30000 µs	200	µs
Min_Powerlink_CycleTime_us	This parameter specifies the minimum POWERLINK cycle time used to calculate the safe response time. • Permissible values: 200 - 30000 µs	200	µs
Min_CPU_CrossLinkTask_CycleTime_us	This parameter specifies the minimum cycle time for copying data between the two POWERLINK networks. A value of 0 means that both SafeLOGIC controllers are in the same POWERLINK network. • Permissible values: 0 - 3000000 µs	0	µs
Worst_Case_Response_Time_us	This parameter specifies the limit value for monitoring the safe response time. • Permissible values: 3000 - 12500000 µs Note: Keep the "Slow_Connection" parameter in mind when entering large values here!	100000	µs
Max_SDG_Cycle_Time_us	This parameter specifies the maximum cycle time of the other SafeLOGIC controller used to calculate the safe response time. • Permissible values: 800 - 20000	5000	µs
Min_SDG_Cycle_Time_us	This parameter specifies the minimum cycle time of the other SafeLOGIC controller used to calculate the safe response time. • Permissible values: 800 - 20000	1600	µs
Slow_Connection	This parameter specifies whether this connection is classified as a slow connection. **Parameter value** / **Description** Yes: This is a connection with a large ratio between the SafeLOGIC cycle time and the telegram runtime (affects the parameter calculation internally). General rule: "Yes" at ratios of 50:1 and higher No: Standard connection. Parameter calculation unchanged.	No	-

Table 38: SafeDESIGNER parameters: Safety_Response_Time

Note:

The CPU_CrossLinkTask_CycleTime_us parameter 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, then the minimal and maximum value should be set to 0.

The entire connection between the controllers must be taken into consideration for this parameter – including the time it takes to copy between interfaces.

Note:

The Slow_Connection parameter can also be used to specify that one of the connections 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.

10 Intended use

10.1 Qualified personnel

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

- Qualified personnel that are familiar with relevant safety concepts for automation technology and the applicable standards and regulations.
- Qualified personnel that 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.

10.2 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 use involving serious risks or hazards that could lead to death, injury or serious environmental damage. In particular, such risks and hazards include the use of these devices to monitor nuclear reactions in nuclear power plants, as well as flight control systems, flight safety, the control of mass transit systems, medical life support systems and the control of weapons systems.

When using safety-related control components, the safety precautions that apply 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 notices, connection descriptions (type plate and documentation) and limit values listed in the technical data are to be read carefully before installation and commissioning and must be observed.

10.3 Disclaimer

It is the user's responsibility to clear the use of B&R safety-related control components with the respective authorities.

B&R will not assume warranty or liability for damages that occur due to:

- Improper use
- Non-observance of standards and guidelines
- Unauthorized modifications to devices, connections and settings
- Operation of unauthorized or unsuitable devices or device groups
- Failure to follow the safety notices covered in this manual

10.4 Installation notes

Products must be protected against impermissible dirt and grime. Products are protected from dirt and grime up to Pollution Level II in the IEC 60664 standard.

Normally, IP54 provides protection up to Pollution Level II, but operation in condensing relative humidity is NOT allowed.

Danger!

Pollution levels stronger than specified by Pollution Level II in the IEC 60664 standard can result in hazardous failures. It is extremely important that you ensure a proper operating environment.

Danger!

In order to guarantee a specific supply voltage, a SELV power supply that conforms to IEC 60204 must be used for the bus, SafeIO and SafeLOGIC supplies.

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

As can be discerned from the following image, X20 potential groups must be protected using a fuse with a maximum of 10 A.

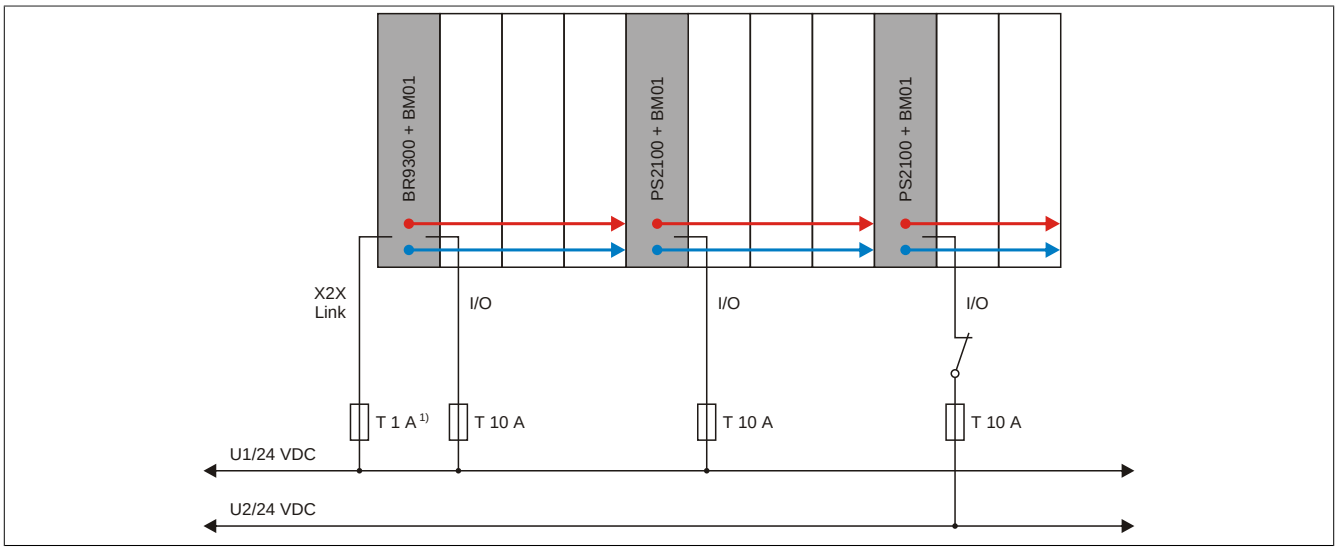


Image 12: Protecting various potential groups

1) Recommended for line protection.

10.5 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 switching off and cannot be modified.

> **Danger!**
>
> **For applications in which the safe state must actively turn on an actuator, additional external safety measures must be present (e.g. mechanical braking in the event of a hanging load).**

10.6 Mission time

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 removed from operation, i.e. replaced by new safety modules before their mission time expires.**

11 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>V1.51
V1.50
V1.42
V1.41
V1.40
V1.20
V1.10</td><td>
<table>
<tr><th>Version</th><th>From</th><th>To</th></tr>
<tr><td>Product set</td><td>Release 1.2</td><td>Release 1.5</td></tr>
<tr><td>SafeDESIGNER</td><td>2.70</td><td>2.99</td></tr>
<tr><td>Firmware</td><td>270</td><td>299</td></tr>
<tr><td>Upgrades</td><td>1.2.0.0</td><td>1.5.999.999</td></tr>
</table>
</td></tr>
<tr><td>V1.02
V1.01
V1.00</td><td>
<table>
<tr><th>Version</th><th>From</th><th>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 39: Release information

12 Manual history

Version	Date	Comment
1.51	March 2012	Section 6.1 Register description - Parameters in the I/O configuration - General group • "Authorization" parameter added Section 7.3 Maintenance scenarios - Confirmation of firmware change • Danger notice regarding permissible firmware versions added Section 8.1 POWERLINK data interface - Remote control • Various corrections and updates Sections 8.2 and 8.3 POWERLINK data interface - Machine options and application download • Interface - POWERLINK V2 objects updated to include Index:Subindex 0x2405:0x09 and 0x2405:0x0A
1.50	February 2012	Section 9 NEW - SafeLOGIC to SafeLOGIC communication
1.42	October 2011	Section 9.4 Intended use - Installation notes • Installation notes concerning approved grounding methods
1.41	February 2011	Section 8.1 POWERLINK data interface - Remote control - Remote control interface • "Index:Subindex" of the POWERLINK V2 object corrected Section 8.1 POWERLINK data interface - Remote control - Reading status • Descriptions of values 19-21 corrected
1.40	February 2011	First edition as a product-specific manual

Table 40: Manual history