X20(c)IF1041-1

1 General information

1.1 Other applicable documents

For additional and supplementary information, see the following documents.

Other applicable documents

Document name	Title
MAX20	X20 system user's manual
MAEMV	Installation / EMC guide

1.2 Coated modules

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

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

The coating has been certified according to the following standards:

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

1.3 Order data

Order number	Short description	Figure
	X20 interface module communication	
X20IF1041-1	X20 interface module, for DTM configuration, 1 CANopen master interface, electrically isolated, order 1x terminal block TB2105 separately!	
X20cIF1041-1	X20 interface module, coated, for DTM configuration, 1 CANopen master interface, electrically isolated, order 1x terminal block TB2105 separately!	
	Required accessories	
	Terminal blocks	
0TB2105.9010	Accessory terminal block, 5-pin, screw clamps 2.5 mm²	
0TB2105.9110	Accessory terminal block, 5-pin, push-in terminal block 2.5 mm²	

Table 1: X20IF1041-1, X20cIF1041-1 - Order data

1.4 Module description

The interface module is equipped with a CANopen interface. This allows third-party components to be integrated in the B&R system and makes it possible to quickly and easily transfer data in both directions.

Functions:

- CANopen master
- Error monitoring

CANopen

CANopen is a higher-layer protocol based on CAN. The standardized protocol offers very flexible configuration options.

Error monitoring

The status of the module and fieldbus is monitored. An error code is returned if an error occurs.

2 Technical description

2.1 Technical data

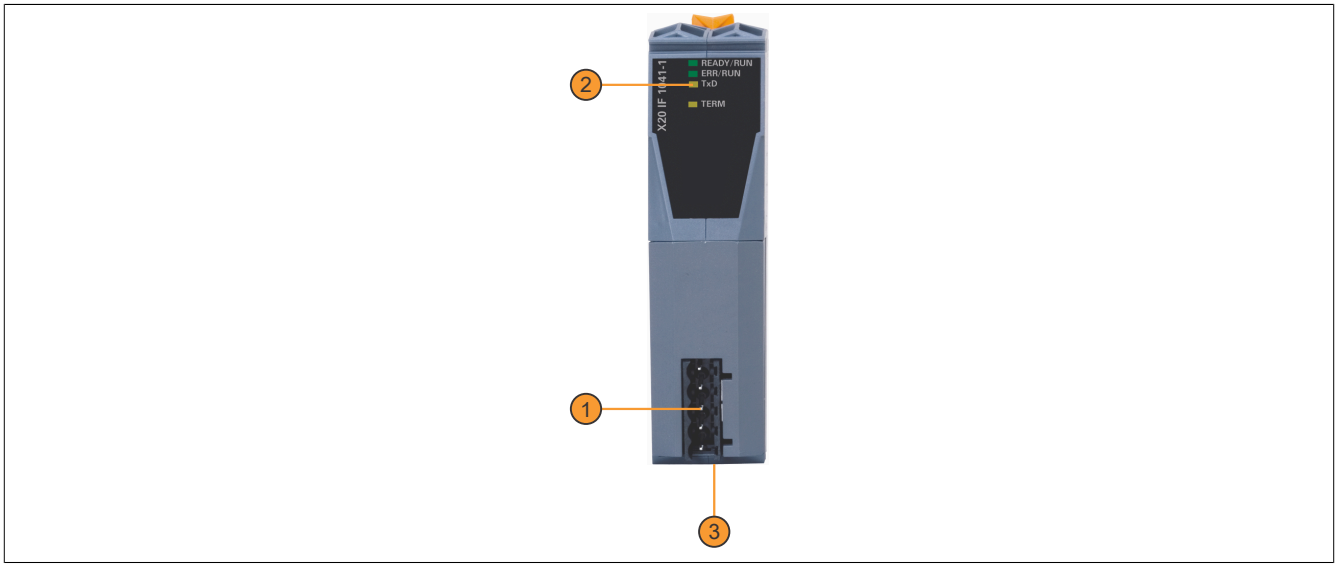
Order number	X20IF1041-1	X20cIF1041-1
Short description		
Communication module	CANopen master	
General information		
B&R ID code	0xA709	0xE505
Status indicators	Module status, network status, data transfer, terminating resistor	
Diagnostics		
Module status	Yes, using LED status indicator and software	
Network status	Yes, using LED status indicator and software	
Data transfer	Yes, using LED status indicator	
Terminating resistor	Yes, using LED status indicator	
Power consumption	1.1 W	
Additional power dissipation caused by actuators (resistive) [W]	-	
Certifications		
CE	Yes	
UKCA	Yes	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
DNV	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)	
LR	ENV1	
KR	Yes	
ABS	Yes	
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4.0 g EMC: Bridge and open deck	
EAC	Yes	
KC	Yes	-
Interfaces		
Interface IF1		
Fieldbus	CANopen master	
Variant	5-pin male multipoint connector	
Max. distance	1000 m	
Transfer rate	Max. 1 Mbit/s	
Terminating resistor	Integrated in module	
Controller	netX100	
Memory	8 MB SDRAM	
Electrical properties		
Electrical isolation	PLC isolated from CANopen (IF1)	

Table 2: X20IF1041-1, X20cIF1041-1 - Technical data

Order number	X20IF1041-1	X20cIF1041-1
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level		
0 to 2000 m	No limitation	
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	-25 to 60°C	
Vertical mounting orientation	-25 to 50°C	
Derating	-	
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, condensing
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x terminal block TB2105 separately.	
Slot	In the X20 PLC and expandable bus controller X20BC1083	In the X20c PLC and expandable bus controller X20cBC1083

Table 2: X20IF1041-1, X20cIF1041-1 - Technical data

2.2 Operating and connection elements



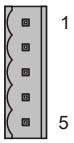
1	IF1 - CANopen	2	LED status indicators
3	Switch for terminating resistor on the bottom of the module	4	-

2.2.1 LED status indicators

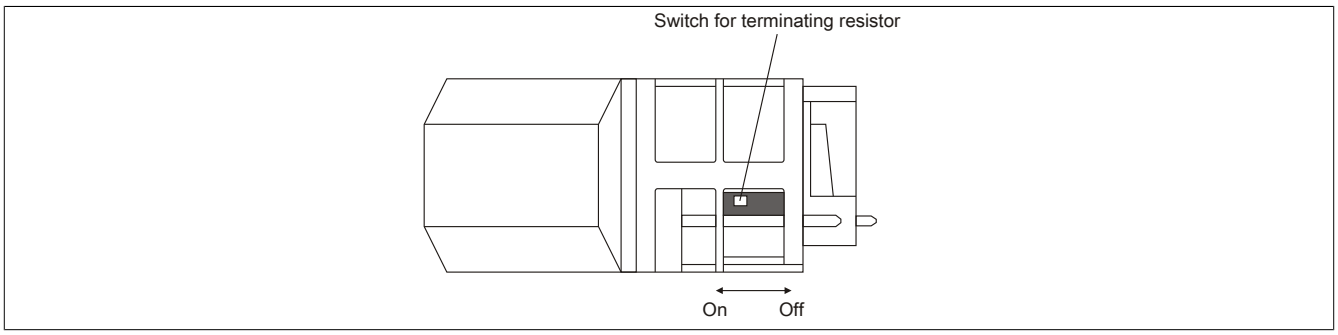
Figure	LED	Color	Status	Description
	READY/RUN	Green/red	Off	No power to module
		Green	On	PCI bus communication in progress
		Red	Blinking	Error when booting
			On	Communication on the PCI bus has not yet been started
	ERR/RUN	Green/red	Off	Module executes a reset
			Green on Red blinking with double pulse	CANopen communication has a disturbance. Possible causes: • The CAN bus cable is broken or the CAN bus controller is in "Bus off" mode • The module is in PREOPERATIONAL mode • At least one configured CANopen slave is not functioning
			Green on Red blinking	Communication was stopped (the module is in STOPPED mode)
		Green	Blinking	Communication is being started (module is being initialized)
			On	Communication is ready
	TxD	Yellow	Flickering or on	The module is transmitting data via the CANopen interface.
	TERM	Yellow	On	Terminating resistor integrated in the module switched on

2.2.2 CAN bus interface

The interface is a 5-pin multipoint connector. Terminal block 0TB2105 must be ordered separately.

Interface	Pinout		
 5-pin male multipoint connector	**Terminal**	**Function**	
	1	CAN⊥	CAN ground
	2	CAN_L	CAN low
	3	SHLD	Shield
	4	CAN_H	CAN high
	5	NC	

2.2.3 Terminating resistor



A terminating resistor is integrated in the interface module. It can be switched on or off with a switch on the bottom of the housing. A switched-on terminating resistor is indicated by LED "TERM".

3 Function description

3.1 CANopen / CAN

CAN (Controller Area Network) topology is based on a line structure and uses twisted wire pairs for data transfer. CANopen is a higher-layer protocol based on CAN. The standardized protocol offers very flexible configuration options.

CANopen operating modes such as synchronous, event and polling are supported as well as PDO linking, life/node guarding, heartbeat, emergency objects and much more.

For additional information, see "CANopen interface" on page 6.

3.2 Error codes

The module returns an error code if an error occurs. A complete list of all error codes in PDF format is available in under item "Communication_Error" in section "Communication / Fieldbus systems / Support with FDT/DTM / Diagnostic functions / Diagnostics on the runtime system / Master diagnostics" in Automation Help.

4 Commissioning

4.1 Firmware

The module comes with preinstalled firmware. The firmware is part of the Automation Studio project. The module is automatically brought up to this level.

A hardware upgrade must be performed to upgrade the firmware included in Automation Studio (see "Project management - Desktop - Upgrades" in Automation Help).

4.2 Operating the module

The interface module can be operated in the slot of a controller or in the slot of an expandable POWERLINK bus controller.

4.2.1 Use in the expandable X20BC1083 POWERLINK bus controller

4.2.1.1 Cyclic data

If this module is connected to the expandable POWERLINK bus controller, the amount of cyclic data is limited by the POWERLINK frame. This is 1488 bytes each in the input and output directions.
When using multiple X20IF10xx-1 interfaces or other X2X modules with a POWERLINK bus controller, the 1488 bytes are divided between all connected modules.

4.2.1.2 Operation

It is important to note the following in order to operate the module with the bus controller without problems:

- A minimum revision ≥ E0 is required for the bus controller.
- The module can only be operated with the POWERLINK V2 setting. V1 is not permitted.
- With SDO access to POWERLINK object 0x1011/1 on the bus controller, the firmware and configuration stored on the bus controller are not reset. They can only be overwritten by accessing them again. This affects objects 0x20C0 and 0x20C8, subindexes 92 to 95.

4.2.1.3 Timing characteristics

The internal data transfer results in an additional runtime shift of one cycle per direction.

> **Information:**
>
> **For additional information about runtime behavior, see section "Runtime shift" in X20BC1083.**

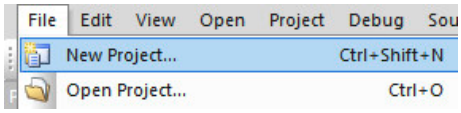
5 CANopen interface

5.1 Settings in Automation Studio

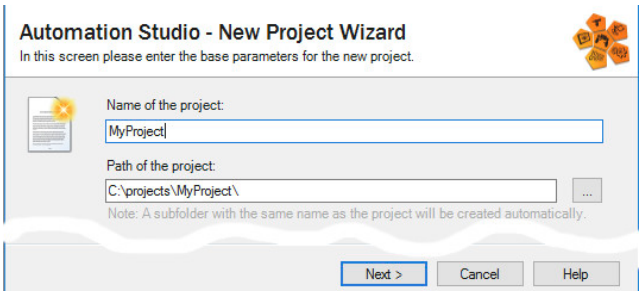
To configure the interface, a new Automation Studio project is created and the suitable settings are made on the module.

5.1.1 Creating an Automation Studio project

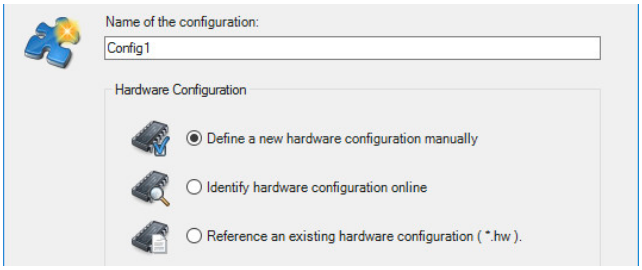
● Create a new Automation Studio project by selecting "New project".



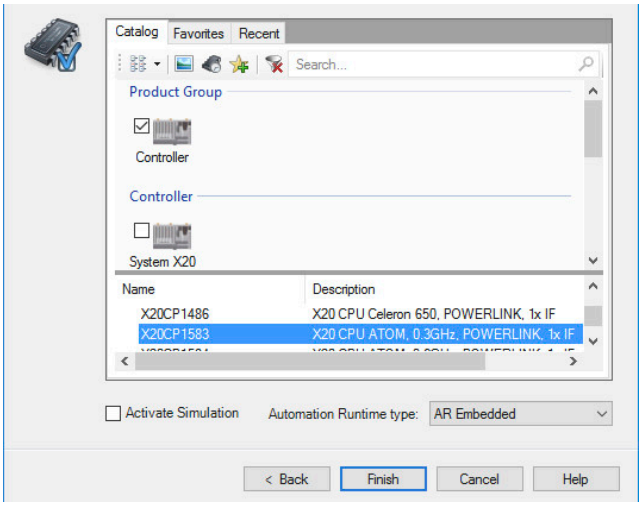
● Assign a project name and set up the project path.



● The type of hardware configuration is selected, and the name of the configuration is assigned.

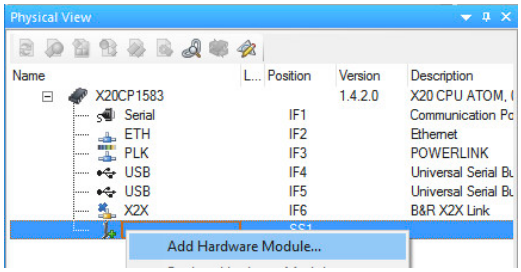


● If "Define a new hardware configuration manually" was selected, the hardware is selected in the next step. In order to simplify the search, different filters can be set for this in the Hardware Catalog. Finally, the Automation Studio project is created by selecting the required hardware is and clicking "Finish".

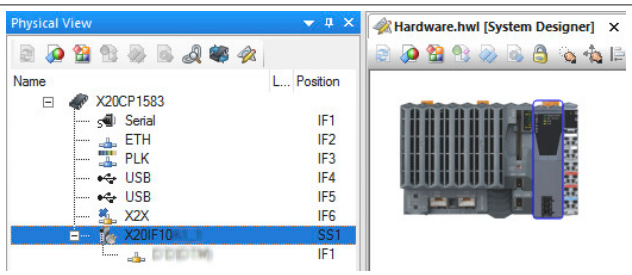


5.1.2 Adding and configuring the interface module

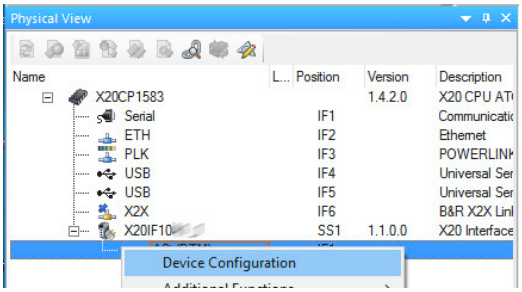
● In this example, the interface card is connected in the slot of a controller. Right-clicking on the slot and selecting "Add hardware module" opens the Hardware Catalog.



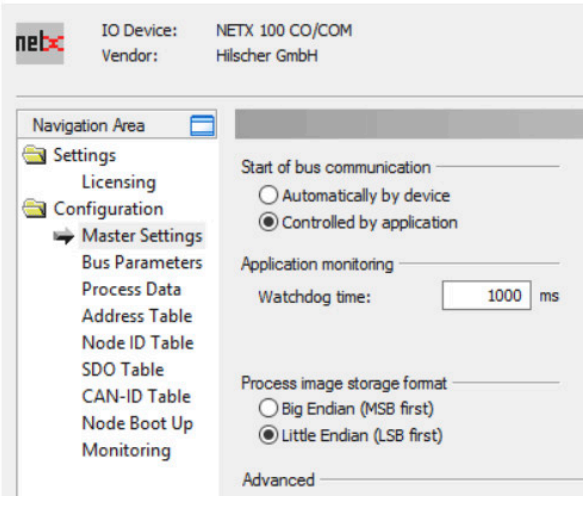
● The module is added to the project via drag-and-drop or by double-clicking on the interface card.



● Additional module settings can be made under "Device configuration". This configuration environment is opened by right-clicking on the IF interface and selecting "Device configuration".



● General settings are made in the device configuration.



5.1.2.1 Master settings

- Start of bus communication

It is possible to select how data exchange is started on the module.

Parameter	Explanation
Automatically by device	Data exchange is started automatically after initializing this module.
Controlled by application	Data exchange is started by Automation Runtime.

- Module alignment

The addressing mode is defined by the process image here. The addresses (offsets) of the process data are always interpreted as byte addresses.

Addressing mode	Explanation
Byte boundaries	The module address can start on any offset.
2 byte boundaries	The module address can only start on even byte offsets.

Information:

This configuration is automatically managed by Automation Runtime and is not permitted to be changed (default setting).

- Application monitoring

The module-internal watchdog time can be set here. If the watchdog has been enabled (watchdog time not equal to 0), the hardware watchdog must be reset after the set time at the latest.

Parameter	Explanation	Values
Watchdog time	Watchdog software disabled	0 ms
	Permissible range of values Default value: 1000 ms	20 to 65535 ms

Information:

The watchdog time is reset automatically by Automation Runtime.

- Process data handshake

This parameter configures the handshake for the data exchange between application and device.
Only "Buffered, host-controlled" is supported here.

- Process image storage format

This is used to define how data is stored in the process image (I/O mapping). The storage format is only applied to data type "Word". This change has no effect on other data types.

Storage format	Explanation
Big-endian	MSB/LSB = Higher/Lower byte (Motorola format)
Little-endian	MSB/LSB = Higher/Lower byte (Intel format)

Information:

This configuration is automatically managed by Automation Runtime and is not permitted to be changed (default setting).

- Advanced

This parameter is not supported.

- Device status offset

Here, the status offset can be set to be calculated automatically or using a predefined value.

Status offset	Explanation
Automatic calculation	The device status always directly follows the input bytes. If input data is added in the configuration, the start address of the device status is moved back in the dual-ported memory.
Static	Here, the distance (free buffer) between the last input byte and the start of the device status can be set. This way, the start address of the device status never changes in the dual-port memory. If additional input data is added, the distance (free buffer) is reduced. If more data is added than free buffer exists, the start address of the device status in the dual-port memory must be shifted. **If the offset is too small, an error is output. To correct the error, the free buffer must be increased to a sufficient size.** 

Information:

This configuration is automatically managed by Automation Runtime and is not permitted to be changed (default setting).

5.1.2.2 Bus parameters

- Device description

The symbolic name of the device can be changed here. However, this is only used by the configuration dialog boxes and not by Automation Studio.

- Node settings

The NodeID, baud rate and behavior at startup and in the event of error can be configured here.

Parameter	Explanation	Values
Node ID	The node ID is used in CANopen for addressing and each ID is only permitted to occur once in a network.	1 to 127
Baud rate	Configures the data transfer rate	10 kbit/s to 1 Mbit/s
Stop in case of monitoring error	How the master behaves in the event of a monitoring error is set here. In each case, the corresponding error code is output. • **Enabled** The master switches to mode Stop, and communication with all other slaves is aborted. • **Not enabled** The master remains in mode Operational, and communication with the other slaves is maintained.	
Send "Global start node"	If enabled, the master transmits a "Global start node" after the startup of all configured slaves. This synchronizes and starts all slaves. If a slave should not be started, both this parameter and "Send the start node command" must be disabled under "Node bootup" on page 12. If only one of the two parameters is disabled, the slave is started.	

- SYNC master settings

The COB ID can be changed here. Each communication object in the network has a unique communication object identifier (COB ID).

In addition, the cyclic period of the SYNC message can be set or switched off (cycle period = 0).

Sync object	Explanation	Values
COB ID	COB ID of the SYNC message Default value: 128	0 to 128 1664 to 1759 1761 to 1792
Cycle period	Period of the SYNC message. The value 0 disables the transmission of messages. Default value: 100	0 to 65535

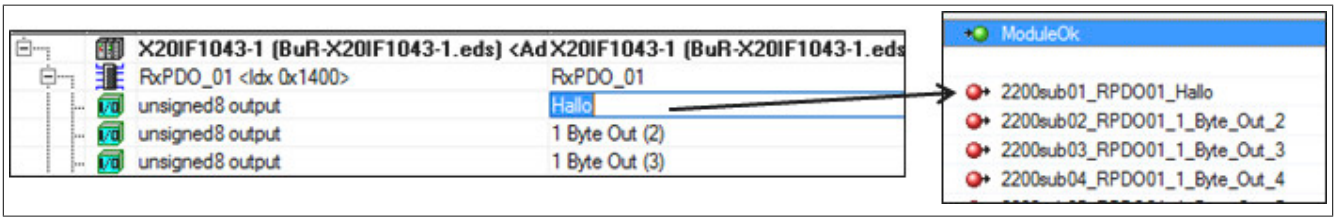
- 29-bit COB ID

This parameter is not supported.

5.1.2.3 Process data

This table lists the process data of the individual slaves.

Parameter	Explanation
Type	Device designation specified by the hardware. Further description of modules configured on the device or the input or output signals.
Tag	The name of the input and output data can be changed in column "Day".
SCADA	This parameter is not supported.



5.1.2.4 Address table

This table provides information about the addresses of the input and output data (in decimal or hexadecimal notation).

"Display mode" allows toggling between decimal and hexadecimal display.

Column	Explanation
Node ID	Node ID of the slave
Device	Device name of the slave
Name	Name of the slave
Obj.Idx	Object index
Obj.Name	Object name
COB ID	COB ID of the CAN message
Type	Data type for the input and output data
Length	Length of the input and output data
Address	Offset for the input and output data

The address table can also be exported as a CSV file.

5.1.2.5 Node ID table

All slaves are listed in this table.

Parameter	Explanation
Activate	This allows the slaves to be enabled or disabled. If a slave is disabled, the master reserves memory in the process data image for the slave, but no data is exchanged. If a slave is enabled, the process memory is reserved and data is exchanged.
Node ID	A node ID can be assigned to the slave here.
Device	Device name of the slave
Name	Name of the slave
Vendor	Device manufacturers

5.1.2.6 SDO table

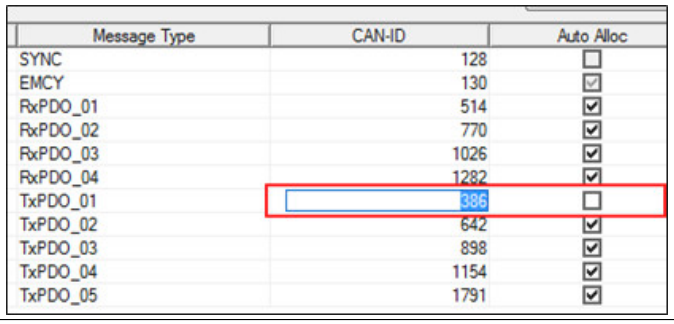
This table lists all objects that are transferred during the startup phase. The object information cannot be changed.

"Display mode" allows toggling between decimal and hexadecimal display.

Column	Explanation
Node ID	Node ID of the slave
Device	Device name of the slave
Name	Name of the slave
Obj.Idx	Object index
Sub.Idx	Subindex
Parameter	Parameter name
Value	Value of the parameter

5.1.2.7 CAN ID table

The CAN IDs used are listed in this table. If "Auto alloc = Enabled", the CAN IDs are assigned automatically. "Display mode" allows toggling between decimal and hexadecimal display.

Column	Explanation	Values
Node ID	Station address of the device in the network	1 to 127
Device	Device name of the slave	
Name	Name of the slave	Any
Message type	Message type	• NODE GUARDING • EMCY • RXPD0[x] • TXPD0[x] • SYNC
CAN ID	COB ID	0 to 2047
Auto alloc	Enables/Disables automatic allocation To change a CAN ID, "Auto alloc" must be disabled. 	

5.1.2.8 Node bootup

The startup phase can be adjusted here. By disabling boot phases, these are not executed, e.g. to be able to transfer another configuration at a later time.

Phase/Status	Parameter	Description
1 - Node reset	Send the reset node command	If enabled, the master first transmits the CANopen-specific node reset command.
2 - Check node, profile and type	Compare the configured profile and type object 1000H with real value	If enabled, the master compares the contents of object 0x1000 on the module with the set data. If the values do not match, the node is not accessed. The configuration is not transferred and a configuration error is reported. This option must be enabled if a node object is not defined in the EDS file. In this case, under "Profiles and type", the values for the profile and the device type of the node must be entered according to the specifications of the device manufacturer. "EDS default" resets the values for the profile and the device type of the node to the original values from the EDS file and disables the comparison.
3 - Configuration, guarding protocol	Configure the guard time and lifetime factor	If enabled, the master writes the two objects 0x100C (guard time) and 0x100D (lifetime factor) into the corresponding node objects during startup. If not enabled, the last configured values are obtained from the slave. If the configuration on the slave has been deleted, these values are 0.
4 - Configuration SYNC COB ID	Configure the COB ID for the synchronization message	If enabled, the master transfers the configuration of the SYNC COB ID set in Automation Studio under "SYNC master settings" on page 9 to the slave (object 0x1005). However, if the set cycle period is 0, no SYNC messages are sent. If not enabled, the last configured value is obtained from the slave. If the configuration on the slave has been deleted, the default value is 0x80.
5 - Configuration EMCY COB ID	Configure the COB ID for the emergency message	If enabled, the master transfers the fixed EMCY COB ID to the slave (object 0x1014). If not enabled, the last configured value is obtained from the slave. If the configuration on the slave has been deleted, the default value is 0x80 + NodeID.
6 - Configuration, download of objects	Download the object configuration to the node	If enabled, the master transfers all relevant configuration objects to the node, e.g. information about the PDO mapping, the COB IDs of the transmit and receive PDOs and all configured objects from the node configuration. If not enabled, the slave receives no configuration from the master. Phases 3, 4 and 5 are also not executed. If the slave supports automatic configuration, only the first 4 PDOs are active. COB IDs are defined and obtained from the slave.
7 - Start node	Send the start node command	If enabled, the master sends the CANopen-specific start node command at the end of the bootup procedure in order to achieve the operating state.
8 - Initiate PDO data	Remote request all TxPDOs and send current RxPDOs once after bootup	If enabled, the master reads and writes the configured PDOs after startup. This transmits all current data from the process output data memory to the nodes in addition to reading all current data from the nodes and then storing it in the process input data area of the master.

Information:

Parameters "Node boot up - Start node" and "Bus parameters - Send 'global start node'" must be considered together.

Example

If all boot phases have been disabled but "Send 'global start node'" is active, the slave is still started. In this case, communication is started without the configuration being applied.

5.1.2.9 Monitoring

Device monitoring can be configured here. The following types of monitoring are possible:

- The master monitors the individual nodes.
- A node monitors the master.
- One node monitors another node.

The node to be monitored can be selected in column "Active" of the table. For each node, the node guarding protocol or heartbeat protocol can be selected.

- Selected node

Contains all configured nodes.

- Node guarding protocol

The master transmits a cyclic poll request to the node to check whether the node still exists on the bus. The node transmits its current status back to the master as response. The node can use the poll request from the master in order to monitor the master.

Parameter	Explanation
Guarding time	Monitoring the slave from the point of view of the master If communication is running, the master queries the node at the set time interval to check whether the node is still present in the network. If the guarding time has the value 0, the monitoring is disabled on both the master and slave.
Lifetime factor	Using the slave to monitor the master. If the communication is running, the node monitors the master in the calculated time interval *"Guarding time * Lifetime factor"* to check whether the node is still present in the network. If the lifetime factor has the value 0, monitoring is disabled on the slave.

Information:

In order to use the node guarding protocol, the node must support this protocol.

- Heartbeat protocol

A "heartbeat producer" transmits cyclic heartbeat requests. One or multiple "heartbeat consumers" can receive the request.

Parameter	Explanation
Producer time	Time interval at which heartbeat requests are transmitted.

Information:

To use the heartbeat protocol, the node must support the heartbeat protocol.

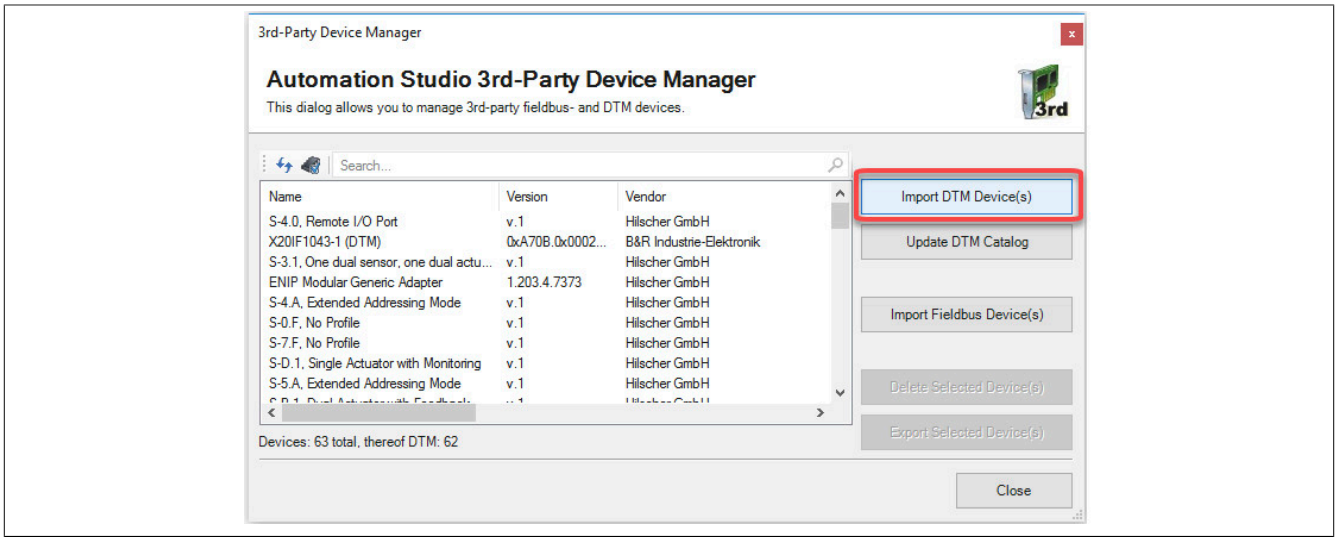
5.2 Adding the EDS file in Automation Studio

A device description file (EDS, DCF file) is required to inform the CANopen master which slaves were connected and how they were configured.

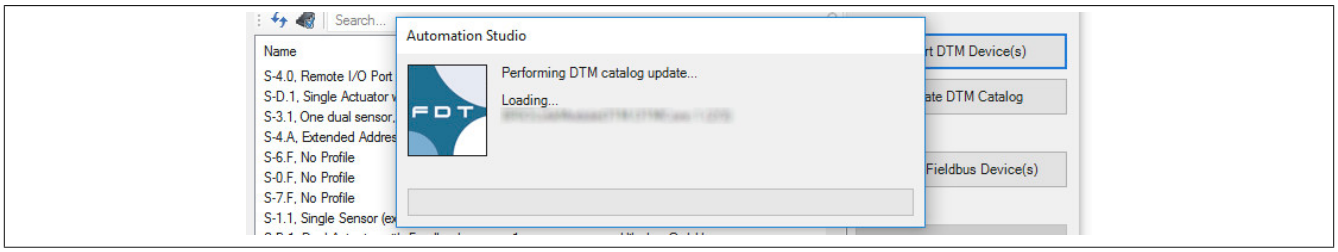
To add and use a device description file in Automation Studio, perform the following steps:

● The device description file (EDS, DCF) must be provided by the manufacturer of the CANopen slave.

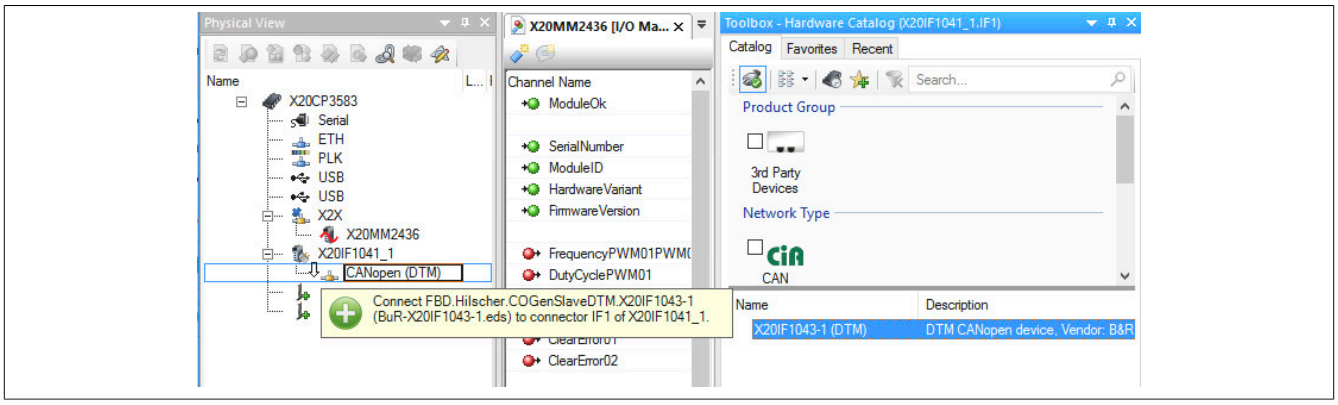
● Open the dialog box in Automation Studio under "Tools - Manage 3rd-party devices" and select "Import DTM Device(s)".



● Select the EDS file to be imported and confirm with OK. The EDS file is imported into Automation Studio.



● Click on "CANopen (DTM)" on the CANopen master X20IF1041-1, drag the EDS file from the Hardware Catalog and attach it to the CANopen master.



● Right-click on the device description file and select "Device configuration" to open the configuration environment for the EDS file.