X20IF1043-1

Data sheet
2.11 (March 2025)

Disclaimer

Version history

B&R makes every effort to keep documents as current as possible. The most current versions are available for download on the B&R website (www.br-automation.com).

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

1.2 Order data

Order number	Short description	Figure
	X20 interface module communication	
X20IF1043-1	X20 interface module, for DTM configuration, 1 CANopen slave interface, electrically isolated, order 1x terminal block TB2105 separately!	
	Required accessories	
	Terminal blocks	
0TB2105.9010	Accessory terminal block, 5-pin, screw clamp terminal block 2.5 mm²	
0TB2105.9110	Accessory terminal block, 5-pin, push-in terminal block 2.5 mm²	

Table 1: X20IF1043-1 - Order data

1.3 Module description

The interface module is equipped with a CANopen (slave) interface. This allows the B&R system (I/O modules, POWERLINK, etc.) to be connected to systems from other manufacturers and makes it possible to quickly and easily transfer data in both directions.

Functions:

- CANopen slave
- 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	X20IF1043-1
Short description	
Communication module	CANopen slave
General information	
B&R ID code	0xA70B
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 to 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)
CCS	Yes
LR	ENV1
KR	Yes
ABS	Yes
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4 g EMC: Bridge and open deck
KC	Yes
Interfaces	
Interface IF1	
Fieldbus	CANopen slave
Variant	5-pin male multipoint connector
Max. distance	1000 m
Transfer rate	Max. 1 Mbit/s
Terminating resistor	Integrated in module
Controller	netX100
Electrical properties	
Electrical isolation	PLC isolated from CANopen (IF1)
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

Table 2: X20IF1043-1 - Technical data

Order number	X20IF1043-1
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x terminal block TB2105 separately.
Slot	In the X20 PLC and expandable bus controller X20BC1083

Table 2: X20IF1043-1 - Technical data

2.2 Operating and connection elements
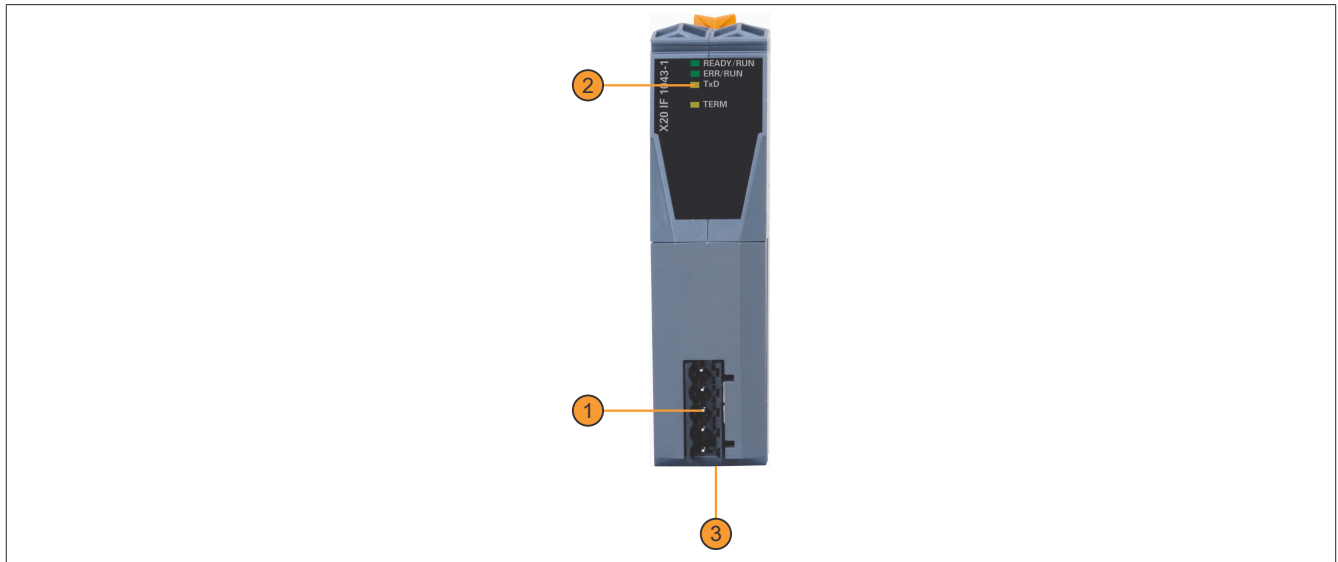
1	IF1 - CANopen	2	LED status indicators
3	Switch for terminating resistor on the bottom of the module	4	-

2.2.1 LED status indicators

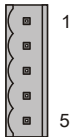| Figure	LED	Color	Status	Description
	READY/RUN	Green/red	Off	No power to module
		Green	On	PCI bus communication in progress
		Red	Blinking	Error when booting
			On	Communication on the PCI bus has not yet been started
	ERR/RUN	Green/red	Off	Module executes a reset
			Blinking green Red double flash	The module is in mode "PREOPERATIONAL". CANopen communication has a disturbance. Possible causes: • The CAN bus cable has been disconnected or the CAN bus controller is in "Bus off" mode. • Heartbeat or monitoring event occurred
			Blinking green Red single flash	The module is in mode "PREOPERATIONAL". The CAN error counter has reached or exceeded the warning limit; e.g. too many error frames.
			Green single flash Red double flash	The module is in mode "STOPPED". CANopen communication was stopped. Possible causes: • Heartbeat or monitoring event occurred
			Blinking green	The module is in mode "PREOPERATIONAL".
			Green on	The module is in mode "OPERATIONAL".
	TxD	Yellow	Flickering or on	Module sending data via the CANopen interface
	TERM	Yellow	On	Terminating resistor integrated in the module switched on

LED blinking diagram
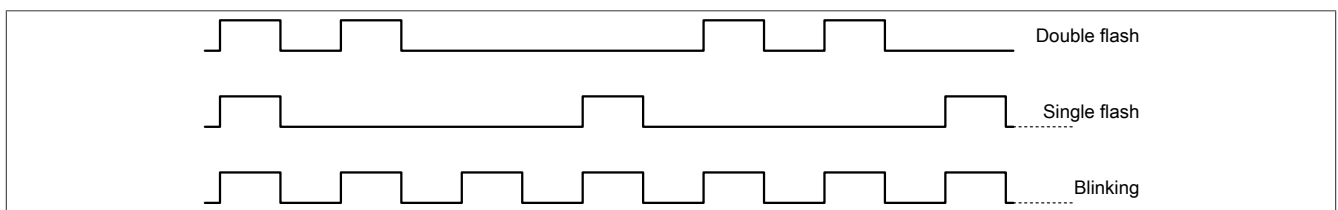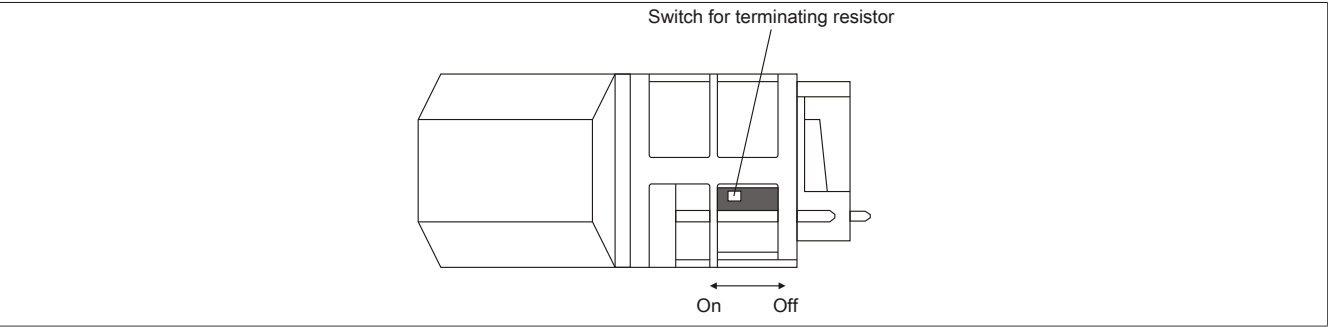
2.2.2 CAN bus interface

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

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

2.2.3 Terminating resistor

Switch for terminating resistor

On   Off

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 .

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 Help "Project management - Workspace - 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 The CANopen interface

> **Information:**
>
> **The settings on the slave must match the settings of the corresponding device description file; otherwise, no connection can be established.**

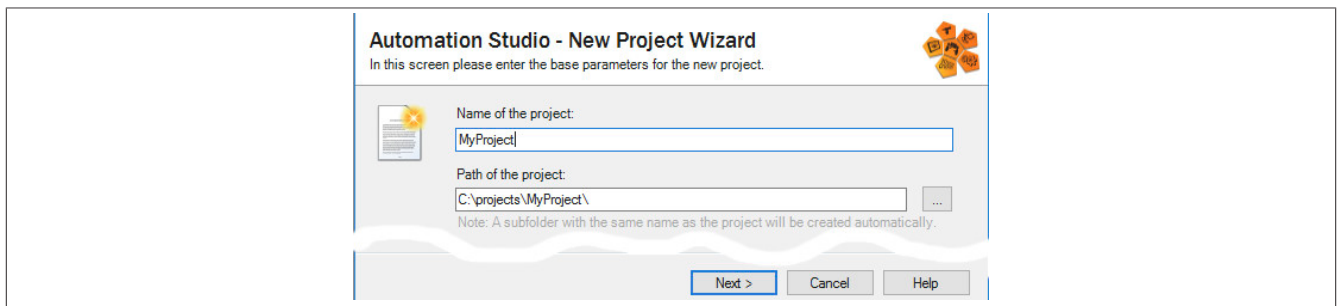## 5.1 Settings in Automation Studio

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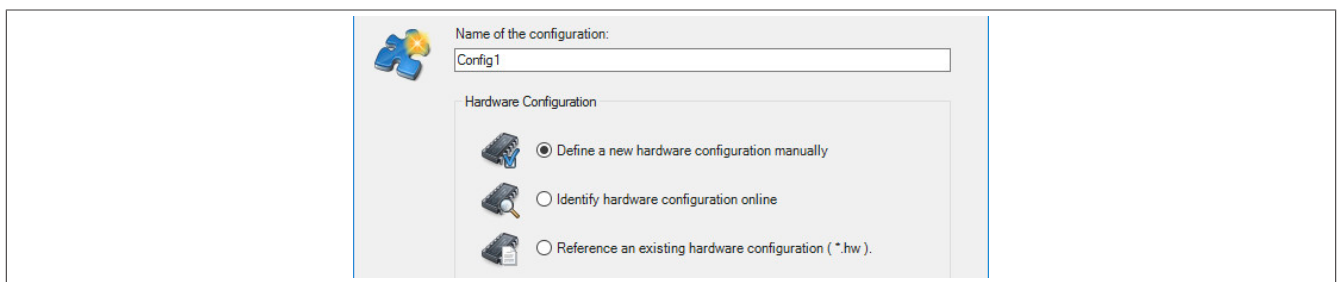
5.1.1 Creating an Automation Studio project

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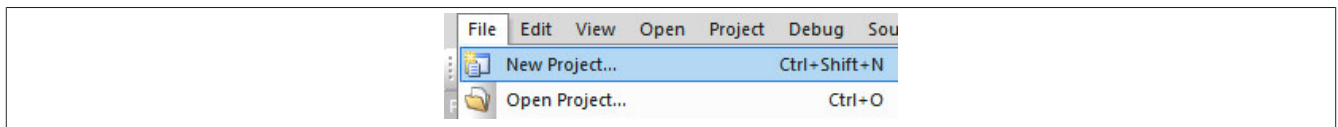

• Assign a project name and set up the project path.

• 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".
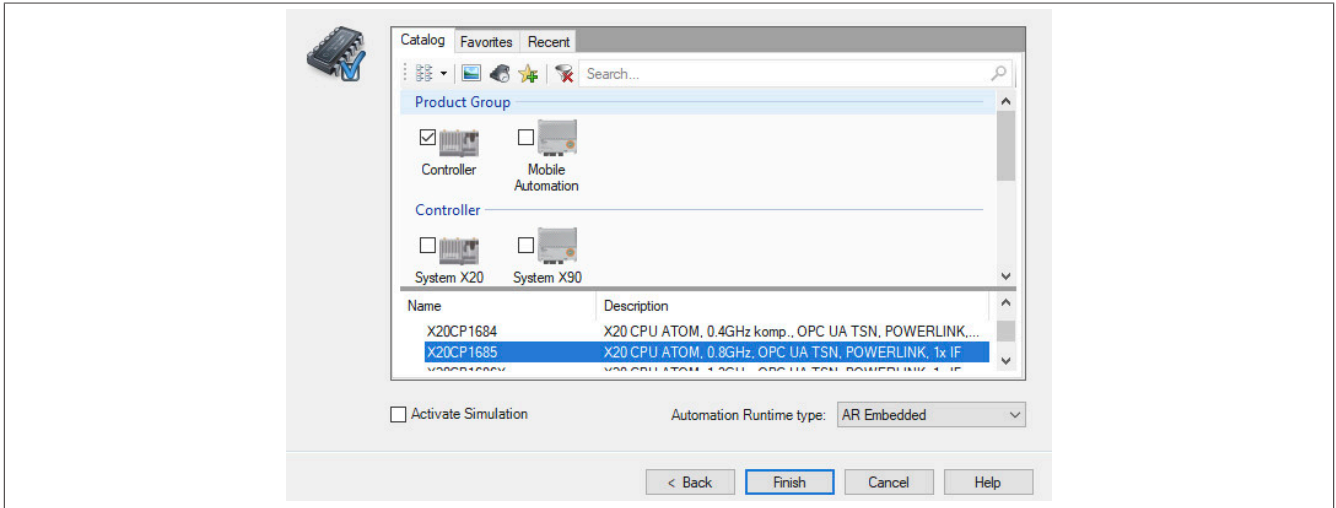
5.1.2 Adding and configuring the interface module

• 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.
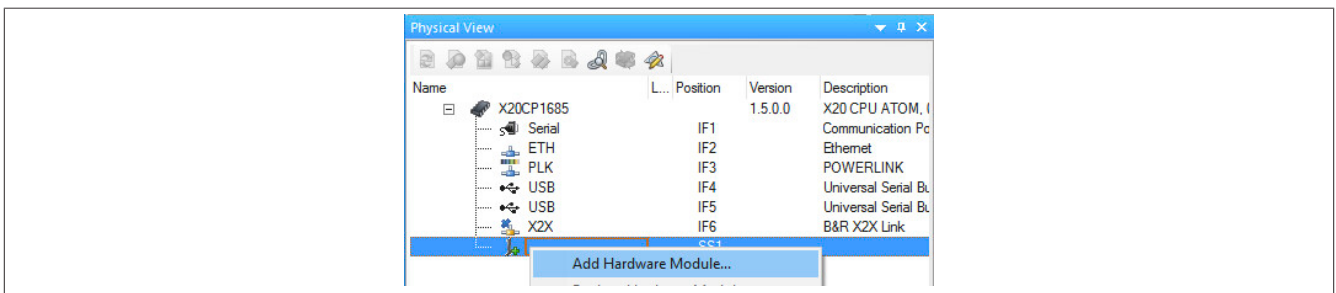
• The module is added to the project via drag-and-drop or by double-clicking on the interface card.
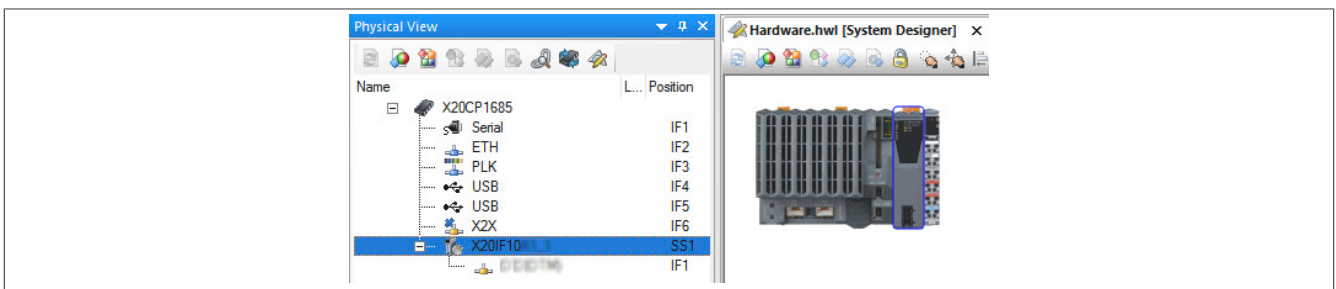
• Additional module settings can be made under "Device configuration". This configuration environment is opened by right-clicking on the IF interface and selecting "Device configuration".
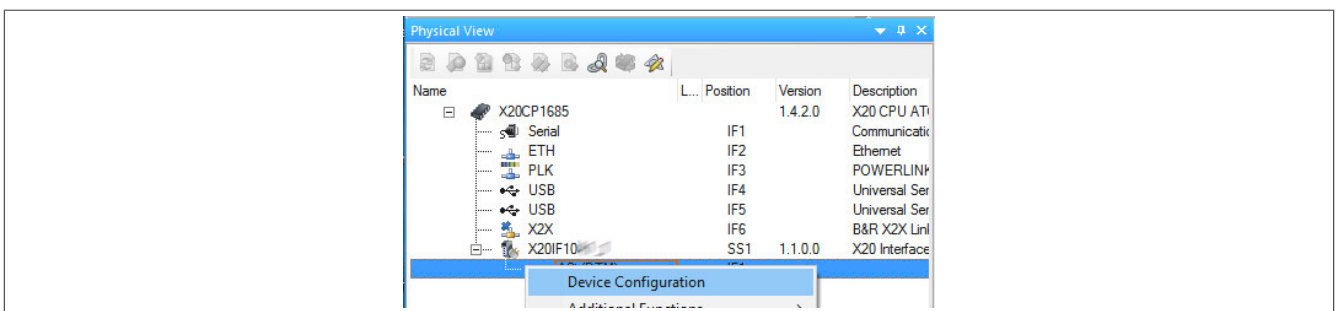
- General settings are made in the device configuration.

5.1.2.1 General

- Bus parameters

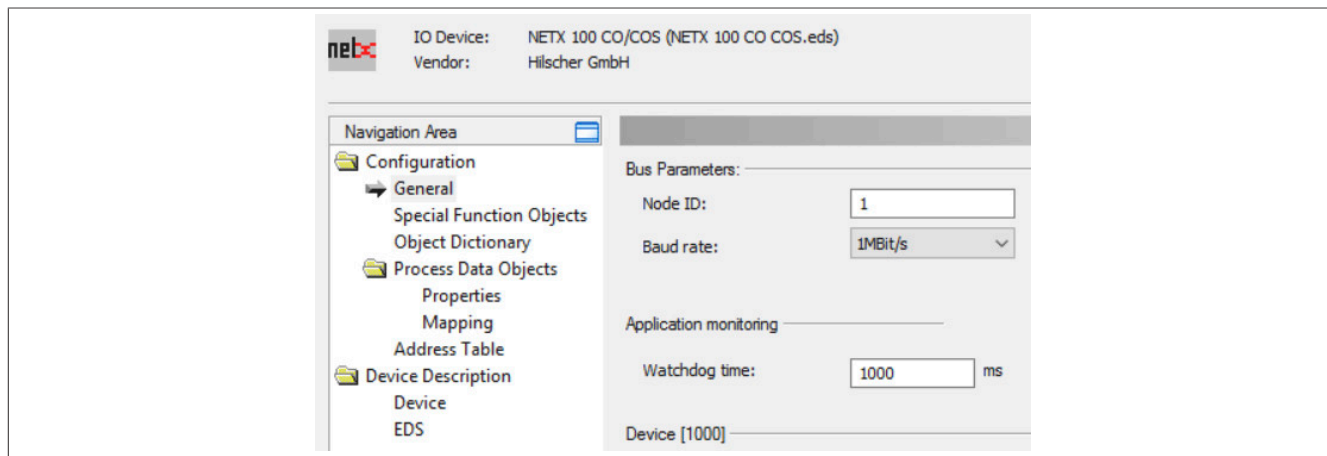The NodeID and baud rate can be configured here.

These values from the EDS description file on the master can only be read.

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. ⓘ The same node ID must be set in the EDS description file on the master and slave.	1 to 127
Baud rate	Sets the transfer rate	10 kbit/s to 1 Mbit/s
Enable address switch	This parameter is not supported.	

- 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	Software watchdog 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.
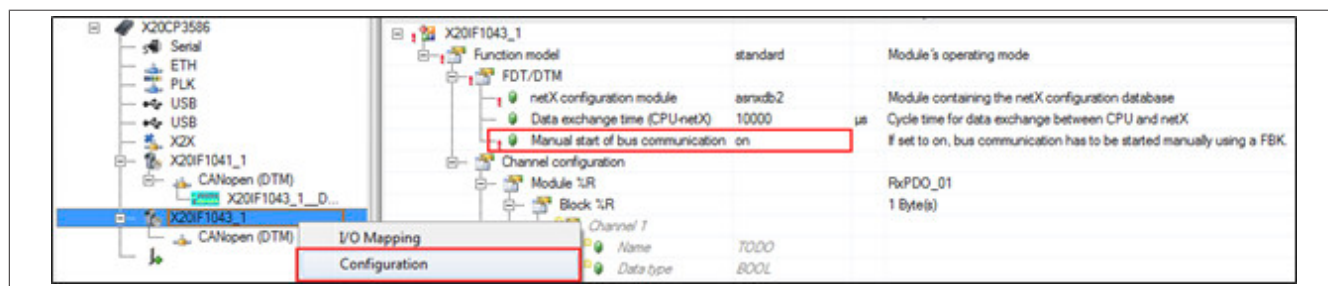
- Start of bus communication

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

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

> **Information:**
>
> **Parameter "Manual start of bus communication" can be enabled under the I/O configuration of the CANopen slave.**

The following settings must be made in order to avoid automatic data exchange:

- In the IF module configuration, "Manual start of bus communication" must be set to "On".
- "Start of bus communication" must be set to "Controlled by application".

With this setting, communication can only be started via function block **AsNxCoS - nxcosStartBusComm()**.

- Device

This data is read from the EDS file (DTM) and not configurable.

- Vendor

This data is read from the EDS file (DTM) and not configurable.

5.1.2.2 Special function objects

The receipt of EMCY messages and transmission of SYNC messages can be set here. However, these settings can only be configured in the device description file on the master.

- Synchronization message

Whether the slave generates the SYNC messages can be set here.

> **Information:**
>
> **On the slave, this setting causes an error message and is therefore not permitted to be used.**

Parameter	Explanation
Device generates SYNC message	Enabling generation of the SYNC message

No other parameters are supported.

- TimeStamp message

This parameter is not supported.

- Emergency message

Whether the master can receive EMCY messages can be set here.

Parameter	Explanation
EMCY exists	Master can receive EMCY messages

No other parameters are supported.

5.1.2.3 Object dictionary

The object dictionary of the device is listed here. These are taken from the EDS file (DTM). PDO objects can be enabled or disabled in the object dictionary. This is done under "Process data objects - Properties" on page 13.

The object dictionary can be filtered with "Area" and "Status"; using "Object", a single object can be searched.

Object dictionary

Parameter	Explanation	Values
Configure	Overview of the configuration options. • Key symbol: Entry cannot be configured in the object dictionary. • Checkbox with check mark: Object can be configured. • Checkbox without check mark: Object is locked for the configuration.	
Index.Subindex	Object address consisting of index and subindex	
Name	Symbolic name of the device	
Access	Access rights of the device	• RO - Read only • RW - Read and write • WO - Write only • CONST - Constant value

When a specific object is selected in the object dictionary, the name, current value, default value, data type and minimum/maximum values of the object are listed. This listing can be in decimal and hexadecimal notation.

5.1.2.4 Process data objects - Properties

Additional PDOs can be enabled here. Each additional PDO enabled is listed under "Process data objects → Mapping → PDO contents mapping for" and in the I/O mapping. The first 4 PDOs (0x1400 to 0x1404 for RxPDO and 0x1800 to 0x1804 for TxPDO) are enabled by default. To update the I/O mapping, additional PDOs must be enabled.

The PDO type can be used to toggle between transmit PDOs (TPDO) and receive PDOs (RPDO).

Parameter	Explanation
Configure	Enables or disables a PDO for configuration and communication
Index	Object index of the PDO
PDO name	Name of the PDO

If a PDO is selected in the table, the properties of the PDO are listed under the table.

The CANopen interface

- COB ID

The COB ID consists of the CAN identifier and additional parameters for the corresponding communication object.

Parameter	Explanation	Values
CAN ID	CAN identifier of the object.	1 to 2047
PDO exists	This parameter is identical to enabling the PDOs in the table under "Configure". If this parameter was enabled, the affected PDO is present in the mapping. By default, only the first 4 PDOs are enabled.	
RTR permitted	If selected, the message trigger mode "Remotely requested" is permitted for this PDO. This means that when an RTR triggered by a PDO consumer is received, the transfer of an event-controlled PDO is triggered. Otherwise, message trigger mode "Remotely requested" is not permitted for this PDO. The value from the EDS file is applied as the default value. ⓘ An RTR is not permitted to be used to query a transfer of emergency objects.	
29-bit	This parameter is not supported.	

- Transmission

The transfer type and rate are defined here.

Parameter	Explanation	Values
Transmission type	For the transmit or receive PDOs, different transfer types are possible. A PDO can be configured for an event-driven, synchronous or asynchronous transfer. Transfer types can be synchronized, e.g. to the synchronization message SYNC, which is transmitted by the master in defined time intervals. Synchronous means that the transfer of the PDOs is related to the synchronization message. Asynchronous means that the transfer of the PDOs is not related to the SYNC synchronization message and can be executed at any time. ⓘ Support for the different transfer types depends on the manufacturer and device. CANopen does not require support from individual or all transfer types. For information about whether a device supports the desired transfer type, check the device description of the device used.	Synchronous acyclic (0) Synchronous cyclic (1 to 240) Event-controlled, manufacturer-specific (254) Event controlled, profile-specific (255)
Transmission rate	For synchronous TPDOs, a number must be specified for transfer type "Synchronous cyclic (1 to 240)" to which the SYNC message of the data transfer refers. • A SYNC number of 1 indicates that the message is transferred with each SYNC synchronization message. • A SYNC number of n indicates that the message is transferred with every nth SYNC synchronization message. Asynchronous TPDOs are not transferred in any timing relationship with a SYNC synchronization message.	1 to 240
Inhibit time	It describes the minimum time period that must elapse between the sending of identical messages. This suppresses sending identical messages too frequently. This parameter is not supported.	
Event timer	Only for TPDOs transfer types 254 and 255. The expiration of the timer is used as an event to transmit the TPDO. Transmitting the TPDO and resetting the event timer can also be caused by an application event, however. This parameter is not supported.	

5.1.2.5 Process data objects - Mapping

The mapping can be configured here.

The PDO type can be used to toggle between transmit PDOs (TPDO) and receive PDOs (RPDO).

The table is divided into two parts. All usable objects are listed in the upper part. The lower part contains a list of the objects that have already been added to the mapping. It is added to the mapping by double-clicking on a PDO in the upper table.

Parameter	Explanation
Index.Subindex	Index or subindex of the relevant object.
Parameter	Name of the object
Data type	Data type of the object
Length	Length of the PDO in bytes
Access	PDO access right

The enabled PDOs can be toggled under "PDO contents mapping for". A maximum of 8 bytes, i.e. 64 bits, can be mapped in a PDO.

> **Information:**
>
> **The number of mappable PDOs is limited. As soon as the limit has been reached, either an object must be removed or additional PDOs must be enabled under "Process data objects - Properties" on page 13.**

5.1.2.6 Address table

Contains a list of all PDOs that are categorized according to their inputs and outputs. The respective length of the inputs and outputs as well as the assigned address can be read in this table.

This table is not available in the EDS description file on the master.

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

Parameter	Explanation
PDO-Name	The RxPDO name or TxPDO name is displayed here. These are defined in the EDS file.
PDO-Idx	Object index of the process data object (PDO)
COB ID	COB ID of the CAN message
Obj.Idx Subidx	All objects are addressed in the object index and, if applicable, in the associated sub-indexes defined by the EDS file.
Obj.Name	Object name
Type	Type of input or output data
Length	Length of the input or output data of the PDO in bytes
Address	Offset address of the input or output data

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

5.1.2.7 Device description

General device information and the entire EDS file can be read here.

5.2 EDS device description file

The module description is made available to the master in an EDS file. This file contains the description of the slave's complete range of functions. The EDS file can be downloaded from the B&R website (www.br-automation.com) in the Downloads section for the interface module and then imported into the respective master environment.

5.3 Configuration example

- In this example, the PDO mapping should be configured to 40 input bytes and 20 output bytes. "Device configuration" on the CANopen slave is opened for this.
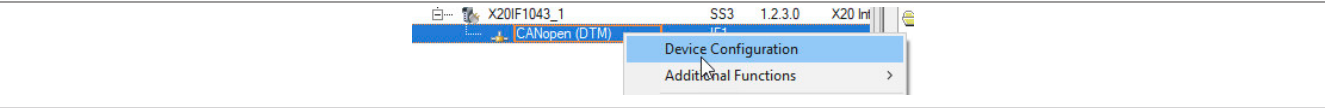
- The number of RPDOs and TPDOs that must be applied is defined under "Process data objects → Properties". 8 bytes of data can be transferred per PDO. The minimum number of RPDOs and TPDOs is always 4, even if less data should be transferred.

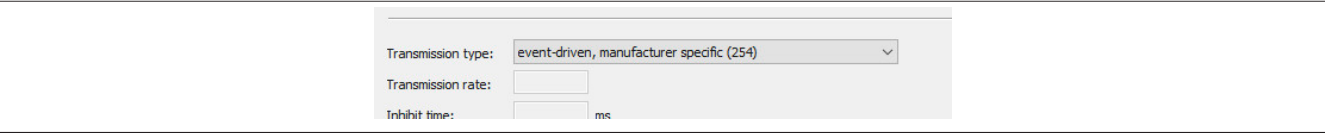### RPDO

For 40 byte input data, 5 RPDOs are selected.
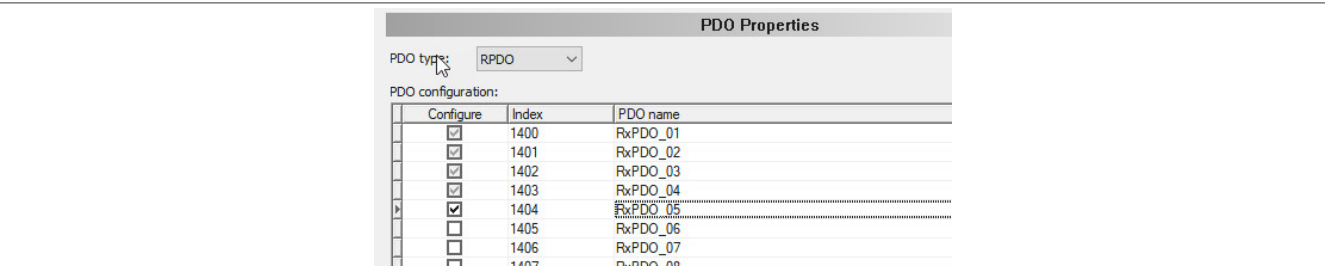
TPDO

To configure the TPDOs, "TPDO" is selected as the PDO type. Since only 20 bytes should be transferred, the default setting of 4 PDOs remains unchanged.

- The type of data transfer is defined (cyclic, acyclic, event-driven, etc.) under "Transmission type". For this example, the PDOs are transferred as "event-driven", i.e. only when data is changed.

- The PDOs are filled under "Process data objects → Mapping". The configured RPDOs are listed under "PDO contents mapping for"; in this example, they are listed as RPDO 1 to RPDO 5.
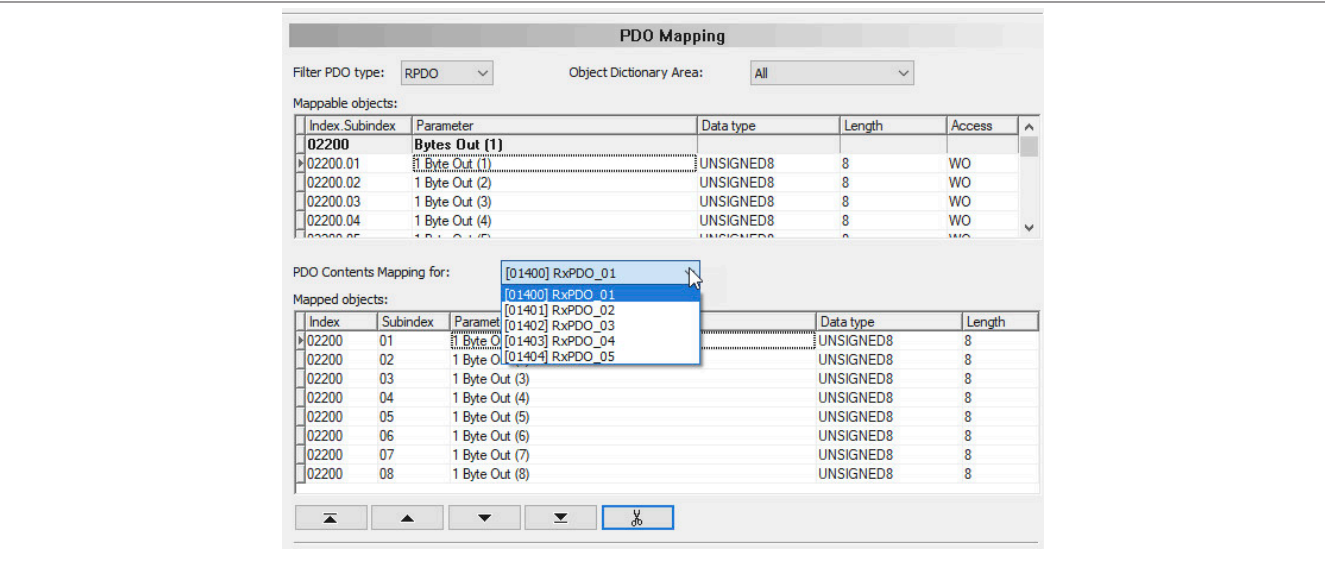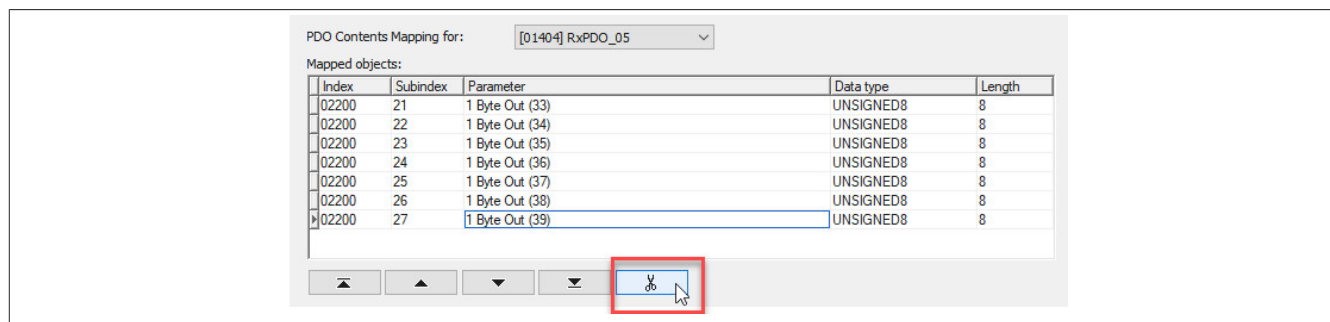

By default, the PDOs are filled with bytes. No change must therefore be made for the input bytes.

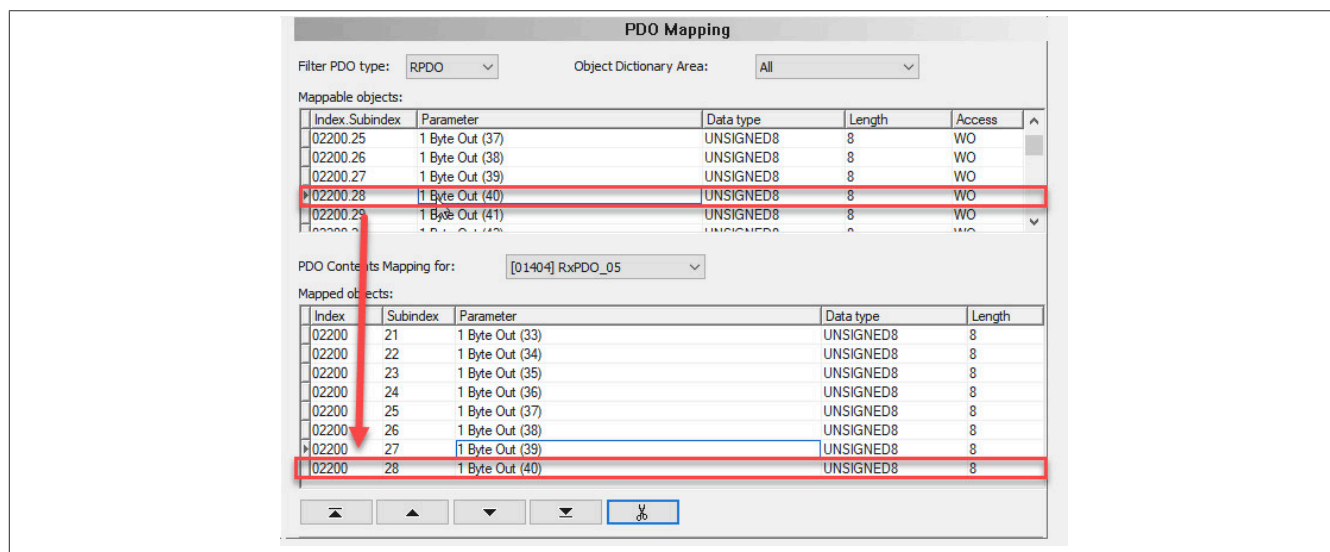If a change is desired, RPDOs can be swapped out or data cut and pasted with "PDO contents mapping for".

Cut

Select the desired object and cut it out with the scissors.

Paste

Double-clicking on an index in "Mappable objects" transfers the object to "Mapped objects".

● TPDOs are edited by changing to "TPDO" under "Filter PDO type". Bytes that are not required can be removed from the assignment. This is done by selecting the relevant TPDOs and cutting the superfluous objects using the scissors.

Because only 20 output bytes are needed in this example, no bytes are mapped in TPDO4 and only 4 bytes in TPDO3.
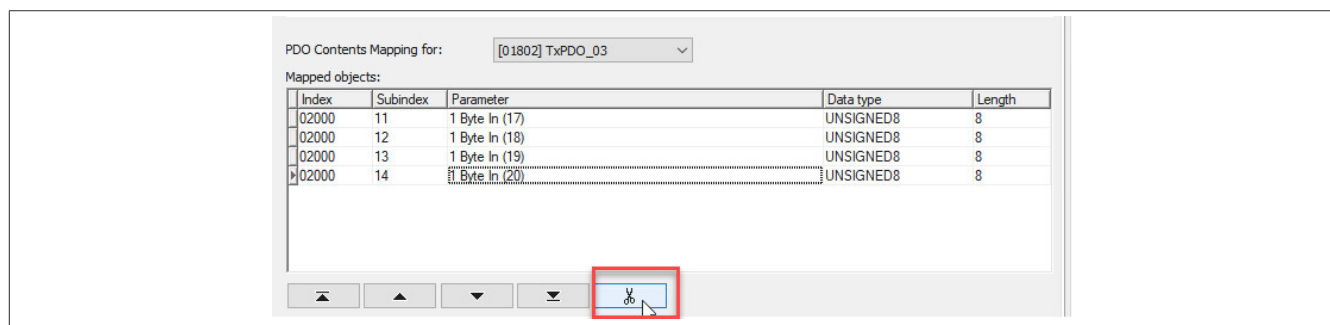
● Configuring the device description file

Identical settings must be made in the device description file. To do this, the device description file attached to the master must be set identically to the device configuration and I/O mapping of the CANopen slave.

If the setting on the CANopen slave and the device description file attached to the master do not match, no connection is established.

> **ⓘ Information:**
>
> **The number of input and output bytes must be set identically on the master and slave. The direction of the data on the slave is applied in the direction opposite to the master since data exchange takes place in the opposite direction.**

In the following example, this means:

- Device description file on the master: 40 bytes in the output direction and 20 bytes in the input direction.
- Slave: 40 bytes in the input direction and 20 bytes in the output direction.