TURCK

Your Global Automation Partner

# TBEN-S2-2COM-4DXP Compact I/O Module for Serial Data Transmission

Instructions for Use

# Table of Contents

# 1 About these instructions

These instructions describe the setup, functions and use of the product and help you to operate the product according to its intended purpose. Read these instructions carefully before using the product. This will prevent the risk of personal injury and damage to property. Keep these instructions safe during the service life of the product. If the product is passed on, pass on these instructions as well.

## 1.1 Target groups

These instructions are written for specifically trained personnel and must be read carefully by anyone entrusted with the installation, commissioning, operation, maintenance, disassembly or disposal of the device.

When using the device in Ex areas, the user must also have knowledge of explosion protection (IEC/EN 60079-14 etc.).

## 1.2 Explanation of symbols

The following symbols are used in these instructions:

| | |
|---|---|
|  | **DANGER**<br>DANGER indicates a hazardous situation with a high level of risk, which, if not avoided, will result in death or serious injury. |
|  | **WARNING**<br>WARNING indicates a hazardous situation with a medium level of risk, which, if not avoided, will result in death or serious injury. |
|  | **CAUTION**<br>CAUTION indicates a hazardous situation with a medium level of risk, which, if not avoided, will result in moderate or minor injury. |
|  | **NOTICE**<br>CAUTION indicates a situation which, if not avoided, may cause damage to property. |
|  | **NOTE**<br>NOTE indicates tips, recommendations and important information about special action steps and issues. The notes simplify your work and help you to avoid additional work. |
| ▶ | **MANDATORY ACTION**<br>This symbol denotes actions that the user must carry out. |
| ⇨ | **RESULT OF ACTION**<br>This symbol denotes the relevant results of an action. |

## 1.3 Other documents

The following additional documents are available online at www.turck.com

- Data sheet
- TBEN-Accessories list (D301367)
- Notes on Use in Ex zone 2 and 22 (100022986)
- Declarations of conformity (current version)
- Approvals

## 1.4 Feedback about these instructions

We make every effort to ensure that these instructions are as informative and as clear as possible. If you have any suggestions for improving the design or if some information is missing in the document, please send your suggestions to techdoc@turck.com.

# 2 Notes on the product

## 2.1 Product identification

These instructions apply to the following compact I/O module for serial data transmission:

- TBEN-S2-2COM-4DXP

## 2.2 Scope of delivery

The delivery consists of the following:

- Closing caps for M8 or M12 sockets
- Labeling clips

## 2.3 Turck service

Turck supports you in your projects — from the initial analysis right through to the commissioning of your application. The Turck product database at www.turck.com offers you several software tools for programming, configuring or commissioning, as well as data sheets and CAD files in many export formats.

For the contact details of our branches worldwide, please see page [▶ 167].

# 3 For your safety

The product is designed according to state of the art technology. Residual hazards, however, still exist. Observe the following safety instructions and warnings in order to prevent danger to persons and property. Turck accepts no liability for damage caused by failure to observe these safety instructions.

## 3.1 Intended use

Due to the Turck multiprotocol technology, the compact multiprotocol I/O modules for Ethernet can be operated in the four Ethernet protocols PROFINET, EtherNet/IP, Modbus TCP and CC-Link IE Field Basic. The modules detect the bus protocol automatically during the start-up.

The block module TBEN-S2-2COM-4DXP offers two serial interfaces for connecting serial RS232 and RS482 data terminal devices. Besides the raw RS232/RS485-communication, the device supports Modbus RTU. 32 Modbus RTU server can be connected per port. Additionally, the module provides 4 universal digital channels.

The device must only be used as described in these instructions. Any other use is not in accordance with the intended use. Turck accepts no liability for any resulting damage.

## 3.2 General safety instructions

- The device must only be fitted, installed, operated, parameterized and maintained by trained and qualified personnel.
- Only use the device in compliance with the applicable national and international regulations, standards and laws.
- The device meets the EMC requirements for the industrial areas. When used in residential areas, take measures to prevent radio frequency interference.
- Change the default password of the integrated web server after the first login. Turck recommends the use of a secure password.

## 3.3 Notes on Ex protection

- When using the device in Ex areas, the user must have knowledge of explosion protection (IEC/EN 60079-14 etc.).
- Observe national and international regulations for explosion protection.
- Only use the device within the permissible operating and ambient conditions (see certification data and Ex approval specifications).
- The document "Notes on Use in Ex Zone 2 and 22" (ID 100022986) contains the approval data for using the device in hazardous areas. Observe the requirements in the document.

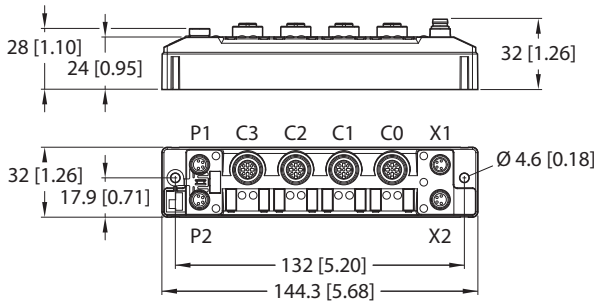
## 3.4 Requirements for Ex approval

- Only disconnect and connect circuits when there is no potentially explosive atmosphere or when the power supply is switched off
- Connect the metal protective cover to the equipotential bonding in the Ex area (cable cross-section: 4 $mm^2$).
- Ensure impact resistance in accordance with EN IEC 60079-0 – alternative measures:
  - Install the device in the TB-SG-S protective housing (ID 100014866).
  - Install the device in an area offering impact protection (e.g. in the robot arm) and attach a warning sign: "DANGER: Do not connect or disconnect the device under power."
- Do not install the device in areas critically exposed to UV light.
- Prevent risks caused by electrostatic charge.
- Provide unused male connectors with suitable sealing or blanking caps in order to ensure degree of protection IP65, IP67 or IP69K The tightening torque for the M4 screws is 0.5 Nm.

# 4 Product description

The devices are designed in a fully encapsulated housing with degree of protection IP65/IP67/IP69K. Two COM ports are available for connecting serial devices. To connect digital sensors and actuators, the device has four universal digital I/O channels that can be used as inputs or outputs without configuration. The connections for serial devices and digital I/Os are designed as M12 sockets. Two 4-pin M8 sockets are provided for the Ethernet connection. The connections for the supply voltage are also designed as 4-pin M8 connectors.

## 4.1 Device overview




mm [Inch]

Fig. 1: TBEN-S2… – dimensions

### 4.1.1 Display elements

The device has the following LED indicators:

- Power supply
- Group and bus errors
- Status
- Diagnostics

## 4.2 Properties and features

- Fiber-glass reinforced housing
- Shock and vibration tested
- Fully potted module electronics
- Degree of protection IP65/IP67/IP69K
- 4-pole M8-connectors for voltage supply
- Two 4-pole M8 connectors for the connection to Ethernet
- Two serial interface which can be used as RS232 or RS485
- Integrated Modbus RTU Client function for connecting up to 32 Modbus RTU Server per port
- Four universal digital channels as PNP inputs or outputs (0,5 A)
- 192 byes per port per write or read operation
- Separated voltage groups
- LED displays and diagnostics
- Multi protocol: PROFINET device, EtherNet/IP device, Modbus TCP server, CC-Link IE Field Basic server

## 4.3 Operating principle

The devices are equipped with a multi-protocol Ethernet interface. Via the Ethernet interface, the multiprotocol device is connected to an (existing) Ethernet network as an EtherNet/IP device, Modbus TCP slave, PROFINET device or CC-Link IE Field Basic server. The process data is exchanged between Ethernet and TBEN-S during operation. The RS232/RS485 interfaces are used for connecting devices with RS232 and/or RS485 interface (e.g. bar code readers, printers, drives, light curtains, etc.).

## 4.4 Functions and operating modes

### 4.4.1 Multiprotocol technology

The device can be used in the following Ethernet protocols:

- PROFINET
- EtherNet/IP
- Modbus TCP
- CC-Link IE Field Basic

The required Ethernet protocol can be detected automatically or determined manually.

#### Automatic protocol detection

A multiprotocol device can be operated without intervention of the user (which means, without changes in the parameterization) in all of the three Ethernet protocols mentioned.

During the system start-up phase (snooping phase), the module detects which Ethernet protocol requests a connection to be established and adjusts itself to the corresponding protocol. After this an access to the device from other protocols is read-only.

#### Manual protocol selection

The user can also define the protocol manually. In this case, the snooping phase is skipped and the device is fixed to the selected protocol. With the other protocols, the device can only be accessed read-only.

#### Protocol-dependent functions

The device supports the following Ethernet protocol-specific features:

**PROFINET**

- Fast Start Up (FSU), prioritized start-up, only digital I/O channels
- S2 redundancy

**EtherNet/IP**

- QuickConnect (QC), only digital I/O channels
- Device Level Ring (DLR)

#### Ethernet ports used

| Port | Protocol |
|---|---|
| 00022 | SFTP |
| 00053 | DNS TCP |
| 00067 | DHCP |
| 00080 | HTTP |
| 00093 | PROFINET DCP |
| 00502 | Modbus TCP |
| 58554 | Turck Services |

### 4.4.2 Serial RS232 or RS485 data communication

In RS232 mode one device can be connected to each serial port. In RS485 mode up to 32 devices can be connected to one port.

The chapter "Operating" contains a description of the transmit an receive sequence [▶ 151].

### 4.4.3 Modbus RTU data communication

The Modbus RTU data communication is coordinated by the TBEN-S2-2COM-4DXP and is transparent for PLC programmers. The TBEN-S2-2COM-4DXP functions as a Modbus RTU Client (Modbus RTU Master). The process values of connected Modbus RTU Servers (Modbus RTU Slaves) are directly available. In addition to that, the state of all Modbus connections can be monitored.

#### Modbus client mode

The Modbus Client Mode RS232, allows the connection of up to 8 Modbus RTU Servers. The Modbus Client Mode RS485 allows the connection of up to 32 Modbus RTU Servers without any programming effort. Depending on the connected devices, applications with up to 64 RS485-devices are possible .

- Standard mode [▶ 148]
  - One Modbus RTU server per server configuration Block (SCB)
  - Max. 8 Modbus RTU Servers per COM port
- Multi server mode [▶ 149]
  - Up to twelve identical Modbus RTU Servers per Server Configuration Block (SCB)
  - Max. 32 Modbus RTU server per COM port, in total max. 64 per TBEN-S2-2COM-4DXP device
- Read/write extension [▶ 150]
  - Connection of Modbus RTU Servers with more than 12 registers, which have to be read or written.

### 4.4.4 Digital modules – extended digital functions

In PROFINET, the extended digital functions are configured via device parameterization via GSDML file. In EtherNet/IP, the functions are provided in special catalog files for RSLogix from Rockwell Automation. In Modbus TCP the extended functions are configured via Modbus registers. In addition to that, the functions are configurable via the device's web server or the device DTMs.

The digital TBEN modules provide the following extended digital functions:

#### Digital filter

The function "digital filter" extends the filter time of digital inputs to 3 ms. Digital input signals can thus be reliably detected even when short-term interfering signals in rough environments occur.

#### Impulse stretch

The function "impulse stretch" allows a detection of short input signals in longer PLC cycle times by means of signal extension.

### 4.4.5 Turck Field Logic Controller function (FLC ARGEE)

The device supports logic processing via the "Turck Field Logic Controller (FLC ARGEE)" function. This enables the device to implement small to medium-sized control tasks in order to reduce the load of the central controller. The FLCs can be programmed in the ARGEE engineering environment.

The ARGEE programming software can be downloaded free of charge from **www.turck.com**.

The "SW_ARGEE_Environment_Vx.x.zip" file also contains the documentation for the programming environment as well as the software.

## 4.5 Possible Ethernet network structures

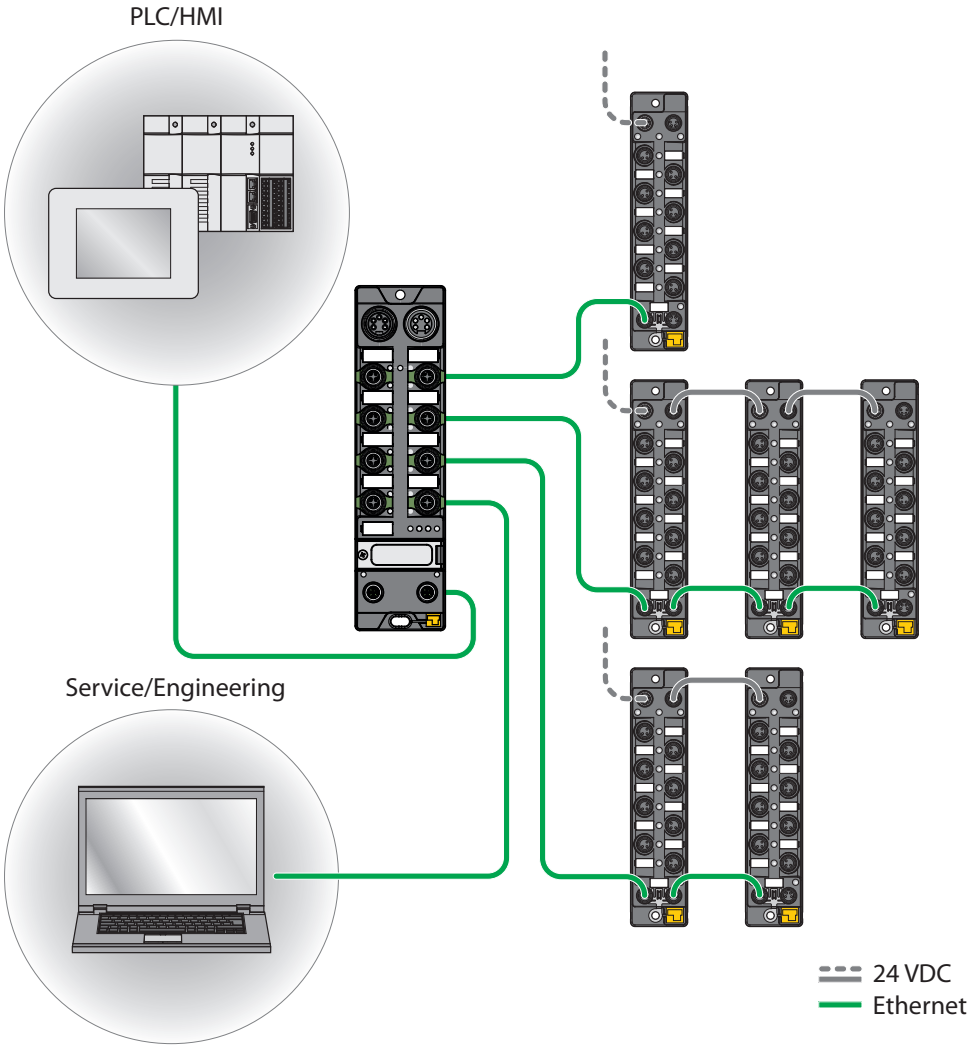



Fig. 2: Network structure, example 1

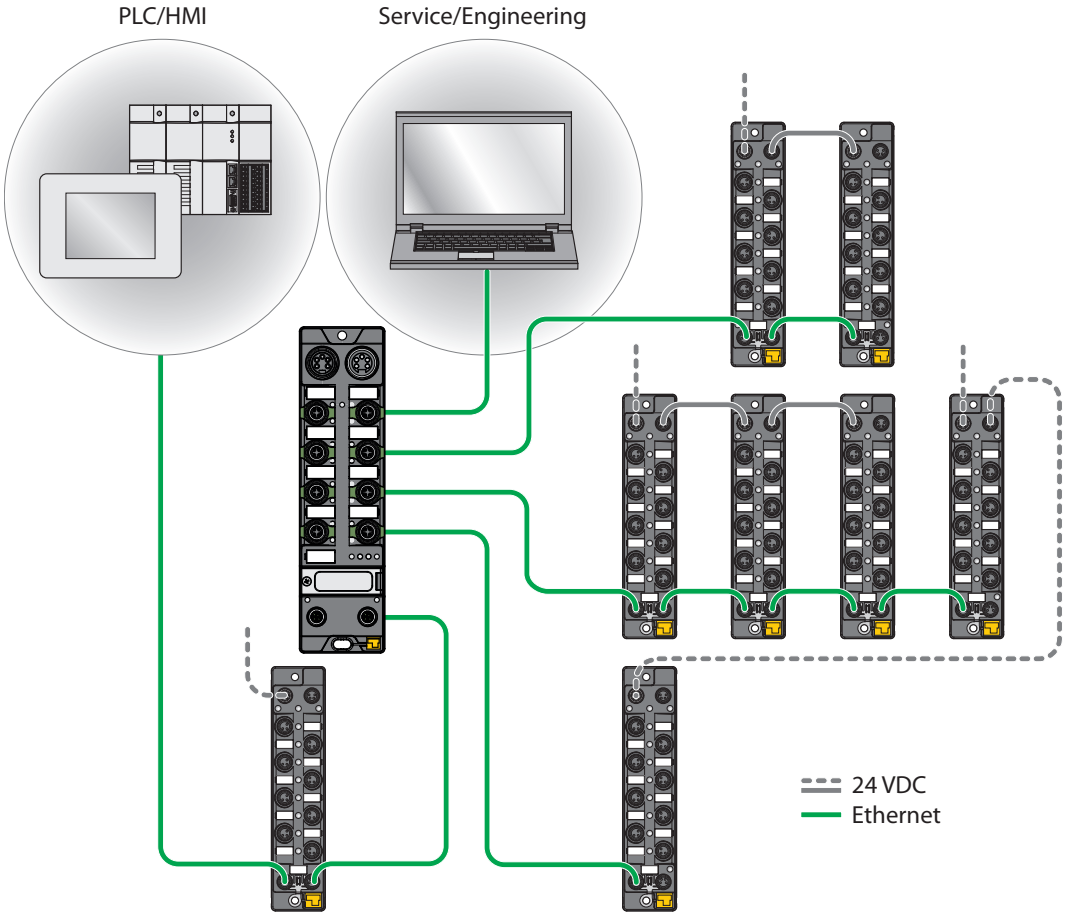



Fig. 3: Network structure, example 2

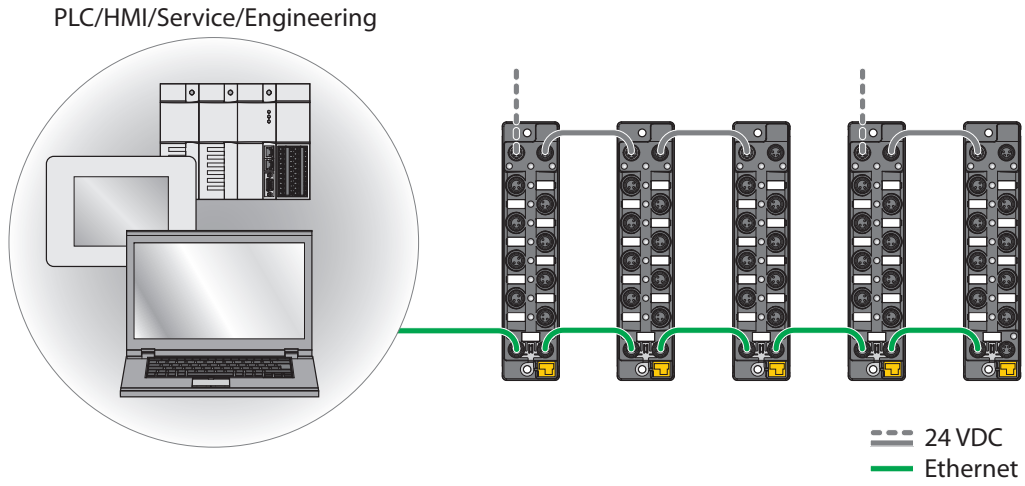



Fig. 4: Network structure, example 3

### 4.5.1 Ethernet daisy chain - max. number of connected modules

Prerequisites:

- Optimized network: only TBEN modules in the daisy chain, no additional switches, no third-party devices
- Exchange of pure cyclical process data, no acyclical data

| Cycle time | Maximum number of TBEN modules |
|---|---|
| 1 ms | 21 |
| 2 ms | 42 |

**NOTE**
Deviations from the specification above may lead to a reduction of possible TBEN modules connected to one daisy chain.

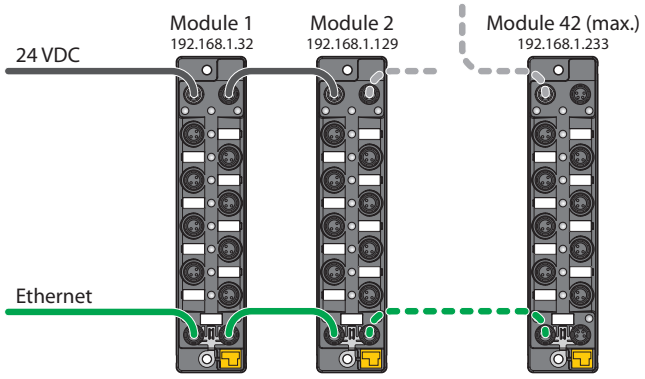



Fig. 5: Daisy chain

## 4.6 Technical accessories

Optionally available accessories for mounting, connecting and parameter setting can be found in the Turck product database at www.turck.com. Accessories are not supplied with the device.

# 5 Mounting

The device can be mounted on a DIN rail according to EN 60715 (TS35) or screwed onto an even mounting plate. Both composite and individual assembly are possible.

## 5.1 Installing a device in zone 2 and zone 22

The devices can be used in combination with the TB-SG-S (ID 100014866) protective housing set in zone 2 and zone 22. Combined mounting is not possible in zone 2 and zone 22.

**DANGER**
Potentially explosive atmosphere
**Risk of explosion due to spark ignition**
**Operation in zone 2 or zone 22:**

- ▶ Only install the device if there is no potentially explosive atmosphere present.
- ▶ Observe the requirements for Ex approval.

- ▶ Screw on the housing. Use a Torx T8 screwdriver.
- ▶ Place the device on the base plate of the protective housing fasten both together on the mounting plate [▷ 19].
- ▶ Connect the device, [▷ 23].
- ▶ Fit the housing cover and screw on as shown in the following figure. The tightening torque for the Torx T8 screw is 0.5 Nm.

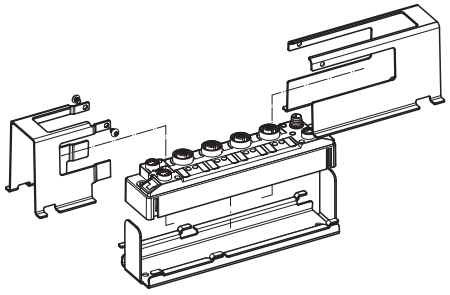


Fig. 6: Installing the device in the TB-SG-S protective housing

## 5.2 Combine TBEN-S modules for mounting

The modules can be mounted individually or in combination as a group of modules on a mounting plate or DIN rail.

### 5.2.1 Combine TBEN-S modules for composite mounting to a mounting plate

The TBNN-S0-STD connector serves for composite mounting of TBEN-S modules on a mounting plate:

- Unlock the cover flap at the connector with a flat tool (e.g. screw driver) (1).
- Open the flap completely (2).
- Connect the module and the connector so that the spring of the connector is inserted into the groove of the TBEN-S module (3).
- Flap back the cover and close it (4). It has to engage audibly.
- Repeat steps 1 to 4 until the module group in complete.

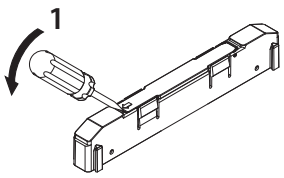



Fig. 7: Step 1

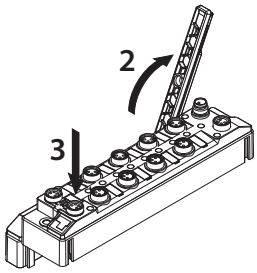



Fig. 8: Step 2

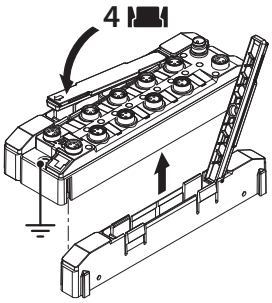



Fig. 9: Step 3

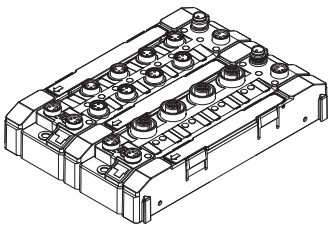


Fig. 10: Step 4

### 5.2.2 Combine TBEN-S modules for single and composite mounting on a DIN rail

The TBNN-S0-DRS adapter serves for single and composite mounting of TBEN-S modules on a DIN rail.

**NOTICE**
Incorrect mounting
**Incorrect grounding may cause malfunction**

- ▶ Align the adapters so that the arrow on the locking lever points in the direction of the M8 Ethernet sockets.
- ▶ Connect the grounding contact of the adapter with the grounding contact of the module.

- ▶ Unlock the cover flap at the connector with a flat tool (e.g. screw driver) (1).
- ▶ Open the flap completely (2).
- ▶ Connect the module and the connector so that the spring of the connector engages in the groove of the module (3).
- ▶ Flap back the cover and close it (4). It has to engage audibly.
- ▶ Repeat steps 1 to 4 until the module group in complete.

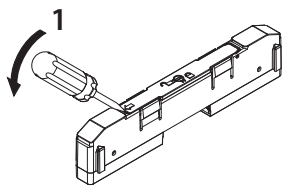



Fig. 11: Step 1

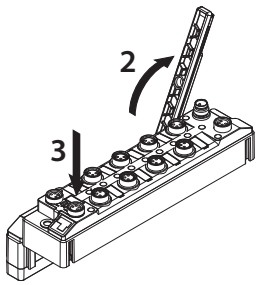



Fig. 12: Step 2

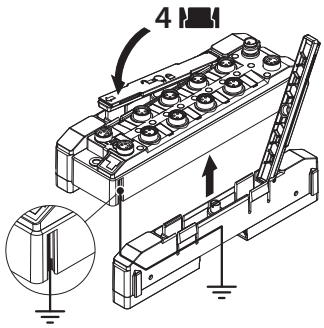



Fig. 13: Step 3

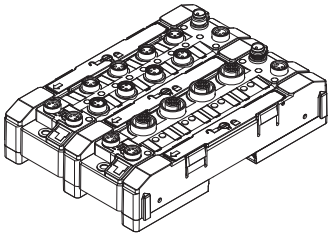


Fig. 14: Step 4

## 5.3 Attach TBEN-S modules to a mounting plate

- Fasten the module or module composite to a mounting plate with two M4 screws per device. The maximum tightening torque for the M4 screws is 1.3 Nm
- Avoid mechanical stresses.
- Optional: Ground the device.

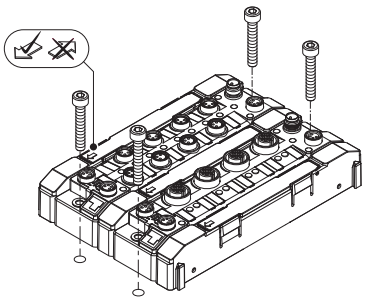


Fig. 15: Mounting the device on a mounting plate

## 5.4 Mounting TBEN-S modules on a DIN rail (TS35)

- For composite or single mounting: Mount connectors to the left and to the right of the module.
- Place the module or module composite on the DIN rail so that the cut-outs in the connector enclose the DIN rail (1).
- Avoid mechanical stresses.
- Close the rotating bolt of the connector with a screwdriver (2).
- Optional: Ground the device.

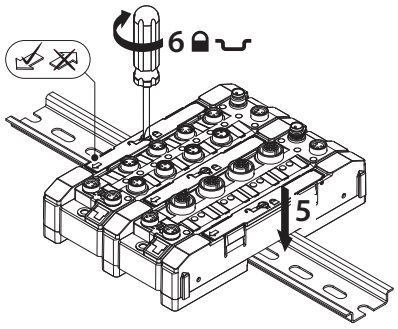



Fig. 16: Mounting a module composite on a DIN rail

**NOTE**
To increase stability on the DIN rail, end brackets can be mounted on the right and left of the module or the module combination.

## 5.5 Outdoor device installation

The device is UV resistant in accordance with DIN EN ISO 4892-2. Direct sunlight may cause material wear and changes in color. The mechanical and electrical properties of the device are not impaired.

- To prevent material wear and color changes: Protect the device from direct sunlight with protective panels.

## 5.6 Grounding the device

### 5.6.1 Equivalent wiring diagram and shielding concept

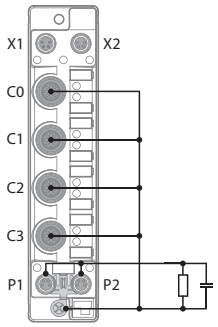



Fig. 17: Equivalent wiring diagram and shielding concept

### 5.6.2 Fieldbus and I/O level shielding

The fieldbus and the I/O level of the TBEN-S modules can be grounded separately.

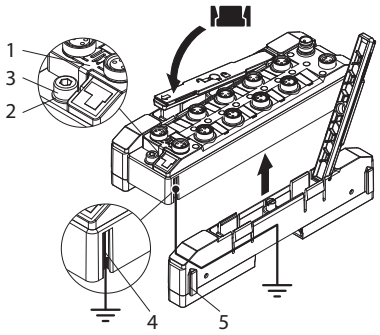



Fig. 18: Fieldbus and I/O level shielding

The grounding ring (2) and the grounding contact (4) are connected to each other and form the module grounding. The shielding of the I/O level is permanently connected to the module grounding. The module grounding is only connected to the reference potential of the installation when the module is mounted.

#### Shielding concept of the I/O modules (I/O level)

In the case of direct mounting on a mounting plate, the module grounding is connected to the reference potential of the system via the metal screw in the lower mounting hole (3). If no module grounding is desired, the electrical connection to the reference potential must be interrupted, e.g. by mounting the device on an insulated mounting plate.

In the case of DIN rail mounting, the module earthing is led through the side grounding contact (4) via connector TBNN-S0-DRS to the top-hat rail and connected to the reference potential of the installation. If no module grounding is desired, the electrical connection to the reference potential must be interrupted, e.g. by removing the grounding spring on the TBNN-S0-DRS.

### Shielding concept of the fieldbus level

On delivery, a grounding clip (1) is provided on the connectors for the fieldbus connection (P1, P2).

In the case of direct mounting on a mounting plate, the shielding of the fieldbus cables is routed directly to the module grounding via the ground clip and the metal screw in the lower mounting hole. In the case of DIN rail mounting, the shielding of the fieldbus cables is connected to the module grounding by the metal screw. The metal screw is supplied with the TBNN-S0-DRS connector.

If direct grounding of the fieldbus shield is not desired, the grounding clip (1) must be removed. In this case, the fieldbus shield is connected to the module ground via an RC element.

## 5.6.3 Grounding the device – I/O and fieldbus level

The grounding of the fieldbus level can either be connected directly via the grounding clip (1) or connected and routed indirectly via an RC element to the grounding of the I/O level. If the fieldbus grounding is to be routed via an RC element, the grounding clip must be removed.

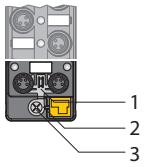



Fig. 19: Grounding clip (1)

### Removing the grounding clip: disconnect the direct grounding of the fieldbus level

- ▶ Use a slim slotted screwdriver in order to lift up and remove the grounding clamp.

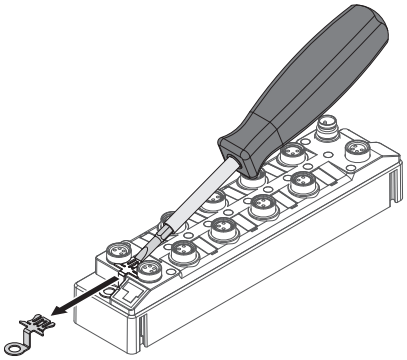


Fig. 20: Use a flat slotted screwdriver to push the grounding clip forwards and remove it.

#### Mounting the grounding clip: grounding the fieldbus level directly

- ▶ Place the grounding clamp between the fieldbus connectors by using a screwdriver in such way that the clamp contacts the metal housing of the connectors.
- ⇨ The shielding of the fieldbus cables is connected to the grounding clip.

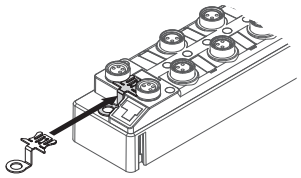


Fig. 21: Mounting the grounding clip

### 5.6.4 Grounding the device – mounting on a DIN rail

- ▶ For mounting on a DIN rail with TBNN-S0-DRS connectors: Screw the enclosed metal screw into the lower mounting hole of the TBEN-S module.
- ⇨ The shielding of the M8 flanges of the I/O level is connected to the reference potential of the installation via the DIN rail and the connector.
- ⇨ With mounted grounding clip: The shielding of the fieldbus is connected to the reference potential of the installation via the module grounding of the I/O level.

### 5.6.5 Grounding the device – mounting on a mounting plate

- ▶ For mounting onto a mounting plate: Fix the TBEN-S module with an M4 metal screw through the lower mounting hole.
- ⇨ The shielding of the M8 flanges for the I/O level is connected to the reference potential of the installation via the M4 metal screw.
- ⇨ With mounted grounding clip: The shielding of the fieldbus is connected to the reference potential of the installation via the module grounding of the I/O level.

# 6 Connecting

**NOTICE**
Penetration of liquids or foreign objects due to leaking connections
**Loss of degree of protection IP65/IP67/IP69K possible**

- Tighten the M8 plug connector with a tightening torque of 0.4 Nm (Ethernet) or 0.6 Nm (I/Os and supply).
- Tighten M12 male connectors with a tightening torque of 0.8 Nm.
- Only use accessories that guarantee the protection class.
- Provide unused male connectors with suitable sealing or blanking caps. The tightening torque for the M4 screws is 0.5 Nm.

## 6.1 Connecting a device in zone 2 and zone 22

**DANGER**
Explosive atmosphere
**Explosion due to ignitable sparks**
**For use in Zone 2 and Zone 22:**

- Only disconnect and connect circuits when there is no potentially explosive atmosphere or when the power supply is switched off.
- Only use connecting cables that are approved for use in potentially explosive atmospheres.
- Use all connectors or seal them with screw caps or blind caps. The tightening torque for the screw caps is 0.5 Nm.
- Observe requirements for Ex approval.

## 6.2 Connecting a device to Ethernet

The device is provided with an autocrossing switch with two 4-pin M8 Ethernet plug connectors for connecting to Ethernet.

**NOTICE**
Reversing the Ethernet and power supply cables
**This may destroy the module electronics**

- When connecting the Ethernet and supply cables make sure that the correct M8 plug connectors are used:
  - Ethernet: P1 and P2
  - power supply: X1 and X2

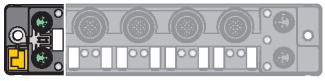


Fig. 22: M8 Ethernet plug connectors for connecting the fieldbus

- Connect the device to the fieldbus according to the pin assignment below.
- Provide unused male connectors with suitable sealing or blanking caps. The tightening torque for the M4 screws is 0.5 Nm.

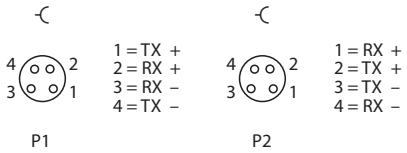



Fig. 23: Ethernet connections — pin assignment of P1 and P2

### 6.2.1 Applications with QuickConnect (QC) and Fast Start Up (FSU)

- Do not use crossover cables in applications with QuickConnect (QC) or Fast Startup (FSU).
- Connect incoming Ethernet cables to P1.
- Connect outgoing Ethernet cables to P2.

## 6.3 Connecting the supply voltage

The device is provided with two 4-pin M8 plug connectors for connecting the power supply. V1 and V2 are galvanically isolated.

**NOTICE**
Reversing the Ethernet and power supply cables
**This may destroy the module electronics**

- When connecting the Ethernet and supply cables make sure that the correct M8 plug connectors are used:
  - Ethernet: P1 and P2
  - power supply: X1 and X2

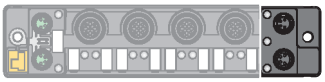


Fig. 24: M8 connector for connecting the supply voltage

- Connect the device to the power supply according to the pin assignment shown below.
- Provide unused male connectors with suitable sealing or blanking caps. The tightening torque for the M4 screws is 0.5 Nm.

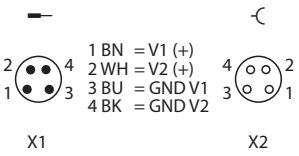



Fig. 25: Pin assignment power supply connectors

| | Meaning |
|---|---|
| X1 | Power feed |
| X2 | Continuation of the power to the next node |
| V1 | Power supply 1 (incl. supply of electronics) |
| V2 | Power supply 2 |

**NOTE**
The system voltage (V1) and the load voltage (V2) are fed in and monitored separately. In case of an undercut of the admissible voltage, the connectors are switched-off according to the module's supply concept. In case of an undervoltage at V2, the LED PWR changes from green to red. In case of an undervoltage at V1, the LED is turned off.

### 6.3.1 Supply concept

All TBEN-S-modules are supplied via two separate voltages V1 and V2. The I/O-channels are therefore consequently separated into the different potential groups "switchable I/O" (supplied through V2) and "non-switchable" I/O (supplied through V1).

V1 = supply of the module electronics and the respective slots

V2 = supply of the respective slots

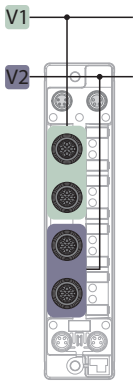



Fig. 26: Supply TBEN-S2-2COM-4DXP

## 6.4 Connecting serial devices

The TBEN-S2-2COM-4DXP has with two 5-pin M12 connectors for connecting serial RS232 or RS485 devices.

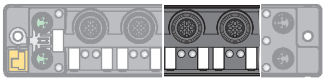


Fig. 27: M12 connector for connecting serial RS232 and RS485 devices

▶ Connect the device to the voltage supply according to the pin assignment below.

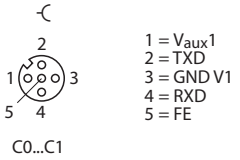



Fig. 28: Pin assignment RS232 connection

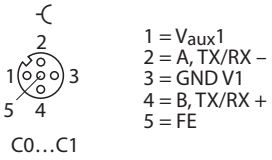



Fig. 29: Pin assignment RS485 connection

### 6.4.1 Activating and deactivating of RS485 termination and biasing

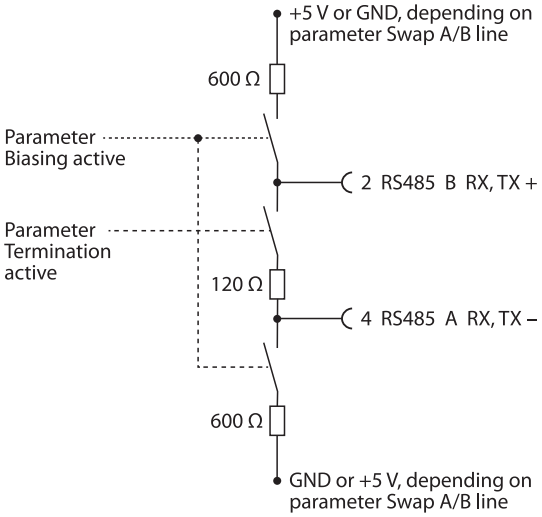



Fig. 30: RS485 termination and biasing

#### RS485 termination

The TBEN-S2 module is provided with one internal bus termination resistor per COM port which can be activated or deactivated via the parameter "Termination active" [▶ 133]. The internal termination is activated per default. The termination can also be done externally. The internal termination has to be deactivated if an external termination is used.

The termination at TBEN-S2-2COM-4DXP is necessary, if the device is mounted at the beginning or the end of the RS485 line.

When building up an RS485 line topology a terminating resistor (e.g. RSE57-TR2/RFID) has to be set at the other end of the RS485 line.

### Biasing

Activating the biasing function suppresses undefined signal levels on both signal lines in the RS485 network by means of a bias resistor.

In the TBEN-S2-2COM-4DXP the biasing is is done with a biasing resistor of 600 Ω.

The biasing function can be activated or deactivated via the parameter [▷ 133]. The biasing function is activated per default. We recommend to activate the biasing function, if the TBEN-S2-2COM-4DXP is mounted at the beginning or the end of the RS485.

## 6.5 Connecting digital sensors and actuators

The TBEN-S2-device has 5-pin M12 connectors for connecting digital sensors and actuators.

The following combinations of sensors an actuators can be connected:

- Two digital inputs
- Two digital outputs
- One digital input and one digital output

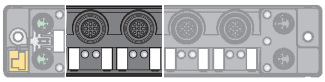


Fig. 31: M12 connectors for connecting digital sensors and actuators

▶ Connect the sensors and actuators to the device according to the pin assignment.

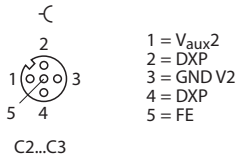



Fig. 32: Pin assignment C2 and C3

The channels are assigned to the connectors as follows:

| Channel | Connector | Pin |
|---|---|---|
| DXP4 (Ch4) | C2 | 4 |
| DXP5 (Ch5) | C2 | 2 |
| DXP6 (Ch6) | C3 | 4 |
| DXP7 (Ch7) | C3 | 2 |

# 7 Commissioning

## 7.1 Adjusting network settings

**NOTE**
Changes to network settings are only applied after restarting the device.

The network settings can be adapted via TAS (Turck Automation Suite), the web server, the DTM, a DHCP server or PROFINET DCP.

### 7.1.1 Adjusting network settings via TAS (Turck Automation Suite)

In the delivery state the device has the IP address 192.168.1.254. The IP address can be set via TAS (Turck Automation Suite). TAS is available free of charge at www.turck.com.

- Connect the device to a PC via the Ethernet interface.
- Open TAS.
- Click **Scan network**.

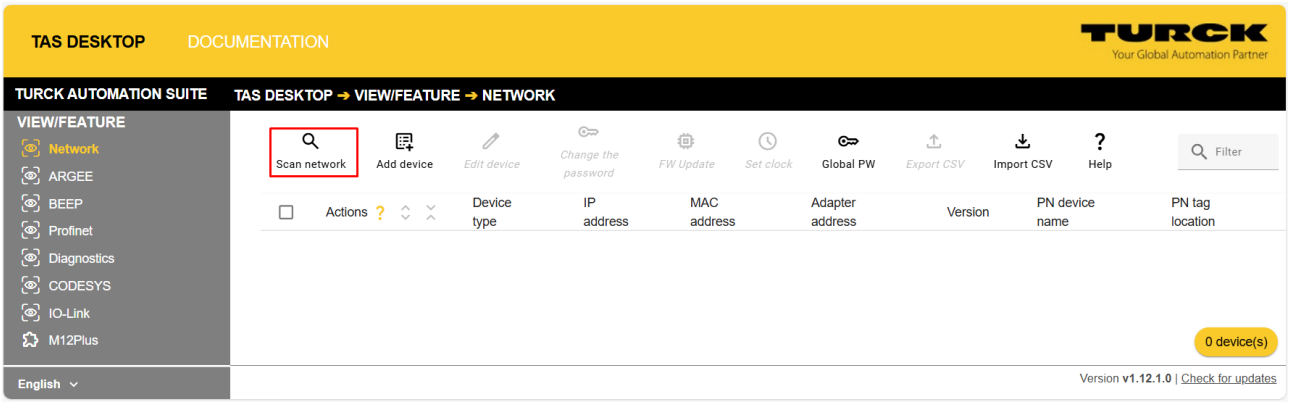



Fig. 33: Home screen in TAS

⇨ TAS shows the connected devices.

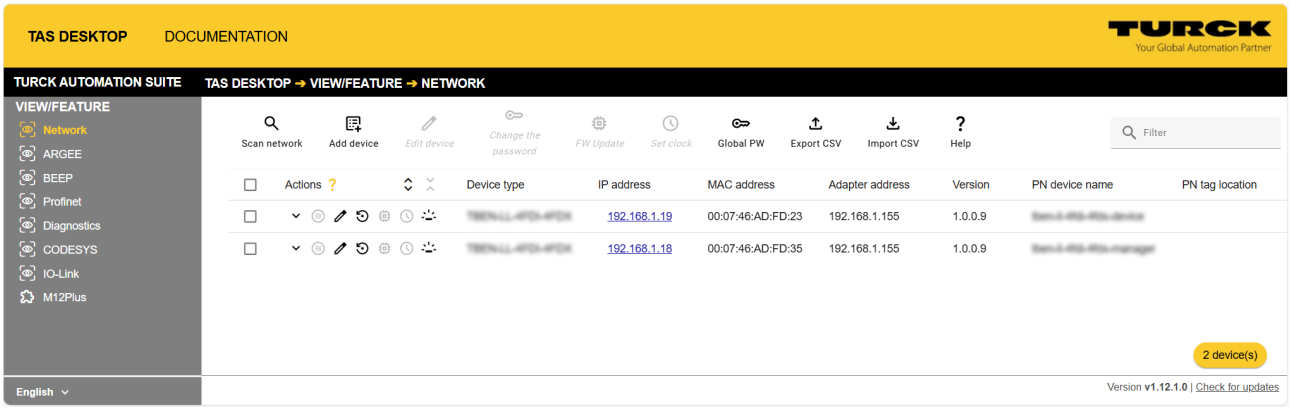



Fig. 34: Found devices in TAS

- ▶ Select the relevant device (check box).
- ▶ Click **Edit device**.

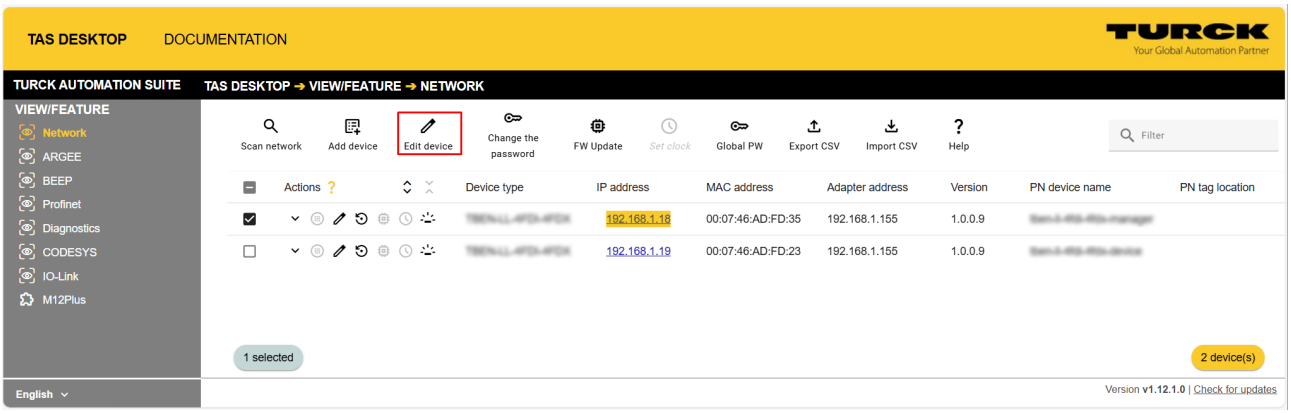



Fig. 35: Selecting the device in TAS

**NOTE**
By clicking on the IP address of the device, the configuration view of the device can be opened either in TAS or on the device website.

- ▶ Enter the device password and click **Login** The default password is "password".
  Note: Turck recommends changing the password after the first login for security reasons.

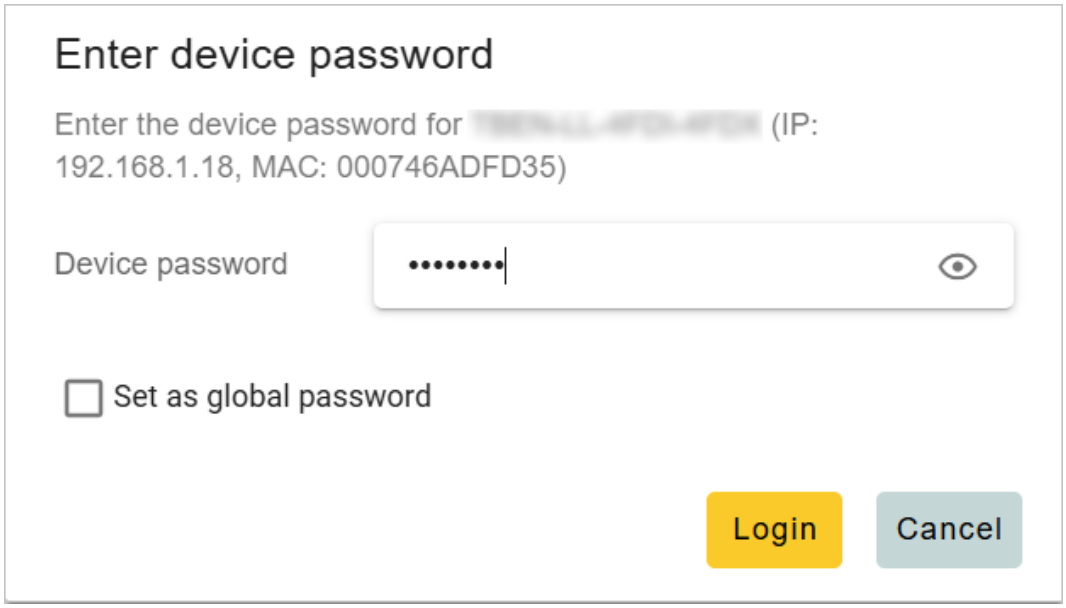



Fig. 36: Entering the device password

- ▶ Change the PN device name, IP address and, if necessary, the default gateway, subnet mask and PN tag location.
- ▶ Save changes by clicking on **Apply** .

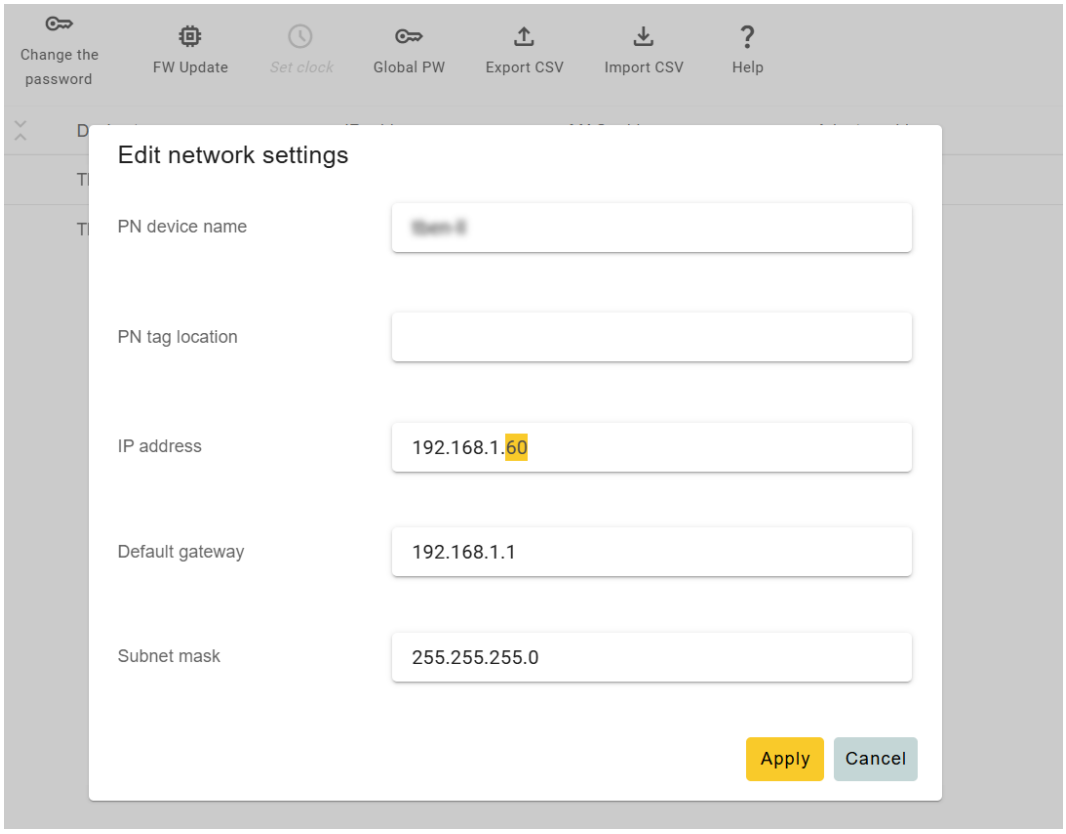



Fig. 37: Changing network settings in TAS

## 7.2 Commissioning the devices in PROFINET

### 7.2.1 Address setting in PROFINET

In IP-based communication, the field devices are addressed by means of an IP address. PROFINET uses the Discovery and Configuration Protocol (DCP) for IP assignment.

**NOTE**
DCP is a standard protocol and is not part of the installation of TAS, for example. The protocol can also be used outside of PROFINET in IPC operating systems (Windows, Linux), for example, and is available in tool packages such as WinPcap, Npcap, Wireshark, etc.

When delivered, each field device has, among other things, a MAC address. The MAC address is sufficient to give the respective field device a unique name.

The address is assigned in two steps:

- Assignment of a unique plant specific name to the respective field device
- Assignment of the IP address from the IO-Controller before the system start-up based on the plant-specific (unique) name

#### PROFINET naming convention

The device name is checked for correct spelling during input. The following rules apply to the use of the device name in accordance with PROFINET specification V2.3.

- All device names must be unique.
- Maximum name size: 240 characters
  Allowed:
  - Lower case letters a…z
  - Numbers 0…9
  - Hyphen and dot
- The name may consist of several components separated by a period. A name component, i.e. a string between two dots, may be a maximum of 63 characters long.
- The device name must not start or end with a hyphen.
- The name must not begin with "port-xyz" (y…z = 0…9).
- The name must not have the form of an IP address (n.n.n.n, n = 0…999).
- Do not use special characters.
- Do not use capital letters.

### 7.2.2 Device model TBEN-S2-2COM-4DXP, slots and sub slots

The TBEN-S2-2COM-4DXP has 28 virtual slots for various device functions (COM port function, data buffer, DXP channels, etc.), diagnostic and status messages (COM diagnostics, MB server status, etc.) and the module status.

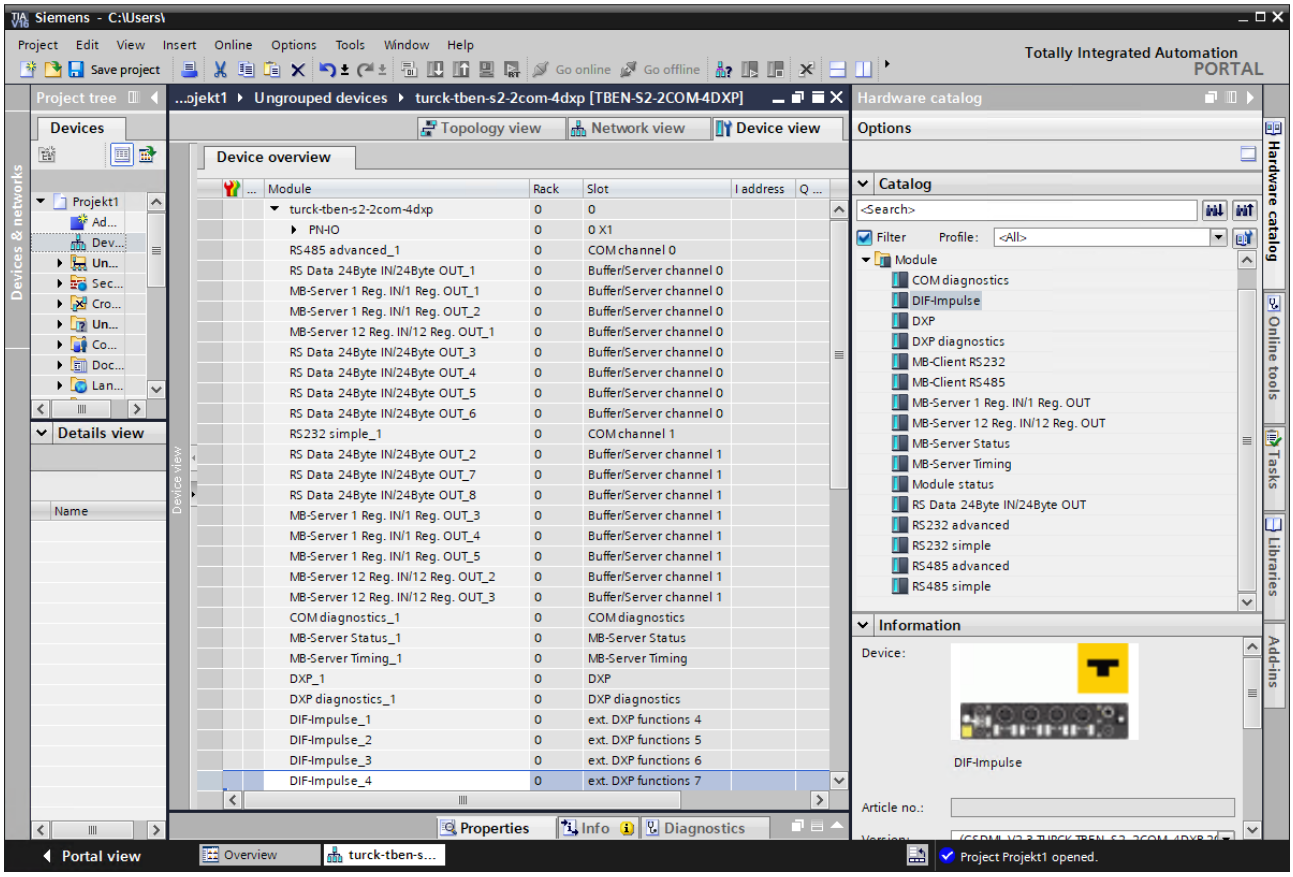



Fig. 38: TIA-Portal – allocation of the slots and sub slots

Besides Slot 0 (DAP) all other slots of the device contain only one sub slot. For this reason slots and sub slots are described as synonyms in the following.

| Slot no. | Name | Description | Pluggable sub modules |
|---|---|---|---|
| 0 | TBEN-S2-2COM-4DXP | Device interface to PROFINET IO, Device Access Point | ■ Device Access Point<br>■ Ethernet interface<br>■ Ethernet port 0<br>■ Ethernet port 1 |
| 1 | COM channel 0 | Function of the first COM port (COM0) | ■ RS232 simple<br>■ RS232 advanced<br>■ RS485 simple<br>■ RS485 advanced<br>■ MB Client RS232<br>■ MB Client RS485 |
| 2…9 | Buffer/server channel 0 | COM0: Configuration of the data buffers for input and output data or configuration of servers that are connected via Modbus RTU Modbus RTU. | ■ RS Data 24Byte IN/24Byte OUT<br>■ MB-Server 1 Reg. IN/1 Reg. OUT<br>■ MB-Server 12 Reg. IN/12 Reg. OUT |
| 10 | COM channel 1 | Function of the second COM port (COM1) | ■ RS232 simple<br>■ RS232 advanced<br>■ RS485 simple<br>■ RS485 advanced<br>■ MB Client RS232<br>■ MB Client RS485 |
| 11…18 | Buffer/server channel 1 | COM1: Configuration of the data buffers for input and output data or configuration of servers that are connected via Modbus RTU Modbus RTU. | ■ RS Data 24Byte IN/24Byte OUT<br>■ MB-Server 1 Reg. IN/1 Reg. OUT<br>■ MB-Server 12 Reg. IN/12 Reg. OUT |
| 19 | COM diagnostics | Cyclic diagnostic data of the COM ports | COM diagnostics |
| 20 | MB Server Status | Status data of connected Modbus RTU server (cyclic) | MB Server Status |
| 21 | MB Server Timing | Timing data of connected Modbus RTU server (cyclic) | MB Server Timing |
| 22 | DXP | Configuration of the four DXP channels | DXP |
| 23 | DXP diagnostics | Diagnostic data of the DXP channels (cyclic) | DXP diagnostics |
| 24…27 | Ext. DXP functions 4…7 | Configuration of input filter times and pulse stretching for the digital input channels 4…7. | DIP pulses |
| 28 | Module status | Module status messages (cyclic) | Module status |

## Sub module TBEN-S2-2COM-4DXP (Device Access Point)

The Device Access Point "TBEN-S2-2COM-4DXP" provides device's PROFINET interface. This module is always plugged in slot 0 and cannot be deleted.

- **PROFINET IO**

| Feature | Description |
|---|---|
| Conformance Class | B |
| Update time [ms] | 1…512 |
| Media Redundancy Protocol (MRP) | MRP client |
| Fast Startup (FSU) | < 500 ms |
| Topology detection via LLDP | Yes |

- **Parameters**

| Parameter name | Value | Meaning | Description |
|---|---|---|---|
| Output behavior at communication loss | **0** | **Set to 0** | The device switches the outputs to "0". No error information is sent. |
| | 1 | Hold current value | The device keeps the current data at the outputs. |
| Disable output power diagnosis | **0** | **No** | Monitoring of voltage V2 is activated. |
| | 1 | Yes | The sending of the diagnosis is deactivated. |
| Deactivate EtherNet/IP | **0** | **No** | Explicit disabling of the Ethernet protocols or the web server |
| | 1 | Yes | |
| Deactivate Modbus TCP | **0** | **No** | |
| | 1 | Yes | |
| Deactivate web server | **0** | **No** | |
| | 1 | Yes | |

- **Process data**
  This sub module has no process data.

## Sub module "RS232 simple" and "RS323 advanced"

The sub module "RS232 simple" and "RS323 advanced" can be plugged into slots 1 (COM 0) and 10 (COM 1). The sub modules switch the COM port to the RS232 mode, provide parameters for the configuration as well as control and status data. More detailed information about the transmit and receive sequence can be found under [▶ 151].

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].

The "RS232 simple" and "RS323 advanced" submodules differ only in the available parameters.

- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| COM – COM port status bits | %IB0 | USINT |
| Transmitter ready | %IX0.0 | BOOL |
| Receive complete | %IX0.1 | BOOL |
| Frame error | %IX0.2 | BOOL |
| Parity/format error | %IX0.3 | BOOL |
| Buffer overflow | %IX0.4 | BOOL |
| Timeout | %IX0.5 | BOOL |
| Invalid TX length | %IX0.6 | BOOL |
| Invalid RX length | %IX0.7 | BOOL |
| Reserved | %IB1 | USINT |
| Received frame length | %IB2 | USINT |

- **Process output data** [▶ 145]

| Process value | Offset | Data type |
|---|---|---|
| COM – COM – COM port control bits | %QB0 | USINT |
| Transmit | %QX0.0 | BOOL |
| Receive | %QX0.1 | BOOL |
| Reserved | %QB01 | USINT |
| Transmitter frame length | %QB2 | USINT |
| Reserved | %QB3 | USINT |
| Receiver frame length | %QB4 | USINT |

## Sub module "RS485 simple" and "RS485 advanced"

The sub module "RS485 simple" and "RS485 advanced" can be plugged into slots 1 (COM 0) and 10 (COM 1). The sub modules switch the COM port to the RS485 mode, provide parameters for the configuration as well as control and status data. More detailed information about the transmit and receive sequence can be found under [▶ 151].

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].

The "RS485 simple" and "RS485 advanced" submodules differ only in the available parameters.

- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| COM – COM port status bits | %IB0 | USINT |
| Transmitter ready | %IX0.0 | BOOL |
| Receive complete | %IX0.1 | BOOL |
| Frame error | %IX0.2 | BOOL |
| Parity/format error | %IX0.3 | BOOL |
| Buffer overflow | %IX0.4 | BOOL |
| Timeout | %IX0.5 | BOOL |
| Invalid TX length | %IX0.6 | BOOL |
| Invalid RX length | %IX0.7 | BOOL |
| Reserved | %IB1 | USINT |
| Received frame length | %IB2 | USINT |

- **Process output data** [▶ 145]

| Process value | Offset | Data type |
|---|---|---|
| COM – COM – COM port control bits | %QB0 | USINT |
| Transmit | %QX0.0 | BOOL |
| Receive | %QX0.1 | BOOL |
| Reserved | %QB01 | USINT |
| Transmitter frame length | %QB2 | USINT |
| Reserved | %QB3 | USINT |
| Receiver frame length | %QB4 | USINT |

## Sub module "RS Data 24Byte IN/24Byte OUT"

The sub module "RS Data 24Byte IN/24Byte OUT" can be plugged into slots 2…9 (COM 0) and slot 11…18 (COM 1). It is used to set up the transmit and receive buffer for the serial communication via RS232 or RS485 modularly in steps of 24 bytes. The maximum length for the transmit and receive buffer for one COM port is 8 × 24 bytes = 192 byte

This sub module can only be used for COM ports which are used as pure RS232 or RS485 interface.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Byte 0 | %IB0 | BYTE | First byte of the receive buffer block |
| … | … | … | … |
| Byte 23 | %IB23 | BYTE | Last byte of the receive buffer block |

- **Process output data** [▶ 145]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Byte 0 | %QB0 | BYTE | First byte of the transmit buffer block |
| … | … | … | … |
| Byte 23 | %QB23 | BYTE | Last byte of the transmit buffer block |

## Sub module "MB-Client RS232" and "MB-Client RS485"

The sub modules "MB-Client RS232" and "MB-Client RS485" can be plugged into slots 1 (COM 0) and 10 (COM 1). They switch the COM port to the RS232 mode and activate the Modbus RTU Client function for the COM port.

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| Reserved - not used for the Modbus RTU Client function | %IB0…%IB3 | USINT |
| MB-Server cycle time (*1 ms) | %IB4 | UINT |

## Sub module "MB server …Reg. IN/…Reg. OUT"

The sub modules "MB-Server 1 Reg. IN/1 Reg. OUT" ands "MB-Server 12 Reg. IN/12 Reg. OUT" can be plugged into slots 2…9 (COM 0) and slot 11…18 (COM 1). These sub modules are used to configure connected Modbus RTU servers and to exchange data with the connected servers. Eight Modbus RTU servers can be configured for each COM port.

These sub module can only be used for COM ports which are used as Modbus RTU Clients.

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].
- **Process input data** "MB-Server 1 Reg. IN/1 Reg. OUT" [▶ 140]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Input register 0 | %IW0 | UINT | Input register of the Modbus server |

- **Process output data** "MB-Server 1 Reg. IN/1 Reg. OUT" [▶ 145]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Output register 0 | %QW0 | UINT | Output register of the Modbus server |

- **Process input data** "MB-Server 12 Reg. IN/12 Reg. OUT" [▶ 140]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Input register 0 | %IW0 | UINT | First input register of the Modbus server |
| … | … | … | … |
| Input register 11 | %IW11 | UINT | Last input register of the Modbus server |

- **Process output data** "MB-Server 12 Reg. IN/12 Reg. OUT" [▶ 145]

| Process value | Offset | Data type | Description |
|---|---|---|---|
| Output register 0 | %QW0 | UINT | First output register of the Modbus server |
| … | … | … | … |
| Output register 11 | %QW11 | UINT | Last output register of the Modbus server |

## Sub module "COM diagnostics"

The sub module "COM diagnostics" can be plugged into slot 19. This sub module provides diagnostic data for the COM ports via cyclic input data.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| Diagnostics for COM0 | %IB0 | USINT |
| Hardware error | %IX0.0 | BOOL |
| Parameterization error | %IX0.1 | BOOL |
| Overcurrent supply VAUX1 | %IX0.7 | BOOL |
| Modbus diagnostics for COM0 | %IB1 | USINT |
| Error MB server 0 | %IX1.0 | BOOL |
| … | … | … |
| Error MB server 7 | %IX1.07 | BOOL |
| Diagnostics for COM1 | %IB2 | USINT |
| Hardware error | %IX2.0 | BOOL |
| Parameterization error | %IX2.1 | BOOL |
| Overcurrent supply VAUX1 | %IX2.7 | BOOL |
| Modbus diagnostics for COM1 | %IB3 | USINT |
| Error MB server 0 | %IX3.0 | BOOL |
| … | … | … |
| Error MB server 7 | %IX3.07 | BOOL |

## Sub module "MB server status"

The sub module Sub module "MB server status" can be plugged into slot 20. This sub module cyclically provides status data for the connected Modbus RTU servers.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| **COM0** | | |
| COM0 MB server status Ch0 | %IB0 | USINT |
| Error code bit 0 Ch0 | %IX0.0 | BOOL |
| Error code bit 1 Ch0 | %IX0.1 | BOOL |
| Error code bit 2 Ch0 | %IX0.2 | BOOL |
| Error code bit 3 Ch0 | %IX0.3 | BOOL |
| Read error Ch0 | %IX0.4 | BOOL |
| Write error Ch0 | %IX0.5 | BOOL |
| Parity/format error Ch0 | %IX0.6 | BOOL |
| MODBUS timeout Ch0 | %IX0.7 | BOOL |
| COM0 MB server status Ch0 | %IB1 | USINT |
| Valid read config. Ch0 Ch0 | %IX1.4 | BOOL |
| Valid write config. Ch0 Ch0 | %IX1.5 | BOOL |
| … | | |
| COM0 MB server status Ch7 | %IB14 | USINT |
| Error code bit 0 Ch7 | %IX14.0 | BOOL |
| Error code bit 1 Ch7 | %IX14.1 | BOOL |
| Error code bit 2 Ch7 | %IX14.2 | BOOL |
| Error code bit 3 Ch7 | %IX14.3 | BOOL |
| Read error Ch7 | %IX14.4 | BOOL |
| Write error Ch7 | %IX14.5 | BOOL |
| Parity/format error Ch7 | %IX14.6 | BOOL |
| MODBUS timeout Ch7 | %IX14.7 | BOOL |
| COM0 MB server status Ch7 | %IB15 | USINT |
| Valid read config. Ch7 | %IX15.4 | BOOL |
| Valid write config. Ch7 | %IX15.5 | BOOL |
| **COM1** | | |
| COM1 MB server status Ch0 | %IB16 | USINT |
| Error code bit 0 Ch0 | %IX16.0 | BOOL |
| Error code bit 1 Ch0 | %IX16.1 | BOOL |
| Error code bit 2 Ch0 | %IX16.2 | BOOL |
| Error code bit 3 Ch0 | %IX16.3 | BOOL |
| Read error Ch0 | %IX16.4 | BOOL |
| Write error Ch0 | %IX16.5 | BOOL |
| Parity/format error Ch0 | %IX16.6 | BOOL |
| MODBUS timeout Ch0 | %IX16.7 | BOOL |

| Process value | Offset | Data type |
|---|---|---|
| COM1 MB server status Ch0 | %IB17 | USINT |
| Valid read config. Ch0 | %IX17.4 | BOOL |
| Valid write config. Ch0 Ch0 | %IX17.5 | BOOL |
| … | | |
| COM1 MB server status Ch7 | %IB30 | USINT |
| Error code bit 0 Ch7 | %IX30.0 | BOOL |
| Error code bit 1 Ch7 | %IX30.1 | BOOL |
| Error code bit 2 Ch7 | %IX30.2 | BOOL |
| Error code bit 3 Ch7 | %IX30.3 | BOOL |
| Read error Ch7 | %IX30.4 | BOOL |
| Write error Ch7 | %IX30.5 | BOOL |
| Parity/format error Ch7 | %IX30.6 | BOOL |
| MODBUS timeout Ch7 | %IX30.7 | BOOL |
| COM1 MB server status Ch7 | %IB31 | USINT |
| Valid read config. Ch7 | %IX31.4 | BOOL |
| Valid write config. Ch7 | %IX31.5 | BOOL |

**NOTE**
The Modbus Exception Codes are defined in the description of "Modbus Application Protocol V1" at http://www.modbus.org

## Sub module "MB server timing"

The sub module "MB server timing" can be plugged into slot 21. This sub module cyclically provides timing data for the connected Modbus RTU servers.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| COM0 MB server timing (*1ms) Ch0 | %IW0 | UINT |
| … | … | … |
| COM0 MB server timing (*1ms) Ch7 | %IW7 | UINT |

## Sub module "DXP"

The sub module "DXP" can be plugged into slot 22. It provides parameters for the configuration as well as process data for the four digital channels (Ch4…Ch7) of the module. The DXP channels can be used as input or output without any configuration.

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| DXP | %IB0 | USINT |
| Input value Ch4 | %IX0.4 | BOOL |
| Input value Ch5 | %IX0.5 | BOOL |
| Input value Ch6 | %IX0.6 | BOOL |
| Input value Ch7 | %IX0.7 | BOOL |

- **Process output data** [▶ 145]

| Process value | Offset | Data type |
|---|---|---|
| DXP | %QB0 | USINT |
| Output value Ch4 | %QX0.4 | BOOL |
| Output value Ch5 | %QX0.5 | BOOL |
| Output value Ch6 | %QX0.6 | BOOL |
| Output value Ch7 | %QX0.7 | BOOL |

## Sub module "DXP diagnostics"

The sub module "DXP diagnostics" can be plugged into slot 23. The sub module cyclically provides diagnostic data for the four digital channels.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| DXP | %IB0 | USINT |
| Overcurrent VAUX2 Ch4/Ch5 | %IX0.2 | BOOL |
| Overcurrent VAUX2 Ch6/Ch7 | %IX0.3 | BOOL |
| DXP | %IB1 | USINT |
| Overcurrent output Ch4 | %IX1.4 | BOOL |
| Overcurrent output Ch5 | %IX1.5 | BOOL |
| Overcurrent output Ch6 | %IX1.6 | BOOL |
| Overcurrent output Ch7 | %IX1.7 | BOOL |

### Sub module "DIF pulses"

The sub module "DIF pulses" (DIF = Digital Input Filter) can be plugged into slots 24…27. It provides parameters for the configuration as well as process data for the four digital channels (Ch4…Ch7) of the module. One slot is assigned to each digital channel. Slot 24 is assigned to the digital channel 4 and slot 27 is assigned to the digital channel 7. This sub module is used to configure the filter times and the pulse stretching for the digital channels.

- **Parameters**
  The parameters are described in chapter "Parameterizing and configuring [▶ 133].
- **Process data**

This sub module has no process data. The configured parameters affect the input process values of the sub module "DXP" (slot 22).

### Sub module "Module status"

The sub module "Module status" can be plugged into slot 28. This sub module cyclically provides device status data.

- **Parameters**
  This sub module requires no configuration and has thus no parameters.
- **Process input data** [▶ 140]

| Process value | Offset | Data type |
|---|---|---|
| Module status – byte 0 | %IB0 | USINT |
| Undervoltage V1 | %IX0.1 | BOOL |
| Internal error | %IX0.2 | BOOL |
| I/O-ASSISTANT Force Mode active | %IX0.6 | BOOL |
| Module status – byte 1 | %IB1 | USINT |
| Module diagnostics pending | %IX1.0 | BOOL |
| Undervoltage V2 | %IX1.7 | BOOL |

### 7.2.3 FSU – Fast Start-Up (prioritized startup)

FSU enables a PLC to build up connections to PROFINET nodes in less than 500 ms after switching-on the network power supply (V1). The fast start-up is necessary for fast tool changing applications at robot arms for example in the automobile industry.

**NOTE**
For the correct cabling in FSU applications please observe the note in the chapter "Connecting the Device to Ethernet" [▶ 24].

#### Fast Start-Up (FSU) in TBEN

The device supports the prioritized start-up (FSU). This function is only guaranteed for digital channels.

#### Activating FSU

In order to enable FSU, the field bus nodes have to be configured respectively, for example in TIA-Portal (Siemens).

| | |
|---|---|
| Autonegotiation: | deactivated |
| Transmission medium/duplex: | set to a fixed value |

- During configuration, please observe that the neighboring devices do also support FSU and that the settings for the ports of neighboring devices are identical.
- Set "Transmission rate/duplex" to a fix value.
- Deactivate auto-negotiation

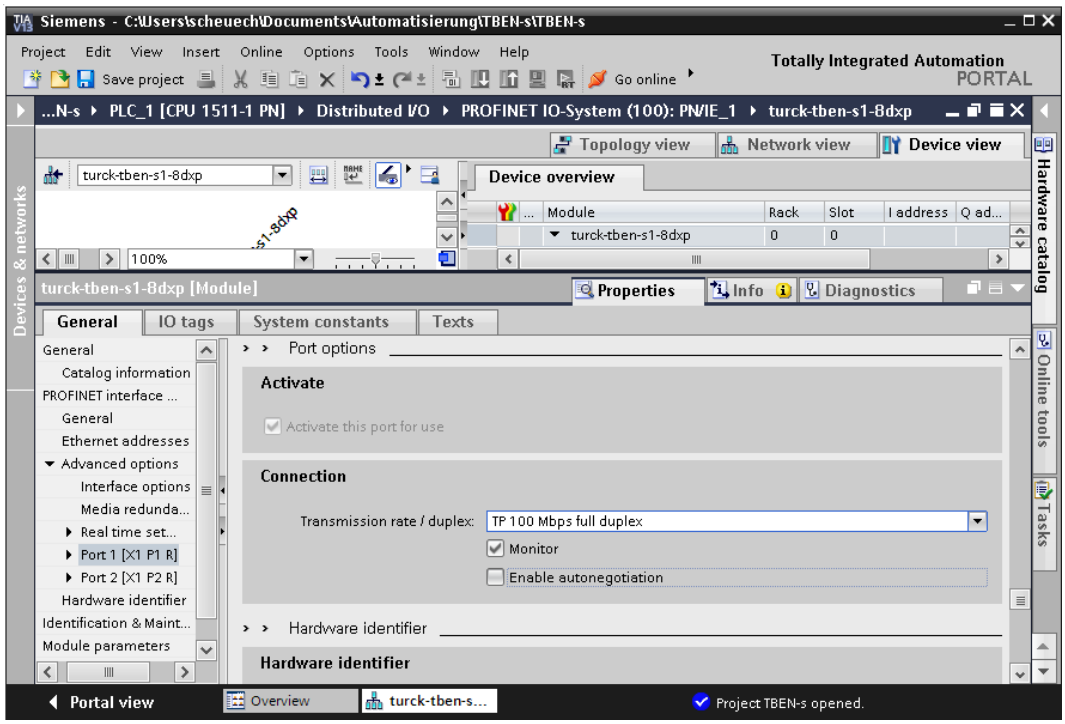



Fig. 39: TIA-Portal – port-settings for FSU

▶ Activate the prioritized start-up at the I/O device.

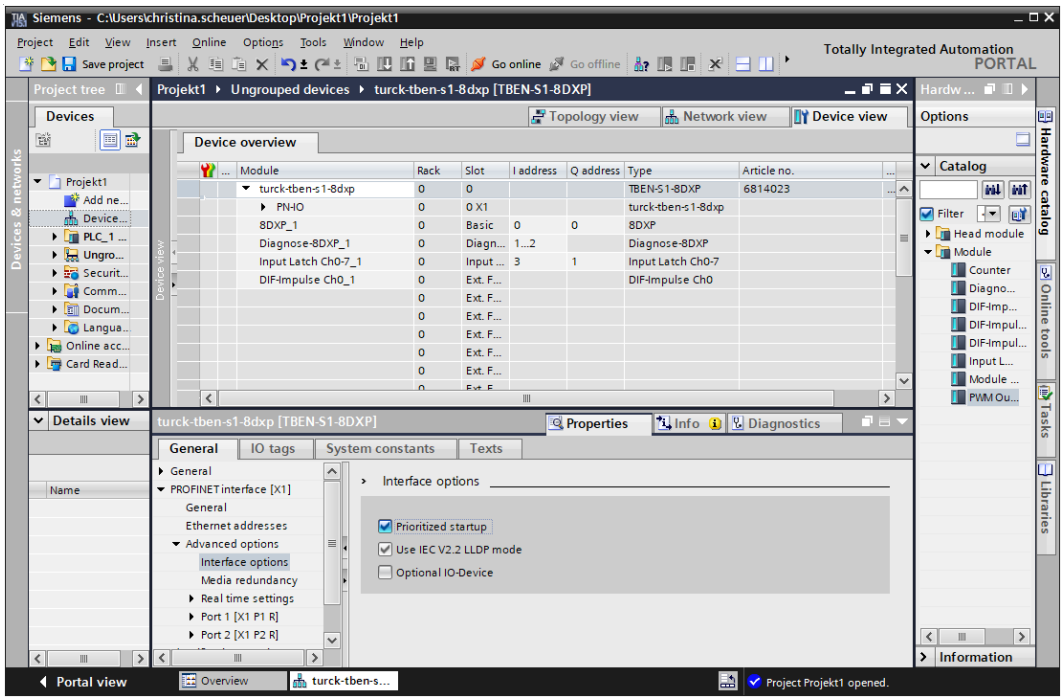



Fig. 40: TIA-Portal – prioritized start-up, activation at the I/O device

## 7.2.4 MRP (Media Redundancy Protocol)

The device supports MRP. MRP is a standardized protocol according to IEC 62439. It describes a mechanism for media redundancy in ring topologies. With MRP, a defective ring topology with up to 50 nodes is detected and reconfigured in the event of an error. With MRP a trouble-free switch-over is not possible.

A Media Redundancy Manager (MRM) checks the ring topology of a PROFINET network defined by the network configuration for functionality. All other network nodes are Media Redundancy Clients (MRC). In the error-free state, the MRM blocks normal network traffic on one of its ring ports, with the exception of the test telegrams. The physical ring structure thus becomes a line structure again at the logical level for normal network traffic. If a test telegram fails to appear, a network error has occurred. In this case, the MRM opens its blocked port and establishes a new functioning connection between all remaining devices in the form of a linear network topology.

The time between ring interruption and recovery of a redundant path is called reconfiguration time. For MRP, this is a maximum of 200 ms. Therefore, an application must be able to compensate for the 200 ms interruption. The reconfiguration time always depends on the Media Redundancy Manager (e.g. the PROFINET PLC) and the I/O cycle and watchdog times set here. For PROFINET, the response monitoring time must be selected accordingly > 200 ms.

It is not possible to use Fast Start-Up in an MRP network.

### 7.2.5 User data for acyclic services

The acyclic data exchange is by using via Record Data CRs (Communication Relation). Via these Record Data CRs the reading and writing of the following services is realized:

- Writing of AR data (AR = Application Relation)
- Writing of configuration data
- Reading and writing of device data
- Reading of diagnostic data
- Reading of I/O data
- Reading of Identification Data Objects (I&M functions)

#### Acyclic device user data

| Index | | Name | Data type | Access | Comment |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Module parameters | WORD | read/ write | Parameter data of the module (slot 0) |
| 2 | 0x02 | Module designation | STRING | read | Designation assigned to the module (slot 0) |
| 3 | 0x03 | Module revision | STRING | read | Firmware revision of the module |
| 4 | 0x04 | Vendor ID | WORD | read | Vendor ID for Turck |
| 5 | 0x05 | Module name | STRING | read | The device name assigned to the module |
| 6 | 0x06 | Module type | STRING | read | Device type of the module |
| 7 | 0x07 | Device ID | WORD | read | Device ID of the module |
| 8…23 | 0x08… 0x17 | reserved | - | - | - |
| 24 | 0x18 | Module diagnostics | WORD | read | Diagnostic data of the module (slot 0). |
| 25…31 | 0x19… 0x1F | reserved | - | - | - |
| 32 | 0x20 | Input list | ARRAY of BYTE | read | List of all module input channels |
| 33 | 0x21 | Output list | ARRAY of BYTE | read | List of all module output channels |
| 34 | 0x22 | Diag. list | ARRAY of BYTE | read | List of all I/O-channel diagnostics |
| 35 | 0x23 | Parameter list | ARRAY of BYTE | read | List of all I/O-channel parameters |
| 36… 28671 | 0x24… 0x6FFF | reserved | - | - | - |
| 28672 | 0x7000 | Module parameters | WORD | read/ write | Activate fieldbus protocol |
| 28673… 45039 | 0x7001 … 0xAFEF | reserved | - | - | - |
| 45040 | 0xAFF0 | I&M0-functions | | read | Identification & Maintaining |
| 45041 | 0xAFF1 | I&M1-functions | STRING[54] | read/ write | I&M Tag function and location |

| Index | | Name | Data type | Access | Comment |
|---|---|---|---|---|---|
| 45042 | 0xAFF2 | I&M2-functions | STRING[16] | read/ write | I&M Installation Date |
| 45043 | 0xAFF3 | I&M3-functions | STRING[54] | read/ write | I&M Description Text |
| 45044 | 0xAFF4 | I&M4-functions | STRING[54] | read/ write | I&M Signature |
| 45045… 45055 | 0xAFF5 … 0xAFFF | I&M5 to I&M15-functions | | - | Not supported |

## Acyclic I/O channel user data

| Index | | Name | Data Type | Access | Comment |
|---|---|---|---|---|---|
| **Dec.** | **Hex.** | | | | |
| 1 | 0x01 | Module parameters | specific | read/ write | Parameters of the module |
| 2 | 0x02 | Module type | ENUM UINT8 | read | Contains the module type |
| 3 | 0x03 | Module version | UINT8 | read | Firmware version of I/O channels |
| 4 | 0x04 | Module ID | DWORD | read | Ident number of the I/O |
| 5…9 | 0x05 … 0x09 | reserved | - | - | - |
| 10 | 0x0A | Controller Version | UINT8 array [8] | read | |
| 11…18 | 0x0B… 0x12 | reserved | - | - | - |
| 19 | 0x13 | Input data | specific | read | Input data of the respective I/O-channel |
| 20…22 | 0x14 … 0x16 | reserved | - | - | - |
| 23 | 0x17 | Output data | specific | read/ write | Output data of the respective I/O-channel |
| … | … | reserved | - | - | - |

## 7.3 Connecting the devices to a PROFINET controller with TIA Portal

The following example describes the connection of the devices to a Siemens PLC in PROFINET by means of the programming software SIMATIC STEP7 Professional V13 (TIA-Portal).

### Used hardware

The following hardware components are used in this example:

- Siemens PLC S7-1500
- TBEN-S… block module

### Used software

The following software tools are used in this example:

- SIMATIC STEP7 Professional V13 (TIA-Portal)
- GSDML file for TBEN-S… (can be downloaded for free as ZIP archive "TBEN-S_PROFINET.zip" under www.turck.com)

### Prerequisites

- The software is started.
- A new project has been created.
- The controller has been added to the project.

### 7.3.1 Installing the GSDML-file

The GSDML file is available for free at www.turck.com.

▶ Adding the GSDML file: Click **Options** → **Manage general station description files (GSD)**.

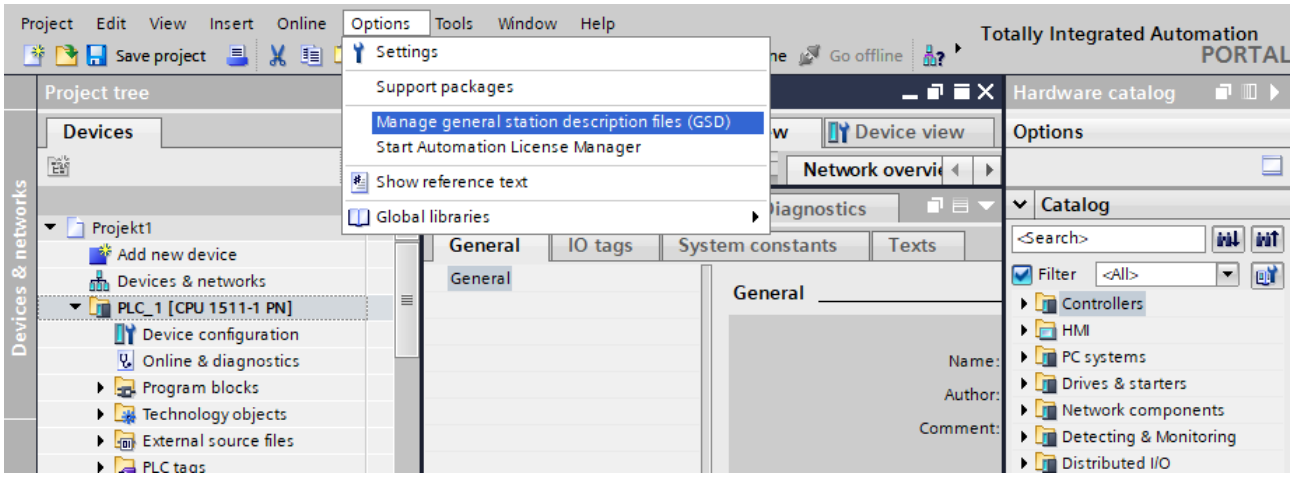



Fig. 41: Manage general station description files (GSD) in TIA Portal

▶ Installing the GSDML file: Define the source path for the GSDML-file and click **Install**.
⇨ The device is added to the hardware catalog.

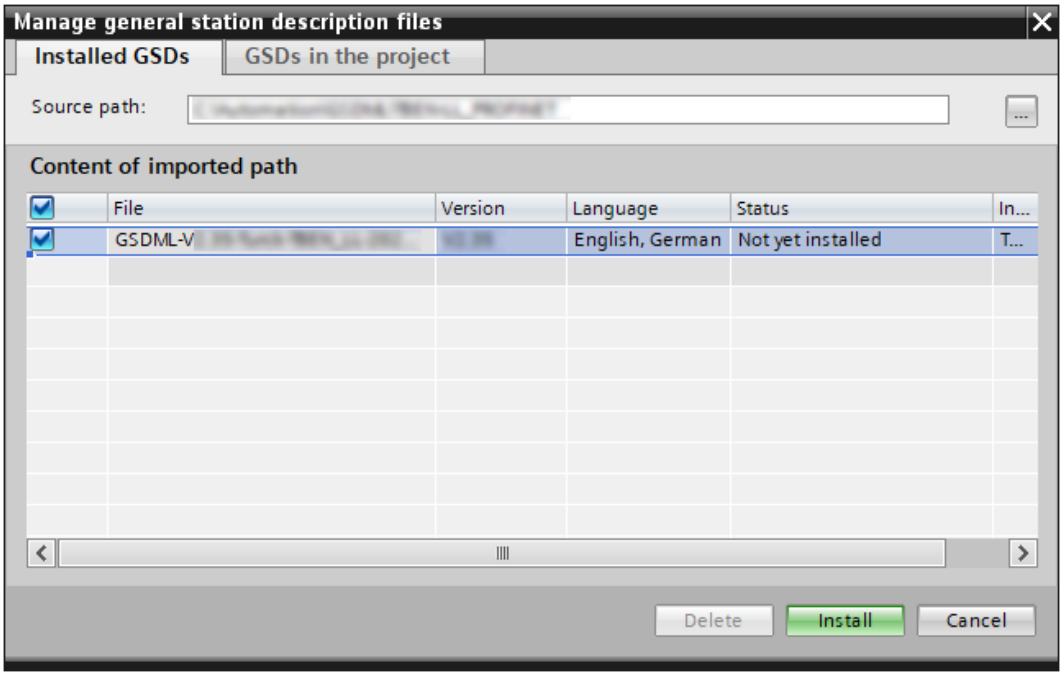



Fig. 42: TIA Portal – installing the GSDML file

### 7.3.2 Connecting the device to the PLC

- Select the TBEN device from the Hardware catalog and drag it into the hardware window.
- Connect the device to the PLC in the "Devices & networks" editor.

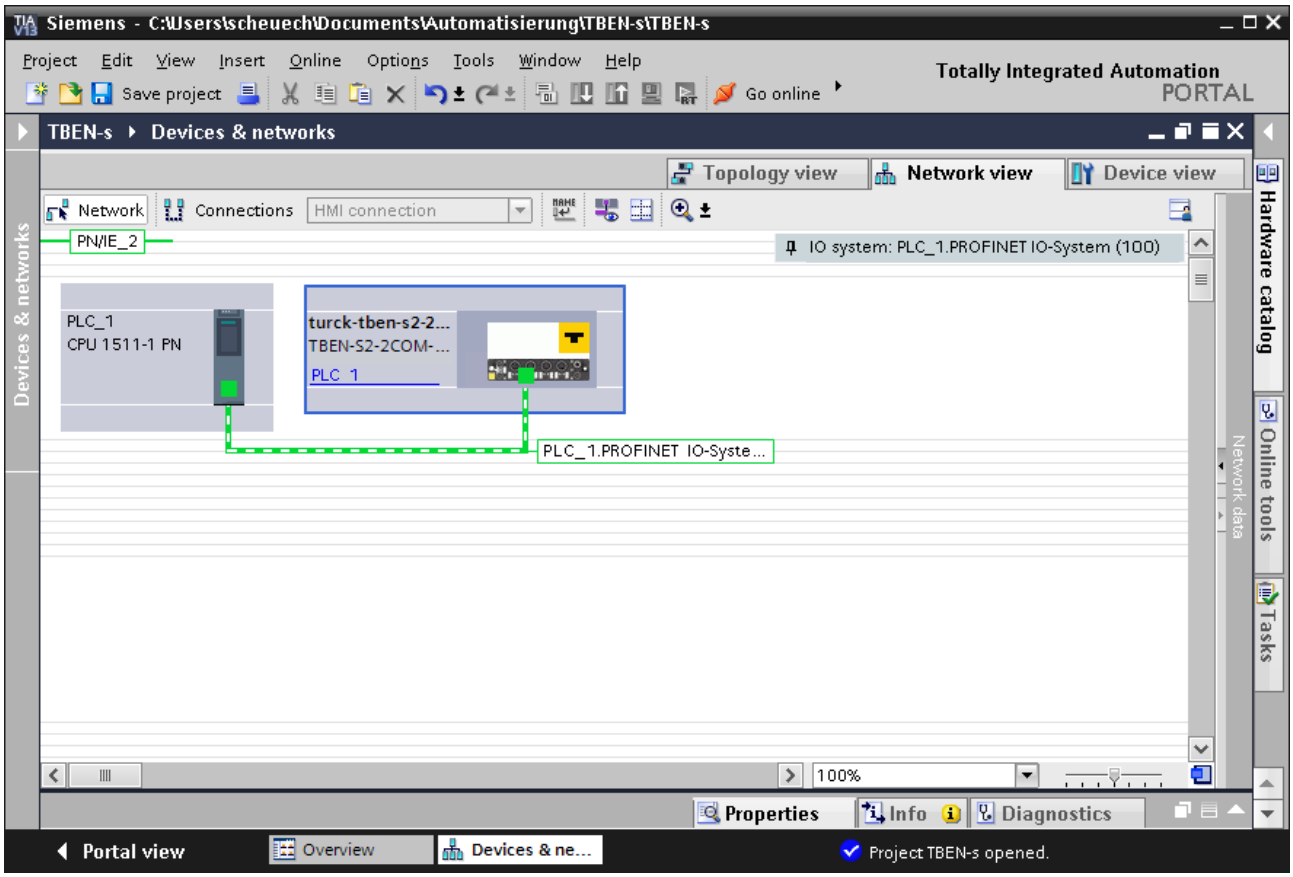



Fig. 43: Connecting the device to the PLC

### 7.3.3 Configuring device functions

- **Select Device view → Device overview.**
- Select functions as operation mode, diagnostics etc. from the hardware catalog and add them to the device slots via drag&drop.

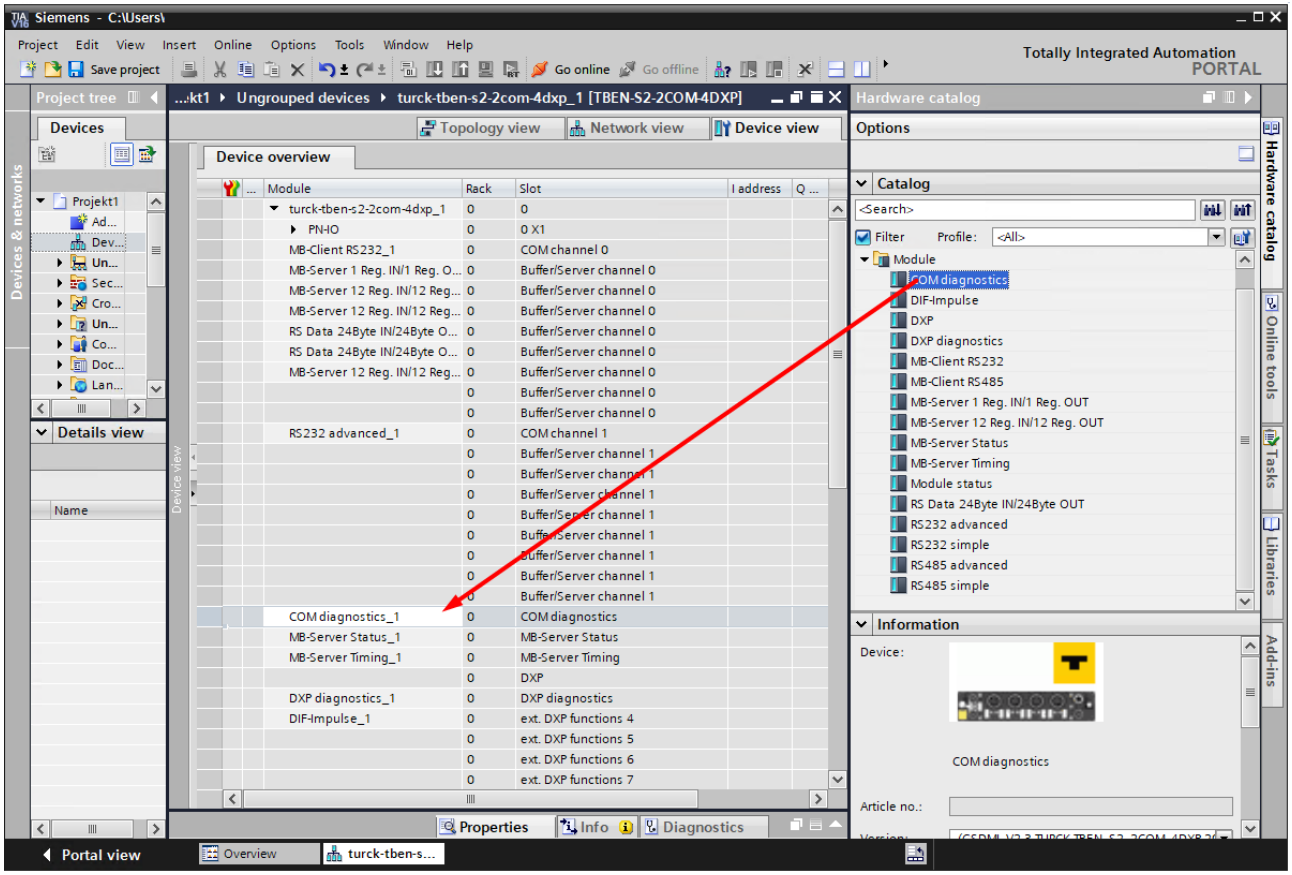



Fig. 44: Configuring the Device Slots

### 7.3.4 Setting the IP address in TIA Portal

- Select **Device view** → register **Properties** → **Ethernet addresses**.
- Assign the desired IP address.

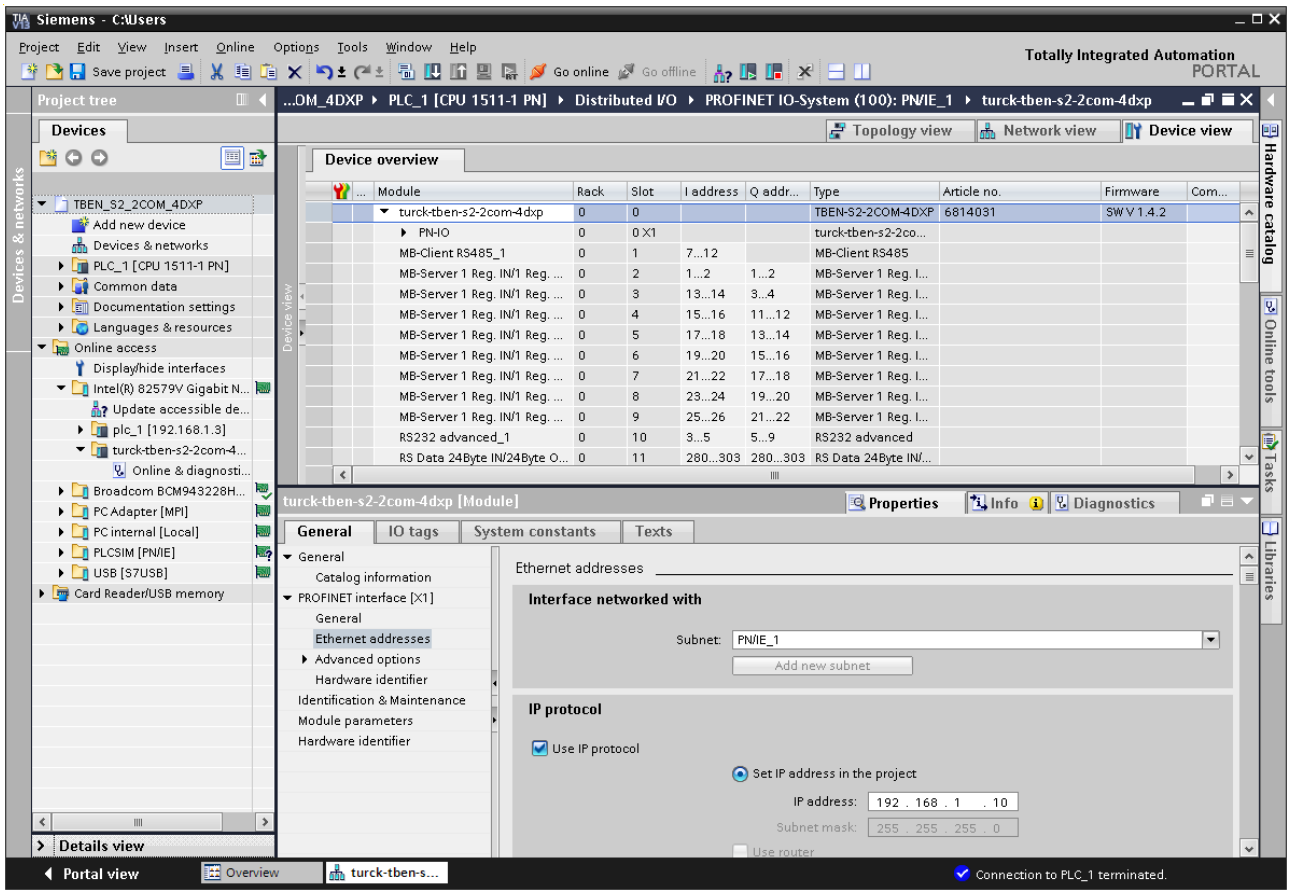



Fig. 45: Setting the IP address in TIA Portal

### 7.3.5 Connecting the device online with the controller

- ▶ Start the online mode (Go online).
- ⇨ The device has been successfully connected to the PLC.

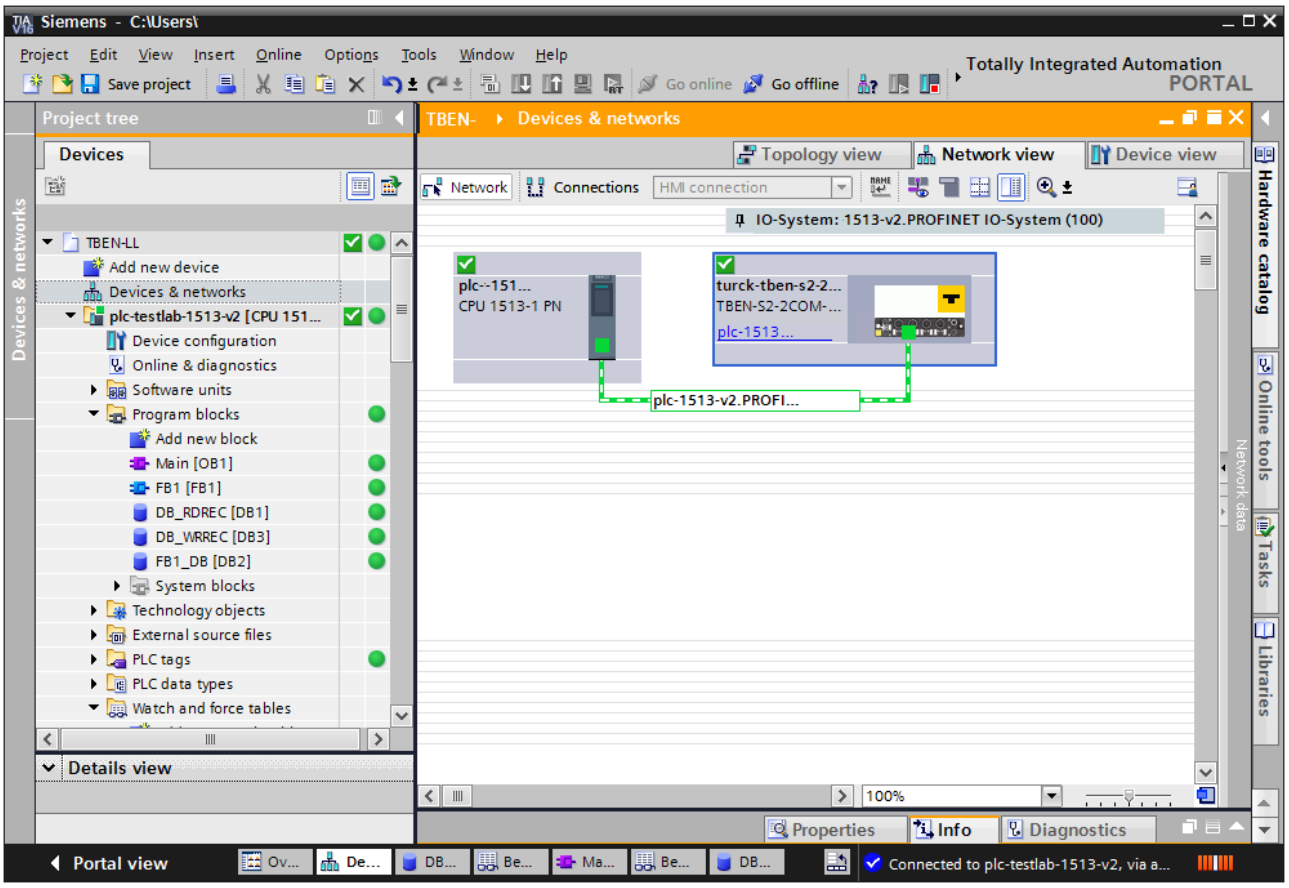



Fig. 46: TIA Portal: Online mode

### 7.3.6 PROFINET – Mapping

The PROFINET mapping corresponds to the data mapping described in "Process input data: overview" [▶ 140] and „Process output: overview" [▶ 145].

## 7.4 Commissioning the devices in Modbus TCP

### 7.4.1 Implemented Modbus functions

The devices support the following functions for accessing process data, parameters, diagnostics and other services.

| Function Code | |
|---|---|
| 3 | Read Holding Registers – reading multiple output registers |
| 4 | Read Input Registers – reading multiple input registers |
| 6 | Write Single Register – writing single output register |
| 16 | Write Multiple Registers – writing multiple output |
| 23 | Read/Write Multiple Registers – reading and writing multiple registers |

### 7.4.2 Modbus registers

| Address | Access | Meaning |
|---|---|---|
| 0x0000…<br>0x01FF | read only | Process data of inputs (identical to registers 0x8000…0x8FFF) |
| 0x0800…<br>0x09FF | read/write | Process data of the outputs (identical to registers 0x9000…<br>0x9FFF) |
| 0x1000…<br>0x100B | read only | Module identifier |
| 0x100C | read only | Module status |
| 0x1017 | read only | Register mapping revision (always 2, if not, mapping is incompatible with this description) |
| 0x1020 | read only | Watchdog, actual time in ms |
| 0x1120 | read/write | Watchdog, predefined time in ms (default: 500 ms) |
| 0x1130 | read/write | Modbus connection mode register |
| 0x1131 | read/write | Modbus Connection Timeout in s. (def.: 0 = never) |
| 0x113C…<br>0x113D | read/write | Modbus parameter restore (reset of parameters to default values) |
| 0x113E…<br>0x113F | read/write | Modbus parameter save (permanent storing of parameters) |
| 0x1140 | read/write | Deactivate protocol<br>Deactivates explicitly the selected Ethernet protocol:<br>■ Bit 0 = deactivate EtherNet/IP<br>■ Bit 1 = deactivate Modbus TCP<br>■ Bit 2 = deactivate PROFINET<br>■ Bit 15 = deactivate web server |
| 0x1141 | read/write | Active protocol<br>■ Bit 0 = EtherNet/IP active<br>■ Bit 1 = Modbus TCP active<br>■ Bit 2 = PROFINET active<br>■ Bit 15 = Web server active |
| 0x2400 | read only | V1 in mV: 0 at < 18 V |
| 0x2401 | read only | V2 in mV: 0 at < 18 V |
| 0x8000…<br>0x8FFF | read only | Process data of the inputs (identical to registers 0x0000…0x01FF) |

| Address | Access | Meaning |
|---|---|---|
| 0x9000…<br>0x9FFF | read/write | Process data of the outputs (identical to registers 0x0800…<br>0x09FF) |
| 0xA000…<br>0xAFFF | read only | Diagnostics |
| 0xB000…<br>0xBFFF | read/write | Parameters |

The following table shows the register mapping for the different Modbus addressing methods

| Description | Hex | Decimal | 5 digit | Modicon |
|---|---|---|---|---|
| Inputs | 0x0000…<br>0x01FF | 0…511 | 40001…<br>40512 | 400001…<br>400512 |
| Outputs | 0x0800…<br>0x09FF | 2048…<br>2549 | 42049…<br>42560 | 402049…<br>402560 |
| Module identifier | 0x1000…<br>0x1006 | 4096…<br>4102 | 44097…<br>44103 | 404097…<br>404103 |
| Module status | 0x100C | 4108 | 44109 | 404109 |
| Watchdog, actual time | 0x1020 | 4128 | 44129 | 404129 |
| Watchdog, predefined time | 0x1120 | 4384 | 44385 | 404385 |
| Modbus connection mode | 0x1130 | 4400 | 44401 | 404401 |
| Modbus connection timeout in s. | 0x1131 | 4401 | 44402 | 404402 |
| Modbus parameter restore | 0x113C…<br>0x113D | 4412…<br>4413 | 44413…<br>44414 | 404413…<br>404414 |
| Modbus parameter save | 0x113E…<br>0x113F | 4414…<br>4415 | 44415…<br>44416 | 404415…<br>404416 |
| Deactivate protocol | 0x1140 | 4416 | 44417 | 404417 |
| Active protocol | 0x1141 | 4417 | 44418 | 404418 |
| V1 in mV | 0x2400 | 9216 | 49217 | 409217 |
| V2 in mV | 0x2401 | 9217 | 49218 | 409218 |
| Process data inputs | 0x8000,<br>0x8001 | 32768,<br>32769 | - | 432769,<br>432770 |
| Process data outputs | 0x9000,<br>0x9001 | 36864,<br>36865 | - | 436865,<br>436866 |
| Diagnostics | 0xA000,<br>0xA001 | 40960,<br>40961 | - | 440961,<br>440962 |
| Parameters | 0xB000,<br>0xB001 | 45056,<br>45057 | - | 445057,<br>445058 |

### Register 0x1130: Modbus connection mode

This register defines the behavior of the Modbus connections.

| Bit | Designation | Value | Meaning |
|---|---|---|---|
| 0 | MB_OnlyOneWrite Permission | **0** | All Modbus connections receive the write authorization. |
| | | 1 | Only one Modbus connection can receive the write permission. A write permission is opened until a disconnect. After the disconnect the next connection which requests a write access receives the write authorization. |
| 1 | MB_ImmediateWrite Permission | **0** | With the first write access, a write authorization for the respective Modbus connection is requested. If this request fails, an exception response with exception-code 0x01 is generated. If the request is accepted, the write access is executed and the write authorization remains active until the connection is closed. |
| | | 1 | The write authorization for the respective Modbus connection is already opened during the connection establishment. The first Modbus connection thus receives the write authorization, all following connections do not (only if bit 0 = 1). |
| 2…15 | Reserved | - | - |

### Register 0x1131: Modbus connection timeout

This register defines after which time of inactivity a Modbus connection is closed through a disconnect.

Value range: 0…65535 s

default: 0 s = never (Modbus connection will never be closed)

**Behavior of the BUS LED**

If Modbus is the active protocol in case of a connection timeout and no further Modbus connections exist, the BUS LED behaves as follows:

| Connection timeout | BUS LED |
|---|---|
| Timeout | Green flashing |

### Register 0x113C and 0x113D: Restore Modbus connection parameters

Registers 0x113C and 0x113D serve for resetting the parameter-register 0x1120 and 0x1130 to 0x113B to the default settings. The service resets the parameters without saving them.

Procedure:

- ▶ Write 0x6C6F to register 0×113C.
- ▶ To activate the reset of the registers, write 0x6164 (”load”) within 30 seconds in register 0x113D. Both registers can also be written with one single request using the function codes FC16 and FC23.
- ⇨ The parameters are reset to default values.
- ▶ Save changes via a subsequent Save service.

### Register 0x113E and 0x113F: Save Modbus connection parameters

Registers 0x113E and 0x113F are used for the non-volatile saving of parameters in registers 0x1120 and 0x1130 to 0x113B.

Procedure:

- ▶ Write 0x7361 to register 0×113E.
- ▶ Write 0x7665 (”save”) within 30 seconds in register 0x113F to activate the reset of the registers. Both registers can also be written with one single request using the function codes FC16 and FC23.
- ⇨ The parameters are saved.

### 7.4.3 Register Mapping

| Register no. | Bit no. | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Input data (packed)** [▶ 140] | | | | | | | | | | | | | | | | |
| 0x0000…0x0002 | Status messages COM | | | | | | | | | | | | | | | |
| 0x0003…0x0062 | Receive data COM0 | | | | | | | | | | | | | | | |
| 0x0063…0x0065 | Status messages COM1 | | | | | | | | | | | | | | | |
| 0x0066…0x00C5 | Receive data COM1 | | | | | | | | | | | | | | | |
| 0x00C6 | COM channel diagnostics COM0 | | | | | | | | | | | | | | | |
| 0x00C7 | COM channel diagnostics COM1 | | | | | | | | | | | | | | | |
| 0x00C8…0x00D7 | Status messages Modus server, (depending on the parameterization, only valid for Modbus client mode) | | | | | | | | | | | | | | | |
| 0x00D8…0x00E7 | Modbus server timing | | | | | | | | | | | | | | | |
| 0x00E8 | DXP channels | | | | | | | | | | | | | | | |
| 0x00E9 | Overcurrent diagnostics | | | | | | | | | | | | | | | |
| 0x00EA | Module status (status word) | | | | | | | | | | | | | | | |
| **Ouptut data (packed)** [▶ 145] | | | | | | | | | | | | | | | | |
| 0x0800…0x0802 | Control data COM0 | | | | | | | | | | | | | | | |
| 0x0803…0x0862 | Transmit data COM0 | | | | | | | | | | | | | | | |
| 0x0863…0x0865 | Control data COM1 | | | | | | | | | | | | | | | |
| 0x0866…0x08C5 | Transmit data COM1 | | | | | | | | | | | | | | | |
| 0x08C6 | DXP channels | | | | | | | | | | | | | | | |
| **Status word** [▶ 155] | | | | | | | | | | | | | | | | |
| 0x100C | Module status | | | | | | | | | | | | | | | |
| **Input registers** [▶ 140] | | | | | | | | | | | | | | | | |
| 0x8000…0x80EA | S. input data (packed) register 0x0000…0x00EA | | | | | | | | | | | | | | | |
| **Output registers** [▶ 145] | | | | | | | | | | | | | | | | |
| 0x9000…0x90C6 | S. output data data (packed) register 0x0800…0x08C6 | | | | | | | | | | | | | | | |
| **Diagnostics** [▶ 156] | | | | | | | | | | | | | | | | |
| 0xA000 | COM channel diagnostics COM0 | | | | | | | | | | | | | | | |
| 0xA001 | COM channel diagnostics COM1 | | | | | | | | | | | | | | | |
| 0xA002 | DXP diagnostics | | | | | | | | | | | | | | | |

| Register no. | Bit no. | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Parameters** [▶ 133] | | | | | | | | | | | | | | | | |
| 0xB000…<br>0xB005 | Parameters COM0 | | | | | | | | | | | | | | | |
| 0xB006…<br>0xB025 | SCBs (Server Configuration Block) COM0 | | | | | | | | | | | | | | | |
| 0xB026…<br>0xB02B | Parameters COM1 | | | | | | | | | | | | | | | |
| 0xB02C…<br>0xB04B | SCBs (Server Configuration Block) COM1 | | | | | | | | | | | | | | | |
| 0xB04C…<br>0xB050 | DXP channels | | | | | | | | | | | | | | | |

### 7.4.4 Error behavior (watchdog)

#### Behavior of outputs

In case of a failure of the Modbus communication, the outputs' behavior is as follows, depending on the defined time for the Watchdog (register 0x1120):

| Watchdog | Behavior of outputs |
|---|---|
| 0 ms | All outputs maintain the actual value in case of an error |
| > 0 ms<br>(default = 500 ms) | Outputs switch to 0 after the watchdog time has expired (setting in register 0x1120). |

**NOTE**
Setting the outputs to predefined substitute values is not possible in Modbus TCP. Eventually parameterized substitute values will not be used.

#### Behavior of the BUS LED

When the watchdog triggers, the BUS LED lights up red.

#### Behavior of the device in case of loss of Modbus communication

If Modbus is the active protocol and all Modbus connections are closed, the watchdog switches all outputs to "0" after the watchdog time has expired, unless another protocol (PROFINET, EtherNet/IP) has been activated in the meantime.

## 7.5 Connecting devices to a Modbus Client with CODESYS

### Naming convention

Turck uses the terms "Modbus client" and "Modbus server" according to Modbus Organization. The following description uses the terms "Modbus TCP Master" (client) and "Modbus TCP Slave" (server) only because of the naming in CODESYS.

### Used hardware

The following hardware components are used in this example:

- TX715-P3CV01 (IP address: 192.168.145.72)
- Block module TBEN-S2-2COM-4DXP (IP address: 192.168.145.200)

### Used software

The following software tools are used in this example:

- CODESYS 3.5.18 (can be downloaded for free under **www.turck.com)**

### 7.5.1 Connecting the device to the PLC

The following components have to be added to CODESYS first, in order to connect the device to the PLC.

- Ethernet adapter
- Modbus TCP client (in CODESYS: Modbus TCP Master)
- Modbus TCP server (in CODESYS: Modbus TCP Slave)

## Adding the Ethernet Adapter

- ▶ Right-click **Device** in the project tree **TX715-P3CV01**.
- ▶ Select **Add Device**.
- ▶ Select **Ethernet Adapter**.
- ▶ Click **Insert device.**
- ⇨ The Ethernet Adapter is added to the project tree as **Ethernet (Ethernet)**.

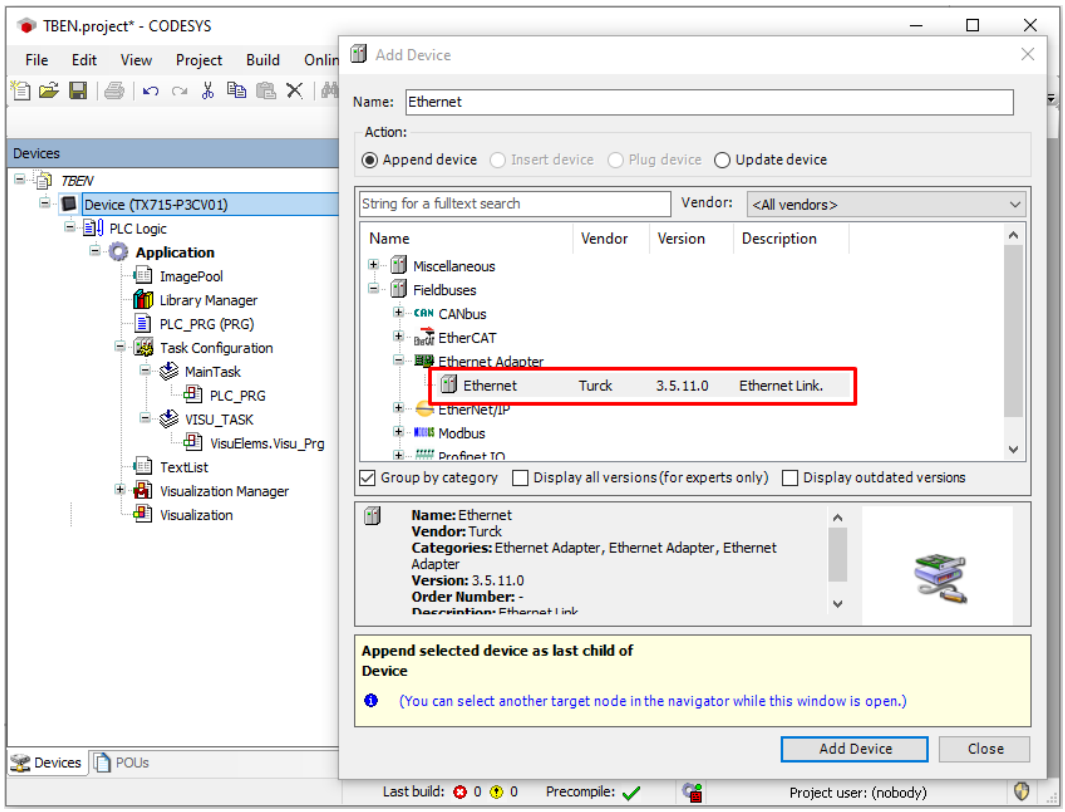



Fig. 47: Adding the Ethernet Adapter

## Adding the Modbus TCP Master

- ▶ Right-click the **Ethernet (Ethernet)** in the project tree.
- ▶ Select **Add Device**.
- ▶ Double-click **Modbus TCP Master**.
- ⇨ The **Modbus_TCP_Master** is added to the project tree.

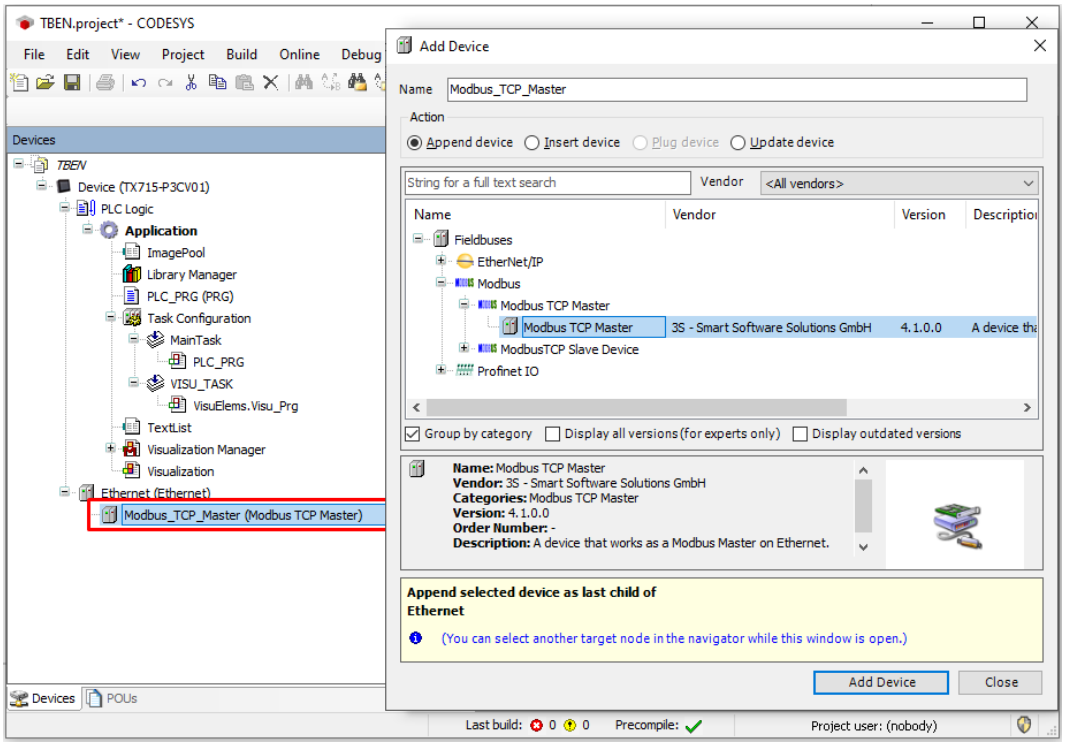



Fig. 48: Adding the Modbus TCP Master

## Adding the Modbus TCP Server (Slave)

- ▶ Right-click the **Modbus TCP Master** in the project tree.
- ▶ Select **Add Device**.
- ▶ Double-click **Modbus TCP Slave**.
- ⇨ The **Modbus_TCP_Slave** is added to the project tree.

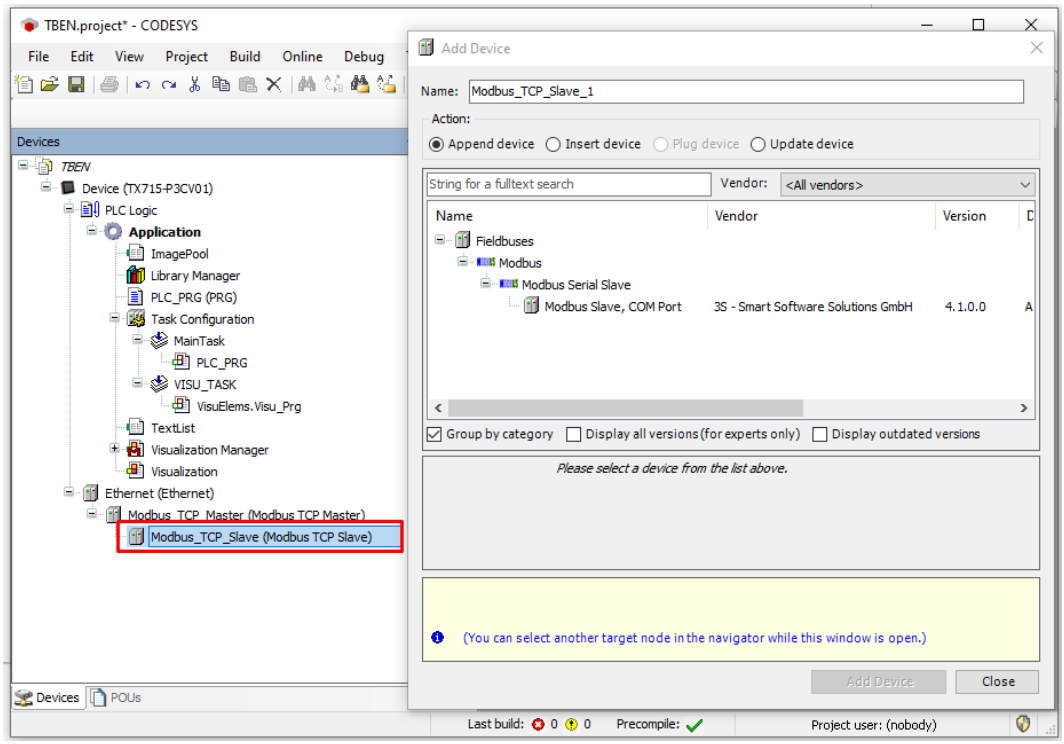



Fig. 49: Adding the Modbus TCP Slave

### 7.5.2 Configuring the Network Interface

- Click **Device → Scan network**.
- Select Modbus TCP Master (here: TX715-P3CV01) and confirm with OK.

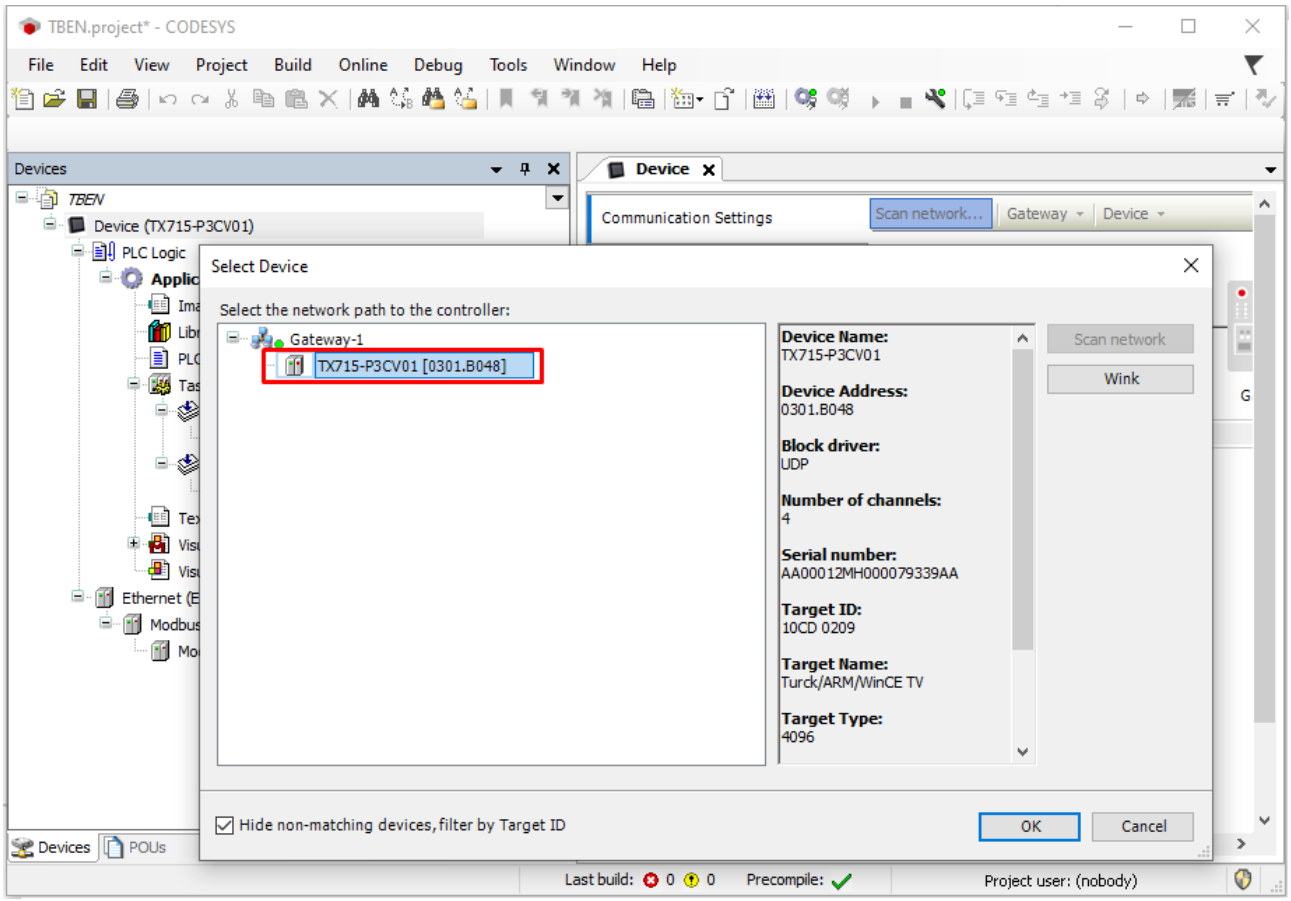



Fig. 50: Configuring the network interface

- Double-click **Ethernet**.
- Open the dialog box **Network Adapter** by clicking the **Browse…** button in the register tab **General**.
- Select the interface TX715-P3CV01 (here: 192.168.145.72)

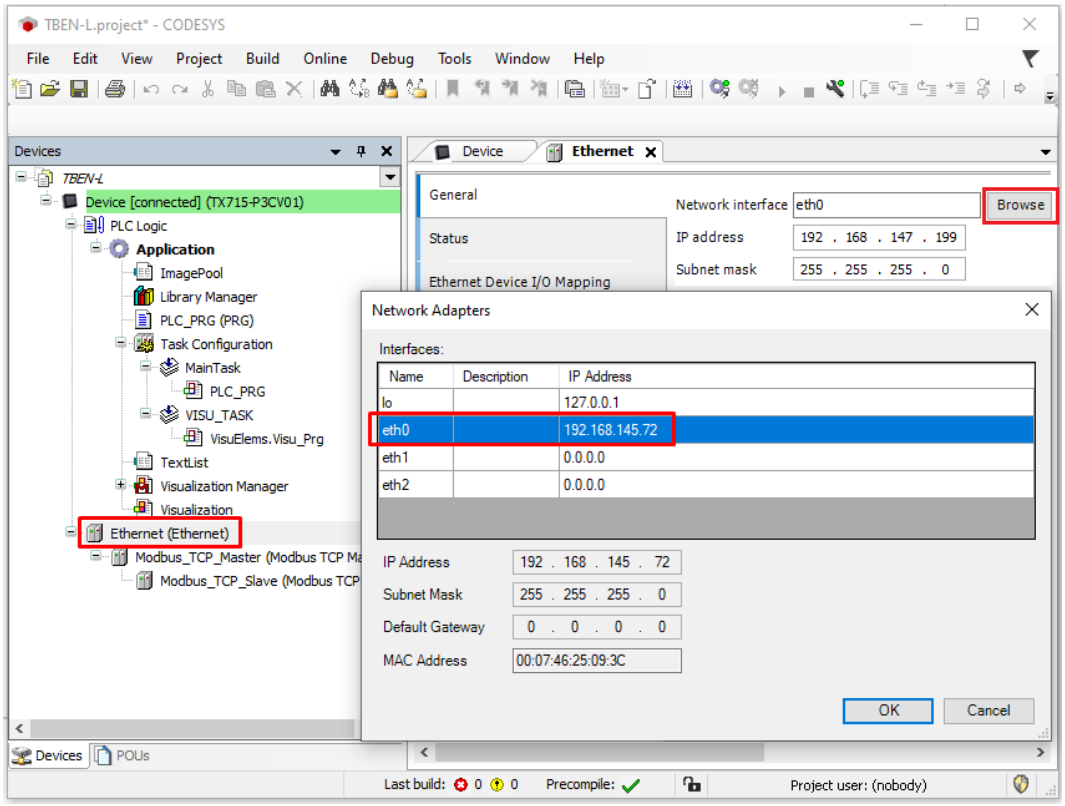



Fig. 51: Selecting the interface

### 7.5.3 Modbus TCP Server (Slave): setting the IP address

- Double click **Modbus TCP Server (Slave)**.
- Enter the **slave IP address** in the **General** register tab (here: 192.168.145.200).

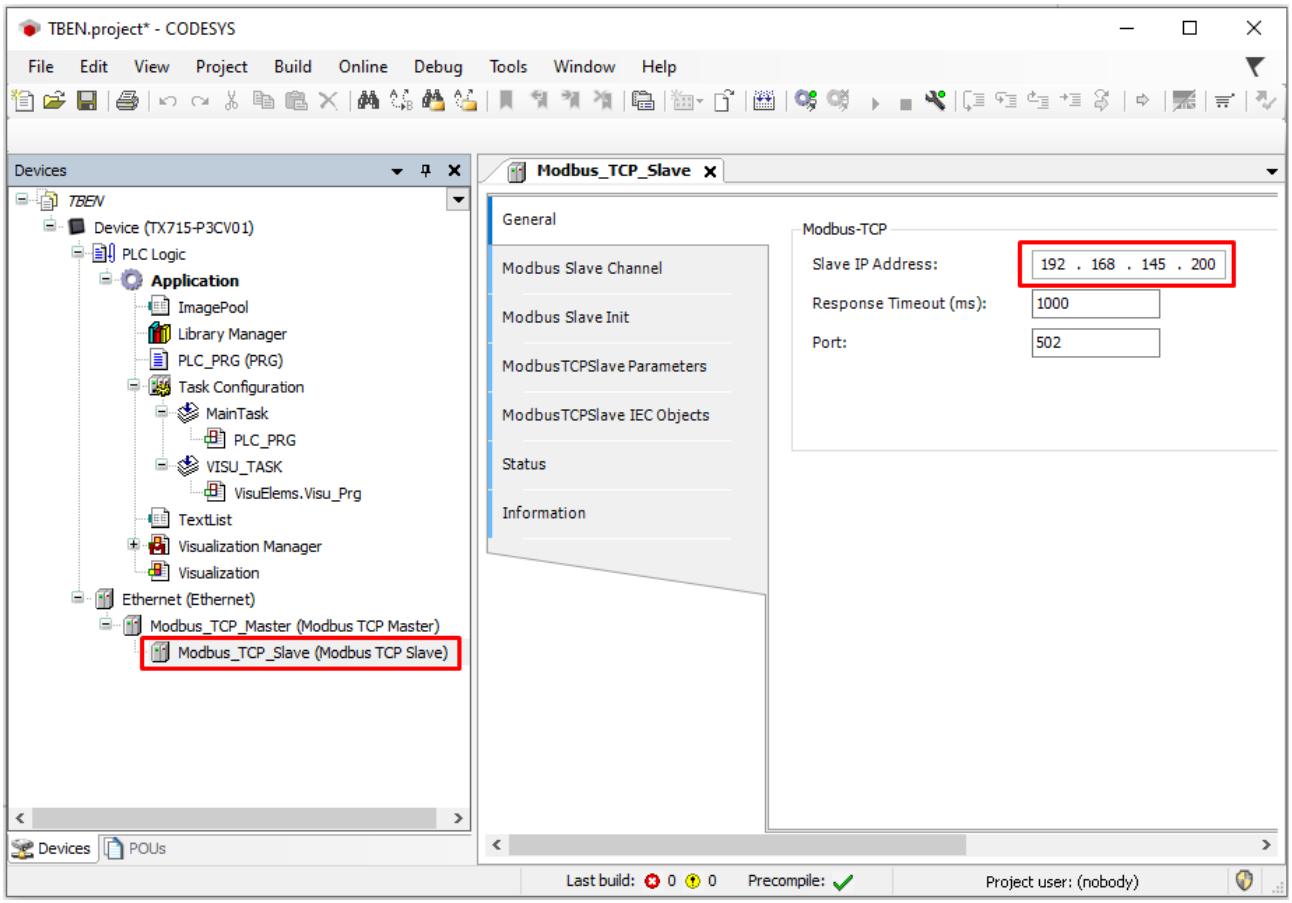



Fig. 52: Modbus TCP Slave: Setting the IP address

## 7.5.4 Defining modbus channels

### Defining channel 0 (input data)

- Double click **Modbus TCP Slave**.
- In the register tab select **Modbus Slave Channel** → **Add Channel**.
- Enter the following values:
  Channel name
  Access type: Read Input Registers
  Offset: 0x0000
  Length: 1 register
- Confirm with OK.

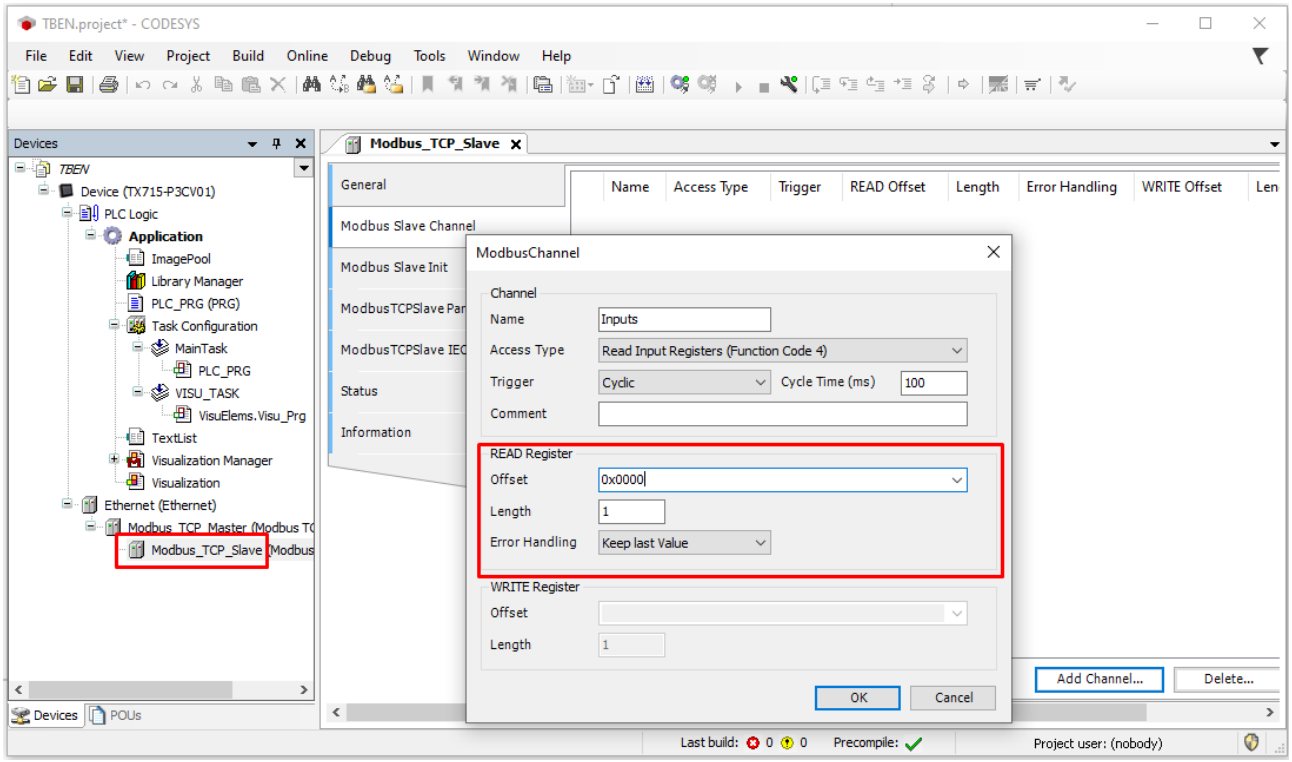



Fig. 53: Defining the input register

### Defining channel 1 (output data)

- Double click **Modbus TCP Slave**.
- In the register tab select **Modbus Slave Channel** → **Add Channel**.
- Enter the following values:
  Channel name
  Access type Write Single Register
  Offset: 0x0800
  Length: 1 register
- Confirm with OK.

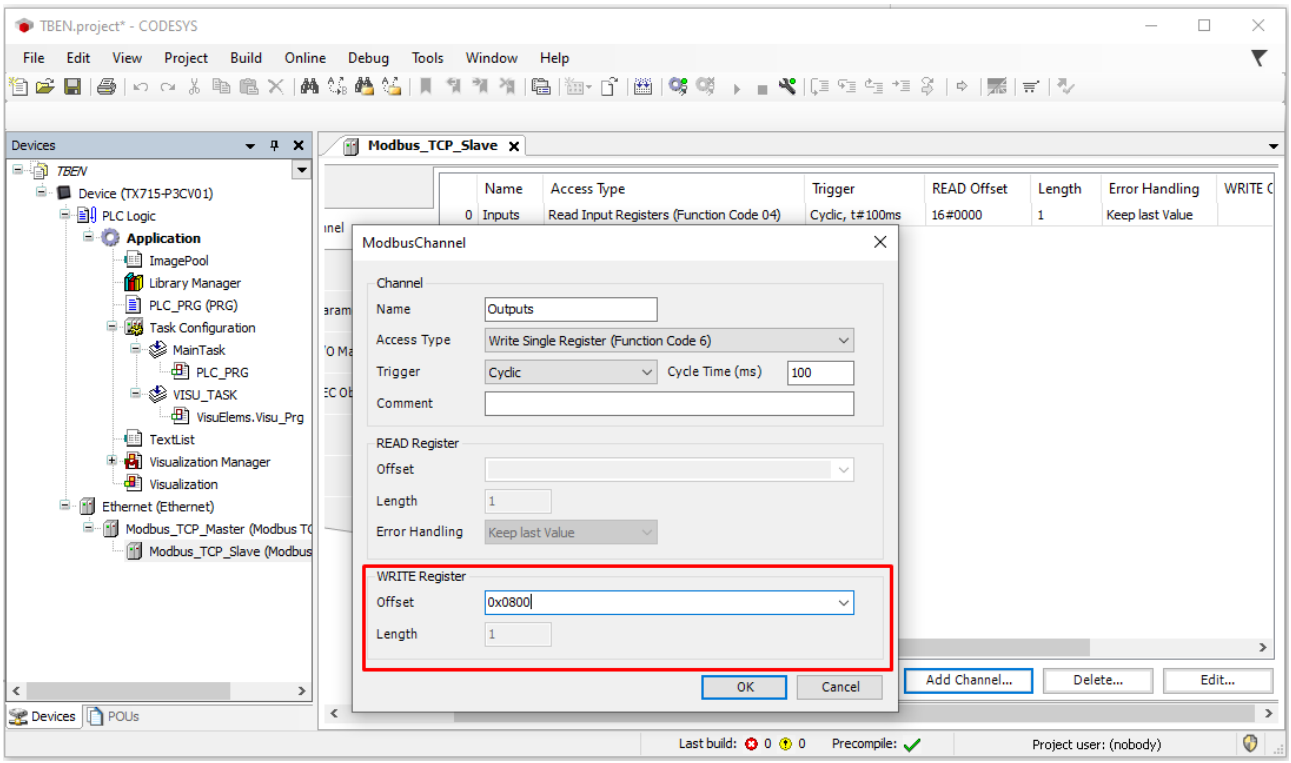



Fig. 54: Defining the output data register

### 7.5.5 Going online with the PLC

- Select the device.
- Click **Online → Login**.

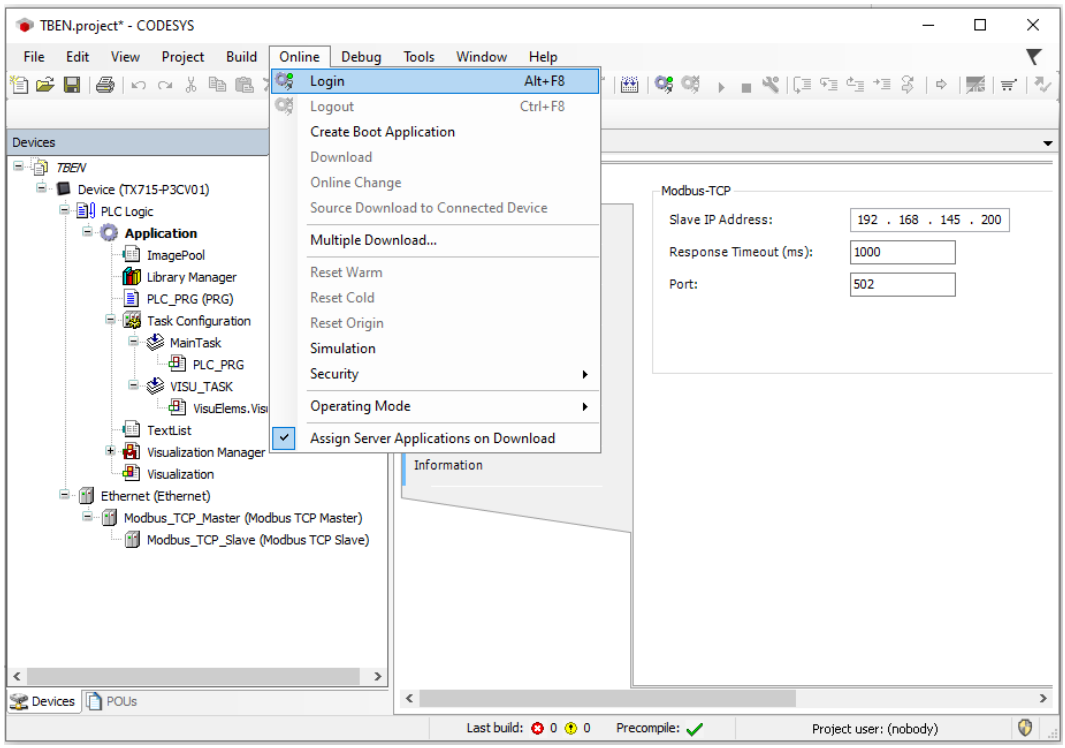



Fig. 55: Login

▶ Download the application to the PLC and start it via **Debug** → **Start**.

⇨ The Modbus TCP communication is setup.

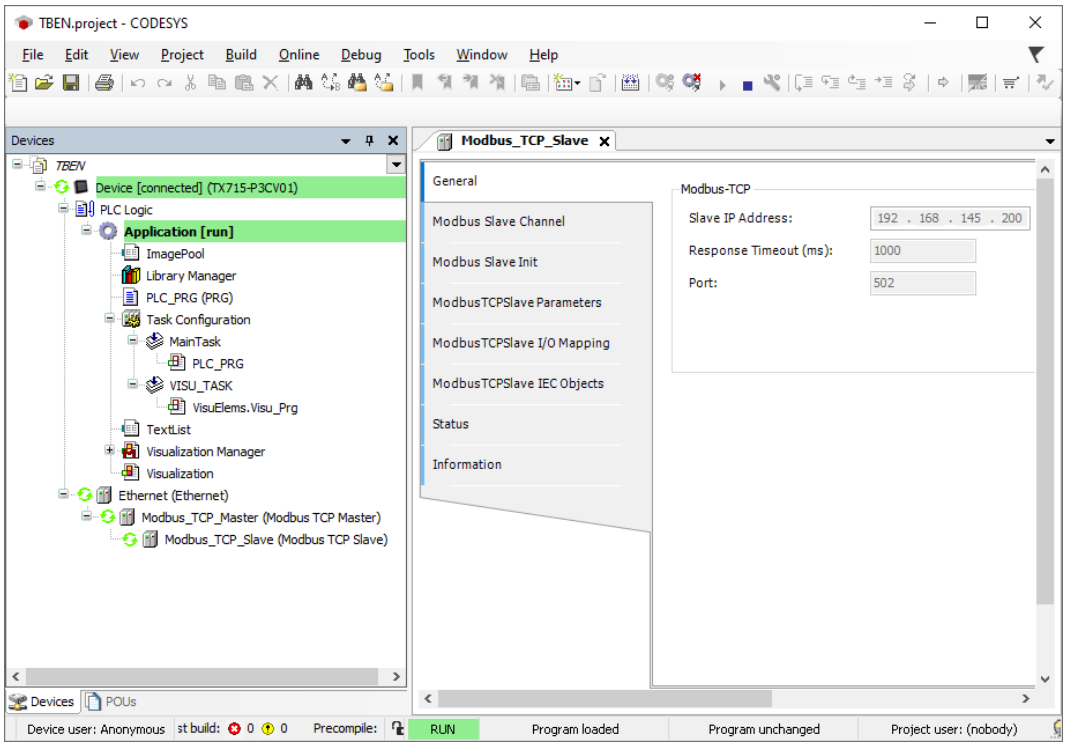



Fig. 56: Modbus TCP communication

### 7.5.6 Reading process data

The process data can be interpreted by means of the mapping (Registermapping) if the device is connected to the PLC.

The process data can be interpreted by means of the mapping ( [▶ 59]) if the device is connected to the PLC.

- ▶ Double click **Modbus TCP Slave**.
- ▶ Click onto register tab **Modbus TCP Slave I/O Mapping**.
- ▶ Set the function **Always update variables** to **Enabled 1 (…)**.
- ⇨ The process data are displayed.

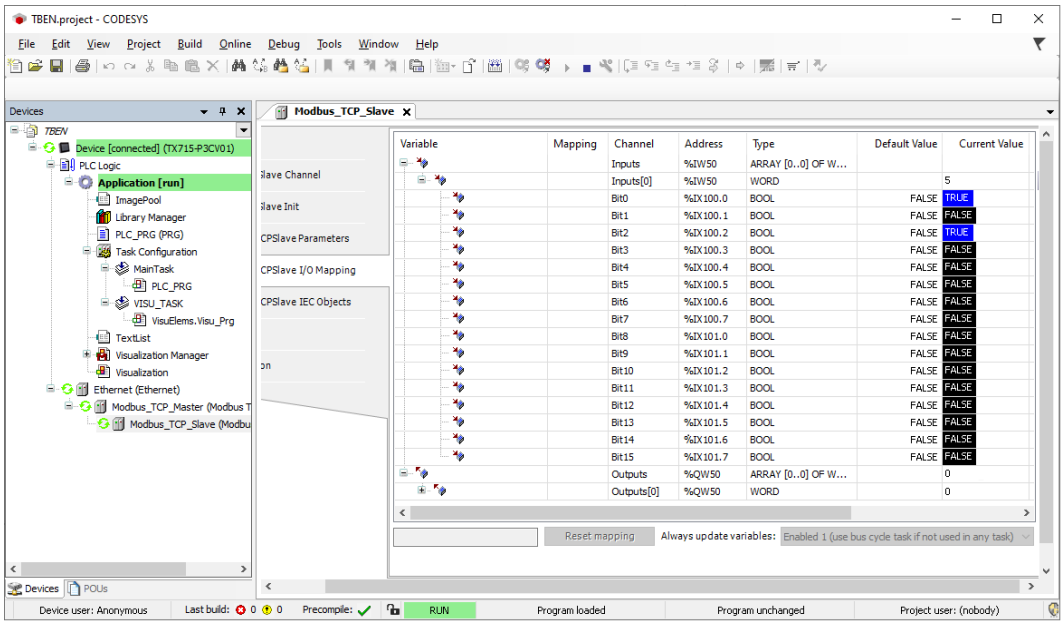



Fig. 57: Process data

## 7.6 Commissioning the devices in EtherNet/IP

### 7.6.1 Common EtherNet/IP features

| Feature | Description |
|---|---|
| QuickConnect | < 500 ms |
| Device Level Ring (DLR) | Yes |
| Number of TCP connections | 3 |
| Number of CIP connections | 10 |
| Input assembly instance | 103 |
| Output assembly instance | 104 |
| Configuration assembly Instance | 106 |

### 7.6.2 EDS files and catalog files

The EDS and catalog files can be downloaded free of charge from www.turck.com.

### 7.6.3 QuickConnect (QC)

The devices support QuickConnect. The maximum start-up times defined for QuickConnect are, however, only guaranteed for the digital channels.

QuickConnect enables a PLC to build up connections to EtherNet/IP nodes in less than 500 ms after switching-on the power supply for the EtherNet/IP network. The fast start-up is necessary for fast tool changing applications at robot arms for example in the automobile industry.

QuickConnect can be activated via the web server of the device, via Configuration Assembly (e.g. in Logix Designer (Studio 5000) or via Class Instance Attribute.

**NOTE**
Activating QuickConnect activated the automatic setting of all necessary port properties.

| Port property | Status |
|---|---|
| Auto negotiation | Deactivated |
| Transmission speed | 100BaseT |
| Duplex | Full duplex |
| Topology | Linear |
| AutoMDIX | Deactivated |

For information on the correct connection of Ethernet cables in QuickConnect applications, please refer to the chapter Connecting [▶ 24].

## Activating QuickConnect via Configuration Assembly

The Configuration Assembly is part of the device's Assembly Class.

- ▶ Configure the Configuration Assembly in Logix Designer (Studio 5000).
- ▶ Activate QuickConnect via byte9, bit 0 = 1 in the Controller Tags.

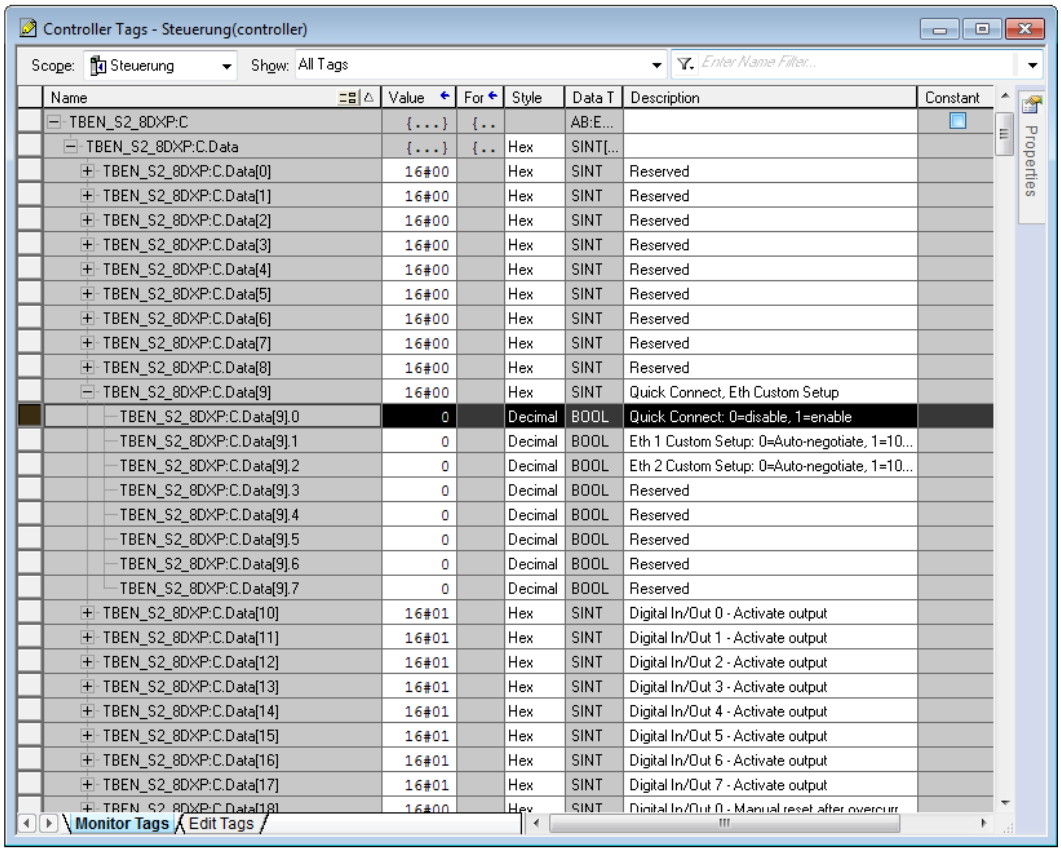



Fig. 58: Configuring QuickConnect in RSLogix

## Activating Quick Connect via Class Instance Attribute

- ▶ Activate Quick Connect via Class Instance Attribute as follows:

| Class | Instance | Attribute | Value |
|---|---|---|---|
| 0xF5 | 0x01 | 0x0C | 0: deactivated (default)<br>1: activated |

Activating QuickConnect via the Webserver.

- ▶ Activate the checkbox **Activate QuickConnect** in the web server.

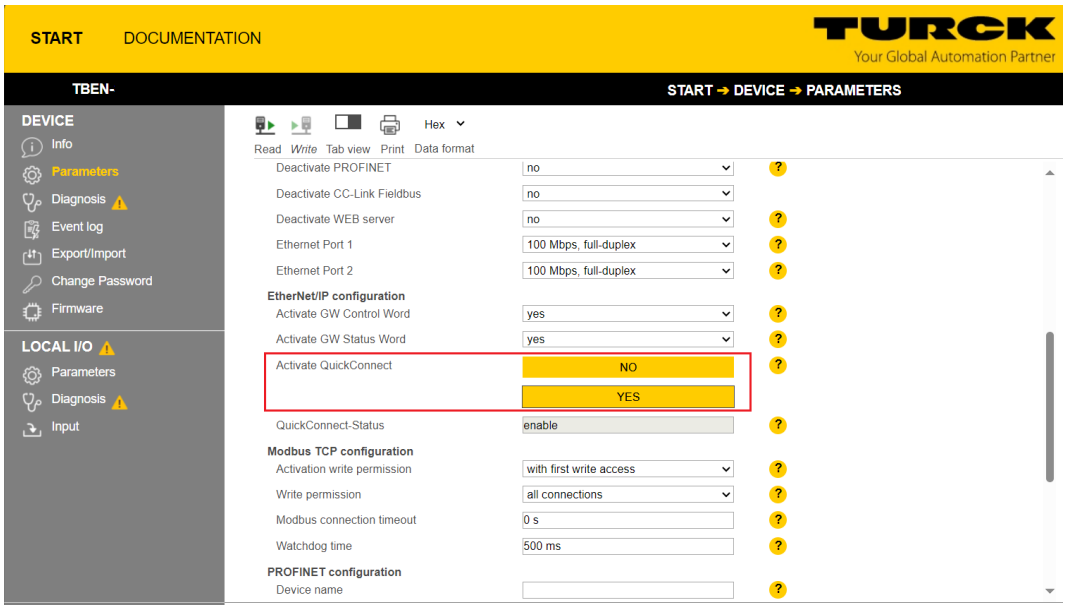



Fig. 59: Activating QuickConnect in the web server

## 7.6.4 Device Level Ring (DLR)

The devices support DLR (Device Level Ring). The DLR redundancy protocol is used to increase the stability of EtherNet/IP networks.

DLR-enabled devices have an integrated switch and can thus be integrated into a ring topology. The DLR protocol is used to detect an interruption in the ring. If the data line is interrupted, data are sent through an alternative network section, so that the network can be reconfigured as soon as possible.

DLR-capable network nodes (DLR supervisor) are provided with extended diagnostic functions which enable the devices to localize errors and thus decrease the time for error search and maintenance. Normally, the controller (i.e. the controller/PLC) assumes the supervisor function, all other network nodes are DLR participants. The supervisor blocks one of its two ports for normal Ethernet traffic, so that a line topology is created for normal Ethernet telegrams. DLR messages can continue to use the ring in both directions and thus continuously check the function of the ring.

### 7.6.5 Diagnostic messages via process data

The diagnostic messages of the I/O channels are directly mapped into the process data [▶ 140]overview Additionally, the device's status word contains the module diagnostics.

The status word contains the module status.

| Byte no. | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 (LSB) | Under-voltage V2 | - | - | - | - | - | ARGEE | Module diagnostics pending |
| 1 (MSB) | - | Force Mode active | - | - | - | - | Under-voltage V1 | - |

#### Control Word

The control word has no function.

### 7.6.6 EtherNet/IP standard classes

The modules support the following EtherNet/IP standard classes in accordance with the CIP specification.

| Class Code | | Object name |
|---|---|---|
| Dec. | Hex. | |
| 01 | 0x01 | Identity Object [▶ 77] |
| 04 | 0x04 | Assembly ObjectAssembly Object |
| 06 | 0x06 | Connection Manager Object [▶ 83] |
| 245 | 0xF5 | TCP/IP Interface Object [▶ 84] |
| 246 | 0xF6 | Ethernet Link Object [▶ 87] |

## Identity Object (0x01)

The following description is taken from the CIP specification, Vol. 2, Rev. 2.1 by ODVA & ControlNet International Ltd. and adapted to the Turck products.

**Instance attributes**

| Attr. no. | | Attribute name | Get/Set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Vendor | G | UINT | Contains the manufacturer ID.<br>Turck = 0x30 |
| 2 | 0x02 | Product type | G | UINT | Shows the general product type.<br>Communications Adapter $12_{dec}$ = 0x0C |
| 3 | 0x03 | Product code | G | UINT | Identifies a special product in a device type.<br>default: $27247_{dec}$ = 0x6A6F |
| 4 | 0x04 | Revision<br>■ Major<br>■ Minor | G | STRUCT OF:<br>■ USINT<br>■ USINT | Revision of the device which is represented by the Indentity Object.<br>■ 0x01<br>■ 0x06 |
| 5 | 0x05 | Device status | G | WORD | WORD |
| 6 | 0x06 | Serial number | G | UDINT | Contains the last 3 bytes of the MAC ID |
| 7 | 0x07 | Product name | G | STRUCT OF:<br>USINT<br>STRING [13] | i.e.: TBEN-S2-2COM-4DXP |

**Device status**

| Bit | Name | Definition |
|---|---|---|
| 0…1 | Reserved | default = 0 |
| 2 | Configured | TRUE = 1:<br>The application in the device has been configured (default setting). |
| 3 | Reserved | default = 0 |
| 4…7 | Extended Device Status | 0011 = no I/O connection established<br>0110 = at least one I/O connection in RUN mode<br>0111 = at least one I/O connection established, all in IDLE mode<br>All other settings = reserved |
| 8 | Minor recoverable fault | Recoverable fault, e.g.:<br>■ Undervoltage<br>■ Force mode of DTM active<br>■ Diagnostics at I/O channel active |
| 9…10 | Reserved | |
| 11 | DIAG | Common error bit |
| 12…15 | Reserved | default = 0 |

**Common services**

| Service code | | Class | Instance | Service name |
|---|---|---|---|---|
| Dec. | Hex. | | | |
| 1 | 0x01 | Yes | Yes | Get_Attribute_All<br>Returns a predefined list of object attributes |
| 5 | 0x05 | No | Yes | Reset<br>Starts the reset service for the device |
| 14 | 0x0E | Yes | Yes | Get_Attribute_Single<br>Returns the content of a specified attribute |
| 16 | 0x10 | No | No | Set_Attribute_Single<br>Changes a single attribute |

## Assembly Object (0x04)

Assembly Objects bind attributes of multiple objects. to allow data to or from each object to be sent or received over a single connection.

The following description of the Ethernet Link Object is taken from the CIP specification, Vol. 2, Rev. 2.1 by ODVA & ControlNet International Ltd. and adapted to the Turck products.

**Class attributes**

| Attr. no. | | Attribute name | Get/set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Revision | G | UINT | 2 |
| 2 | 0x02 | Max. object instance | G | UINT | 104 |

**Instance Attributes**

| Attr. no. | | Attribute name | Get/set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 3 | 0x03 | Data | S | ARRAY OF BYTE | Identifies a special product in a device type.<br>default: $27247_{dec.}$ = 0x6A6F |
| 4 | 0x04 | Size | G | UINT | Number of bytes in attribute 3:<br>256 or variable |

**Common services**

| Service code | | Class | Instance | Service name |
|---|---|---|---|---|
| Dec. | Hex. | | | |
| 1 | 0x01 | Yes | Yes | Get_Attribute_All<br>Returns a predefined list of object attributes. |
| 14 | 0x0E | Yes | Yes | Get_Attribute_Single<br>Returns the content of a specified attribute. |

## Configuration Assembly (Instance 106)

The modules support Configuration Assembly.

The Configuration Assembly contains:

10 bytes module configuration data (EtherNet/IP specific)

+ x Byte (parameter data, depending on device type)

The chapter "Parameterizing and Configuring" [▶ 133] contains a detailed parameter description.

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Device configuration data** | | | | | | | | | |
| 0…8 | 0x00…0x08 | - | - | - | - | - | - | - | - |
| 9 | 0x09 | - | - | - | - | - | Eth2 Port-Setup | Eth1 Port-Setup | Quick Connect |
| **Parameter data COM0** | | | | | | | | | |
| 10 | 0x0A | - | | | | | Operation mode | | |
| 11 | 0x0B | - | | | | | | | Swap A/B line |
| 12 | 0x0C | - | | | | Data rate | | | |
| 13 | 0x0D | - | | | | | | Character format | |
| 14 | 0x0E | - | | | | | | | Stop bits |
| 15 | 0x0F | - | | | | | | EOF detection | |
| 16 | 0x10 | - | | | | | | | Termination active |
| 17 | 0x11 | - | | | | | | | Biasing active |
| 18 | 0x12 | - | | | | | | VAUX1 | |
| 19 | 0x13 | - | | | | | | | |
| 20 | 0x14 | Character timeout | | | | | | | |
| 21 | 0x15 | | | | | | | | |
| 22 | 0x16 | Response timeout | | | | | | | |
| 23 | 0x17 | | | | | | | | |
| 24 | 0x18 | 1st end delimiter | | | | | | | |
| 25 | 0x19 | 2nd end delimiter | | | | | | | |
| 26 | 0x1A | MB server cycle time (* ms) | | | | | | | |
| 27 | 0x1B | | | | | | | | |
| **COM 0 – SCB 0** | | | | | | | | | |
| 28 | 0x1C | Server address | | | | | | | |
| 29 | 0x1D | - | | | | Number reg./server write access | | | |
| 30 | 0x1E | - | | | | Number reg./server read access | | | |
| 31 | 0x1F | Read access | | | | | | | |
| 32 | 0x20 | Write access | | | | | | | |
| 33 | 0x21 | - | | | | | | | |
| 34 | 0x22 | Start address for read access | | | | | | | |
| 35 | 0x23 | | | | | | | | |

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 36 | 0x24 | Start address for write access | | | | | | | |
| 37 | 0x25 | | | | | | | | |
| **COM 0 – SCB 1** | | | | | | | | | |
| 38… 47 | 0x26…0x2F | Assignment (similar to bytes 29… 37) | | | | | | | |
| … | | … | | | | | | | |
| **COM 0 – SCB 7** | | | | | | | | | |
| 98… 107 | 0x62…0x6B | Assignment (similar to bytes 29… 37) | | | | | | | |
| **Parameter data COM1** | | | | | | | | | |
| 108 | 0x6C | - | | | | | Operation mode | | |
| 109 | 0x6D | - | | | | | | | Swap A/B line |
| 110 | 0x6E | - | | | | Data rate | | | |
| 111 | 0x6F | - | | | | | Character format | | |
| 112 | 0x70 | - | | | | | | | Stop bits |
| 113 | 0x71 | - | | | | | | EOF detection | |
| 114 | 0x72 | - | | | | | | | Termination active |
| 115 | 0x73 | - | | | | | | | Biasing active |
| 116 | 0x74 | - | | | | | | VAUX1 | |
| 117 | 0x75 | - | | | | | | | |
| 118 | 0x76 | Character timeout | | | | | | | |
| 119 | 0x77 | | | | | | | | |
| 120 | 0x78 | Response timeout | | | | | | | |
| 121 | 0x79 | | | | | | | | |
| 122 | 0x7A | 1st end delimiter | | | | | | | |
| 123 | 0x7B | 2nd end delimiter | | | | | | | |
| 124 | 0x7C | MB server cycle time (* ms) | | | | | | | |
| 125 | 0x7D | | | | | | | | |
| **COM 1 – SCB 0** | | | | | | | | | |
| 126 | 0x7E | Server address | | | | | | | |
| 127 | 0x7F | - | | | | Number reg./server write access | | | |
| 128 | 0x80 | - | | | | Number reg./server read access | | | |
| 129 | 0x81 | Read access | | | | | | | |
| 130 | 0x82 | Write access | | | | | | | |
| 131 | 0x83 | - | | | | | | | |
| 132 | 0x84 | Start address for read access | | | | | | | |
| 133 | 0x85 | | | | | | | | |
| 134 | 0x86 | Start address for write access | | | | | | | |
| 135 | 0x87 | | | | | | | | |

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM 1 – SCB 1** | | | | | | | | | |
| 136… 145 | 0x88…0x91 | Assignment (similar to bytes 29… 37) | | | | | | | |
| … | … | … | | | | | | | |
| **COM 1 – SCB 7** | | | | | | | | | |
| 196… 205 | 0xC4…0xCD | | | | | | | | |
| **DXP channels** | | | | | | | | | |
| 206 | 0xCE | | | | | | | | SRO4 |
| … | … | … | | | | | | | |
| 209 | 0xD1 | | | | | | | | SRO7 |
| 210 | 0xD2 | | | | | | | | EN_DO4 |
| … | … | … | | | | | | | |
| 213 | 0xD5 | | | | | | | | EN_DO7 |
| 214 | 0xD6 | | | | | | | DIF pulses (DXP4) | |
| 215 | 0xD7 | | | | | | | | Input filter (DXP4) |
| 216 | 0xD8 | Pulse stretch (DXP4) | | | | | | | |
| … | … | … | | | | | | | |
| 226 | 0xE2 | | | | | | | DIF pulses (DXP7) | |
| 227 | 0xE3 | | | | | | | | Input filter (DXP7) |
| 228 | 0xE4 | Pulse stretch (DXP7) | | | | | | | |

**Device Configuration Data**

The default values are written in **bold**.

| Designation | Value | | Meaning |
|---|---|---|---|
| QuickConnect | 0 | **disabled** | QuickConnect is deactivated. |
| | 1 | Activated | QuickConnect is activated. |
| Eth … Port-Setup | 0 | **Auto negotiation** | The port is set to autonegotiation. |
| | 1 | 100BT/FD | Defined setting of communication parameters for the Ethernet port to:<br>■ 100BaseT<br>■ Full duplex |

## Process data instances

### Instance 103 and Instance 104

Instances 103 and 104 are input and output instances with variable size. The size of the assembly data is calculated exactly beforehand to ensure the station configuration, diagnostics etc.. The effective size of the Assembly Instance can be determined using the Assembly Object (instance 0×67, attribute 0x04).

| **Input data** | |
|---|---|
| Input Assembly Instance 103 | 0…470 bytes, default: 470 bytes |

| **Output data** | |
|---|---|
| Output Assembly Instance 104 | 0…400 bytes, default: 400 bytes |

## Process data mapping

The process data mapping the the TBEN-S2-2COM-4DXP for EtherNet/IP corresponds the the process data mapping in chapter "Operating" [▶ 140]. In EtherNet/IP, however, the status and control words are mapped before the process data in the default setting.

The mapping of status and control word can be deactivated via the Gateway Class VSC 100 [▶ 88].

**NOTE**
Activating or deactivating the status and control Word in EtherNet/IP changes the process data mapping.

▶ Observe the offset in the device's process data mapping.

**Process input data**

| **Word offset** | **Bit no.** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | Status word (module status) | | | | | | | | | | | | | | | |
| 1…99 | COM0 | | | | | | | | | | | | | | | |
| 100…198 | COM1 | | | | | | | | | | | | | | | |
| 199, 200 | Diagnostics for COM ports | | | | | | | | | | | | | | | |
| 201…232 | Modbus servers | | | | | | | | | | | | | | | |
| 233 | DXP channels | | | | | | | | | | | | | | | |
| 234 | Diagnostics DXP channel | | | | | | | | | | | | | | | |

**Process output data**

| **Word offset** | **Bit no.** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0 | Control Word | | | | | | | | | | | | | | | |
| 1…99 | COM0 | | | | | | | | | | | | | | | |
| 100…198 | COM1 | | | | | | | | | | | | | | | |
| 199 | DXP channels | | | | | | | | | | | | | | | |

## Connection Manager Object (0x06)

This object is used for connection and connectionless communications, including establishing connections across multiple subnets.

The following description of the Ethernet Link Object is taken from the CIP specification, Vol. 2, Rev. 2.1 by ODVA & ControlNet International Ltd. and adapted to the Turck products.

**Common services**

| Service code | | Class | Instance | Meaning |
|---|---|---|---|---|
| **Dec.** | **Hex.** | | | |
| 84 | 0x54 | No | Yes | FWD_OPEN_CMD<br>(opens a connection) |
| 78 | 0x4E | No | Yes | FWD_CLOSE_CMD<br>(closes a connection) |
| 82 | 0x52 | No | Yes | UNCONNECTED_SEND_CMD |

## TCP/IP Interface Object (0xF5)

The following description of the Ethernet Link Object is taken from the CIP specification, Vol. 2, Rev. 1.1 by ODVA & ControlNet International Ltd. and adapted to the Turck products.

### Class attributes

| Attr. no. | | Designation | Get/Set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Revision | G | UINT | 1 |
| 2 | 0x02 | Max. object instance | G | UINT | 1 |
| 3 | 0x03 | Number of instances | G | UINT | 1 |
| 6 | 0x06 | Max. class identifier | G | UINT | 7 |
| 7 | 0x07 | Max. instance attribute | G | UINT | 6 |

### Instance Attributes

| Attr. no. | | Designation | Get/Set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Status | G | DWORD | Interface status |
| 2 | 0x02 | Configuration capability | G | DWORD | Interface capability flag |
| 3 | 0x03 | Configuration control | G/S | DWORD | Interface control flag |
| 4 | 0x04 | Physical link object | G | STRUCT | |
| | | Path size | | UINT | Number of 16 bit words: 0x02 |
| | | Path | | Padded EPATH | 0x20, 0xF6, 0x24, 0x01 |
| 5 | 0x05 | Interface configuration | G | Structure of: | TCP/IP network interface configuration |
| | | IP address | G | UDINT | Actual IP address |
| | | Network mask | G | UDINT | Actual network mask |
| | | Gateway addr. | G | UDINT | Actual default gateway |
| | | Name server | G | UDINT | 0 = no server address configured |
| | | Name server 2 | G | UDINT | 0 = no secondary server address configured |
| | | Domain name | G | UDINT | 0 = no Domain Name configured |
| 6 | 0x06 | Host name | G | STRING | 0 = no host name configured |
| 12 | 0x0C | QuickConnect | G/S | BOOL | 0 = deactivate<br>1 = activate |

### Common services

| Service code | | Class | Instance | Meaning |
|---|---|---|---|---|
| Dec. | Hex. | | | |
| 1 | 0x01 | Yes | Yes | Get_Attribute_All |
| 2 | 0x02 | No | No | Set_Attribute_All |
| 14 | 0x0E | Yes | Yes | Get_Attribute_Single |
| 16 | 0x10 | No | Yes | Set_Attribute_Single |

### Interface Status

The Status attribute indicates the status of the TCP/IP network interface.

| Bit | Designation | Meaning |
|---|---|---|
| 0…3 | Interface configuration status | Indicates the status of the Interface Configuration attribute:<br>0 = The Interface Configuration attribute has not been configured<br>1 = The Interface Configuration attribute contains valid configuration.<br>2…15 = reserved |
| 4…31 | Reserved | |

### Configuration Capability

The Configuration Capability indicates the device's support for optional network configuration capability.

| Bit | Designation | Meaning | Value |
|---|---|---|---|
| 0 | BOOTP client | The device is capable of obtaining its network configuration via BOOTP. | 1 |
| 1 | DNS client | The device is capable of resolving host names by querying a DNS server. | 0 |
| 2 | DHCP client | The device is capable of obtaining its network configuration via DHCP. | 1 |

### Configuration control

The Configuration Control attribute is used to control network configuration options.

| Bit | Designation | Meaning |
|---|---|---|
| 0…3 | Startup configuration | Determines how the device shall obtain its initial configuration.<br>0 = The device shall use the interface configuration values previously stored (for example, in non-volatile memory or via hardware switches, etc).<br>1…3 = reserved |
| 4 | DNS Enable | Always 0 |
| 5…31 | Reserved | Set to 0 |

**Interface Configuration**

This attribute contains the configuration parameters required to operate a TCP/IP device.

To change this attribute, proceed as follows:

- ▶ Read out the attribute.
- ▶ Change the parameters.
- ▶ Set the attribute.
- ⇨ The TCP/IP Interface Object applies the new configuration upon completion of the Set service. If the value of the Startup Configuration bits (Configuration Control attribute) is 0, the new configuration is stored in non-volatile memory.

The device does not reply to the set service until the values are safely stored to non-volatile memory.

An attempt to set any of the components of the Interface Configuration attribute to invalid values results in an error (status code 0x09) returned from the Set service. If initial configuration is obtained via BOOTP or DHCP, the Interface Configuration attribute components are all 0 until the BOOTP or DHCP reply is received. Upon receipt of the BOOTP or DHCP reply, the Interface Configuration attribute shows the configuration obtained via BOOTP/DHCP.

**Host name**

This attribute contains the device's host name. The host name attribute is used when the device supports the DHCP-DNS Update capability and has been configured to use DHCP upon start up. The mechanism allows the DHCP client to transmit its host name to the DHCP server. The DHCP server then updates the DNS records on behalf of the client.

## Ethernet Link Object (0xF6)

The following description of the Ethernet Link Object is taken from the CIP specification, Vol. 2, Rev. 1.1 by ODVA & ControlNet International Ltd. and adapted to the Turck products.

### Class attributes

| Attr.-no. | | Designation | Get/Set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Revision | G | UINT | 1 |
| 2 | 0x02 | Max. object instance | G | UINT | 1 |
| 3 | 0x03 | Number of instances | G | UINT | 1 |
| 6 | 0x06 | Max. class identifier | G | UINT | 7 |
| 7 | 0x07 | Max. instance attribute | G | UINT | 6 |

### Instance attributes

| Attr.-no. | | Designation | Get/Set | Type | Value |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Interface speed | G | UDINT | Speed in megabit per second (e.g. 10, 100, 1000 etc.) |
| 2 | 0x02 | Interface flags | G | DWORD | Interface capability flag |
| 3 | 0x03 | Physical address | G | ARRAY OF USINT | Contains the interface's MAC address (Turck: 00:07:46:xx:xx:xx) |
| 6 | 0x06 | Interface control | G | 2 WORD | Allows port-wise changes of the Ethernet-settings |
| 7 | 0x07 | Interface type | G | | |
| 10 | 0x0A | Interface label | G | | |

### Interface flags

| Bit | Designation | Meaning | Default value |
|---|---|---|---|
| 0 | Link status | Indicates whether or not the Ethernet communications interface is connected to an active network.<br>0 = inactive link<br>1 = active link | Depends on application |
| 1 | Half/full duplex | 0 = Half duplex<br>1 = Full duplex<br>If the Link Status flag is 0, the value of the Half/Full Duplex flag is indeterminate. | Depends on application |
| 2…4 | Negotiation status | Indicates the status of the automatic autonegotiation<br>0 = autonegotiation in progress<br>1 = autonegotiation and speed detection failed, using default values for speed and duplex (10 Mbps/half duplex).<br>2 = auto-negotiation failed but detected speed (default: half duplex).<br>3 = successfully negotiated speed and duplex<br>4 = autonegotiation not started, yet. Forced speed and duplex. | Depends on application |

| Bit | Designation | Meaning | Default value |
|---|---|---|---|
| 5 | Manual setting requires reset | 0 = interface can activate changes to link parameters (auto-negotiate, duplex mode, interface speed) automatically<br>1 = device requires a Reset service to be issued to its Identity Object in order to adapt the changes. | 0 |
| 6 | Local Hardware Fault | 0 = interface detects no local hardware fault<br>1 = local hardware error detected | 0 |

**Common services**

| Service code | | Class | Instance | Meaning |
|---|---|---|---|---|
| Dec. | Hex. | | | |
| 1 | 0x01 | Yes | Yes | Get_Attribute_All |
| 14 | 0x0E | Yes | Yes | Get_Attribute_Single |
| 76 | 0x4C | No | Yes | Enetlink_Get_and_Clear |

### 7.6.7 VSC – Vendor Specific Classes

In addition to supporting the above named CIP Standard Classes, the device support the vendor specific classes (VSCs) described in the following.

**NOTE**
The chapters “Parameterizing and configuring” [▶ 133] and “Operating” [▶ 133] contain detailed information on parameters, process data, diagnostics and a detailed description of the transmit and receive sequence.

| Class Code | | Name | Description |
|---|---|---|---|
| Dec. | Hex. | | |
| 100 | 0x64 | Gateway | Data and parameters for the field bus specific part of the device. |
| 139 | 0x8B | COM Class | Data and parameters for the COM ports |
| 140 | 0x8C | RS Data/SCB Class | Data of the connected serial devices, data and parameters for the connected Modbus servers |
| 141 | 0x8D | MB Server Timing | Timing data for connected Modbus servers |
| 142 | 0x8E | DXP Class | Data and parameters for the DXP channels |
| 164 | 0xA4 | Ext. function digital | Data and parameters for the extended digital functions of the DXP channels |

#### Gateway Class (VSC 100)

**Object instance**

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 100 | 0x64 | Max. object attribute | G | USINT | Number of the last object attribute to be implemented |
| 101 | 0x65 | Hardware revision | G | STRUCT | Hardware revision number of the device (USINT Maj./USINT Min.) |
| 102 | 0x66 | Firmware revision | G | STRUCT | Firmware revision of the boot firmware (maj./min.). |

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 103 | 0x67 | Service tool ident number | G | UDINT | BOOT-ID (identification number) |
| 104 | 0x68 | Hardware Info | G | STRUCT | Module hardware information (UINT) |

**Object instance 2, gateway instance**

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 109 | 0x6D | Device status | G | STRUCT | Contains the device status. |
| 115 | 0x73 | On IO connection timeout | G/S | ENUM USINT | Reaction when the time limit for an I/O connection is exceeded: |
| | | | | | 0: SWITCH IO FAULTED (0): The channels are switched to substitute value. |
| | | | | | 1: SWITCH IO OFF (1): The outputs are switched to 0. |
| | | | | | 2: SWITCH IO HOLD (2): No further changes to I/O data. The outputs are held. |
| 138 | 0x8A | GW status register | G/S | DWORD | Activates or deactivates the mapping of the status word into the device's input data. Activating or deactivating of the status word is only possible in Assembly Instance 103. |
| 139 | 0x8B | GW control register | G/S | DWORD | Activates or deactivates the mapping of the control word into the device's output data. Activating or deactivating of the control word is only possible in Assembly Instance 104. |
| 140 | 0x8C | Disable protocols | G/S | UINT | Deactivation of the used Ethernet protocol. |
| | | | | | Bit 0: Deactivates EtherNet/IP (cannot be deactivated via the EtherNet/IP interface). |
| | | | | | Bit 1: Deactivates Modbus TCP |
| | | | | | Bit 2: Deactivates PROFINET |
| | | | | | Bit 15: Deactivates the web server |

## COM Class (VSC 139)

This class contains 2 object instances, one for COM0 and one for COM1.

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Operation mode | G/S | USINT | 0 = RS485<br>1 = RS232<br>2 = MB client RS485<br>3 = MB client RS232 |
| 2 | 0x02 | Swap A/B line | G/S | USINT | 0 = no (A = Pin 2, B = Pin 4)<br>1 = yes (A = Pin 4, B = Pin 2) |
| 3 | 0x03 | Data rate | G/S | USINT | 0…3 = reserved<br>4 = 2400 bps<br>5 = 4800 bps<br>6 = 9600 bps<br>7= 14400 bps<br>8 = 19200 bps<br>9 = 28800 bps<br>10 = 38400 bps<br>11 = 57600 bps<br>12 = 115200 bps<br>13 = 230400 bps |
| 4 | 0x04 | Character format | G/S | USINT | 0 = 7O<br>1 = 7E<br>2 = 8N<br>3 = 8O<br>4 = 8E |
| 5 | 0x05 | Stop bits | G/S | USINT | 0 = 1 bit<br>1 = 2 bit |
| 6 | 0x06 | EOF detection | G/S | USINT | 0 = character timeout<br>1 = 1 end delimiter<br>2 = 2 end delimiter<br>3 = frame length |
| 7 | 0x07 | Termination active | G/S | USINT | 0 = No<br>1 = Yes |
| 8 | 0x08 | Biasing active | G/S | USINT | 0 = No<br>1 = Yes |
| 9 | 0x09 | Power supply VAUX1 | G/S | USINT | 0 = 0 V (High-Z)<br>1 = V1(24 VDC)<br>2 = +5 VDC |
| 10 | 0x0A | Character timeout | G/S | INT | Character timeout in ms |
| 11 | 0x0B | Response timeout | G/S | INT | Response timeout in ms |
| 12 | 0x0C | 1st end delimiter | G/S | USINT | Is only interpreted if the parameter "EOF detection" is set to 1 end delimiter or 2 end delimiter. |
| 13 | 0x0D | 2nd end delimiter | G/S | USINT | |
| 14 | 0x0E | MB server cycle time (*1 ms) | G/S | INT | Default:<br>0 = best update time possible |
| 15 | 0x0F | Hardware error | G | USINT | 0 = inactive<br>1 = active |

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 16 | 0x10 | Parameterization error | G | USINT | 0 = inactive<br>1 = active |
| 17 | 0x11 | Overcurrent supply VAUX1 | G | USINT | 0 = inactive<br>1 = active |
| 18 | 0x12 | Error MB server 0 | G | USINT | 0 = inactive<br>1 = active |
| … | … | … | G | USINT | 0 = inactive<br>1 = active |
| 25 | 0x19 | Error MB server 7 | G | USINT | 0 = inactive<br>1 = active |
| 26 | 0x1A | Transmitter ready | G | USINT | 0 = inactive<br>1 = active |
| 27 | 0x1B | Receive complete | G | USINT | 0 = inactive<br>1 = active |
| 28 | 0x1C | Frame error | G | USINT | 0 = inactive<br>1 = active |
| 29 | 0x1D | Parity/format error | G | USINT | 0 = inactive<br>1 = active |
| 30 | 0x1E | Buffer overflow | G | USINT | 0 = inactive<br>1 = active |
| 31 | 0x1F | Timeout | G | USINT | 0 = inactive<br>1 = active |
| 32 | 0x20 | Invalid TX length | G | USINT | 0 = inactive<br>1 = active |
| 33 | 0x21 | Invalid RX length | G | USINT | 0 = inactive<br>1 = active |
| 34 | 0x22 | Received frame length | G | USINT | 0 = inactive<br>1 = active |
| 35 | 0x23 | MB server cycle time (*1 ms) | G | USINT | 0 = inactive<br>1 = active |
| 36 | 0x24 | Transmit | G | USINT | 0 = inactive<br>1 = active |
| 37 | 0x25 | Receive | G | USINT | 0 = inactive<br>1 = active |
| 38 | 0x26 | Transmitter frame length | G | USINT | 0 = inactive<br>1 = active |
| 39 | 0x27 | Receiver frame length | G | USINT | 0 = inactive<br>1 = active |

## RS Data/SCB Class (VSC 140)

This class contains 2 object instances, one for COM0 and one for COM1.

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Server address | G/S | USINT | Address of the connected Modbus RTU servers or start address of the first connected Modbus RTU server<br>Default: 0x01 |
| 2 | 0x02 | Number reg./ server read access | G/S | USINT | Number of registers to be read or number of servers from which data have to be read<br>0…12 |
| 3 | 0x03 | Number reg./ server write access | G/S | USINT | Number of registers to be written or number of servers to which data have to be written<br>0…12 |
| 4 | 0x04 | Read access | G/S | USINT | 0 = deactivated<br>3 = read holding registers (FC 3)<br>4 = read input registers (FC 4)<br>23 = read/write multiple registers (FC 23)<br>128 = read extension<br>Multi server mode:<br>131 = read 1 holding registers (FC3)<br>132 = read 1 input register (FC 4)<br>151 = read/write 1 register (FC 23)<br>163 = read 2 holding registers (FC3)<br>164 = read 2 input register (FC 4)<br>183 = read/write 2 register (FC 23)<br>195 = read 3 holding registers (FC3)<br>196 = read 3 input register (FC 4)<br>215 = read/write 3 register (FC 23)<br>227 = read 4 holding registers (FC3)<br>228 = read 4 input register (FC 4)<br>247 = read/write 4 register (FC 23) |
| 5 | 0x05 | Write access | G/S | USINT | 0 = deactivated<br>6 = write single registers (FC 6)<br>16 = write multiple registers (FC 16)<br>23 = read/write multiple registers (FC 23)<br>128 = write extension<br>Multi server mode:<br>134= Multi server mode: write single register (FC6)<br>144 = write 1 registers (FC16)<br>151 = read/write 1 register (FC 23)<br>176 = write 2 registers (FC16)<br>183 = read/write 2 register (FC 23)<br>208 = write 3 registers (FC16)<br>215 = read/write 3 register (FC 23)<br>240 = write 4 registers (FC16)<br>247 = read/write 4 register (FC 23) |
| 6 | 0x06 | Start address read access | G/S | USINT | Address of the 1st register from which data have to be read<br>0… 65535 |
| 7 | 0x07 | Start address Write access | G/S | USINT | |

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 8 | 0x08 | Input register 0 | G | UINT | Input data of the connected serial device or the Modbus-Server, 1 or 12 register(s) per server. |
| … | … | … | | | |
| 19 | 0x13 | Input register 11 | | | |
| 20 | 0x14 | Output register 0 | G | UINT | |
| … | … | … | | | |
| 31 | 0x1F | Output register 11 | G | UINT | |

## MB server timing class (VSC 141)

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | COM0 – MB server timing, server 0 | G | UINT | Update time [ms] of the connected Modbus RTU servers at COM0 or COM1. |
| 2 | 0x02 | COM1 – MB server timing, server 0 | G | UINT | |
| 3 | 0x03 | COM0 – MB server timing, server 1 | G | UINT | |
| 4 | 0x04 | COM1 – MB server timing, server 1 | G | UINT | |
| … | … | … | | | |
| 15 | 0x0F | COM0 – MB server timing, server 7 | G | UINT | |
| 16 | 0x10 | COM1 – MB server timing, server 7 | G | UINT | |

## DXP Class (VSC 142)

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | DXP4 – manual reset after overcurr. | G/S | USINT | 0 = no<br>1 = yes |
| 2 | 0x02 | DXP5 – manual reset after overcurr. | G/S | USINT | 0 = no<br>1 = yes |
| 3 | 0x03 | DXP6 – manual reset after overcurr. | G/S | USINT | 0 = no<br>1 = yes |
| 4 | 0x04 | DXP7 – manual reset after overcurr. | G/S | USINT | 0 = no<br>1 = yes |
| 5 | 0x05 | DXP4 – activate output | G/S | USINT | 0 = no<br>1 = yes |
| 6 | 0x06 | DXP5 – activate output | G/S | USINT | 0 = no<br>1 = yes |
| 7 | 0x07 | DXP6 – activate output | G/S | USINT | 0 = no<br>1 = yes |
| 8 | 0x08 | DXP7 – activate output | G/S | USINT | 0 = no<br>1 = yes |
| 9 | 0x09 | Overcurrent VAUX2 Ch4/5 | G | USINT | 0 = inactive<br>1 = active |
| 10 | 0x0A | Overcurrent VAUX2 Ch6/7 | G | USINT | 0 = inactive<br>1 = active |
| 11 | 0x0B | DXP4 – overcurrent output | G | USINT | 0 = inactive<br>1 = active |
| 12 | 0x0C | DXP5 – overcurrent output | G | USINT | 0 = inactive<br>1 = active |
| 13 | 0x0D | DXP6 – overcurrent output | G | USINT | 0 = inactive<br>1 = active |
| 14 | 0x0E | DXP7 – overcurrent output | G | USINT | 0 = inactive<br>1 = active |
| 15 | 0x0F | DXP4 – input value | G | USINT | 0 = on<br>1 = off |
| 16 | 0x10 | DXP5 – input value | G | USINT | 0 = on<br>1 = off |
| 17 | 0x11 | DXP6 – input value | G | USINT | 0 = on<br>1 = off |
| 18 | 0x12 | DXP7 – input value | G | USINT | 0 = on<br>1 = off |
| 19 | 0x13 | Output value | G | BYTE | 0 = DXP4<br>1 = DXP5<br>3 = DXP6<br>3 = DXP7 |

### Extended DXP Functions Class (VSC 164)

This class provides four instances, one per DXP-channel.

| Attr. no. | | Designation | Get/Set | Type | Meaning |
|---|---|---|---|---|---|
| Dec. | Hex. | | | | |
| 1 | 0x01 | Extended digital function | G/S | USINT | 0 = deactivated<br>1 = input filter and pulse stretch |
| 2 | 0x02 | Input filter | G/S | USINT | 0 = 0.2 ms<br>1 = 3 ms |
| 3 | 0x03 | Impulse stretch | G/S | USINT | 0…255<br>Pulse stretching 10 to 2550 ms in multiples of 10 ms |

## 7.7 Connecting the devices to an EtherNet/IP scanner with Studio 5000

### Used hardware

The following hardware components are used in this example:

- Rockwell Contoller ControlLogix 1756-L72, Logix 5572
- Rockwell Scanner 1756-EN2TR
- Block module TBEN-S2-2COM-4DXP

### Used software

The following software tools are used in this example:

- Rockwell Studio 5000
- Catalog file for Turck compact devices "TURCK_BLOCK_STATIONS_V….L5K" as part of the ZIP file "TBEN-…_ETHERNETIP.zip" (can be downloaded for free at www.turck.com)

### Prerequisites

- An instance of the software with the catalog files is opened.
- A new project has been created in a second instance of Studio 5000.
- The PLC and the scanner mentioned above have been added to the project in the second instance of Studio 5000.

### 7.7.1 Adding the devices from the Catalog files to the new project

- ▶ Right-click the device entry and use **Copy**.

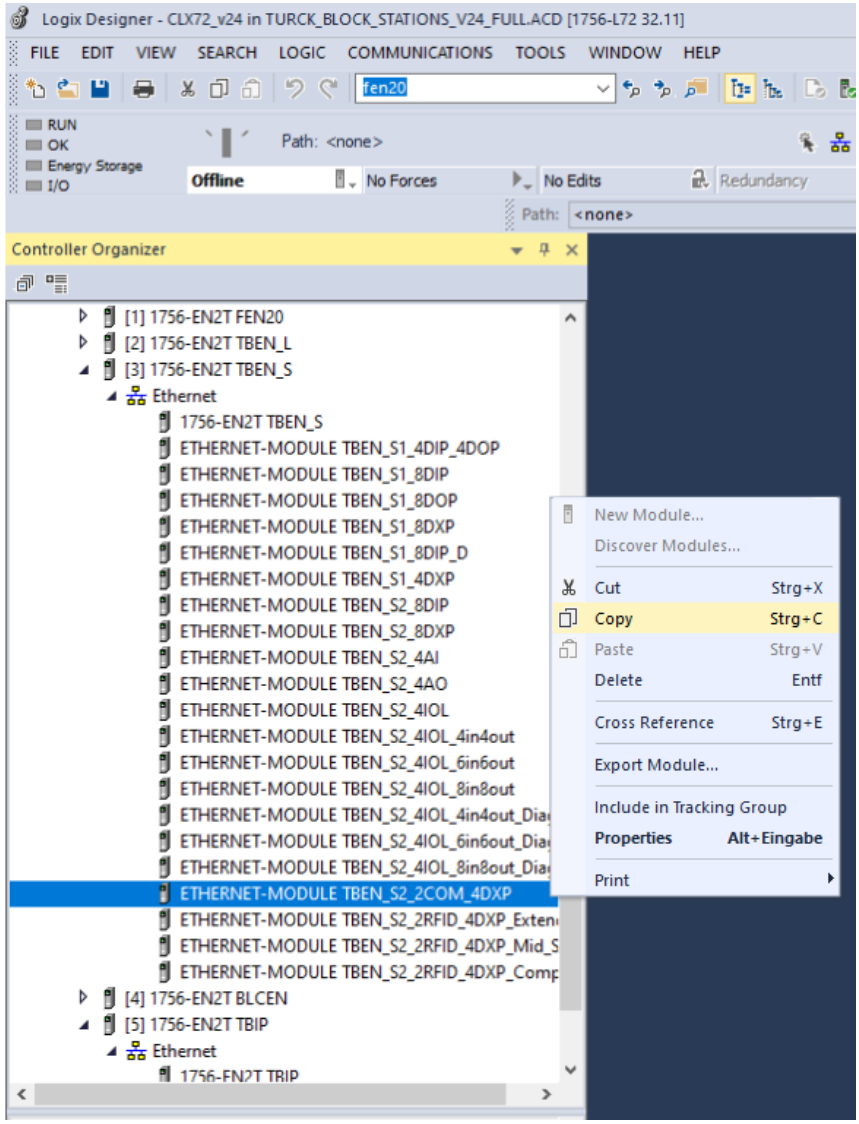



Fig. 60: Copying the device from catalog file in Logix Designer

▶ Right-click the EtherNet/IP Scanner in the 2nd instance of Logix Designer and add the device to the project via **Paste**.

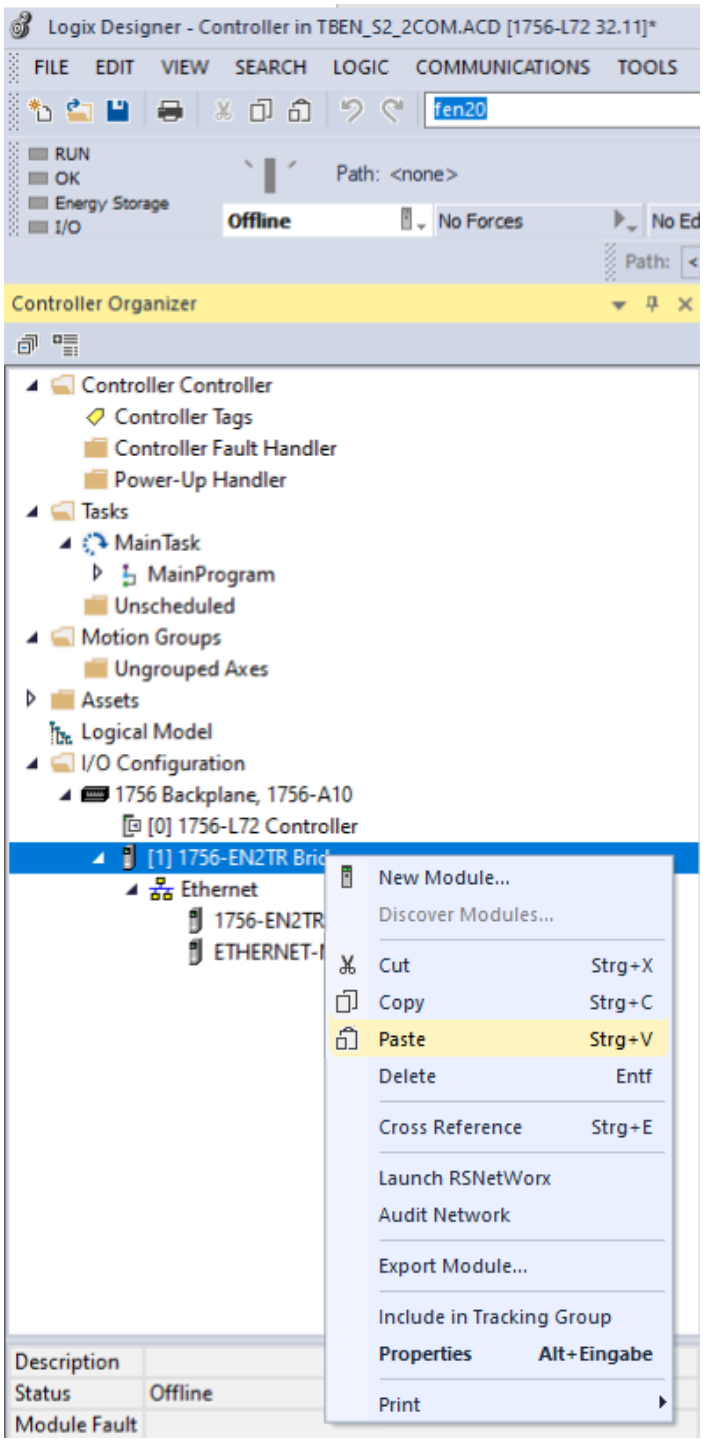



Fig. 61: Adding the device to the project in Logix Designer

### 7.7.2 Configuring the device in Logix Designer

- Open the device entry by double-clicking.
- Assign a module name.
- Set the IP address of the device (example: 192.168.145.181).

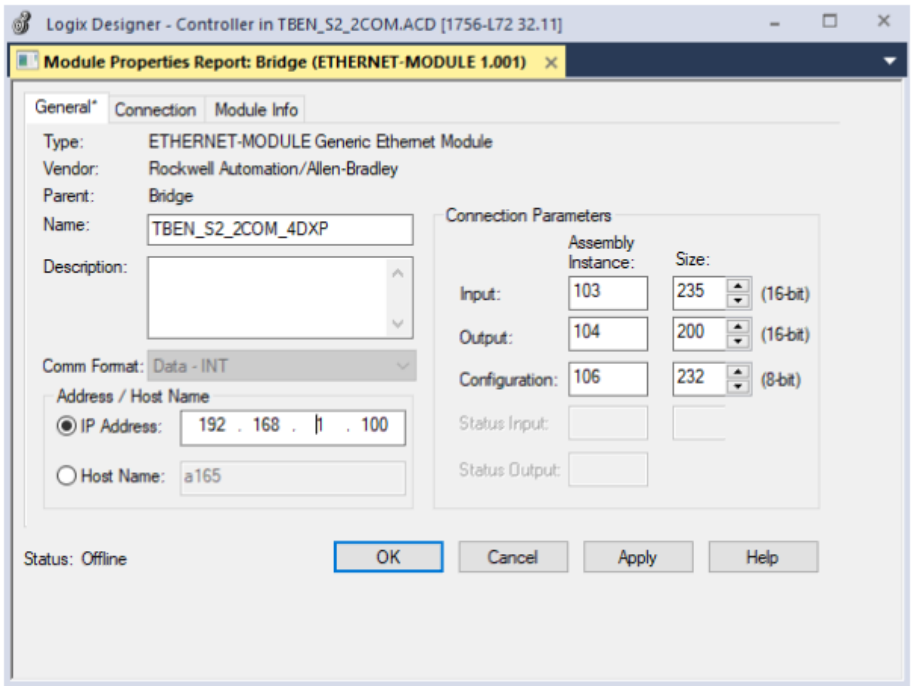



Fig. 62: Setting module name and IP address

- Optional: Set the connection parameters.

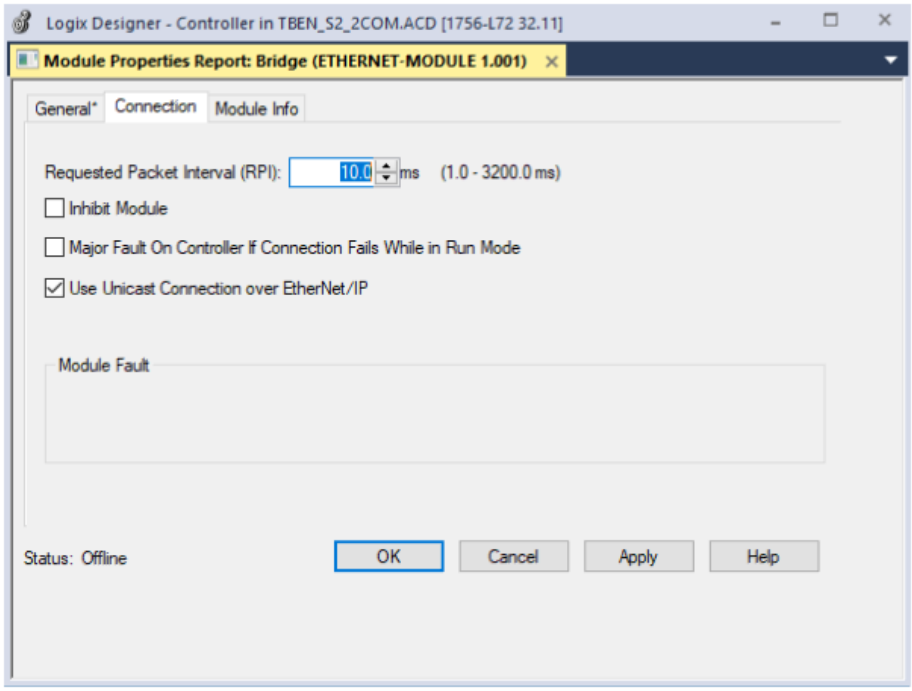



Fig. 63: Setting the connection parameters

### 7.7.3 Parameterizing the device

- Open the Controller Tags of the device.
- Parameterize the device via the Controller Tags **TBEN_S2_2COM_4DXP:C**.

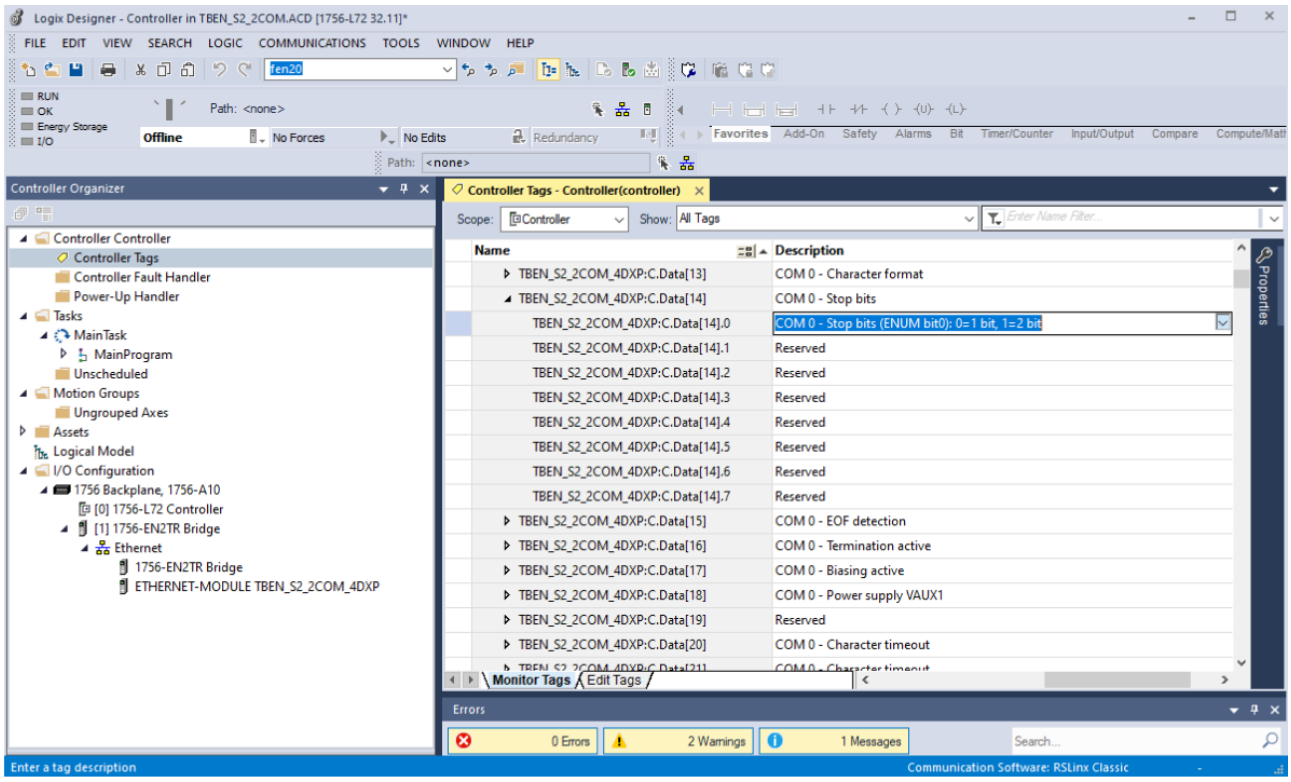



Fig. 64: Parameterizing the device

### 7.7.4 Going online with the PLC

- ▶ Search the network via the **Who Active** button.
- ▶ Select the PLC.
- ▶ Set the communication path via **Set Project Path**.
- ⇨ The communication path is set.

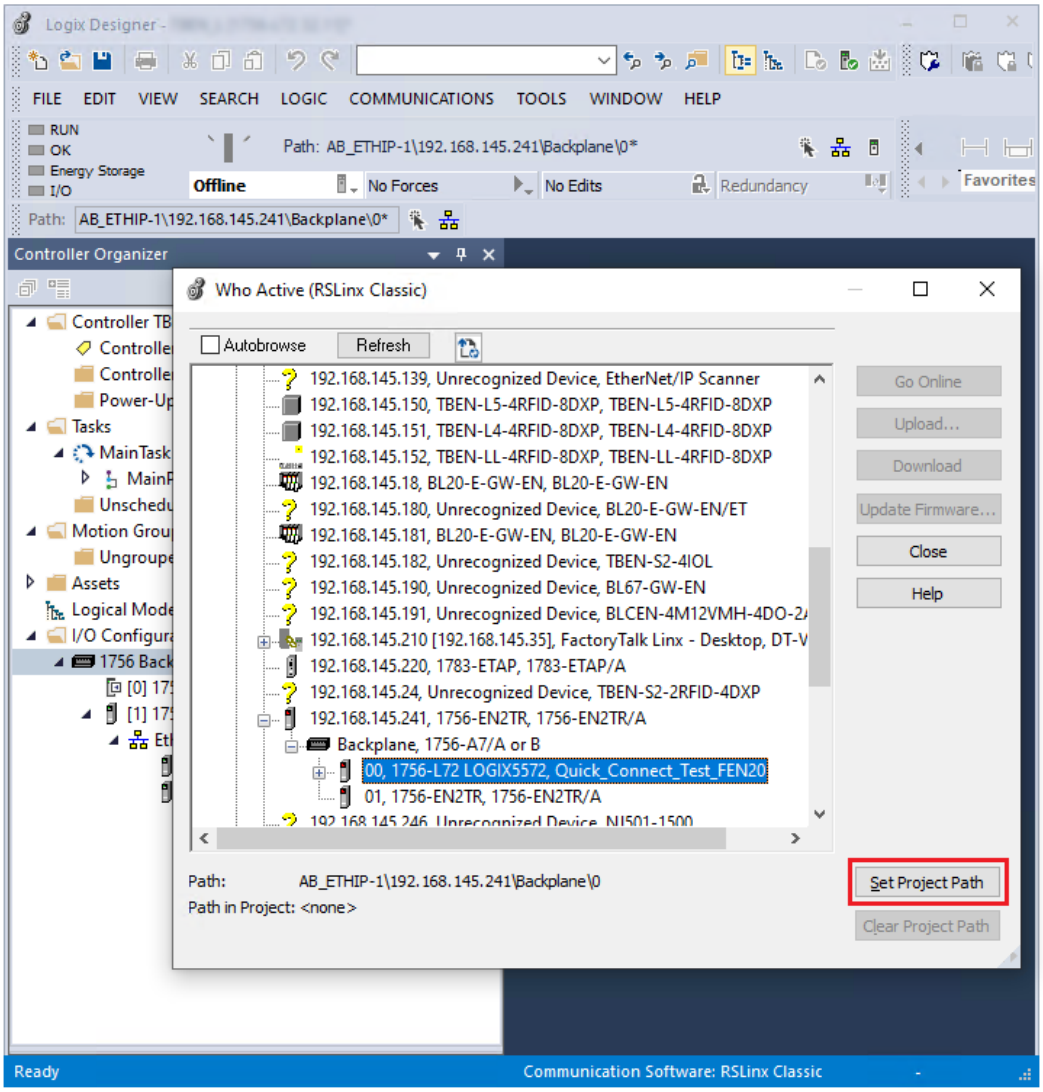



Fig. 65: Setting the communication path

- ▶ Select the PLC.
- ▶ Click **Go online**.

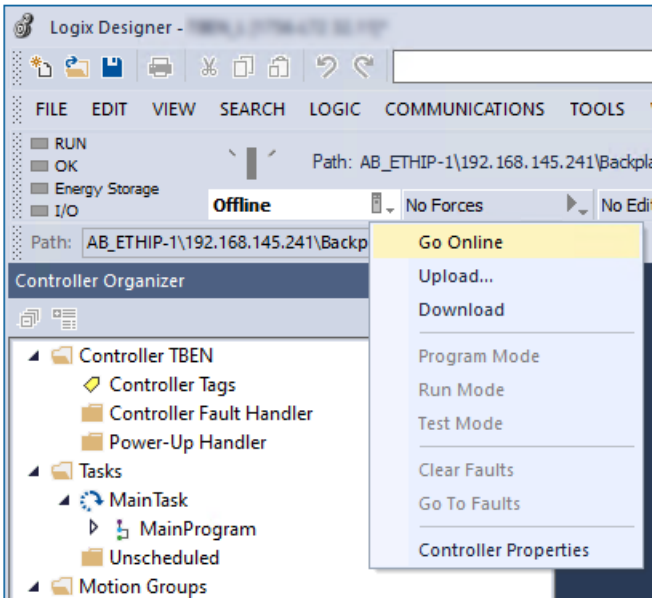



Fig. 66: Going online with the device

- ▶ Click **Download** in the following dialog (Connect To Go Online).
- ▶ Confirm all following messages.
- ⇨ The program is downloaded to the PLC. The online connection is established.

### 7.7.5 Reading process data

- ▶ Open the Controller Tags in the project tree by double-clicking the entry.
- ⇨ The access to the input data (TBEN_S_…:I) and output data (TBEN_S_…:O) is possible.

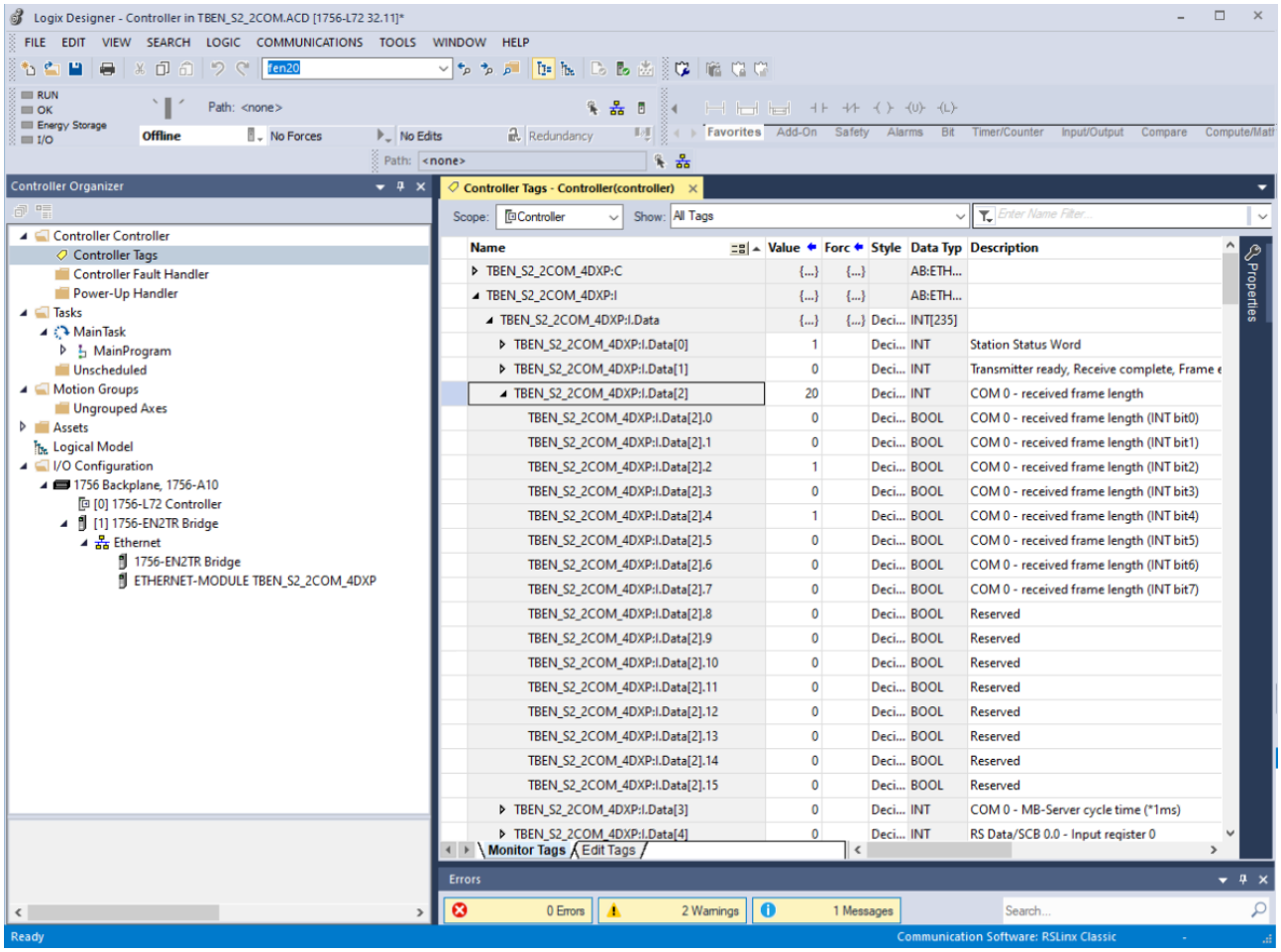



Fig. 67: Controller tags in project tree

# 7.8 Commissioning the device with CC-Link IE Field Basic

## 7.8.1 General features CC-Link IE Field Basic

CC-Link IE Field Basic works with a client/server communication model. A maximum data width of 64 × 64 bits is available for communication between a client station and several server stations, whereby a unit of 64 bits is referred to as an occupied station. A CC-Link Field Basic network can consist of a maximum of 64 occupied stations. I/O modules can occupy one or more of the 64 occupied stations, depending on their complexity and data width.

| **CC-Link IE Field Basic** | | |
|---|---|---|
| Maximum number of stations in a network | Max. 64 occupied stations | An I/O module can occupy several occupied stations. |
| Groups: | Max. 16 occupied stations | To optimize process data traffic, devices can be combined into groups according to their function. A group can consist of a maximum of 16 occupied stations. |
| Acyclic data | | Cyclical data is mapped bit by bit or word by word into registers. |
| | RX | Register for bit-by-bit access to digital inputs (DI) |
| | RY | Register for bit-by-bit access to digital Outputs (DO) |
| | RWr | Register for word-by-word, read access to process data (e.g. IO-Link) |
| | RWw | Register for word-by-word, write access to process data (e.g. IO-Link) |
| Port numbers | 61450 (cyclic data) | |
| | 61451 (port number of server station for NodeSearch and IPAddressSet) | |

## 7.8.2 CSP+ files

The CSP+ files can be downloaded free of charge at www.turck.com.

### 7.8.3 Cyclic data transmission

The cyclic process image of the devices is divided into a bit area and a word area. The bit area contains, for example, the process data of the DXP channels, DXP channel diagnostics and the module status (status word) Prozess-Eingangsdaten. In the TBEN-S2-2COM-4DXP, the word area contains, for example, the process data of the COM ports (status information in RS323/485 mode), the process data of the Modbus server in Modbus client mode, etc.

### 7.8.4 Occupied Stations

Profiles 2 and 4 are defined for RS232 and RS485 mode respectively, profiles 1 and 3 for Modbus client mode [▷ 12].

| Profile | Occupied Stations | Size of process input data | | Size of process output data | |
|---|---|---|---|---|---|
| | | Bit area (RX) | Register area (RWr) | Bit area (RY) | Registers area (RWw) |
| RS232 or RS485 mode | | | | | |
| 2 | 2 | **6 byte**<br>■ DXP input data 16 bit, 4 bit occupied<br>■ DXP diagnostics: 16 bit<br>■ Module status: 16 bit | **112 byte**<br>■ Input data.: 6 byte per COM port<br>■ Receive data buffer: 48 byte per COM port<br>■ Com port diagnostics 2 byte per COM port | **2 byte**<br>■ DXP output data: 16 bit, 4 bit occupied | **108 byte**<br>■ Output data 6 byte per COM port<br>■ Transmit data buffer: 48 byte |
| 4 | 4 | | **232 byte**<br>■ Input data: 6 byte per COM port<br>■ Receive data buffer:<br>– COM0: 120 byte<br>– COM1: 96 byte<br>■ Com port diagnostics 2 byte per COM port | | **228 byte**<br>■ Output data: 6 byte per COM port<br>■ Transmit data buffer:<br>– COM0: 120 byte<br>– COM1: 96 byte |
| Modbus client mode | | | | | |
| 1 | 1 | **6 byte**<br>■ DXP input data: 16 bit, 4 bit occupied<br>■ DXP diagnostics: 16 bit<br>■ Module status: 16 bit | **36 byte**<br>■ 2 byte per connected Modbus server<br>■ COM port diagnostics 2 byte per COM port | **2 byte**<br>■ DXP output data: 16 bit, 4 bit occupied | **32 byte**<br>■ 2 byte per connected Modbus server |
| 3 | 3 | | **144 byte**<br>■ Input data: 6 byte per COM port<br>■ Receive data: 48 byte per COM port (6 byte per SCB)<br>■ COM port diagnostics 2 byte per COM port<br>■ Modbus server status 16 byte per COM port (2 byte per Modbus server) | | **108 byte**<br>■ Output data: 6 byte per COM port<br>■ Transmit data: 48 byte per COM port (6 byte per SCB) |

### 7.8.5 Bit area

| RX | Signal | RY | |
|---|---|---|---|
| RX4 | Input value DXP4 | RY4 | Output value DXP4 |
| … | … | … | … |
| RX7 | Input value DXP7 | RY7 | Output value DXP7 |
| RX12 | Overcurrent VAUX2 Ch4/Ch5 | | |
| RX13 | Overcurrent VAUX2 Ch6/Ch7 | | |
| RX1C | Overcurrent output Ch4 | | |
| RX1D | Overcurrent output Ch5 | | |
| RX1E | Overcurrent output Ch6 | | |
| RX1F | Overcurrent output Ch7 | | |
| **Module status (status word)** | | | |
| RX20 | DIAG | | |
| RX21 | ARGEE | | |
| RX22 | - | | |
| RX23 | | | |
| RX24 | | | |
| RX25 | | | |
| RX26 | | | |
| RX27 | V2 | | |
| RX28 | - | | |
| RX29 | V1 | | |
| RX2A | Internal error | | |
| RX2B | - | | |
| RX2C | | | |
| RX2D | | | |
| RX2E | FCE | | |
| RX2F | - | | |

### 7.8.6 Word area

The data in the word area has different data sizes and content depending on the profile.

Description of process data: Process input data [▶ 140] and process output data [▶ 145].

**1 Occupied Station (profile 1)** [▶ 104]

Profile 1 is used for Modbus client mode.

| **RWr** | | **Process input data** | **RWw** | | **Process output data** |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **Receive data Modbus server** | | | **Transmit data Modbus server** | | |
| RWr0 | | Modbus server 0, input register 0 | RWw0 | | Modbus server 0, output register 0 |
| RWr1 | | Modbus server 1, input register 0 | RWw1 | | Modbus server 1, output register 0 |
| … | | … | … | | … |
| RWrE | | Modbus server 14, input register 0 | RWwE | | Modbus server 14, output register 0 |
| RWrF | | Modbus server 15, input register 0 | RWwF | | Modbus server 15, output register 0 |
| **COM port diagnostics** | | | | | |
| **COM0** | | | | | |
| RWr10 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…<br>0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |
| **COM1** | | | | | |
| RWr11 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…<br>0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |

**2 Occupied Stations (profile 2)** [▶ 104]

Profile 2 is used for the RS232/RS485 mode.

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **COM0** | | | | | |
| RWr0 | 0x0 | Transmitter ready | RWw0 | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr1 | | Received frame length | RWw1 | | Transmit frame length |
| RWr2 | | MB server cycle time COM0 | RWw2 | | Receive frame length |
| RWr3 | | Receive data buffer | RWw3 | | Transmit data buffer |
| … | | | … | | |
| RWr1A | | | RWw1A | | |
| **COM1** | | | | | |
| RWr1B | 0x0 | Transmitter ready | RWw1B | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr1C | | Received frame length | RWw1C | | Transmit frame length |
| RWr1D | | MB server cycle time COM0 | RWw1D | | Receive frame length |
| RWr1E | | Receive data buffer | RWw1E | | Transmit data buffer |
| … | | | … | | |
| RWr35 | | | RWw35 | | |
| **COM port diagnostics** | | | | | |
| **COM0** | | | | | |
| RWr36 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| RWr37 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |

**3 Occupied Stations (profile 3)** [▶ 104]

Profile 3 is used for Modbus client mode.

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **Input data COM0** | | | **Output data COM0** | | |
| RWr0 | 0x0 | Transmitter ready | RWw0 | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr1 | | Received frame length | RWw1 | | Transmit frame length |
| RWr2 | | MB server cycle time COM0 | RWw2 | | Receive frame length |
| **Receive data Modbus server at COM 0** | | | **Transmit data Modbus server at COM0** | | |
| RWr3 | | Modbus server 0, input register 0 | RWw3 | | Modbus server 0, output register 0 |
| RWr4 | | Modbus server 0, input register 1 | RWw4 | | Modbus server 0, output register 1 |
| RWr5 | | Modbus server 0, input register 2 | RWw5 | | Modbus server 0, output register 2 |
| … | | … | … | | … |
| RWr18 | | Modbus server 7, input register 0 | RWw18 | | Modbus server 7, output register 0 |
| RWr19 | | Modbus server 7, input register 1 | RWw19 | | Modbus server 7, output register 1 |
| RW1A | | Modbus server 7, input register 2 | RWw1A | | Modbus server 7, output register 2 |
| **Input data COM1** | | | **Output data COM1** | | |
| RWr1B | 0x0 | Transmitter ready | RWw1B | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr1C | | Received frame length | RWw1C | | Transmit frame length |
| RWr1D | | MB server cycle time COM0 | RWw1D | | Receive frame length |
| **Receive data Modbus server at COM 1** | | | **Transmit data Modbus server at COM1** | | |
| RWr1E | | Modbus server 0, input register 0 | RWw1E | | Modbus server 0, output register 0 |
| RWr1F | | Modbus server 0, input register 1 | RWw1F | | Modbus server 0, output register 1 |
| RWr20 | | Modbus server 0, input register 2 | RWw20 | | Modbus server 0, output register 2 |
| … | | … | … | | … |
| RWr33 | | Modbus server 7, input register 0 | RWw33 | | Modbus server 7, output register 0 |
| RWr34 | | Modbus server 7, input register 1 | RWw34 | | Modbus server 7, output register 1 |
| RWr35 | | Modbus server 7, input register 2 | RWw35 | | Modbus server 7, output register 2 |

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **COM port diagnostics** | | | | | |
| **COM0** | | | | | |
| RWr36 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2… 0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |
| **COM1** | | | | | |
| RWr37 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2… 0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |
| **Modbus server status COM0** | | | | | |
| RWr38 | 0x0… 0x3 | Error code Ch0 | | | |
| | 0x4 | Read error Ch0 | | | |
| | 0x5 | Write error Ch0 | | | |
| | 0x6 | Parity or format error Ch0 | | | |
| | 0x7 | MODBUS timeout Ch0 | | | |
| | 0xC | Valid read config. Ch0 | | | |
| | 0xD | Valid write config. Ch0 | | | |
| … | … | | | | |
| RWr45 | 0x0… 0x3 | Error code Ch7 | | | |
| | 0x4 | Read error Ch7 | | | |
| | 0x5 | Write error Ch7 | | | |
| | 0x6 | Parity or format error Ch7 | | | |
| | 0x7 | MODBUS timeout Ch7 | | | |
| | 0xC | Valid read config. Ch7 | | | |
| | 0xD | Valid write config. Ch7 | | | |

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **Modbus server status COM1** | | | | | |
| RWr46 | 0x0…<br>0x3 | Error code Ch0 | | | |
| | 0x4 | Read error Ch0 | | | |
| | 0x5 | Write error Ch0 | | | |
| | 0x6 | Parity or format error Ch0 | | | |
| | 0x7 | MODBUS timeout Ch0 | | | |
| | 0xC | Valid read config. Ch0 | | | |
| | 0xD | Valid write config. Ch0 | | | |
| … | … | | | | |
| RWr53 | 0x0…<br>0x3 | Error code Ch7 | | | |
| | 0x4 | Read error Ch7 | | | |
| | 0x5 | Write error Ch7 | | | |
| | 0x6 | Parity or format error Ch7 | | | |
| | 0x7 | MODBUS timeout Ch7 | | | |
| | 0xC | Valid read config. Ch7 | | | |
| | 0xD | Valid write config. Ch7 | | | |

**4 Occupied Stations (profile 4)** [▶ 104]

Profile 4 is used for the RS232/RS485 mode.

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| **COM0** | | | | | |
| RWr0 | 0x0 | Transmitter ready | RWw0 | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr1 | | Received frame length | RWw1 | | Transmit frame length |
| RWr2 | | MB server cycle time COM0 | RWw2 | | Receive frame length |
| RWr3 | | Receive data buffer | RWw3 | | Transmit data buffer |
| … | | | … | | |
| RWr3E | | | RWw3E | | |
| **COM1** | | | | | |
| RWr3F | 0x0 | Transmitter ready | RWw3F | 0x0 | Receive |
| | 0x1 | Receive complete | | 0x1 | Transmit |
| | 0x2 | Frame error | | 0x2 | Reserved |
| | 0x3 | Parity/format error | | 0x3 | |
| | 0x4 | Buffer overflow | | 0x4 | |
| | 0x5 | Timeout | | 0x5 | |
| | 0x6 | Invalid TX length | | 0x6 | |
| | 0x7 | Invalid RX length | | 0x7 | |
| RWr40 | | Received frame length | RWw40 | | Transmit frame length |
| RWr41 | | MB server cycle time COM0 | RWw41 | | Receive frame length |
| RWr42 | | Receive data buffer | RWw42 | | Transmit data buffer |
| … | | | … | | |
| RWr71 | | | RWw71 | | |
| **COM port diagnostics** | | | | | |
| **COM0** | | | | | |
| RWr72 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |

| RWr | | Process input data | RWw | | Process output data |
|---|---|---|---|---|---|
| **Word (hex)** | **Bit** | | **Word (hex)** | **Bit** | |
| RWr73 | 0x0 | Hardware error | | | |
| | 0x1 | Parameterization error | | | |
| | 0x2…0x6 | Reserved | | | |
| | 0x7 | Overcurrent supply VAUX1 | | | |
| | 0x8 | Error MB server 0 | | | |
| | 0x9 | Error MB server 1 | | | |
| | … | … | | | |
| | 0xE | Error MB server 6 | | | |
| | 0xF | Error MB server 7 | | | |

### 7.8.7 Parameter mapping

The chapter "Parameterizing and configuring" [▶ 133] contains a detailed parameter description.

| Parameter ID | Offset | Parameter name | Channel | Value | Meaning |
|---|---|---|---|---|---|
| **Parameters for COM port COM0** [▶ 133] | | | | | |
| B000 | 0.0 | Operation mode | COM0 | **0x00** | **RS485** |
| | | | | 0x01 | RS232 |
| | | | | 0x02 | MB client RS485 |
| | | | | 0x03 | MB client RS232 |
| | 0.3 | Swap A/B line | | **0x00** | **No** |
| | | | | 0x01 | Yes |
| | 0.4 | Data rate | | 0x04 | 2400 bps |
| | | | | 0x05 | 4800 bps |
| | | | | **0x06** | **9600 bps** |
| | | | | 0x07 | 14400 bps |
| | | | | 0x08 | 19200 bps |
| | | | | 0x09 | 28800 bps |
| | | | | 0x0A | 38400 bps |
| | | | | 0x0B | 57600 bps |
| | | | | 0x0C | 115200 bps |
| | | | | 0x0D | 230400 bps |
| | 0.8 | Character format | | 0x00 | 7O |
| | | | | 0x01 | 7E |
| | | | | 0x02 | 8N |
| | | | | 0x03 | 8O |
| | | | | **0x04** | **8E** |
| | 0.B | Stop bits | | **0x00** | **1 bit** |
| | | | | 0x01 | 2 bit |
| | 0.C | EOF detection | | 0x00 | Character timeout |
| | | | | 0x01 | 1 end delimiter |
| | | | | 0x02 | 2 end delimiter |
| | | | | **0x03** | **Frame length** |
| | 0.E | Termination active | | **0x00** | **Yes** |
| | | | | 0x01 | No |
| | 0.F | Biasing active | | **0x00** | **Yes** |
| | | | | 0x01 | No |
| | 1.0 | VAUX1 | | **0x00** | **0 V (High-Z)** |
| | | | | 0x01 | V1 (24 VDC) |
| | | | | 0x02 | +5 VDC |
| | 2.0 | Character timeout | | 0x0000… 0xFFFF (**0x0064**) | Character timeout in ms |

| Parameter ID | Offset | Parameter name | Channel | Value | Meaning |
|---|---|---|---|---|---|
| B000 | 3.0 | Response timeout | COM0 | 0x0000… 0xFFFF (**0x03E8**) | |
| | 4.0 | 1st end delimiter | | 0x00 …0xFF (**0x03**) | First end character for recognizing the end of a data packet |
| | 4.8 | 2nd end delimiter | | **0x00**…0xFF | Second end character for recognizing the end of a data packet |
| | 5.0 | MB server cycle time (* ms) | | **0x0000**… 0xFFFF | Time between requests from the Modbus client to the Modbus server in ms. |
| **Parameters for Server Configuration Block (SCB) at COM0** [▶ 136] | | | | | |
| B001 | 0.0 | Server address | COM0/ SCB0 | 0x00… 0x0F (**0x01**) | ■ Standard mode: address of the connected Modbus RTU server<br>■ Multi server mode: start address of the first of the connected Modbus RTU server |
| | 0.8 | Number reg. /server write access | | 0x0…**0xC** | ■ Standard mode: number of registers to be read or written<br>■ Multi server mode: number of servers from which data have to be read or to which data have to be written |
| | 0.C | Number reg./ server read access | | 0x0…**0xC** | |
| | 1.0 | Read access | | 0x00 | Deactivated |
| | | | | 0x03 | Read holding registers (FC3) |
| | | | | 0x04 | Read input registers (FC4) |
| | | | | 0x17 | Read and write multiple registers (FC23) |
| | | | | 0x80 | Read extension |
| | | | | 0x83 | Multi server mode: Read 1 holding register (FC3) |
| | | | | 0x84 | Multi server mode: Read 1 input register (FC4) |
| | | | | 0x97 | Multi server mode: Read and write 1 registers (FC23) |
| | | | | 0xA3 | Multi server mode: Read 2 holding register (FC3) |
| | | | | 0xA4 | Multi server mode: Read 2 input register (FC4) |
| | | | | 0xB7 | Multi server mode: Read and write 2 registers (FC23) |

| Parameter ID | Offset | Parameter name | Channel | Value | Meaning |
|---|---|---|---|---|---|
| B001 | 1.0 | Read access | COM0/ SCB0 | 0xC3 | Multi server mode: Read 3 holding register (FC3) |
| | | | | 0xC4 | Multi server mode: Read 3 input register (FC4) |
| | | | | 0xD7 | Multi server mode: Read and write 3 registers (FC23) |
| | | | | 0xE3 | Multi server mode: Read 4 holding register (FC3) |
| | | | | 0xE4 | Multi server mode: Read 4 input register (FC4) |
| | | | | 0xF7 | Multi server mode: Read and write 4 registers (FC23) |
| | 1.8 | Write access | | 0x00 | **Deactivated** |
| | | | | 0x06 | Write one output register (FC6) |
| | | | | 0x10 | Write multiple output registers (FC16) |
| | | | | 0x17 | Read and write multiple registers (FC23) |
| | | | | 0x80 | Write extension |
| | | | | 0x86 | Multi server mode: Write one output register (FC6) |
| | | | | 0x90 | Multi server mode: Write one output registers (FC16) |
| | | | | 0x97 | Multi server mode: Read and write 1 registers (FC23) |
| | | | | 0xB0 | Multi server mode: Write two output registers (FC16) |
| | | | | 0xB7 | Multi server mode: Read and write 2 registers (FC23) |
| | | | | 0xD0 | Multi server mode: Write three output registers (FC16) |
| | | | | 0xD7 | Multi server mode: Read and write 3 registers (FC23) |
| | | | | 0xE3 | Multi server mode: Write four output registers (FC16) |
| | | | | 0xF7 | Multi server mode: Read and write 4 registers (FC23) |
| B001 | 2.0 | Start address for read access | COM0/ SCB0 | **0x0000**… 0xFFF | Address of the 1st register from which data have to be read |
| | 3.0 | Start address for write access | | **0x0000**… 0xFFF | |
| B002 | 0.0… 3.0 | | COM0/ SCB1 | Assignment similar to B001 for COM0/SCB0 | |
| … | … | … | … | | |
| B008 | 0.0… 3.0 | | COM0/ SCB7 | | |

| Parameter ID | Offset | Parameter name | Channel | Value | Meaning |
|---|---|---|---|---|---|
| **Parameters for COM port COM1** [▶ 133] | | | | | |
| B009 | 0.0… 5.0 | | COM1 | Assignment similar to B000 for SCB0 | |
| **Parameters for Server Configuration Block (SCB) at COM0** [▶ 136] | | | | | |
| B00A | | | COM0/ SCB1 | Assignment similar to B001 for COM0/SCB0 | |
| … | … | … | … | | |
| B011 | | | COM1/ SCB7 | | |
| **Parameters of DXP channels** | | | | | |
| B015 | 0.4 | Manual reset after overcurr. ch4 | DXP4 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.5 | Manual reset after overcurr. ch5 | DXP5 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.6 | Manual reset after overcurr. ch6 | DXP6 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.7 | Manual reset after overcurr. ch7 | DXP7 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.C | Activate output ch4 | DXP4 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.D | Activate output ch5 | DXP5 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.E | Activate output ch6 | DXP6 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| | 0.F | Activate output ch7 | DXP7 | 0x00 | **No** |
| | | | | 0x01 | Yes |
| **Parameters for the extended digital function** | | | | | |
| B017 | 0.0 | DIF pulses (DXP4) | 0x00 | **Deactivated** | |
| | | | 0x01 | Input filter and pulse stretch | |
| | 0.7 | Input filter (DXP4) | **0x00** | **0.2 ms** | |
| | | | 0x01 | 3 ms | |
| | 0.8 | Pulse stretch (DXP4) | **0x00**… 0xFF | Pulse stretching 10 to 2550 ms in multiples of 10 ms | |
| B018 | Extended digital function (DXP5) | | | Assignment similar to B017 | |
| B019 | Extended digital function (DXP6) | | | | |
| B01A | Extended digital function (DXP7) | | | | |

### 7.8.8 Acyclic communication via SLMP – supported functions

The devices support acyclical access via SLMP command Device Read (0x0401) and Device Write (0x1401).

#### Supported Device Codes

| Device Code | Description |
|---|---|
| 0x0011 | Device information (vendor ID, device code, device name, etc.) |
| 0x00AC | Acyclic communication |
| 0x00D8 | Input data |
| 0x00D9 | Output data |
| 0x00DD | Diagnostic data |

#### Supported End Codes

| End Code | Description |
|---|---|
| 0x0000 | Command successfully executed |
| 0xC059 | Command/subcommand: not supported command or subcommand |
| 0xC05C | Wrong data: data content does not fit to the command data content does not fit to the command |
| 0xC061 | Data length: data length does not fit to the command |

#### Read device information (Device Code 0x0011)

| Address (Add) | Content | Access type | Data length in word (Len) | Description |
|---|---|---|---|---|
| 0x0001 | Vendor code | ro | 1 | Vendor ID<br>Turck: 0x3355 |
| 0x0002 | Model code | ro | 2 | ID of the device |
| 0x0003 | Model name | ro | 2 | Device name |
| 0x0004 | FW version | ro | 2 | Firmware version of the device |
| 0x0005 | Stack version | ro | 2 | Version of the CC-Link component |

## Acyclic I/O communication (Device Code 0x00AC)

| Address (Add) | Read access | Write access | Data length in word (Len) | Content | Description |
|---|---|---|---|---|---|
| 0xACAC | Open Connection | | 1 | 0xAD00… 0xADFF, 0x0000 | A read access to this address opens an acyclic connection or returns an error. A valid connection handle is 0xAD00…0xADFF, or 0 in case of failure. |
| 0xACAC | | Close Connection | 1 | 0xAD00… 0xADFF, 0xFFFF | Write access to this address closes an acyclic connection. Writing a previously opened connection address (0xAD00…0xADFF) closes this connection. If the value -1 (0xFFFF) is written, all acyclic connections opened for CC-Link are closed. |
| 0xAD00 … 0xADFF | | | 1…240 | Acyclic data | |

**Example access:**

1. **Open Connection**:
   Device Read (0x0401)
   Device Code = 0xAC
   Add = 0xACAC
   Len =1
   Result: 0xAD00 = Connection address: must be used for the following connection accesses, like read, write and close.

2. **Read Connection:**
   Device Read (0x0401)
   Device Code = 0xAC
   Add = 0xAD00
   Len =1
   Result: n words of received frame. The requested length is the maximum buffer size. If the available acyclic data does not fit in the buffer, the exceeding data is truncated.

3. **Write Connection**:
   Device Read (0x1401)
   Device Code = 0xAC
   Add = 0xAD00
   Len =1
   Result: n words of data to be sent.

4. **Close Connection**:
   Device Write (0x1401)
   Device Code = 0xAC
   Add=0xACAC,
   Len=1
   Data: 0xADxx (address of the previously used Open Connection)

### Read input data (Device Code 0x00D8)

| Address (Add) | Access type | Data length in word (Len) | Description |
|---|---|---|---|
| 0x0000 | ro | 1…n | Access to all input data of the device regardless of profiles and restrictions due to the number of occupied stations, order:<br>1. Data from RWr area<br>2. Data from RX area |
| 0x0001<br>…<br>0x00… | ro | 1…n | Accesses the input data of one (sub)module. Data is structured in the native order of that (sub)module. |

### Write output data (Device Code 0x00D9)

| Address (Add) | Access type | Data length in word (Len) | Description |
|---|---|---|---|
| 0x0000 | rw | 1…n | Access to all output data of the device regardless of profiles and restrictions due to the number of occupied stations,<br>order:<br>1. Data from RWw area<br>2. Data from RY area |
| 0x0001<br>…<br>0x00… | rw | 1…n | Accesses the output data of one (sub)module. Data is structured in the native order of that (sub)module. |

### Read Diagnostic data data (Device Code 0x00DD)

| Address (Add) | Access type | Data length in word (Len) | Description |
|---|---|---|---|
| 0x0000 | ro | 1…n | Access to all diagnostic data of the device regardless of profiles and restrictions due to the number of occupied stations |
| 0x0001<br>…<br>0x00… | ro | 1…n | Accesses the diagnostic data data of one (sub)module. Data is structured in the native order of that (sub) module. |

## 7.9 Connecting devices to a CC-Link IE Field Basic client with GX Works3

### Naming convention

Turck uses the terms "client" and "server". The following description uses the terms "Master Station" (client) and "Slave Station" (server) only because of the naming in Melsoft GX Works.

### Used hardware

The following hardware components are used in this example:

- Mitsubishi MELSEC iQ-R controller
- Mitsubishi CPU 04ENCPU with local CC-Link IOs
- TBEN modules (as example)
  - TBEN-LL-8DIP-8DOP (IP address: 192.168.3.10)
  - TBEN-S2-4IOL (IP address: 192.168.3.12)

### Used software

The following software tools are used in this example:

- Melsoft GX Works3

### Prerequisites

- The GX Works3 software is open and a new project has been created.
- The controller including CPU and local IOs is configured in GX Works3.

### 7.9.1 Register the CSP+ files in GXWorks3

▶ Select and register CSP+ files via **Tools** → **Profile Management** → **Register**.
Note: CSP+ files can only be registered in GX Works3 if no project is open.

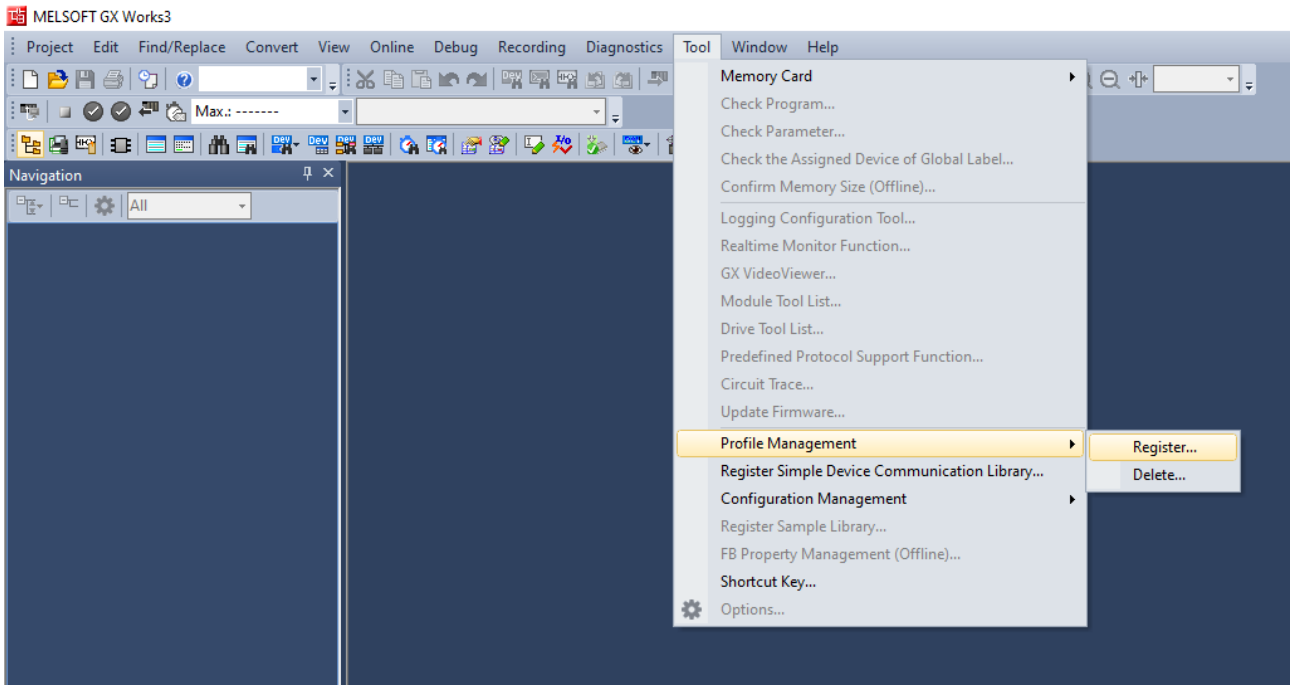



Fig. 68: Profile Management, Register Profile

## 7.9.2 Configuring the network settings

The network settings are configured at the CPU used under **Parameter → used CPU** (here: R04ENCPU) → **Module Parameters**.

### Setting the IP address of the CPU

- Set the IP address of the CPU under **Own Node Settings → IP Address**.

### Activate CC-Link Field Basic

The CC-Link IEF Basic protocol must be activated in the CPU.

- Under **CC-Link IEF Baisc Settings**, set the option **To Use or Not to Use CC-Link IEF Basic Setting** to **Use** in order to activate

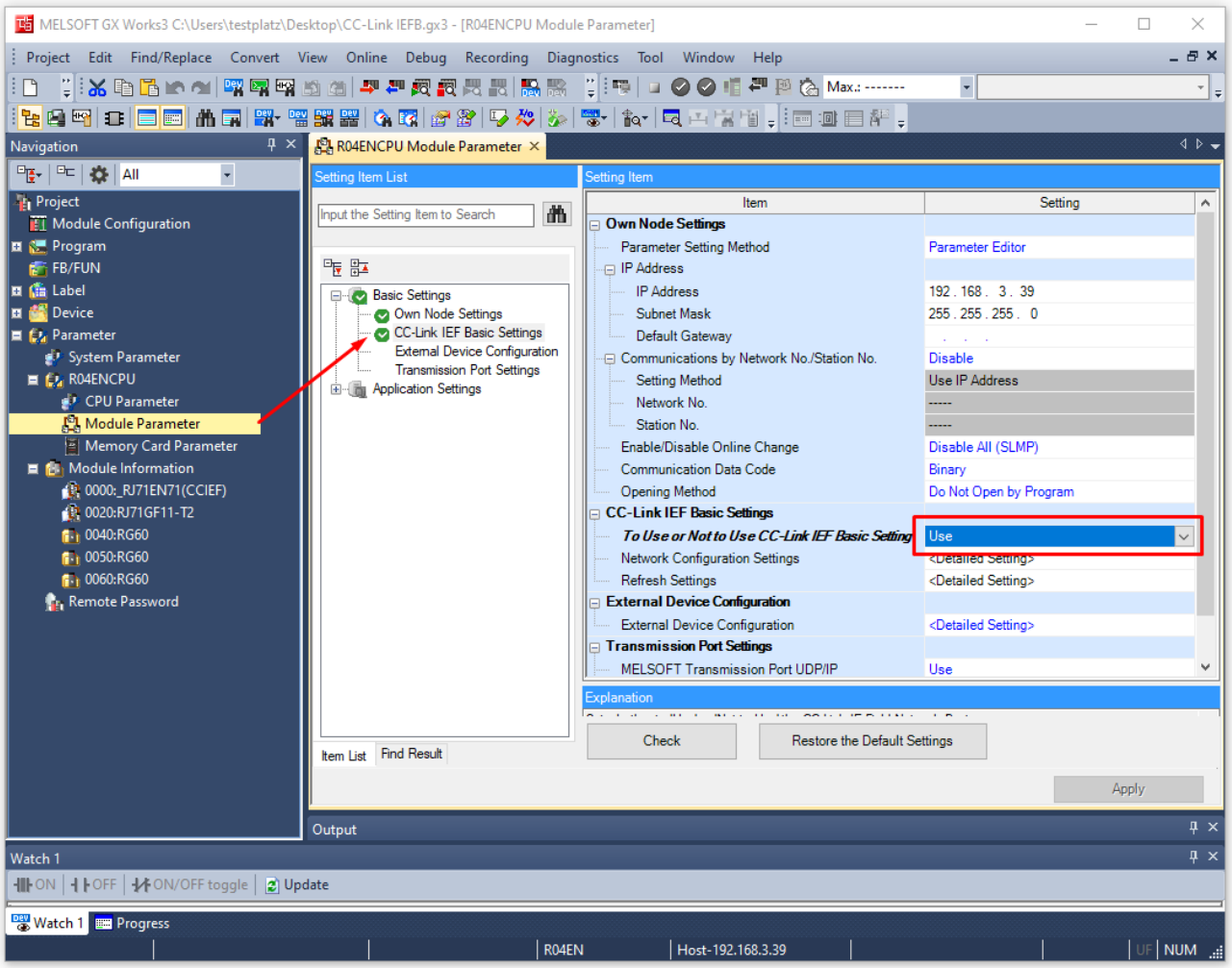



Fig. 69: GX Works3: Activate CC-Link IEF Basic on CPU

### 7.9.3 Configuring the CC-Link IE Field Basic network

#### Scanning the network

- Open **Network Configuration Settings** under **Module Parameters → CC-Link IEF Basic Settings**.

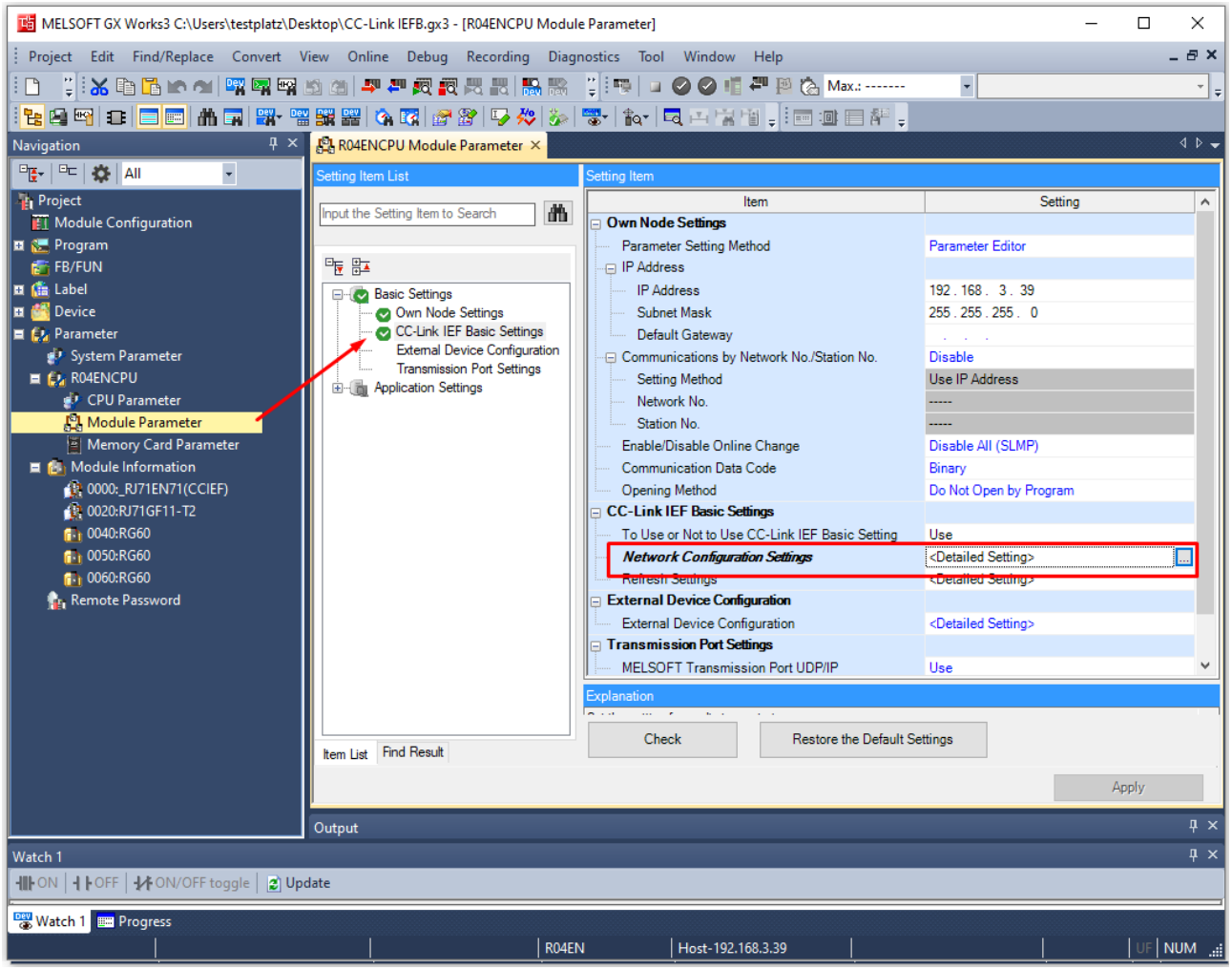



Fig. 70: GX Works3: Network Configuration Settings

▶ Scan the CC-Link IEF Basic network under **CC-Link IEF Basic Configuration** via **Detect Now**.

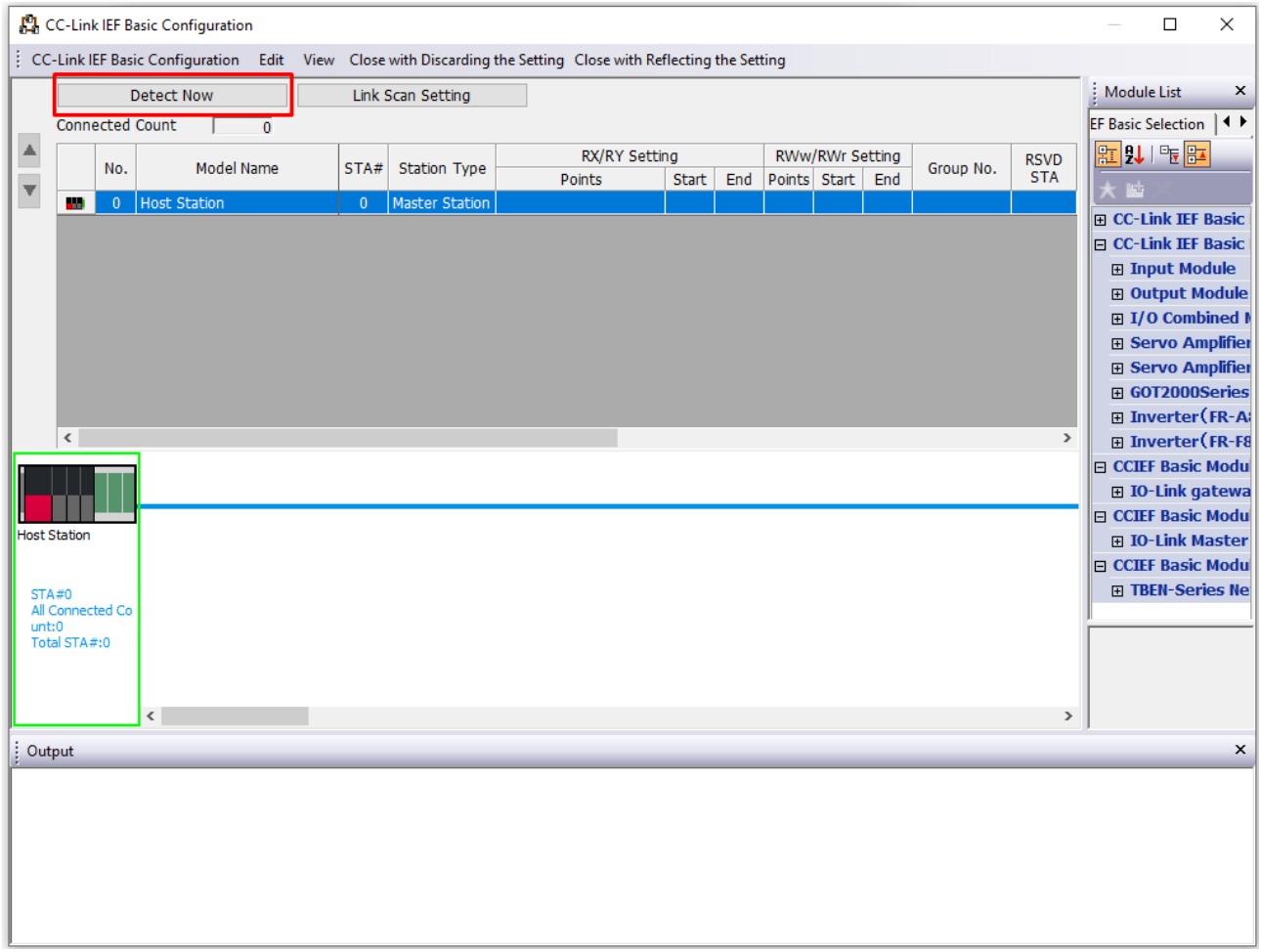



Fig. 71: GX Works3: scanning the CC-Link IEF Basic network

⇨ All CC-Link devices found in the Ethernet network are displayed in the order in which they are integrated in the network

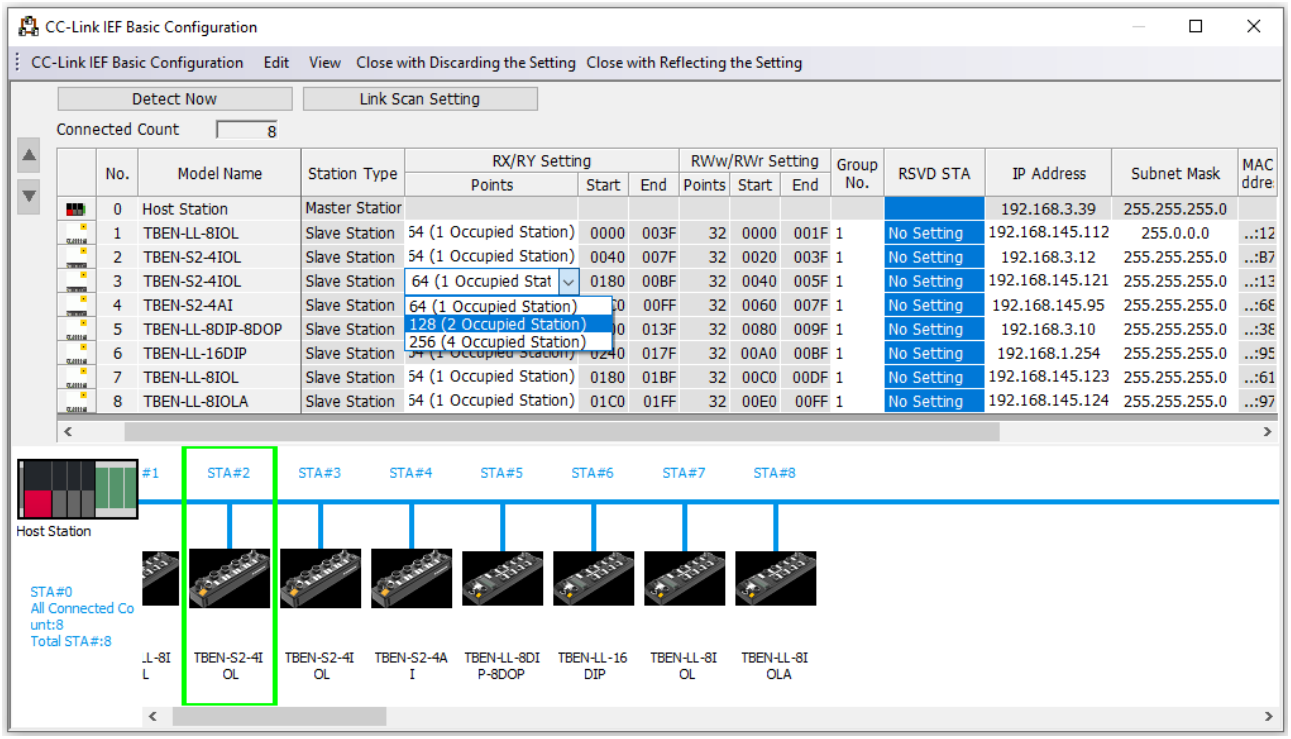


Fig. 72: GX Works3: Devices in the CC-Link IEF Basic network

Devices that do not match the IP address range of the controller cannot be added to the project.

- ▶ Remove devices with an IP address outside the IP address range of the controller from the list of network participants by right-clicking on the device → **Delete** or adjust the IP address of the devices in the **IP Address** column.
- ▶ For devices that can be integrated with different process data variables (profiles): select the profile under **RX/RY Setting** → **Points**.

## Parameterizing CC-Link nodes

- Right-click on the device to be parameterized and select the device parameters via **Online → Parameter Processing of Slave Station**.

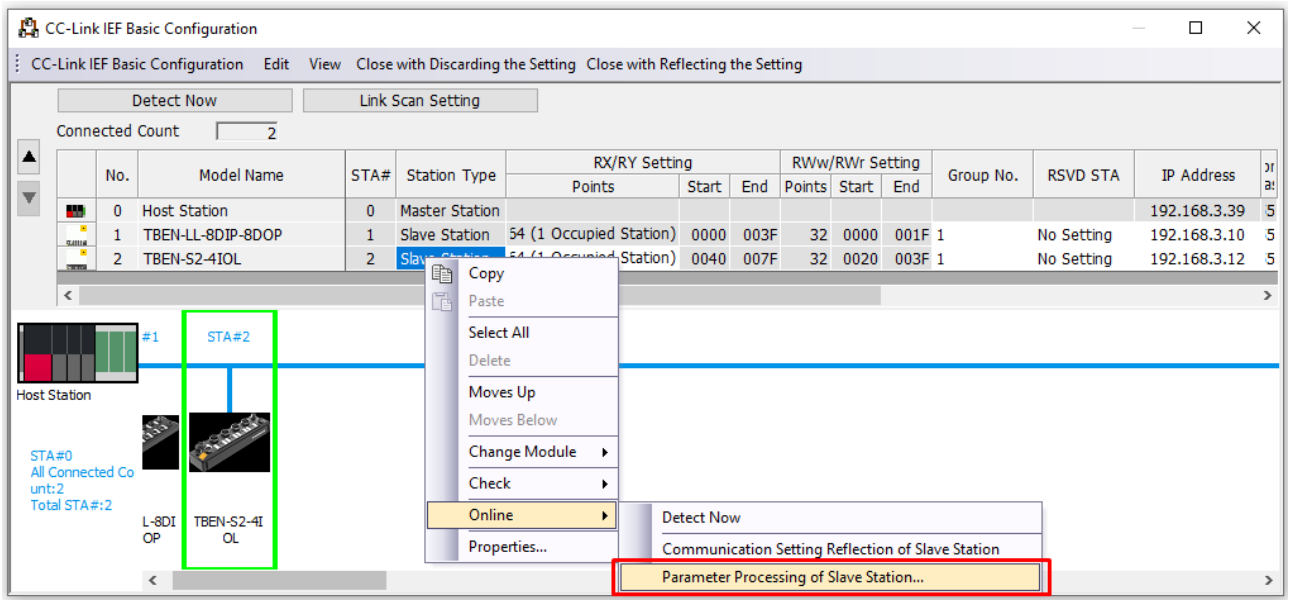



Fig. 73: GX Works3: Opening parameterization

- Activate the writing of parameters via **Method selection → Parameter write**.

**NOTE**
All parameters for one slot (in the example below: Slot 1) must be set. It is not possible to set individual parameters for a slot.

▶ Set the parameters and store the settings via **Execute**.

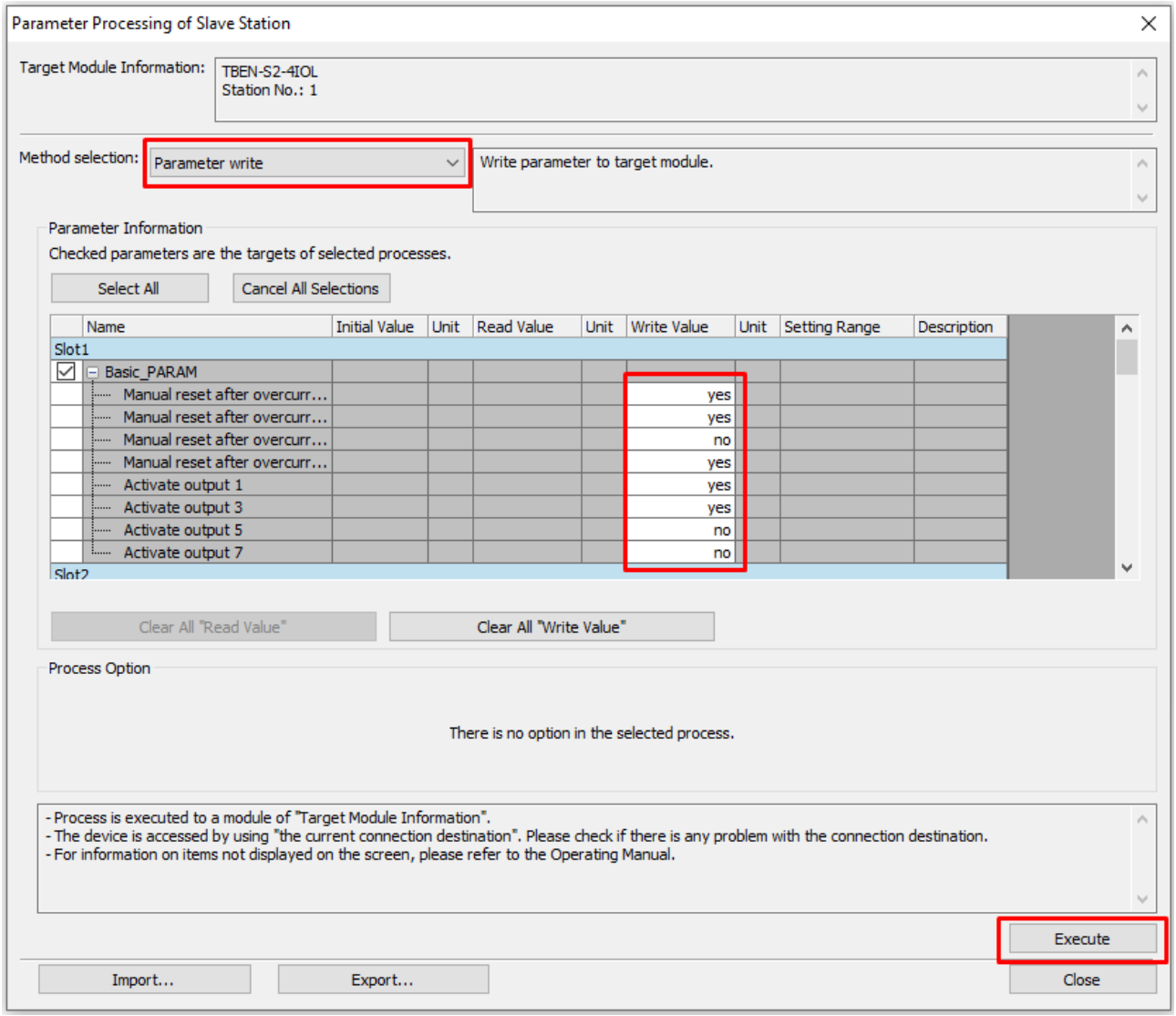



Fig. 74: GX Works3: Parameterizing the device

- Optional: Export the parameter settings as CSV file via **Method selection** → **Parameter read** and re-import them via **Method selection** → **Parameter write** in order to fill the **Write Values** column with the current parameter settings and then set individual parameters.
- Close the window and store the network structure via **CC-Link IEF Basic Configuration** über **Close with Reflecting the Setting**.

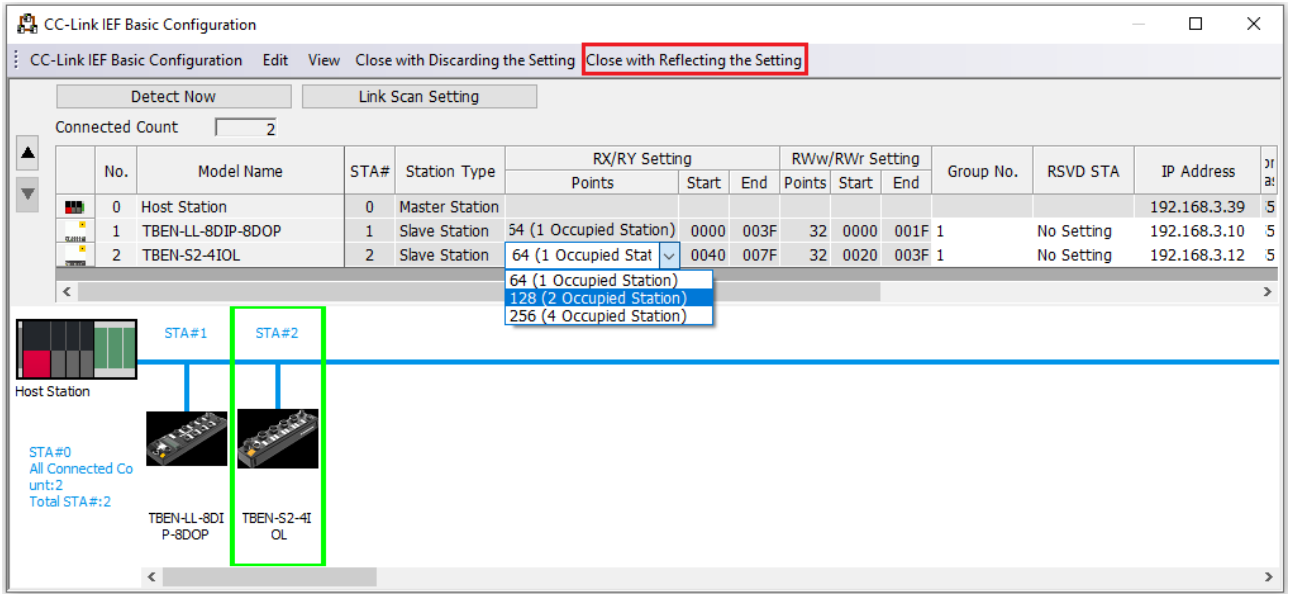



Fig. 75: GX Works3: Storing the network structure

- Apply changes to the network structure under **Module Parameters** with**Apply** .

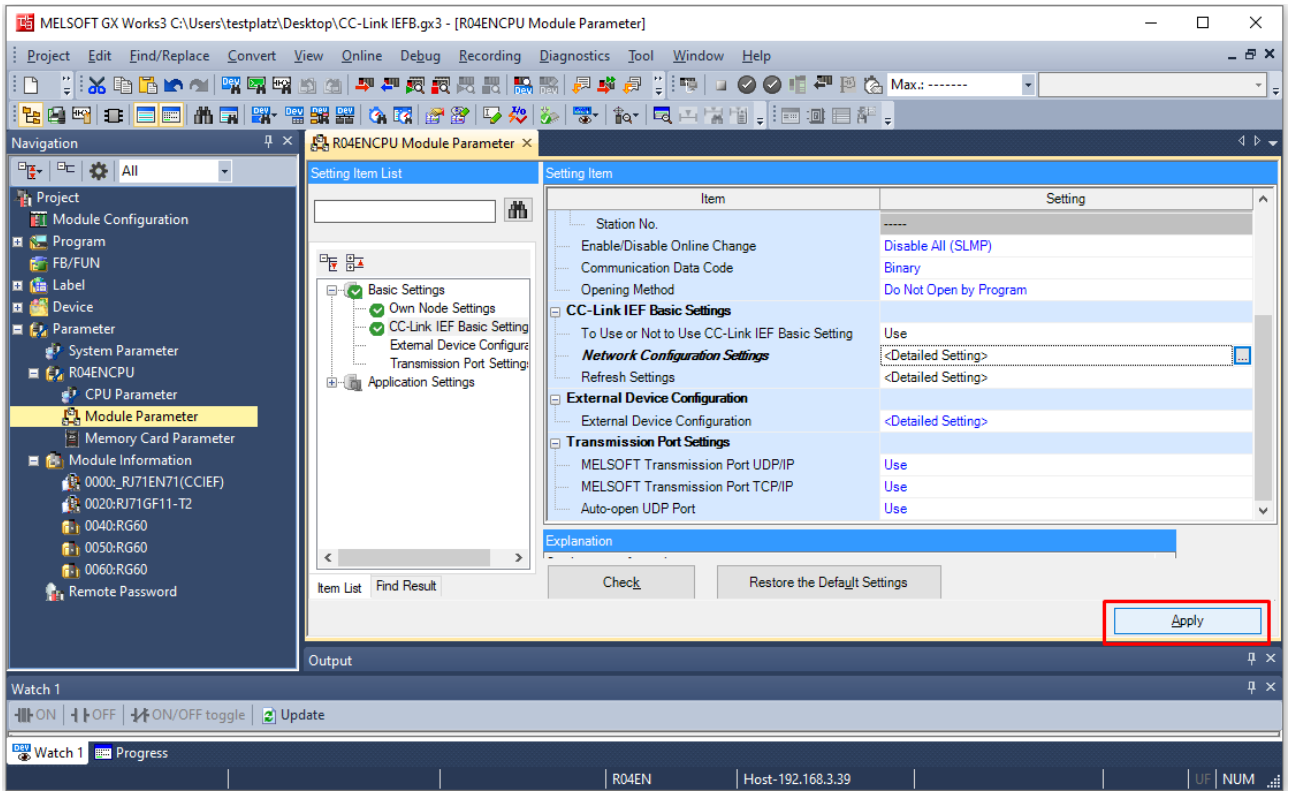



Fig. 76: GX Works3: Module Parameters, accept changes

### 7.9.4 Defining the process data mapping for CC-Link devices in the network

The start addresses of the process data for the devices that follow the **Master Station (Client)** (controller + local IOs) in the network are defined under **Module Parameters → CC-Link IEF Basic Settings** using the **Refresh Settings** function.

- Open the **Refresh Settings** function under **Module Parameters → CC-Link IEF Basic Settings**.
- Define the start addresses for the process data of the CC-Link devices in **CPU side**. **Check** can be used to verify whether the addresses are valid or overlap with the memory area occupied by the control unit.
- Accept the mapping settings with **Apply**.

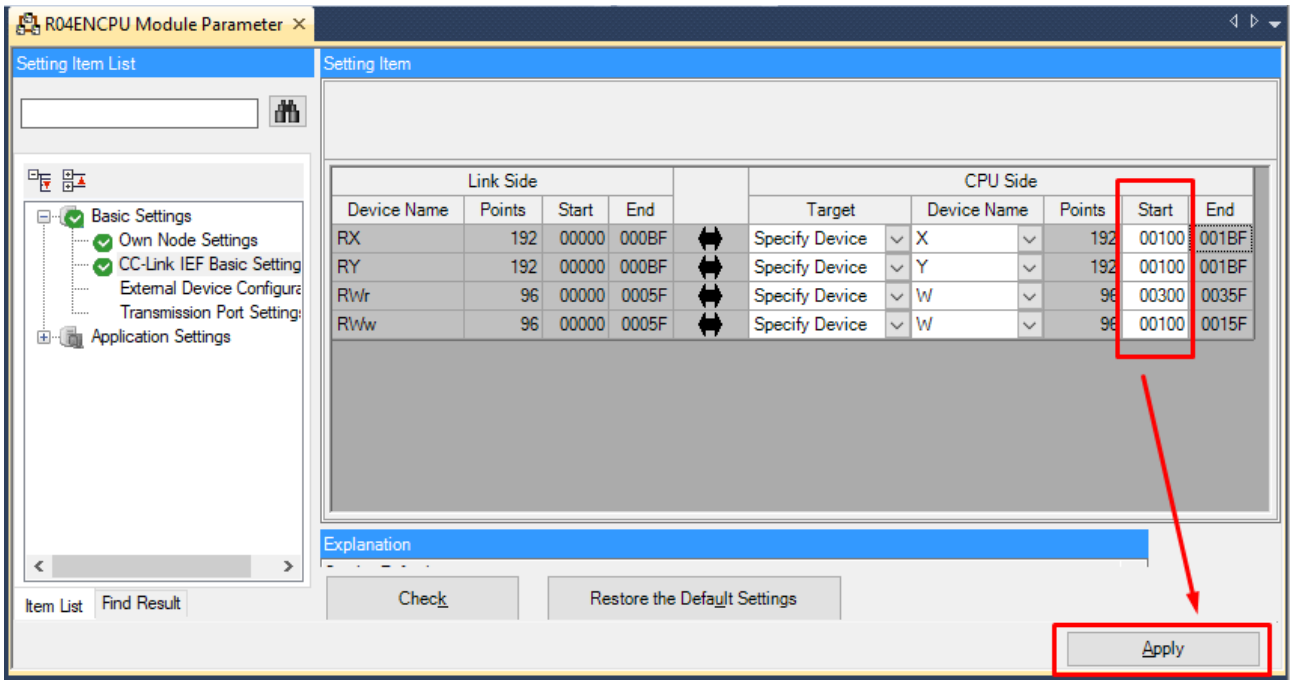



Fig. 77: GX Works3: Process data mapping in Refresh Settings

**NOTE**
Adjusting the mapping may require a voltage reset of the control unit.

### 7.9.5 Going online with the PLC

▶ Write the configuration to the PLC via **Online → Write to PLC**.

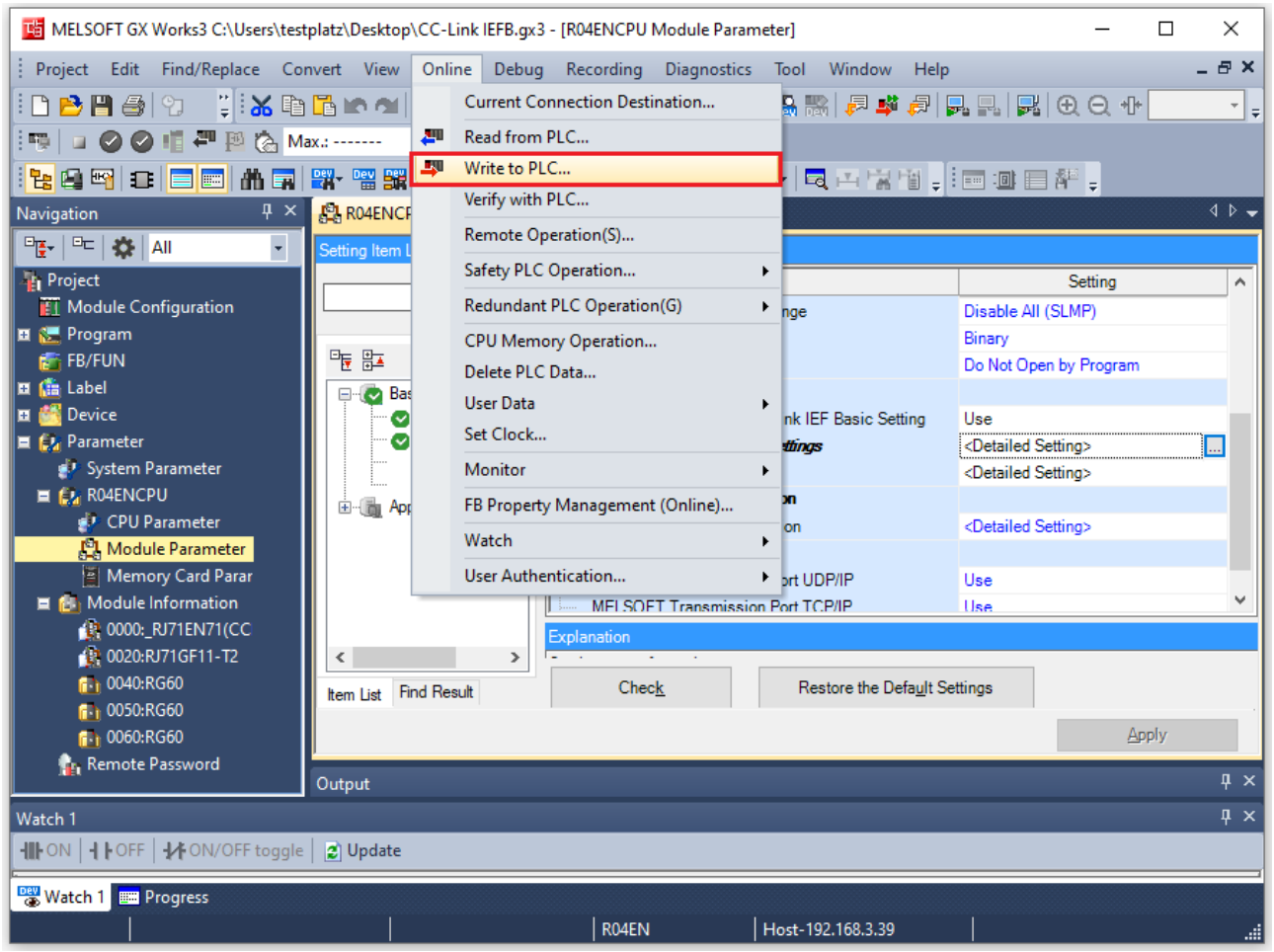



Fig. 78: GX Works3: Writing the configuration to the PLC

▶ If necessary, define which data have to be written and click **Execute**.

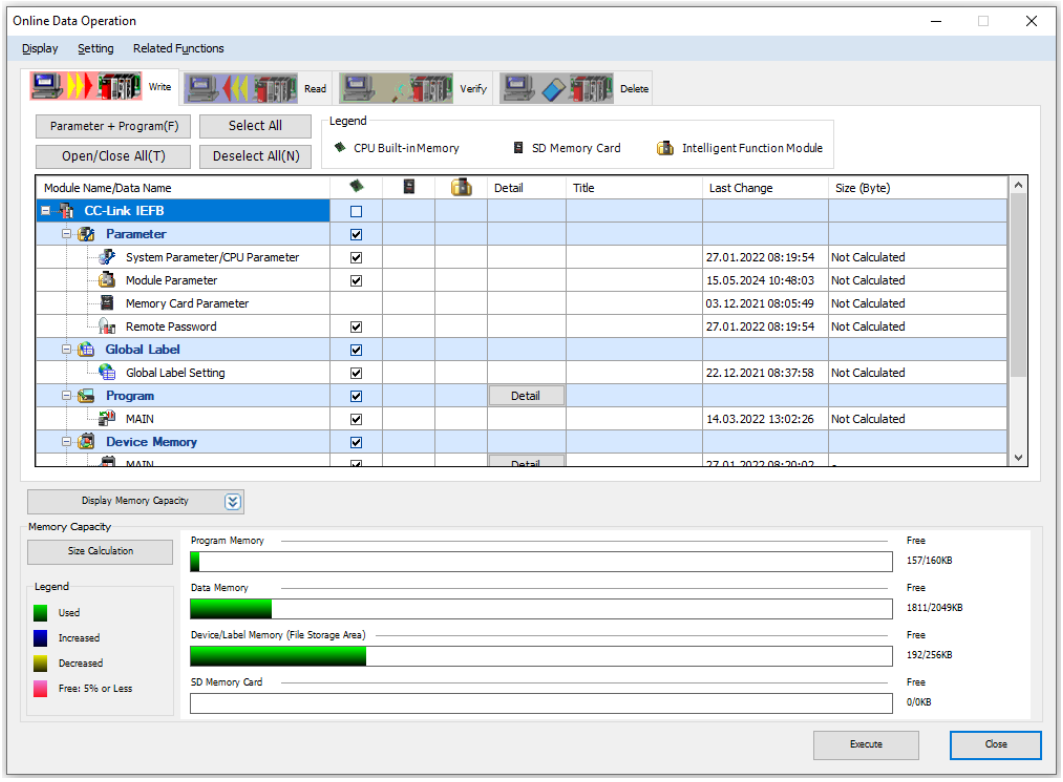



Fig. 79: GX Works3: Selecting data to be written

### 7.9.6 Reading process data

The monitoring of process data is done in the **Device/Buffer Memory Batch Monitor**.

▶ Open the monitoring via **Online** → **Monitor** → **Device/Buffer Memory Batch Monitor**.

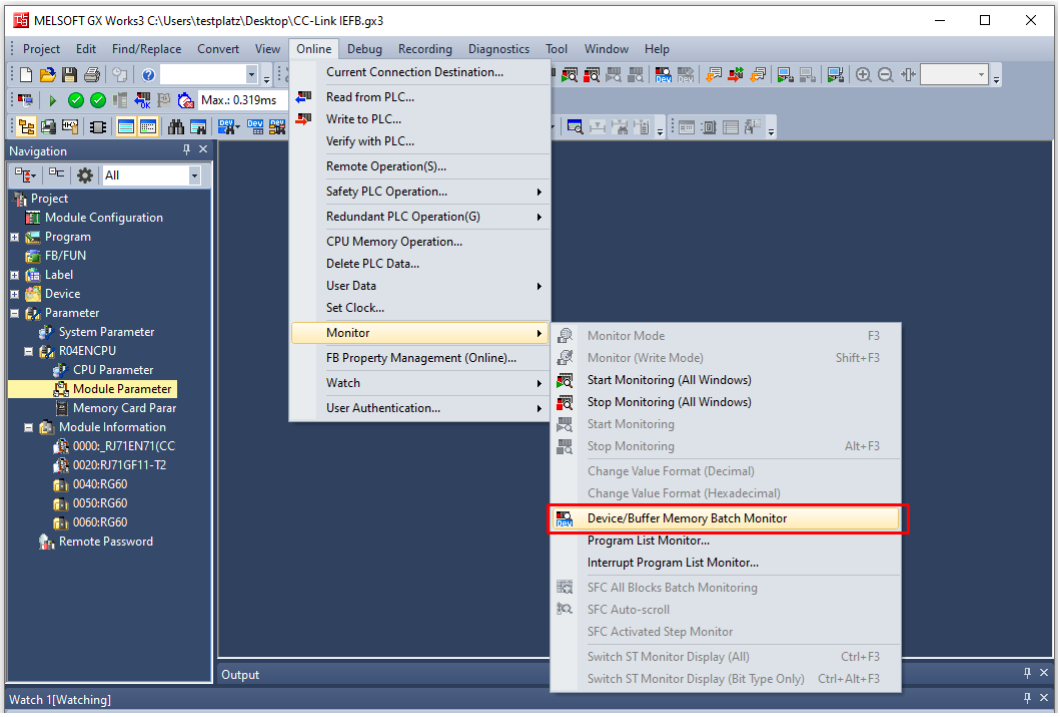



Fig. 80: GX Works3: Starting the monitoring of process data

▶ Enter the address of the process data to be read under **Device Name**. According to the defined process data mapping [▷ 129] **X100** is selected as start address.

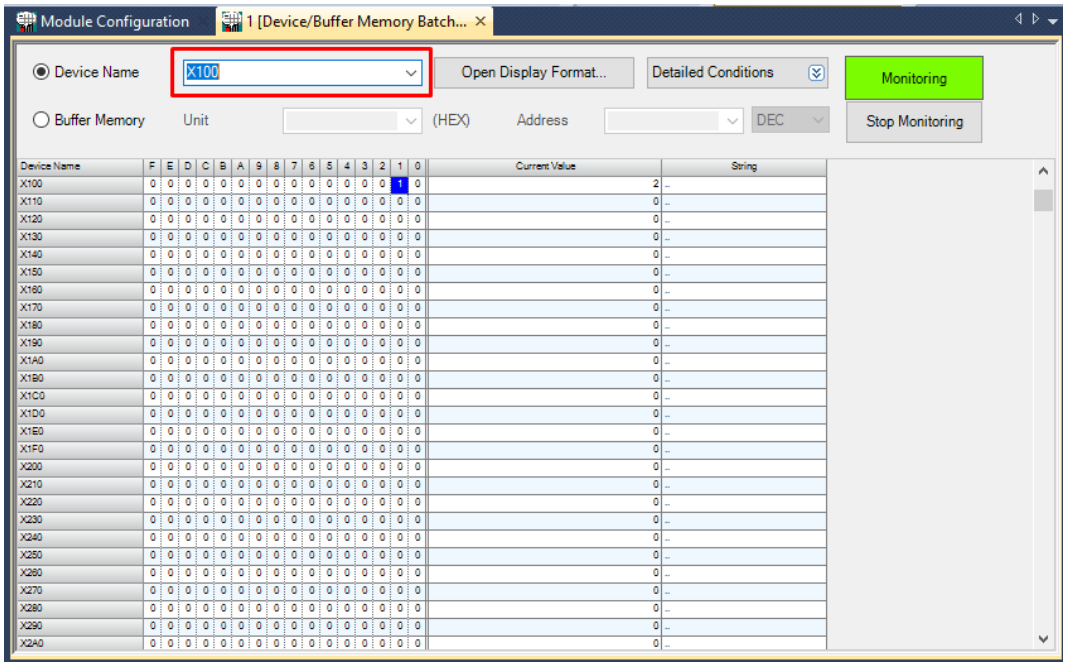



Fig. 81: GX Works3: Monitoring of process data

⇨ The mapping shows a signal at the 2nd Digital input of the 1st CC-Link device (station address 2, TBEN-LL-8DIP-8DOP) [▷ 123].

# 8 Parameterizing and configuring

## 8.1 Parameters for COM0/COM1

| Word no. | | Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM0** | | | | | | | | | | | |
| 0 | 0x00 | 0 | 0x00 | Data rate | | | | Swap A/B line | Operation mode | | |
| | | 1 | 0x01 | Biasing active | Termination active | EOF detection | | Stop bits | Character format | | |
| 1 | 0x01 | 2 | 0x02 | Reserved | | | | | | VAUX1 | |
| | | 3 | 0x03 | Reserved | | | | | | | |
| 2 | 0x02 | 4 | 0x04 | Character timeout | | | | | | | |
| | | 5 | 0x05 | | | | | | | | |
| 3 | 0x03 | 6 | 0x06 | Response timeout | | | | | | | |
| | | 7 | 0x07 | | | | | | | | |
| 4 | 0x03 | 8 | 0x08 | 1st end delimiter | | | | | | | |
| | | 9 | 0x09 | 2nd end delimiter | | | | | | | |
| 5 | 0x05 | 10 | 0x0A | MB server cycle time (* ms) | | | | | | | |
| | | 11 | 0x0B | | | | | | | | |
| **SCB (Server Configuration Block) COM0** | | | | | | | | | | | |
| 6…9 | 0x06 … 0x09 | 12… 19 | 0x0C … 0x13 | SCB0 | | | | | | | |
| … | … | … | … | … | | | | | | | |
| 34… 37 | 0x22 … 0x25 | 68… 75 | 0x44 … 0x4B | SCB7 | | | | | | | |
| **COM1** | | | | | | | | | | | |
| 38… 43 | 0x26 … 0x2B | 76… 87 | 0x4C … 0x57 | Parameters COM1 (assignment acc. to COM0, byte 0…11) | | | | | | | |
| 44… 75 | 0x2C … 0x4B | 88… 151 | 0x58 … 0x97 | SCB0…SCB7, COM1 (assignment acc. to COM0, byte 12…75) | | | | | | | |
| **DXP channels** | | | | | | | | | | | |
| 76 | 0x4C | 152 | 0x98 | SRO7 | SRO6 | SRO5 | SRO4 | - | | | |
| | | 153 | 0x99 | EN_DO7 | EN_DO6 | EN_DO5 | EN_DO4 | - | | | |
| 77 | 0x4D | 154 | 0x9A | - | | | | Input filter (DXP4) | DIF pulses (DXP4) | | |
| | | 155 | 0x9B | Pulse stretch (DXP4) | | | | | | | |
| … | … | … | … | … | | | | | | | |
| 80 | 0x50 | 160 | 0xA0 | - | | | | Input filter (DXP7) | DIF pulses (DXP7) | | |
| | | 161 | 0xA1 | Pulse stretch (DXP7) | | | | | | | |

## Meaning of parameter bits (COM0/COM1)

Default values are **shown in bold.**

| Parameter | Value | | Description | |
|---|---|---|---|---|
| | Dec. | Hex. | | |
| Operation mode | **0** | **0x00** | **RS485** | Operation mode for the COM0 or COM1 channel. |
| | 1 | 0x01 | RS232 | |
| | 2 | 0x02 | MB client RS485 | |
| | 3 | 0x03 | MB client RS232 | |
| Swap A/B line | Changes the outputs polarity of the A/B lines and switches the bias level. | | | |
| | **0** | **0x00** | **No** | Standard configuration, A = pin 2, B = pin 4 |
| | 1 | 0x01 | Yes | A = pin 4, B = pin 2 |
| Data rate | 0…3 | 0x00… 0x03 | Reserved | Data rate of the serial interface |
| | 4 | 0x04 | 2400 bps | |
| | 5 | 0x05 | 4800 bps | |
| | **6** | **0x06** | **9600 bps** | |
| | 7 | 0x07 | 14400 bps | |
| | 8 | 0x08 | 19200 bps | |
| | 9 | 0x09 | 28800 bps | |
| | 10 | 0x0A | 38400 bps | |
| | 11 | 0x0B | 57600 bps | |
| | 12 | 0x0C | 115200 bps | |
| | 13 | 0x0D | 230400 bps | |
| | 14, 15 | 0x0E, 0x0F | Reserved | |
| Character format | 0 | 0x00 | 7O | Parity and the number of bits per sign<br>■ N: no parity<br>■ O: odd parity (1 bit error detection)<br>■ E: even parity (1 bit error detection) |
| | 1 | 0x01 | 7E | |
| | 2 | 0x02 | 8N | |
| | 3 | 0x03 | 8O | |
| | **4** | **0x04** | **8E** | |
| Stop bits | **0** | **0x00** | **1 bit** | Number of stop bits |
| | 1 | 0x01 | 2 bit | |
| EOF detection | 0 | 0x00 | Character timeout | If, after receiving a character, no further character is received for a period longer than the character delay time, the end of a data packet is recognized. |
| | 1 | 0x01 | 1st end delimiter | The end of the frame is detected, as soon as the end delimiter was received. |
| | 2 | 0x02 | 2nd end delimiter | The end of the frame is detected, as soon as the two end delimiters were received. |
| | **3** | **0x03** | **Frame length** | The end of the frame is detected, as soon as the defined frame length was received. |

| Parameter | Value | | Description | |
|---|---|---|---|---|
| | Dec. | Hex. | | |
| Termination active | **0** | **0x00** | **Yes** | Activates or deactivates the termination resistor of the RS485-line. If the termination resistor is deactivated, the module can be operated inside a RS485-line. |
| | 1 | 0x01 | No | |
| Biasing active | **0** | **0x00** | **Yes** | Activates the biasing resistor. |
| | 1 | 0x01 | No | Deactivated the biasing resistor. |
| Power supply | **0** | **0x00** | **0 V (High-Z)** | Voltage level at pin 1 referred to GND at pin 3 |
| | 1 | 0x01 | 24 VDC | |
| | 2 | 0x02 | +5 VDC | |
| Character timeout | 0… 65535 | 0x0000… 0xFFFF (**0x0064**) | Character timeout in ms | |
| Response timeout | 0… 65535 | 0x0000… 0xFFFF (**0x03E8**) | The response timeout is used to configure a timeout for the receiving of a frame. The timeout starts whenever the receive sequence for the COM port is activated via the control bit "receive". The status bit "timeout" is set to TRUE for one cycle whenever the the response time expires. The "RX" LED shortly flashes red. The receive sequence has to be restarted.<br>0: no timeout<br>0x03E8 = 1000 ms | |
| 1st end delimiter | 0…255 | 0x00… 0xFF (**0x03**) | Defines first end character for recognizing the end of a data packet. Is only interpreted if the parameter "EOF detection" is set to 1 end delimiter or 2 end delimiter. | |
| 2nd end delimiter | **0**…255 | **0x00**… 0xFF | Defines second end character for recognizing the end of a data packet. Is only interpreted if the parameter "EOF detection" is set to 2 end delimiters. | |
| Time between frames | **0**… 65535 | **0x0000**… 0xFFFF | Time between requests from the Modbus client to the Modbus server [ms].<br>In exceptional cases, Modbus servers cannot process requests that are too fast. This leads to communication errors. In this case, the time has to be be increased.<br>0 = best update time possible | |

## 8.2 Parameters for Server Configuration Block (SCB)

| Byte offset | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Server address | | | | | | | |
| 1 | Number reg./server read access | | | | Number reg./server write access | | | |
| 2 | Read access | | | | | | | |
| 3 | Write access | | | | | | | |
| 4 | Start address for read access | | | | | | | |
| 5 | | | | | | | | |
| 6 | Start address for write access | | | | | | | |
| 7 | | | | | | | | |

### Meaning of parameter bits (Server Configuration Block)

Default values are **shown in bold.**

| Parameter | Value | | Description |
|---|---|---|---|
| | Dec. | Hex. | |
| Server address | 0… 255 | 0x00… 0xFF (**0x01**) | ■ Standard mode [▶ 148]: address of the connected Modbus RTU server<br>■ Multi server mode [▶ 149]: start address of the first of the connected Modbus RTU server |
| Number reg./ server read access | 0…**12** | 0x0… **0xC** | ■ Standard mode [▶ 148]: number of registers to be read or written<br>■ Multi server mode [▶ 149]: number of servers from which data have to be read |
| Number reg./ server write access | 0…**12** | 0x0… **0xC** | ■ Standard mode [▶ 148]: number of registers to be read or written<br>■ Multi server mode [▶ 149]: number of servers to which data have to be sent |

| Parameter | Value | | Description | |
|---|---|---|---|---|
| | Dec. | Hex. | | |
| Read access | 0 | 0x00 | Deactivated | Valid for the standard mode [▶ 148]: Defines the read access to the configured Modbus server. |
| | 3 | 0x03 | Read holding registers (FC3) | |
| | 4 | 0x04 | Read input registers (FC4) | |
| | 23 | 0x17 | Read and write multiple registers (FC23) | |
| | 128 | 0x80 | Read extension | Valid for the multi server mode [▶ 149]: Extension of the read command for the connection of Modbus RTU servers with more than 12 registers. The write extension provides up to 12 further registers for process input data and can only be selected as an extension of a “MB_Server” in the preceding slot (Server Configuration Block). |
| | 131 | 0x83 | Multi server mode: Read 1 holding register (FC3) | 1, 2, 3 or 4 registers are read or read and written per connected Modbus RTU server. |
| | 132 | 0x84 | Multi server mode: Read 1 input register (FC4) | |
| | 151 | 0x97 | Multi server mode: Read and write 1 registers (FC23) | |
| | 163 | 0xA3 | Multi server mode: Read 2 holding register (FC3) | |
| | 164 | 0xA4 | Multi server mode: Read 2 input register (FC4) | |
| | 183 | 0xB7 | Multi server mode: Read and write 2 registers (FC23) | |
| | 195 | 0xC3 | Multi server mode: Read 3 holding register (FC3) | |
| | 196 | 0xC4 | Multi server mode: Read 3 input register (FC4) | |
| | 215 | 0xD7 | Multi server mode: Read and write 3 registers (FC23) | |
| | 227 | 0xE3 | Multi server mode: Read 4 holding register (FC3) | |
| | 228 | 0xE4 | Multi server mode: Read 4 input register (FC4) | |
| | 247 | 0xF7 | Multi server mode: Read and write 4 registers (FC23) | |

| Parameter | Value | | Description | |
|---|---|---|---|---|
| | Dec. | Hex. | | |
| Write access | 0 | 0x00 | **Deactivated** | Valid for the standard mode [▶ 148]:<br>Defines the write access to the configured Modbus server. |
| | 6 | 0x06 | Write 1 output register (FC6) | |
| | 16 | 0x10 | Write multiple output registers (FC16) | |
| | 23 | 0x17 | Read and write multiple registers (FC23) | |
| | 128 | 0x80 | Write extension | Valid for the multi server mode [▶ 149]:<br>Extension of the write command for the connection of Modbus RTU Servers with more than 12 registers. The write extension provides up to 12 further registers for process input data and can only be selected as an extension of a "MB_Server" in the preceding slot (Server Configuration Block). |
| | 134 | 0x86 | Multi server mode:<br>Write 1 output register (FC6) | 1, 2, 3 or 4 registers are written or read and written per connected Modbus RTU server. |
| | 144 | 0x90 | Multi server mode:<br>Write 1 output registers (FC16) | |
| | 151 | 0x97 | Multi server mode:<br>Read and write 1 registers (FC23) | |
| | 176 | 0xB0 | Multi server mode:<br>Write 2 output registers (FC16) | |
| | 183 | 0xB7 | Multi server mode:<br>Read and write 2 registers (FC23) | |
| | 208 | 0xD0 | Multi server mode:<br>Write 3 output registers (FC16) | |
| | 215 | 0xD7 | Multi server mode:<br>Read and write 3 registers (FC23) | |
| | 240 | 0xE3 | Multi server mode:<br>Write 4 output registers (FC16) | |
| | 247 | 0xF7 | Multi server mode:<br>Read and write 4 registers (FC23) | |
| Start address for read access | **0**… 65535 | **0x0000**… 0xFFFF | Address of the 1st register from which data have to be read | |
| Start address for write access | **0**… 65535 | **0x0000**… 0xFFFF | Address of the 1st register to which data have to be written | |

## 8.3 Parameters for the DXP channels

| Byte offset | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | SRO7 | SRO6 | SRO5 | SRO4 | - | | | |
| 1 | EN_DO7 | EN_DO6 | EN_DO5 | EN_DO4 | - | | | |
| 2 | - | | | | Input filter (DXP4) | DIF pulses (DXP4) | | |
| 3 | Pulse stretch (DXP4) | | | | | | | |
| 4 | - | | | | Input filter (DXP5) | DIF pulses (DXP5) | | |
| 5 | Pulse stretch (DXP5) | | | | | | | |
| 6 | - | | | | Input filter (DXP6) | DIF pulses (DXP6) | | |
| 7 | Pulse stretch (DXP6) | | | | | | | |
| 9 | - | | | | Input filter (DXP7) | DIF pulses (DXP7) | | |
| 10 | Pulse stretch (DXP7) | | | | | | | |

### Meaning of parameter bits (DXP channels)

The default values are written in **bold**.

| Parameter | Value | | Description | |
|---|---|---|---|---|
| | Dec. | Hex. | | |
| Manual reset after overcurr. Ch… (SRO) | 0 | 0x00 | **No** | The output switches on automatically after an overload. |
| | 1 | 0x01 | Yes | The output is manually switched-off after an overload until a new set command is given. |
| Activate output Ch… (EN_DO) | 0 | 0x00 | No | The output at pin 2 is deactivated. |
| | 1 | 0x01 | **Yes** | The output at pin 2 is activated. |
| DIF pulses (DXP…) | 0 | 0x00 | **Deactivated** | Activates or deactivates the extended functions (input filter and pulse stretch) for the respective digital channel [▶ 12]. |
| | 1 | 0x01 | Input filter and pulse stretch | |
| Input filter (DXP…) | 0 | **0x00** | **0.2 ms** | Filter time 0.2 ms |
| | 1 | 0x01 | 3 ms | The filter time of digital inputs is extended to 3 ms. Digital input signals can thus be reliably detected even when short-term interfering signals in rough environments occur. |
| Pulse stretch (DXP…) (*10 ms) | **0**… 255 | **0x00**… 0xFF | | Configures the duration of the pulse stretching of digital input edges from 10 to 2550 ms in multiples of 10 ms.<br>10 = pulse of 100 ms<br>0 = Impulse stretch deactivated |

# 9 Operating

## 9.1 Process input data: overview

| Word no. | | Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Process input data COM ports** | | | | | | | | | | | |
| 0…2 | 0x00…<br>0x02 | 0…5 | 0x0000…<br>0x0005 | Status messages COM0,<br>RS232/RS485 mode [▶ 141] or Modbus client mode [▶ 143] | | | | | | | |
| 3…98 | 0x03…<br>0x62 | 6…<br>197 | 0x0006…<br>0x00C5 | Receive data COM0,<br>192 bytes depending on the configuration of COM0 | | | | | | | |
| 99…<br>101 | 0x63…<br>0x65 | 198…<br>203 | 0x00C6…<br>0x00CB | Status messages COM1,<br>RS232/RS485 mode [▶ 141] or Modbus client mode [▶ 143] | | | | | | | |
| 102…<br>197 | 0x66…<br>0xC5 | 204…<br>395 | 0x00CC…<br>0x018B | Receive data COM1<br>192 bytes depending on the configuration of COM1 | | | | | | | |
| **Diagnostics for COM ports** [▶ 156] | | | | | | | | | | | |
| 198 | 0xC6 | 396 | 0x018C | COM channel diagnostics COM0 | | | | | | | |
| | | | | Over-current VAUX1 | Reserved | | | | | Parame-terization error | Hard-ware error |
| | | 397 | 0x018D | Error MB server 7 | Error MB server 6 | Error MB server 5 | Error MB server 4 | Error MB server 3 | Error MB server 2 | Error MB server 1 | Error MB server 0 |
| 199 | 0xC7 | 398 | 0x018E | COM channel diagnostics COM1 | | | | | | | |
| | | | | Over-current VAUX1 | Reserved | | | | | Parame-terization error | Hard-ware error |
| | | 399 | 0x018F | Error MB server 7 | Error MB server 6 | Error MB server 5 | Error MB server 4 | Error MB server 3 | Error MB server 2 | Error MB server 1 | Error MB server 0 |
| **Process Input data data Modbus server** [▶ 143] | | | | | | | | | | | |
| 200…<br>215 | 0xC8…<br>0xD7 | 400…<br>431 | 0x0190…<br>0x01AF | Status messages Modus server,<br>(depending on the parameterization, only valid for Modbus client mode) | | | | | | | |
| 216…<br>231 | 0xD8…<br>0xE7 | 432…<br>463 | 0x01B0…<br>0x01CF | Modbus server timing | | | | | | | |
| **Process input data DXP channels** | | | | | | | | | | | |
| 232 | 0xE8 | 464 | 0x01D0 | DXP7 | DXP6 | DXP5 | DXP4 | Reserved | | | |
| | | 465 | 0x01D1 | Reserved | | | | | | | |
| **DXP channel diagnostics** [▶ 156] | | | | | | | | | | | |
| 233 | 0xE9 | 466 | 0x01D2 | Reserved | | | | Over-current VAUX2 Ch6/Ch7 | Over-current VAUX2 Ch4/Ch5 | Reserved | |
| | | 467 | 0x01D3 | Overcurrent output | | | | Reserved | | | |
| | | | | Ch7 | K6 | K5 | K4 | | | | |
| **Module status** [▶ 155] | | | | | | | | | | | |
| 234 | 0xEA | 468 | 0x01D4 | V2 | Reserved | | | | - | ARGEE | DIAG |
| | | 469 | 0x01D5 | res. | FCE | Reserved | | | COM | V1 | - |

### 9.1.1 Process input data: RS232 or RS485 mode

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM0** | | | | | | | | | |
| 0 | 0x0000 | Invalid RX length | Invalid TX length | Timeout | Buffer overflow | Parity/format error | Frame error | Receive complete | Transmitter ready |
| 1 | 0x0001 | Reserved | | | | | | | |
| 2 | 0x0002 | Received frame length | | | | | | | |
| 3…5 | 0x0003… 0x0005 | Reserved | | | | | | | |
| 6… 197 | 0x0006… 0x00C5 | Receive data COM0, maximum length 8 × 24 bytes | | | | | | | |
| **COM1** | | | | | | | | | |
| 198 | 0x00C6 | Invalid RX length | Invalid TX length | Timeout | Buffer overflow | Parity/format error | Frame error | Receive complete | Transmitter ready |
| 199 | 0x00C7 | Reserved | | | | | | | |
| 200 | 0x00C8 | Received frame length | | | | | | | |
| 201 … 203 | 0x00C9… 0x00CB | Reserved | | | | | | | |
| 204 … 395 | 0x00CC… 0x018B | Receive data COM1, maximum length 8 × 24 bytes | | | | | | | |

#### Meaning of the process input data in RS232/RS485 mode

| Process value | Value | Description |
|---|---|---|
| Receive data | | [▶ 142] |
| Receive complete | 0 | No valid message received |
| | 1 | The bit is set to TRUE after a message was sent. The bit remains TRUE until the bit "Receive" is set to FALSE. A new receive sequence (Bit "Receive" FALSE → TRUE) resets the bit. |
| Received frame length | 0…192 | Length of the last message which has been received |
| Parity/format error | 0 | No error |
| | 1 | Parity or format error<br>A new receive sequence (Bit "Receive" FALSE → TRUE) resets the bit. |
| Frame error | 0 | No error |
| | 1 | Frame error<br>Possible causes:<br>■ 1st or 2nd end delimiter not valid<br>■ The effective frame length does not match the parameterized frame length.<br><br>A new receive sequence (Bit "Receive" FALSE → TRUE) resets the bit. |

| Process value | Value | Description |
|---|---|---|
| Transmitter ready | 0 | The transmitter is ready. |
| | 1 | The bit is set to TRUE after a message was sent. It indicates that the transmission was completed and that he next send sequence can be started. The bit remains TRUE until the bit "send" has been reset to FALSE (acknowledgment). |
| Buffer overflow | 0 | No error |
| | 1 | Buffer overflow during receive sequence<br>A new receive sequence (Bit "Receive" FALSE → TRUE) resets the bit. |
| Timeout | 0 | No error |
| | 1 | Response timeout: This bit is only used in case of a response time set to > 0.<br>A new receive sequence (Bit "Receive" FALSE → TRUE) resets the bit. |
| Invalid TX length | 0 | No error |
| | 1 | Invalid transmit length, permitted length: 1 to 192 bytes |
| Invalid RX length | 0 | No error |
| | 1 | Invalid receive length, permitted length: 1 to 192 bytes |

### Receive data in RS232/RS485 mode for COM0/COM1

| Byte offset | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dec. | Hex. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COM0 | | | | | | | | | |
| 0 | 0x00 | Byte 0: Receive buffer block 1 | | | | | | | |
| … | … | … | | | | | | | |
| 24 | 0x18 | Byte 23: Receive buffer block 1 | | | | | | | |
| 25 | 0x19 | Byte 0: Receive buffer block 2 | | | | | | | |
| … | … | … | | | | | | | |
| 48 | 0x30 | Byte 23: Receive buffer block 2 | | | | | | | |
| … | … | … | | | | | | | |
| 168 | 0xA8 | Byte 0: Receive buffer block 8 | | | | | | | |
| … | … | … | | | | | | | |
| 192 | 0xC0 | Byte 23: Receive buffer block 8 | | | | | | | |

### 9.1.2 Process input data: Modbus client mode

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM0** | | | | | | | | | |
| 0…3 | 0x0000…<br>0x0003 | Reserved | | | | | | | |
| 4 | 0x0004 | MB server cycle time COM0 | | | | | | | |
| 5 | 0x0005 | | | | | | | | |
| 6…197 | 0x0006…<br>0x00C5 | Receive data of the Modbus servers COM0, maximum length 8 × 12 byte | | | | | | | |
| **COM1** | | | | | | | | | |
| 198…<br>201 | 0x00C6…<br>0x00C9 | Reserved | | | | | | | |
| 202 | 0x00CA | MB server cycle time COM1 | | | | | | | |
| 203 | 0x00CB | | | | | | | | |
| 204…<br>395 | 0x00CC…<br>0x018B | Receive data of the Modbus servers COM1, maximum length 8 × 12 byte | | | | | | | |
| **Modbus server status** | | | | | | | | | |
| | | Server 0 (COM0) | | | | | | | |
| 400 | 0x0190 | MODBUS timeout Ch0 | Parity or format error Ch0 | Write error Ch0 | Read error Ch0 | Error code Ch0 | | | |
| 401 | 0x0191 | Reserved | | Valid write config. Ch0 | Valid read config. Ch0 | Reserved | | | |
| | | Server 0 (COM1) | | | | | | | |
| 402 | 0x0192 | MODBUS timeout Ch1 | Parity or format error Ch1 | Write error Ch1 | Read error Ch1 | Error code Ch1 | | | |
| 103 | 0x0193 | Reserved | | Valid write config. Ch1 | Valid read config. K1 | Reserved | | | |
| 404…<br>415 | 0x0194…<br>0x019F | 2 bytes each for server 2… server 7 (COM0) | | | | | | | |
| 416…<br>431 | 0x0120…<br>0x01AF | 2 bytes each for server 0…server 7 (COM1) | | | | | | | |
| **Modbus server timing** | | | | | | | | | |
| 432 | 0x01B0 | Server 0 (COM0) | | | | | | | |
| 433 | 0x01B1 | | | | | | | | |
| … | … | … | | | | | | | |
| 446 | 0x01BE | Server 7 (COM0) | | | | | | | |
| 447 | 0x01BF | | | | | | | | |
| 448 | 0x01C0 | Server 0 (COM1) | | | | | | | |
| 449 | 0x01C1 | | | | | | | | |
| … | … | … | | | | | | | |
| 462 | 0x01CE | Server 7 (COM1) | | | | | | | |
| 463 | 0x01CF | | | | | | | | |

### Meaning of the process input data in Modbus client mode

| Process value | Value | Description |
|---|---|---|
| Receive data | | s. [▶ 144] |
| Modbus server status | | |
| Error code | | Modbus Exception Code |
| Read error Ch… | 0 | No error |
| | 1 | Modbus read error |
| Write error Ch… | 0 | No error |
| | 1 | Modbus write error |
| Parity or format error Ch… | 0 | No error |
| | 1 | Modbus parity or format error |
| Modbus timeout Ch… | 0 | No error |
| | 1 | The Modbus server did not respond within the defined time. |
| Valid read config. Ch… | 0 | Invalid read configuration |
| | 1 | Read configuration valid |
| Valid write config. Ch… | 0 | Invalid write configuration |
| | 1 | Write configuration valid |
| MB-Server cycle time (*1 ms) | | Update time [ms] with which the Modbus RTU client requests data from all connected Modbus RTU servers. |
| MB-Server Timing time (*1 ms) | | Update time [ms] of the connected Modbus RTU server at COM0 or COM1. |

**NOTE**
The Modbus Exception Codes are defined in the description of "Modbus Application Protocol V1" at http://www.modbus.org

### Receive data of the Modbus servers at COM0 and COM1

Depending on the parameterization, 1 or 12 registers are received per Modbus server.

| Register offset | | Bit no. | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dec. | Hex. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COM0 | | | | | | | | | | | | | | | | | |
| 0 | 0x00 | Modbus server 0, input register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 11 | 0x0B | Modbus server 0, input register 11 | | | | | | | | | | | | | | | |
| 12 | 0x0C | Modbus server 1, input register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 23 | 0x17 | Modbus server 1, input register 11 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 84 | 0x54 | Modbus server 7, input register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 95 | 0x5F | Modbus server 7, input register 11 | | | | | | | | | | | | | | | |

## 9.2 Process output data: overview

| Word no. | | Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Process output data COM ports** | | | | | | | | | | | |
| 0…2 | 0x00…<br>0x02 | 0…5 | 0x0000…<br>0x0005 | Control data COM0,<br>RS232/RS485 mode [▶ 145] or Modbus client modbus client mode [▶ 146] | | | | | | | |
| 3…98 | 0x03…<br>0x62 | 6…<br>197 | 0x0006…<br>0x00C5 | Receive data COM0,<br>192 bytes depending on the configuration of COM0 | | | | | | | |
| 99…<br>101 | 0x63…<br>0x65 | 198…<br>203 | 0x00C6…<br>0x00CB | Control data COM0,<br>RS232/RS485 mode [▶ 145] or Modbus client mode [▶ 146] | | | | | | | |
| 102…<br>197 | 0x66…<br>0xC5 | 204…<br>395 | 0x00CC<br>…0x018B | Receive data COM0,<br>192 bytes depending on the configuration of COM0 | | | | | | | |
| **DXP channels** | | | | | | | | | | | |
| 198 | 0xC6 | 396 | 0x018C | DXP7 | DXP6 | DXP5 | DXP4 | Reserved | | | |
| | | 397 | 0x018D | Reserved | | | | | | | |

### 9.2.1 Process output data: RS232 or RS485 mode

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM0** | | | | | | | | | |
| 0 | 0x0000 | Reserved | | | | | | Receive | Transmit |
| 1 | 0x0001 | Reserved | | | | | | | |
| 2 | 0x0002 | Transmit frame length | | | | | | | |
| 3 | 0x0003 | Reserved | | | | | | | |
| 4 | 0x0004 | Receive frame length | | | | | | | |
| 5 | 0x0005 | Reserved | | | | | | | |
| 6…197 | 0x0006…<br>0xC5 | Receive data COM0, maximum length 8 × 24 bytes | | | | | | | |
| **COM1** | | | | | | | | | |
| 198 | 0x00C6 | Reserved | | | | | | Receive | Transmit |
| 199 | 0x00C7 | Reserved | | | | | | | |
| 200 | 0x00C8 | Transmit frame length | | | | | | | |
| 201 | 0x00C9 | Reserved | | | | | | | |
| 202 | 0x00CA | Receive frame length | | | | | | | |
| 203 | 0x00CB | Reserved | | | | | | | |
| 204…395 | 0x00CC…<br>0x018B | Receive data COM0, maximum length 8 × 24 bytes | | | | | | | |

Meaning of the process Output data data in RS232/RS485 mode

| Process value | Value | Description |
|---|---|---|
| Receive | 0 | Preparation for new receive sequence |
| | 1 | The bit is set to TRUE to start the receive sequence. This bit has to be set to FALSE after every received frame until the status bit "Receive complete" is FALSE. |
| Receive frame length | 1…192 | Number of the characters to be send in bytes |
| Transmit | 0 | New transmit sequence possible |
| | 1 | The bit is set to TRUE to start the transmission. |
| Transmit frame length | 1…192 | Defines the number of characters to be received for the next message. Is only evaluated if the parameter "EOF detection" is set to the value "frame length". |

Transmit data in RS232/RS485 mode for COM0/COM1

| Byte offset | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dec. | Hex. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **COM0** | | | | | | | | | |
| 0 | 0x00 | Byte 0: transmit buffer block 1 | | | | | | | |
| … | … | … | | | | | | | |
| 24 | 0x18 | Byte 23: transmit buffer block 1 | | | | | | | |
| 25 | 0x19 | Byte 0: transmit buffer block 2 | | | | | | | |
| … | … | … | | | | | | | |
| 48 | 0x30 | Byte 23: transmit buffer block 2 | | | | | | | |
| … | … | … | | | | | | | |
| 168 | 0xA8 | Byte 0: transmit buffer block 8 | | | | | | | |
| … | … | … | | | | | | | |
| 192 | 0xC0 | Byte 23: transmit buffer block 8 | | | | | | | |

### 9.2.2 Process output data: Modbus client mode

| Byte no. | | Bit no. | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dec. | Hex. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **COM0** | | | | | | | | | |
| 0…5 | 0x0000…<br>0x0005 | Reserved | | | | | | | |
| 6…197 | 0x0006…<br>0x00C5 | Transmit data of the Modbus servers COM0, maximum length 8 × 12 byte | | | | | | | |
| **COM1** | | | | | | | | | |
| 198…203 | 0x00C6…<br>0x00CB | Reserved | | | | | | | |
| 204…395 | 0x00CC…<br>0x018B | Transmit data of the Modbus servers COM1, maximum length 8 × 12 byte | | | | | | | |

## Receive data of the Modbus servers at COM0 and COM1

Depending on the parameterization, 1 or 12 registers are received per Modbus server.

| **Register offset** | | **Bit no.** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dec.** | **Hex.** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **COM0** | | | | | | | | | | | | | | | | | |
| 0 | 0x00 | Modbus server 0, output register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 11 | 0x0B | Modbus server 0, output register 11 | | | | | | | | | | | | | | | |
| 12 | 0x0C | Modbus server 1, output register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 23 | 0x17 | Modbus server 1, output register 11 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 84 | 0x54 | Modbus server 7, output register 0 | | | | | | | | | | | | | | | |
| … | … | … | | | | | | | | | | | | | | | |
| 95 | 0x5F | Modbus server 7, output register 11 | | | | | | | | | | | | | | | |

## 9.3 Application example: Standard mode

- 1 Modbus RTU server per server configuration Block (SCB)
- max. 8 Modbus RTU servers per COM port

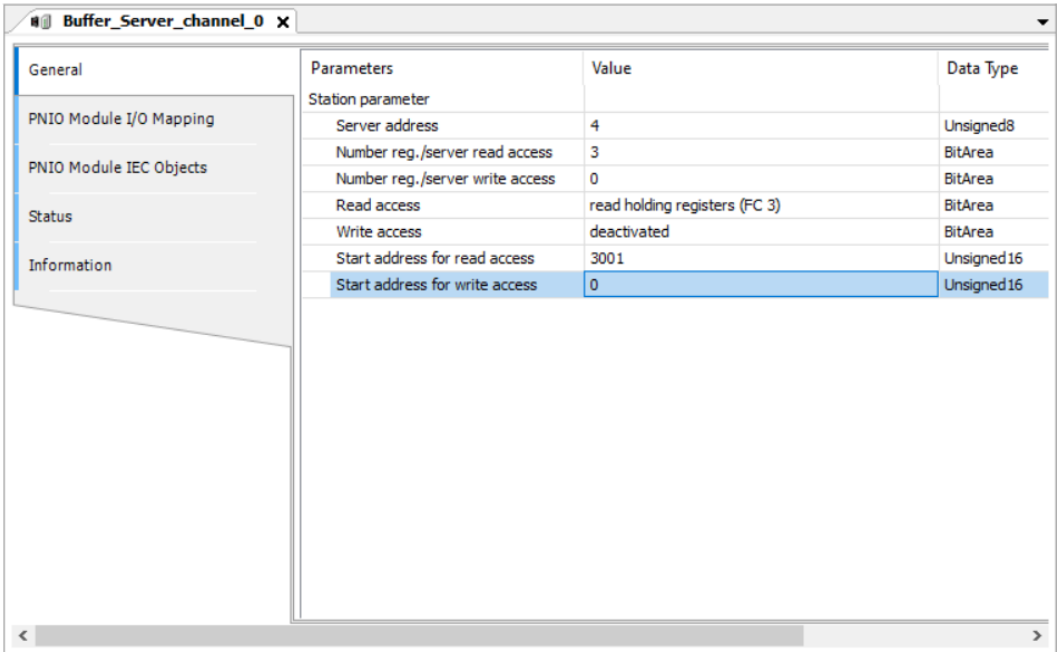



Fig. 82: Example configuration: Standard mode

| Parameter | Value | Meaning |
|---|---|---|
| Server address | 4 | Data of the Modbus RTU server with address 4 are read. |
| Number reg. read access | 3 | Three registers of the addressed Modbus RTU server are read. |
| Number reg. write access | 0 | No write access in the example |
| Read access | Read 1 holding register (FC3) | The holding register of the addressed Modbus RTU server is read. |
| Write access | Deactivated | No write access in the example |
| Start address for read access | 3001 | Address of the register in the addressed Modbus RTU server from which data have to be read |
| Start address for write access | 0 | No write access in the example |

## 9.4 Application example: Multi server mode

Recommended for applications with more than eight identical Modbus RTU servers per port.

- Data exchange with up to twelve identical Modbus RTU Servers per Server Configuration Block (SCB)
- Max. 32 Modbus RTU servers per COM port, in total max. 64 per TBEN-S2-2COM-4DXP device
  Note:
  Depending on the technical characteristics of the Modbus RTU Servers, the connection of up to 64 Modbus RTU Servers per port (128 per device) is possible.
- Prerequisite:
  The parameters “Read access” and “Write access” have both to be set to the Multi server mode or unused functions have been deactivated.
- Mixing of standard and multi-server mode is not permitted.

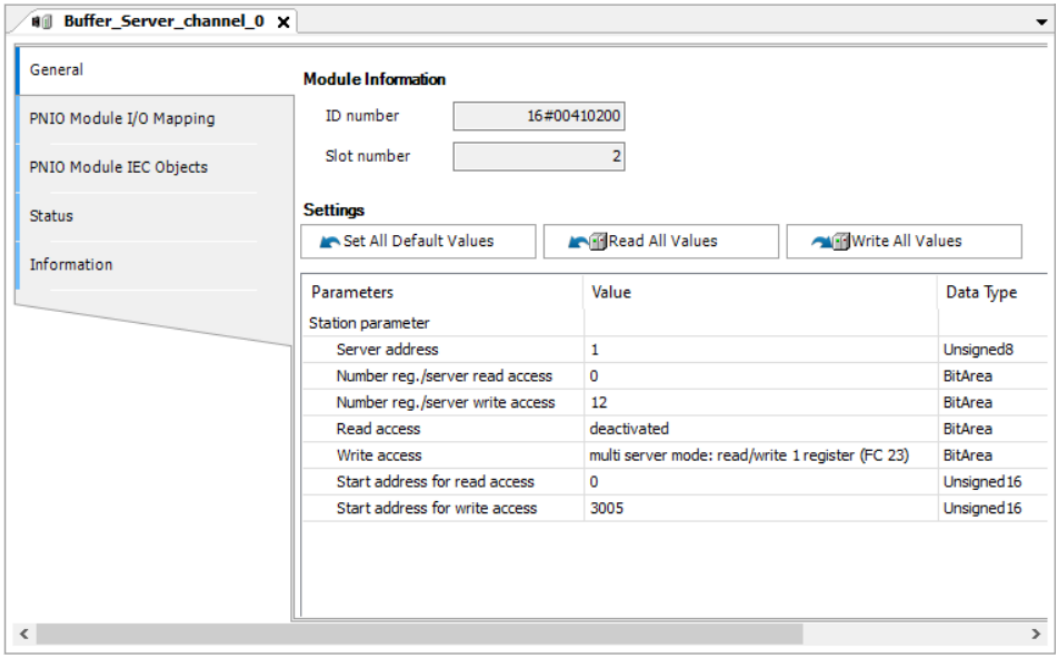



Fig. 83: Example configuration: Multi server mode

| Parameter | Value | Meaning |
|---|---|---|
| Server address | 1 | Address of the 1st Modbus RTU Server in the RS485 line |
| Number server read access | 0 | No read access in the example |
| Number server write access | 12 | Number of Modbus RTU servers from which data have to be read |
| Read access | Deactivated | No read access in the example<br>Can be used in parallel with the write access, but the setting has to correspond to the multi server mode (example: “Multi server mode: read 4 input registers“) |
| Write access | Multi server mode: Read and write 1 register (FC23) | One register is read and written from each of the twelve Modbus RTU servers (server 1 to server 12 in the RS485 line) |
| Start address for read access | 3005 | No read access in the example |
| Start address for write access | 0 | Address of the first register of all connected, identical Modbus RTU servers |

## 9.5 Application example: read/write extension

- Connection of Modbus RTU Servers with more than 12 registers, which have to be read or written.
- Extension of the read or write command to a Modbus RTU Server, which has been configured in the preceding SCB.

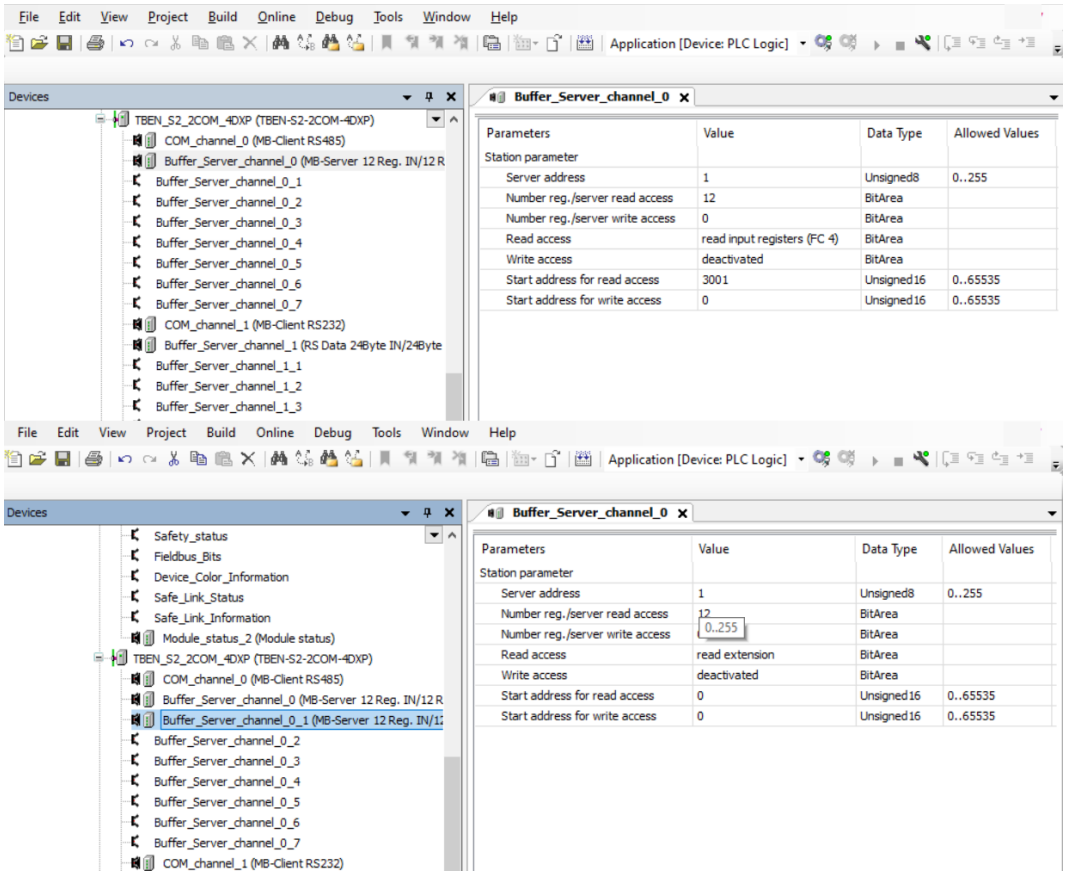



Fig. 84: Example configuration: Read extension

| Parameter 1st SCB | Value | Meaning |
|---|---|---|
| Server address | 1 | Address of the Modbus RTU server from which data have to be read |
| Number reg. read access | 12 | Number of registers to be read |
| Read access | Read input register (FC4) | |
| Start address for read access | 30001 | Address of the first register from which data have to be read |

| Parameter 2nd SCB | Value | Meaning |
|---|---|---|
| Server address | 0 | Not defined, automatically set |
| Number reg. read access | 12 | Number of registers to be read additionally |
| Read access | Read extension | Defines this SCB as extension for the preceding SCB. |
| Start address for read access | 0 | Not defined, automatically set |

## 9.6 Transmitting and receiving data

### 9.6.1 Transmit sequence

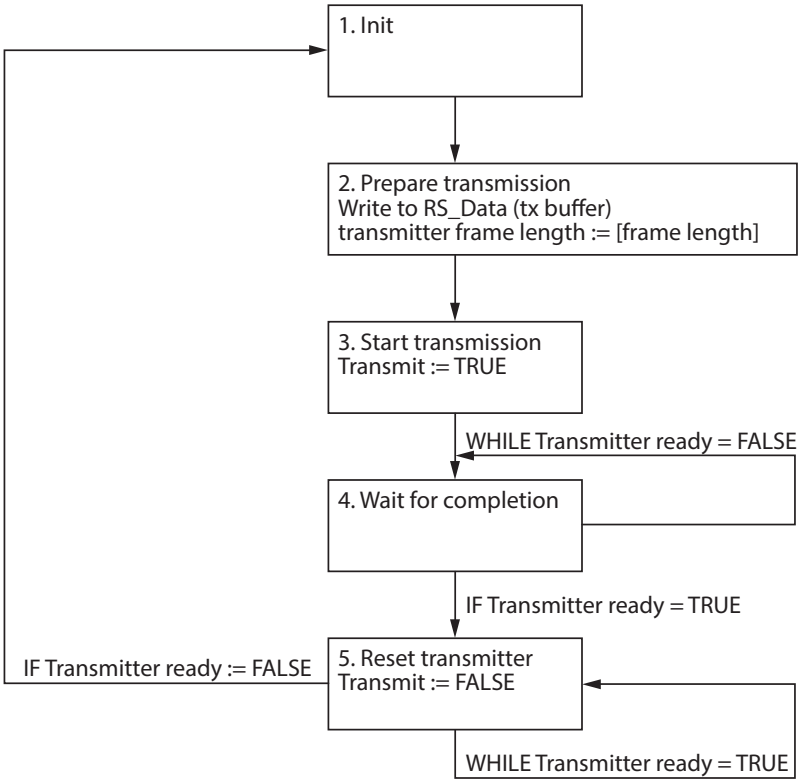



Fig. 85: Transmit sequence

Prerequisite:
Init: "Transmitter ready" is FALSE.

- Write transmit data (RS_Data) to the transmit buffer (TX buffer) and write the transmit data length in bytes to the process output value "Transmitter frame length".
- Set the process output value "Transmit" to TRUE.
- Wait until the process input value "Transmitter ready" = TRUE.
- Set the process output value "Transmit" to FALSE and and skip to the beginning for the next transmit sequence.

### 9.6.2 Receive sequence

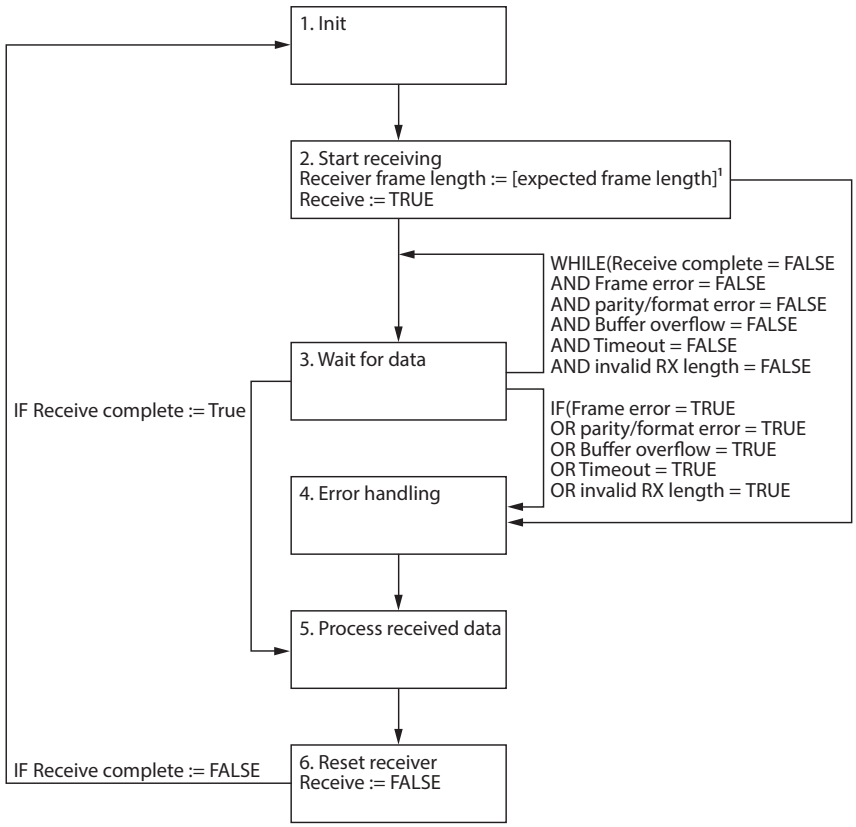



Fig. 86: Receive sequence

Prerequisite:
Init: "Receive complete" is FALSE.

- ▶ Set the process output value "Receive" to TRUE (starts the receiver).
- ▶ Wait until the process input value "Receive complete" = TRUE or until an error is signaled.
- ▶ Perform error handling if an error has been signaled.
- ▶ If no error occurs, read and process received data from the receive buffer.
- ▶ Set the process output value "Receive" to FALSE (stops the receiver) and wait until the process input value "Receive complete" = FALSE.
- ▶ Skip to the beginning for the next reception sequence.

Note:

- The receiver temporarily has to be deactivated between two transmit sequences (refer to steps 5 and 6). The duration for the deactivation depends on the update time set and the PLC cycle time. During this time, no data can be received.
- The data reception is limited to 192 bytes per telegram.

## 9.7 LED displays

The device has the following LED indicators:

- Power supply
- Group and bus errors
- Status
- Diagnostics

| LED PWR | Meaning |
|---|---|
| Off | No voltage connected or under voltage at V1 |
| Green | Voltage on V1 or respectively V1 and V2 OK |
| Red | No voltage connected or under voltage at V2 (only valid for devices with V2 power supply) |

| BUS LED | Meaning |
|---|---|
| Off | No voltage present |
| Green | Connection to a master active |
| Flashing 3 × green in 2 s | ARGEE active |
| Green flashing (1 Hz) | Device is operational |
| Red | IP address conflict, Restore mode active, F_Reset active or Modbus connection timeout |
| Red flashing | Wink command active |
| Red/green (1 Hz) | Autonegotiation and/or wait for IP address allocation in DHCP or BootIP mode |

| ERR LED | Meaning |
|---|---|
| Off | No voltage present |
| Green | No diagnostics |
| Red | Diagnostics present |

| ETH1 and ETH2 LEDs | Meaning |
|---|---|
| Off | No Ethernet connection |
| Green | Ethernet connection established, 100 Mbit/s |
| Green flashing | Data transfer, 100 Mbit/s |
| Yellow | Ethernet connection established, 10 Mbit/s |
| Yellow flashing | Data transfer, 10 Mbit/s |

| TX0 and TX1 | Meaning |
|---|---|
| Off | No serial communication |
| Green flashing | Serial data is sent. |

| RX0 and RX1 | Meaning |
|---|---|
| Off | No serial communication |
| Green flashing | Serial data is received. |
| Red flashing | Serial data is received, parity or format error. |
| Red | Overflow at receive buffer or timeout |

| RX and TX (in combination) | Meaning |
|---|---|
| Red flashing, simultaneously (1 Hz) | Overload of auxiliary voltage |
| Red flashing, alternately | Parameterization error |

| DXP channel LEDs | Meaning (input) | Meaning (output) |
|---|---|---|
| Off | No input signal | Output inactive |
| Green | input signal at DI | Output active |
| Red | – | Actuator overload |
| Red flashing (1 Hz) | Overload of the sensor supply<br>In devices with group diagnostics, all connector LEDs of the supply group flash simultaneously in case of an error. | |

| DXP7 | Meaning |
|---|---|
| White flashing | Wink command active |

## 9.8 Software diagnostic messages

The device provides the following software diagnostic messages:

- COM channel diagnostics
- DXP diagnostics
  diagnostic messages of the universal digital channels of the device
- V1/V2 overcurrent diagnostics
  Overcurrent diagnostics for VAUX1 and VAUX2

### 9.8.1 Status- and control word

#### Status word

| EtherNet/IP/ Modbus | PROFINET | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| Byte 0 | Byte 1 | V2 | - | - | - | - | - | ARGEE | DIAG |
| Byte 1 | Byte 0 | - | FCE | - | - | - | COM | V1 | - |

| Bit | Description |
|---|---|
| ARGEE | ARGEE program running |
| COM | Internal error<br>The device-internal communication is disturbed. |
| DIAG | Diagnostic messages at the device |
| FCE | The DTM Force Mode is activated. The actual output values may no match the ones defined and sent by the fieldbus. |

The status word is mapped into the module's process data.

In EtherNet/IP the mapping can be deactivated via the Gateway Class (VSC 100).

**NOTE**
Activating or deactivating the status and control word modifies the process data mapping in den standard Assembly Instances 103 and 104 EtherNet/IP standard classes, Assembly Object (0x04).

#### Control word

The control word has no function.

### 9.8.2 Diagnostic telegram

| Channel | Byte no. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| **COM channel diagnostics** | | | | | | | | | |
| COM0 | 0 | Over-current VAUX1 | - | | | | | Parame-terization error | Hardware error |
| | 1 | Error MB server 7 | Error MB server 6 | Error MB server 5 | Error MB server 4 | Error MB server 3 | Error MB server 2 | Error MB server 1 | Error MB server 0 |
| COM1 | 2 | Over-current VAUX1 | - | | | | | Parame-terization error | Hardware error |
| | 3 | Error MB server 7 | Error MB server 6 | Error MB server 5 | Error MB server 4 | Error MB server 3 | Error MB server 2 | Error MB server 1 | Error MB server 0 |
| **DXP channel diagnostics** | | | | | | | | | |
| DXP4… DXP7 | 4 | - | | | | Over-current VAUX2 Ch6/Ch7 | Over-current VAUX2 Ch4/Ch5 | Reserved | |
| | | Overcurrent output | | | | | | | |
| | 5 | K7 | K6 | K5 | K4 | Reserved | | | |

#### Meaning of diagnostic bits

| Bit | Value | Meaning |
|---|---|---|
| Error MB server… | 0 | No error |
| | 1 | Error at Modbus server… at respective COM port or Modbus server not accessible, possibly incorrect parameterization of the connection to the server. |
| Hardware error | 0 | No error |
| | 1 | Hardware error,, device replacement may be necessary |
| Parameterization error | 0 | No error |
| | 1 | Parameterization error<br>Possible causes:<br>■ Parameter "Termination active" activated in operation mode "RS232"<br>■ Parameter "Biasing active" activated in operation mode "RS232"<br>■ Invalid parameterization |
| Overcurrent supply VAUX1 | 0 | No error |
| | 1 | Overcurrent at COM port supply (pin 1) |
| Overcurrent VAUX2 Ch… | 0 | No error |
| | 1 | Overcurrent at the supply voltage at connector C2 (channel 4 or channel 5) or at connector C3 (channel 6 or channel 7). |
| Overcurrent output Ch… | 0 | No error |
| | 1 | Overcurrent at output of the channel |

# 10 Troubleshooting

If the device does not function as expected, first check whether ambient interference is present. If there is no ambient interference present, check the connections of the device for faults.

If there are no faults, there is a device malfunction. In this case, decommission the device and replace it with a new device of the same type.

If the device does not work as expected, proceed as follows:

- Exclude environmental disturbances.
- Check the connections of the device for errors.
- Check device for parameterization errors.

If the malfunction persists, the device is faulty. In this case, decommission the device and replace it with a new device of the same type.

# 11 Maintenance

Ensure regularly that the plug connections and cables are in good condition.

The devices are maintenance-free, clean dry if required.

## 11.1 Updating the firmware via TAS

**NOTICE**
Interruption of the power supply during the firmware update
**Risk of device damage due to faulty firmware update**

- ▶ Do not interrupt the power supply during the firmware update.
- ▶ During the firmware update do not reset the power supply.
- ▶ Do not interrupt the Ethernet connection during the firmware update.

**NOTE**
The firmware update function in TAS is locked when the controller connection is active. The device must first be disconnected from the controller before performing the update.

### Starting a firmware update for a device

- ▶ Open TAS.
- ▶ Open the network view and scan the network.
- ▶ Select the device.
- ▶ Click **Firmware update**.

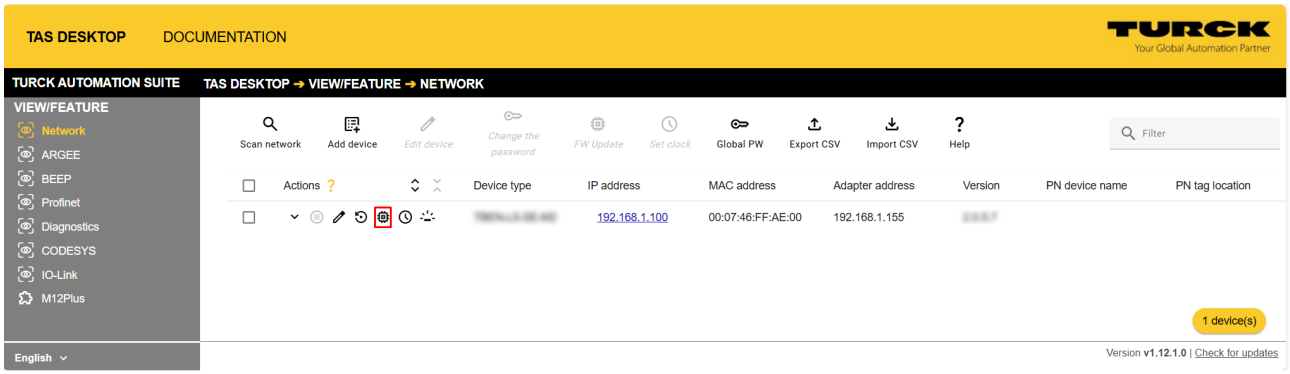



Fig. 87: Firmware update network view

- ▶ In the following dialog: Click **Select file** and open the directory of the firmware file.
- ▶ Select the new firmware file and load it via **Open**.
- ▶ Click **Start** to start the firmware update.
- ▶ Enter the device password and click **Login**

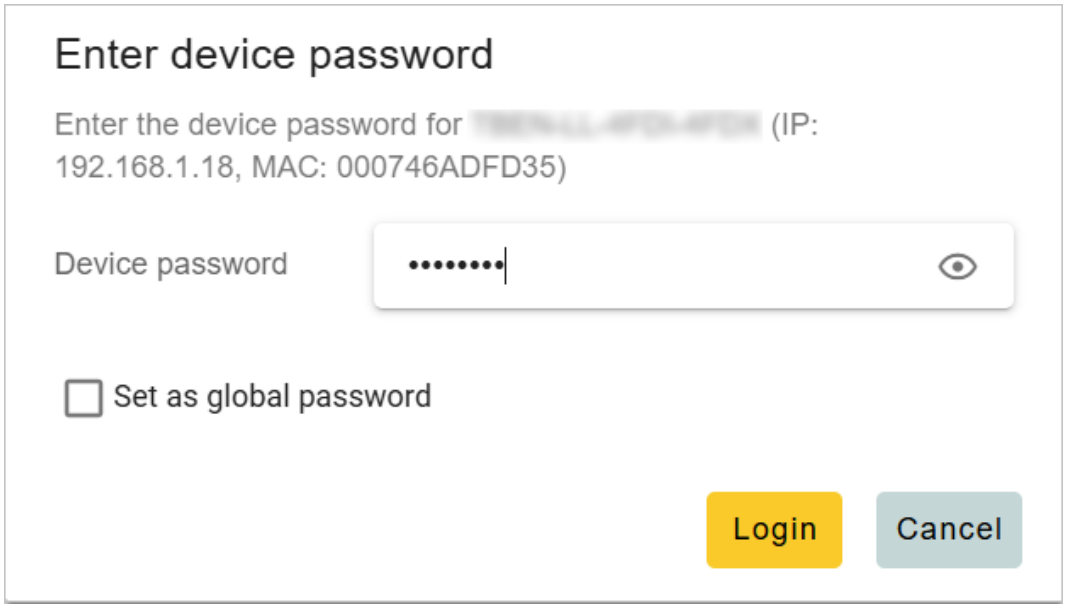



Fig. 88: Entering the device password

➪ The progress of the firmware update is displayed.

**NOTE**
TAS makes it possible to set a global password with which all devices can be unlocked. This requires that all selected devices have the same device password and are in the same TCP network.

As an alternative to selecting a single device, it is also possible to select multiple devices. To do so, all devices to be updated must correspond to the same device type and be in the same TCP network.

This enables a firmware update to be performed for multiple devices at once.

### Starting a firmware update for multiple devices

- ▶ Select all desired devices in the network view using the checkbox.
- ▶ Click **FW update** in the header.

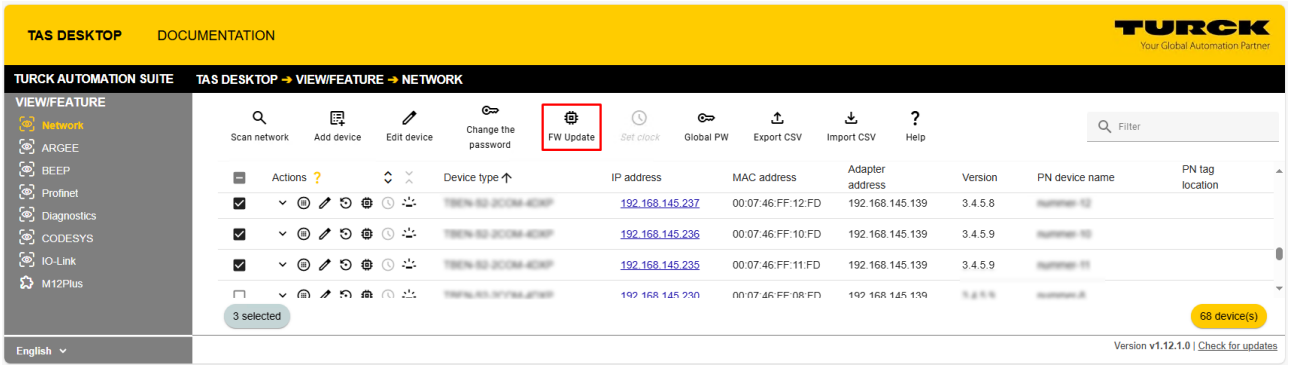



Fig. 89: Firmware update network view multiple devices

- ▶ In the following dialog: Click **Select file** and open the directory of the firmware file.
- ▶ Select the new firmware file and load it via **Open**.
- ▶ Click **Start** to start the firmware update.

- If a global password has not yet been defined: Enter the password and activate the **Set as global password** option.
  Note: If a global password has not yet been defined and the **Set as global password** option is not activated, the password is requested individually for each device.
- Click **Login**.

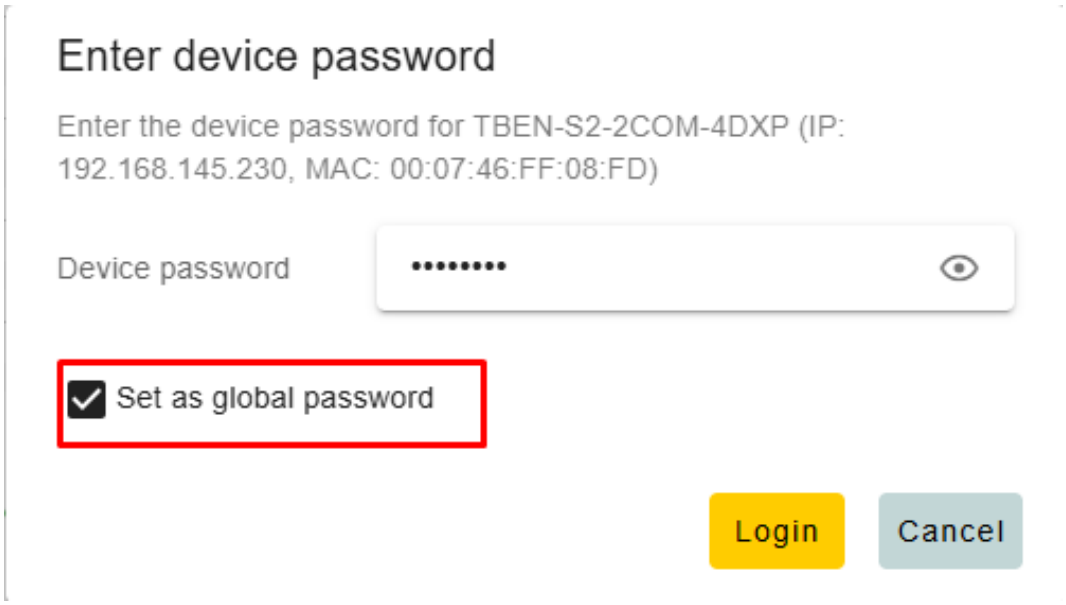



Fig. 90: Entering the device password and setting it as global password

➪ The progress of the firmware update is displayed.

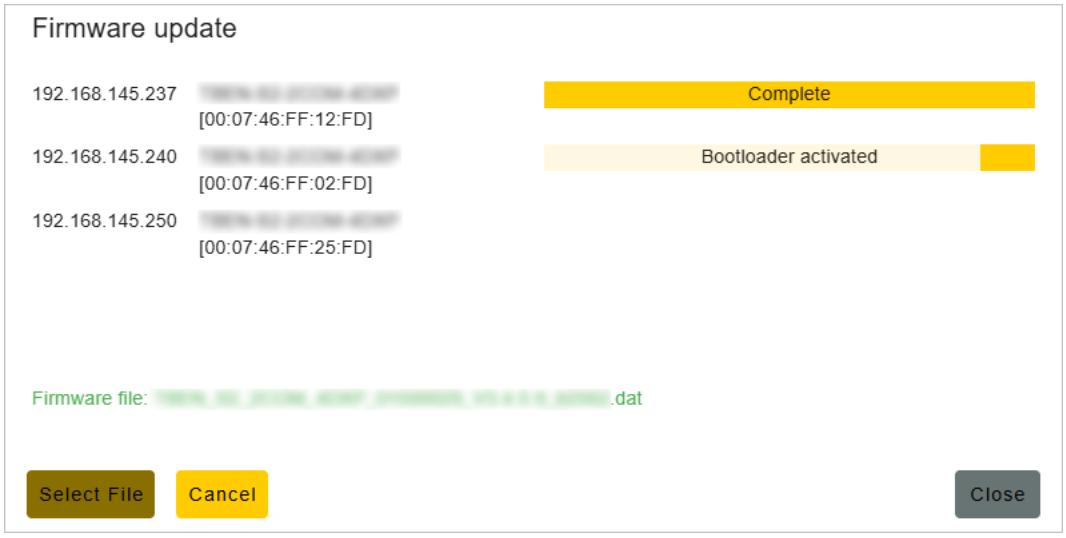



Fig. 91: Firmware update, progress

## 11.2 Updating the firmware via web server

**NOTICE**
Interruption of the power supply during the firmware update
**Risk of device damage due to faulty firmware update**

- Do not interrupt the power supply during the firmware update.
- During the firmware update do not reset the power supply.
- Do not interrupt the Ethernet connection during the firmware update.

- Open the web server.
- Log in to the device as admin (authorization level 0).
  Note: The default password for the admin is “password”.
- Click **Firmware** → **SELECT FIRMWARE FILE**.
- Select the new firmware file and load it via **Open**.

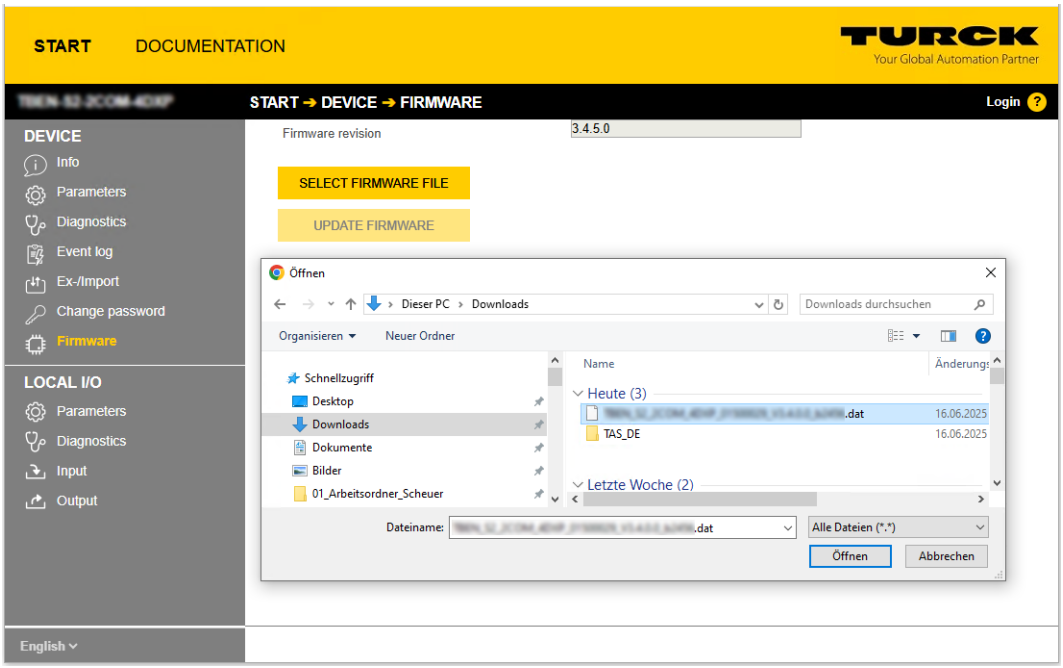



Fig. 92: Web server – selecting the firmware file

- ▶ Click **UPDATE FIRMWARE**.
- ▶ In the following message window, start the firmware update by clicking **OK**.
- ⇨ The progress of the firmware update is displayed.

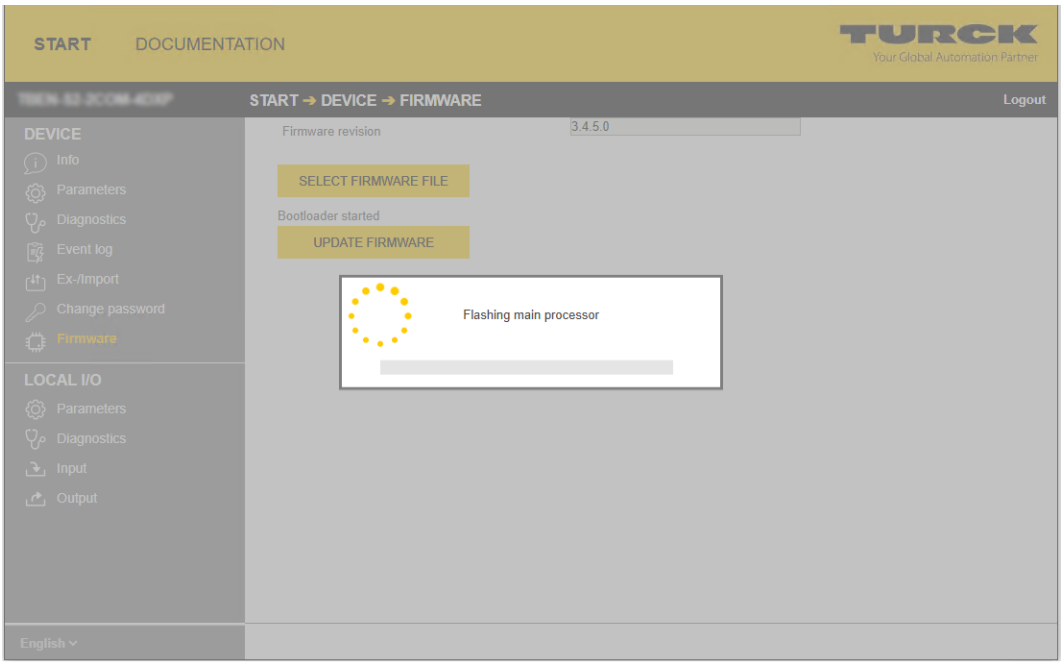



Fig. 93: Web server – firmware update running

- ▶ Restart the device after the update process has been completed via **OK**.

# 12 Repair

The device is not intended for repair by the user. The device must be decommissioned if it is faulty. Observe our return acceptance conditions when returning the device to Turck.

## 12.1 Returning devices

If a device has to be returned, bear in mind that only devices with a decontamination declaration will be accepted. This is available for download at
**https://www.turck.de/en/return-service-6079.php**
and must be completely filled in, and affixed securely and weather-proof to the outside of the packaging.

# 13 Disposal

The devices must be disposed of properly and do not belong in the domestic waste.

# 14 Technical data

| Technical data | |
|---|---|
| **Supply** | |
| Power supply | 24 VDC |
| Permissible range | 18…30 VDC |
| Total current | Max. 4 A per voltage group<br>V1 + V2 max. 5.5 A at 70 °C per module |
| ■ UL rating | V1 + V2 max. 4 A per module |
| ■ Ex derating | S. document "Notes on Use in Ex zone 2 and 22" (ID 100022986) |
| Sensor-/actuator supply VAUX1 | Power supply Connectors C0…C1 from V1, short-circuit proof<br>■ ≤ 55 °C:<br>24 V: ≤ 1.2 A per port<br>5 V: 0.5 A per port<br>■ > 55 °C:<br>24 V: 0.5 A per port<br>5 V: 0.5 A per port |
| Sensor-/actuator supply VAUX2 | Power supply Connectors C2…C3 from V2, short-circuit proof<br>■ ≤ 55 °C: ≤ 1.4 A per port<br>■ > 55 °C: 0.5 A per port |
| Potential isolation | Galvanic isolation from V1 and V2 voltage group voltage proof up to 500 VDC |
| **Connectors** | |
| Ethernet | 1× M8 (IN),<br>1× M8 (OUT) |
| Supply | M8, 4-pin |
| Digital in-/outputs | M12, 5-pin |
| Permissible torques<br>■ Ethernet (M8)<br>■ I/O channels/supply (M8)<br>■ I/O channels (M12)<br>■ Mounting (M4 screws) | <br>0.4 Nm<br>0.6 Nm<br>0.8 Nm<br>1.3 Nm |
| **Max. cable length** | |
| ■ Ethernet | 100 m (per segment) |
| **System data** | |
| Transmission rate | 10 Mbps/<br>100 Mbps |
| Protocol detection | Automatic |
| Web server | Integrated, default: 192.168.1.254 |
| Service interface | Ethernet via P1 or P2 |
| Max. number of parameter changes | $10^6$ (persistant memory) |
| **Modbus TCP** | |
| Address assignment | Static IP, BOOTP, DHCP |
| Supported Function Codes | FC3, FC4, FC6, FC16, FC23 |

| Technical data | |
|---|---|
| Number of TCP connections | 8 |
| Input register start address | 0 (0x0000) |
| Output register start address | 2048 (0x8000) |
| Local port | Port 502, fixed setting |
| **EtherNet/IP** | |
| Address assignment | According to EtherNet/IP standard |
| Device Level Ring (DLR) | Supported |
| Quick Connect (QC) | < 500 ms |
| Min. RPI (Requested Packet Interval) | 2 ms |
| Number of TCP connections | 3 |
| Number of CIP connections | 10 |
| Input Assembly Instance | 103 |
| Output Assembly Instance | 104 |
| Configuration Assembly Instance | 106 |
| **PROFINET** | |
| Address assignment | DCP |
| MinCycle Time | 1 ms |
| Fast Start Up (FSU) | < 500 ms |
| Diagnostics | According to PROFINET Alarm Handling |
| Topology discovery | Supported |
| Automatic address setting | Supported |
| Media Redundancy Protocol (MRP) | Supported |
| Cable length | Max. 30 m |
| **Serial interface** | |
| Signal type | RS232 or RS485 |
| No. of channels | 2 |
| **Operation mode RS232** | |
| Signal low level | -18…-3 VDC |
| Signal high level | 3…18 VDC |
| Transmission signals | TxD, RxD |
| Transmission rate | 300… 230400 bps |
| Transmission type | Full duplex |
| Cable length | 15 m at 19200 baud (max. line capacity < 2000 pF) |
| **Operation mode RS485** | |
| Transmission signals | TX/RX+, TX/RX |
| Transmission rate | 300…230400 bps |
| Transmission type | 2-wire half duplex |
| Biasing | internal or external |
| Termination | internal or external |
| Line impedance | 120 Ω |
| Cable length | Twisted Pair up to 1000 m |
| **Digital inputs** | |
| No. of channels | 4 |

| Technical data | |
|---|---|
| Connection technology | M12, 5-pin |
| Input type | EN 61131-2 type 3, PNP |
| Type of input diagnostics | Channel diagnostics |
| Signal voltage, low level | < 5 V |
| Signal voltage, high level | > 11 V |
| Signal current, low level | 1.5 mA |
| Signal current, high level | > 2 mA |
| Max. input frequency | 100 Hz (for fieldbus communication) |
| Potential isolation | Galvanic isolation to P1/P2, voltage proof up to 500 VDC |
| **Digital outputs** | |
| No. of channels | 4 |
| Connection technology | M12, 5-pin |
| Output type | PNP |
| Type of output diagnostics | Channel diagnostics |
| Output voltage | 24 VDC from potential group V2 |
| Output current per channel | 0.5 A, short-circuit-proof |
| Load type | EN 60947-5-1: DC-13 |
| Load type (UL) | Resistive, pilot duty |
| Short-circuit protection | Yes |
| Potential isolation | Galvanic isolation to P1/P2, voltage proof up to 500 VDC |
| **Standard/directive conformity** | |
| Vibration test | According to EN 60068-2-6 |
| Acceleration | Up to 20 g |
| Shock test | According to EN 60068-2-27 |
| Drop and topple | According to IEC 60068-2-31/IEC 60068-2-32 |
| Electro magnetic compatibility | According to EN 61131-2 |
| Approvals and certificates | CE, UKCA, FCC Statement<br>UV-resistant according to DIN EN ISO 4892-2A (2013) |
| UL cond. | cULus LISTED 21 W2, Encl.Type 1 IND.CONT.EQ. |
| **General information** | |
| Dimensions (w × l × h) | 32 × 144 × 32 mm |
| Operating temperature | -40…+70 °C |
| Storage temperature | -40…+85 °C |
| Operating altitude | Max. 5000 m |
| Degree of protection | IP65/IP67/IP69K (not evaluated by UL) |
| MTTF | 179 years acc. to SN 29500 (Ed. 99) 20 °C |
| Housing material | PA6-GF30 |
| Housing color | Black |
| Material screw | 303 stainless steel |
| Material label | Polycarbonate |
| Halogen free | Yes |
| Mounting | 2 mounting holes, Ø 4.6 mm |

# 15 Turck branches — contact data

| | |
|---|---|
| **Germany** | Hans Turck GmbH & Co. KG<br>Witzlebenstraße 7, 45472 Mülheim an der Ruhr<br>**www.turck.de** |
| **Australia** | Turck Australia Pty Ltd<br>Building 4, 19-25 Duerdin Street, Notting Hill, 3168 Victoria<br>**www.turck.com.au** |
| **Austria** | Turck GmbH<br>Graumanngasse 7/A5-1, A-1150 Vienna<br>**www.turck.at** |
| **Belgium** | Turck Multiprox N. V.<br>Lion d'Orweg 12, B-9300 Aalst<br>**www.multiprox.be** |
| **Brazil** | Turck do Brasil Automação Ltda.<br>Rua Anjo Custódio Nr. 42, Jardim Anália Franco, CEP 03358-040 São Paulo<br>**www.turck.com.br** |
| **Canada** | Turck Canada Inc.<br>140 Duffield Drive, CDN-Markham, Ontario L6G 1B5<br>**www.turck.ca** |
| **China** | Turck (Tianjin) Sensor Co. Ltd.<br>18,4th Xinghuazhi Road, Xiqing Economic Development Area, 300381 Tianjin<br>**www.turck.com.cn** |
| **Czech Republic** | TURCK s.r.o.<br>Na Brne 2065, CZ-500 06 Hradec Králové<br>**www.turck.cz** |
| **France** | TURCK BANNER S.A.S.<br>11 rue de Courtalin Bat C, Magny Le Hongre, F-77703 MARNE LA VALLEE Cedex 4<br>**www.turckbanner.fr** |
| **Hungary** | TURCK Hungary kft.<br>Árpád fejedelem útja 26-28., Óbuda Gate, 2. em., H-1023 Budapest<br>**www.turck.hu** |
| **India** | TURCK India Automation Pvt. Ltd.<br>401-403 Aurum Avenue, Survey. No 109 /4, Near Cummins Complex, Baner-Balewadi Link Rd., 411045 Pune - Maharashtra<br>**www.turck.co.in** |
| **Italy** | TURCK BANNER S.R.L.<br>Via San Domenico 5, IT-20008 Bareggio (MI)<br>**www.turckbanner.it** |
| **Japan** | TURCK Japan Corporation<br>ISM Akihabara 1F, 1-24-2, Taito, Taito-ku, 110-0016 Tokyo<br>**www.turck.jp** |

| | |
|---|---|
| **Korea** | Turck Korea Co, Ltd.<br>A605, 43, Iljik-ro, Gwangmyeong-si<br>14353 Gyeonggi-do<br>**www.turck.kr** |
| **Malaysia** | Turck Banner Malaysia Sdn Bhd<br>Unit A-23A-08, Tower A, Pinnacle Petaling Jaya, Jalan Utara C,<br>46200 Petaling Jaya Selangor<br>**www.turckbanner.my** |
| **Mexico** | Turck Comercial, S. de RL de CV<br>Blvd. Campestre No. 100, Parque Industrial SERVER, C.P. 25350 Arteaga,<br>Coahuila<br>**www.turck.com.mx** |
| **Netherlands** | Turck B. V.<br>Ruiterlaan 7, NL-8019 BN Zwolle<br>**www.turck.nl** |
| **Poland** | TURCK sp.z.o.o.<br>Wroclawska 115, PL-45-836 Opole<br>**www.turck.pl** |
| **Romania** | Turck Automation Romania SRL<br>Str. Siriului nr. 6-8, Sector 1, RO-014354 Bucuresti<br>**www.turck.ro** |
| **Sweden** | Turck AB<br>Fabriksstråket 9, 433 76 Jonsered<br>**www.turck.se** |
| **Singapore** | TURCK BANNER Singapore Pte. Ltd.<br>25 International Business Park, #04-75/77 (West Wing) German Centre,<br>609916 Singapore<br>**www.turckbanner.sg** |
| **South Africa** | Turck Banner (Pty) Ltd<br>Boeing Road East, Bedfordview, ZA-2007 Johannesburg<br>**www.turckbanner.co.za** |
| **Turkey** | Turck Otomasyon Ticaret Limited Sirketi<br>Inönü mah. Kayisdagi c., Yesil Konak Evleri No: 178, A Blok D:4,<br>34755 Kadiköy/ Istanbul<br>**www.turck.com.tr** |
| **United Kingdom** | TURCK BANNER LIMITED<br>Blenheim House, Hurricane Way, GB-SS11 8YT Wickford, Essex<br>**www.turckbanner.co.uk** |
| **USA** | Turck Inc.<br>3000 Campus Drive, USA-MN 55441 Minneapolis<br>**www.turck.us** |

TURCK

Your Global Automation Partner

Over 30 subsidiaries and
60 representations worldwide!

D301438 | 2025/04

www.turck.com