



TB20 – CANopen® Slave Coupler Manual

Version 6 / 09.03.2020 for HW 1-1 & FW 1.12

Manual order no.: 960-160-1AA11/en

Notes

All rights reserved, including those related to the translation, reprinting, and reproduction of this manual or of parts thereof.

No part of this manual may be reproduced, processed, duplicated, or distributed in any form (photocopy, microfilm, or any other methods), even for training purposes or with the use of electronic systems, without written approval from Helmholtz GmbH & Co. KG.

All rights reserved in the event of the granting of a patent or the registration of a utility model.

To download the latest version of this manual, please visit our website at www.helmholtz.de/goto/600-160-1AA11.



We welcome all ideas and suggestions.

Copyright © 2020 by

Helmholtz GmbH & Co. KG

Hannberger Weg 2, 91091 Großenseebach, Germany

Revision record:

Version	Date	Change
1	3/8/2013	First version
2	3/15/2013	Emergency supplemented/corrected; Bit rates table corrected; other smaller corrections
3	7/2/2013	SDOs 200x described; emergencies completely revised; 24 TPDOs & 24 RPDOs
4	2/3/2014	UL; LED in simulation operation; red LED due to diagnoses
5	1/19/2017	FW V1.08: Emergency messages for diagnostics changed Introductory chapter updated DIP switch labeling adjusted Factory reset explained PDO mapping addenda TB20 ToolBox explanation added (PDO mapping, EDS/DCF export) Emergency messages revised
6	3/6/2020	Various corrections and addenda for FW V1.12 Adjustments to layout WEEE note added

Table of contents

1.	General.....	9
1.1.	Target audience for this manual	9
1.2.	Safety instructions	9
1.3.	Note symbols and signal words in the manual	10
1.4.	Intended use	11
1.5.	Improper use	11
1.6.	Installation	12
1.6.1.	Access restriction.....	12
1.6.2.	Electrical installation	12
1.6.3.	Protection against electrostatic discharges.....	12
1.6.4.	Overcurrent protection	12
1.6.5.	EMC protection.....	12
1.6.6.	Operation.....	12
1.6.7.	Liability	12
1.6.8.	Disclaimer of liability	13
1.6.9.	Warranty	13
1.6.10.	Recycling / WEEE	13
2.	System overview.....	14
2.1.	General	14
2.2.	The components that make up the TB20 I/O system	14
2.2.1.	Bus coupler.....	14
2.2.2.	Peripheral modules	14
2.2.3.	Power and isolation module	15
2.2.4.	Power module	16
2.2.5.	Final cover.....	17
2.2.6.	Components in a module	17
2.2.7.	Module Coding	18
3.	Installation and removal.....	19
3.1.	Installation position	19
3.2.	Minimum clearance	19
3.3.	Installing and removing peripheral modules	20
3.3.1.	Installation.....	20
3.3.2.	Removal	21
3.4.	Replacing an electronic module	24
3.5.	Installing and removing the coupler	28
3.5.1.	Installation.....	28

3.5.2.	Removal	29
3.6.	Installing and Removing the Final Bus Cover.....	31
3.6.1.	Installation.....	31
3.6.2.	Removal	31
4.	Setup and wiring.....	32
4.1.	EMC/safety/shielding.....	32
4.2.	Front connectors	33
4.3.	Wiring the coupler	34
4.4.	Using power and isolation modules	35
4.5.	Separate power supply segments for the coupler and the I/O components.....	36
4.6.	Using power modules	37
4.7.	Electronic nameplate.....	38
4.8.	Fusing.....	38
5.	CANopen Coupler Characteristics	39
6.	CAN bus.....	40
6.1.	CAN bus wiring	40
6.2.	CAN bus plug.....	40
6.3.	The CANopen® protocol.....	41
6.3.1.	CANopen Objects	41
6.3.2.	CANopen Functions.....	42
6.3.3.	Network management	43
7.	Setup and use.....	45
7.1.	Address switches.....	45
7.2.	Bit rate	45
7.3.	EDS file-based configuration	45
7.4.	CAN bus connection	45
7.5.	Factory Reset (Restoring the Coupler to Its Factory Settings).....	46
7.6.	LED-based diagnosis.....	46
7.6.1.	CANopen Coupler LEDs.....	46
7.6.2.	Module LEDs.....	47
7.7.	Startup behavior	47
7.8.	SDOs Assigned to I/O Data (SDOs 21xx, 22xx, 6000, 6200, 6401, 6411).....	48
7.9.	Process Data Objects (PDOs).....	49
7.9.1.	PDO Mapping (SDOs 16xx/1Axx).....	49
7.9.2.	PDO Default Mapping.....	50
7.9.3.	PDO Communication Parameters (SDOs 14xx/18xx).....	51
7.9.4.	CAN Identifier for PDOs (SDOs 14xx/18xx, Subindex 1).....	51

7.9.5.	PDO Transmission Type (SDOs 14xx/18xx, Subindex 2)	52
7.9.6.	PDO Inhibit Timer (SDOs 14xx/18xx, Subindex 3)	53
7.9.7.	PDO Event Timer (SDOs 14xx/18xx, Subindex 5)	53
7.10.	Module Parameter Configurations (SDO 30xx)	54
7.11.	Storing a Parameter Configuration (SDO 1010)	55
7.12.	Restoring Factory Settings (SDO 1011)	55
7.13.	Module Diagnostics (SDO 31xx)	56
7.14.	Behavior in the Event of an Error (SDO 1029)	57
7.15.	Hot-swap function (SDO 2020)	58
8.	SDO object table	59
8.1.	System objects (SDO 1000–1200, 2020)	59
8.1.1.	Device Type (1000)	59
8.1.2.	Error Register (1001)	59
8.1.3.	Pre-Defined Error Field (1003)	59
8.1.4.	Number of PDOs (1004)	59
8.1.5.	SYNC identifier (1005)	59
8.1.6.	Device Name (1008)	59
8.1.7.	Hardware version (1009)	60
8.1.8.	Firmware version (100A)	60
8.1.9.	Node ID (100B)	60
8.1.10.	Guard Time (100C)	60
8.1.11.	Life Time Factor (100D)	60
8.1.12.	Store (1010) (Used to Store Parameters)	60
8.1.13.	Restore (1011) (Used to Load Parameters)	60
8.1.14.	Emergency identifier (1014)	61
8.1.15.	Consumer Heartbeat (1016)	61
8.1.16.	Producer Heartbeat (1017)	61
8.1.17.	Identity Object (1018)	61
8.1.18.	Module List (1027)	62
8.1.19.	Object Behavior in Case of Error (1029)	62
8.1.20.	SDO server parameters (1200)	62
8.1.21.	Hot-swap behavior (2020)	62
8.2.	RPDO Communication Parameters (SDO 14xx)	63
8.3.	RPDO Mapping (SDO 16xx)	63
8.4.	TPDO Communication Parameters (SDOs 18xx)	63
8.5.	TPDO Mapping (SDO 1Axx)	63
8.6.	Coupler identification (SDO 200x)	64
8.6.1.	Module ID of the coupler (2001)	64

8.6.2.	Order number of the coupler (2002)	64
8.6.3.	Series number of the coupler (2003).....	64
8.6.4.	Firmware version of the coupler (2004)	64
8.6.5.	Hardware version of the coupler (2006)	64
8.6.6.	Active bit rate (2010)	64
8.7.	Module Configuration (SDO 2F0x)	65
8.7.1.	Module Models (2F00)	65
8.7.2.	Module Codes (2F01)	65
8.7.3.	Module Order Numbers (2F02)	65
8.7.4.	Module Serial Numbers (2F03)	65
8.7.5.	Firmware versions of the modules (2F04).....	66
8.7.6.	Module Hardware Versions (2F06)	66
8.8.	Module Data (SDO 21xx, 22xx, 6xxx)	67
8.8.1.	Input Data (21xx).....	67
8.8.2.	Output Data (22xx)	67
8.8.3.	Digital Inputs (6000)	68
8.8.4.	Digital Outputs (6200)	68
8.8.5.	Analog Inputs (6401)	69
8.8.6.	Analog Outputs (6411).....	69
8.9.	Module Parameters (SDO 30xx).....	70
8.10.	Module diagnostic states (SDO 31xx)	70
9.	Emergency messages.....	71
10.	Configuration with the TB20 ToolBox	74
10.1.	Installation of the TB20 ToolBox.....	74
10.2.	Start and create a new project (“Welcome”)	74
10.3.	Configurator.....	75
10.3.1.	Set coupler parameters	76
10.3.2.	Set PDO mapping	76
10.3.3.	Load project into the coupler (“Upload”)	77
10.3.4.	EDS/DCF export	78
10.4.	Online diagnostics.....	79
10.4.1.	Load project from the coupler (“Download”)	80
10.4.2.	Simulation	81
11.	Technical data	82
12.	TB20 System Dimensions	83
13.	Spare parts	84

13.1.	Base modules	84
13.1.1.	14 mm width standard base module	84
13.1.2.	25 mm width base module.....	84
13.1.3.	Power and isolation base module.....	84
13.1.4.	Power base module	85
13.2.	Front connectors	85
13.2.1.	10-terminal front connector	85
13.2.2.	20-terminal front connector	85
13.3.	Electronic modules	86
13.4.	Final cover	86

1. General

This operating manual applies only to devices, assemblies, software, and services of Helmholz GmbH & Co. KG (“Helmholz”).

1.1. Target audience for this manual

This description is only intended for trained personnel qualified in control and automation engineering who are familiar with the applicable national standards. For installation, commissioning, and operation of the components, compliance with the instructions and explanations in this operating manual is essential.



Configuration, execution, and operating errors can interfere with the proper operation of the TB20 devices and result in personal injury, as well as material or environmental damage. Only suitably qualified personnel may operate the TB20 devices!

Qualified personnel must ensure that the application and use of the products described meet all the safety requirements, including all relevant laws, regulations, provisions, and standards.

1.2. Safety instructions

The safety instructions must be observed in order to prevent harm to living creatures, material goods, and the environment. The safety notes indicate possible hazards and provide information about how hazardous situations can be prevented.

1.3. Note symbols and signal words in the manual



HAZARD

If the hazard warning is ignored, there is an imminent danger to life and health of people from electrical voltage.



WARNING

If the hazard warning is ignored, there is a probable danger to life and health of people from electrical voltage.



CAUTION

If the hazard warning is ignored, people can be injured or harmed.



ATTENTION

Draws attention to sources of error that can damage equipment or the environment.



NOTE

Gives an indication for better understanding or preventing errors.

1.4. Intended use

The TB20 I/O system is an open, modular, and distributed peripheral system designed to be mounted on a 35 mm DIN rail.

Communication with a higher-level control system takes place via a bus system / network through a TB20 bus coupler. Up to 64 modules from the TB20 range can be set up on a bus coupler. The bus couplers support hot swap for replacing modules during ongoing operation.

All components are supplied with a factory hardware and software configuration. The user must carry out the hardware and software configuration for the conditions of use. Modifications to hardware or software configurations that extend beyond the documented options are not permitted and nullify the liability of Helmholz GmbH & Co. KG.

The TB20 devices should not be used as the only means for preventing hazardous situations on machinery and equipment.

Successful and safe operation of the TB20 devices requires proper transport, storage, installation, assembly, installation, commissioning, operation, and maintenance.

The ambient conditions indicated in the technical specifications must be adhered to.

The TB20 systems have a protection rating of IP20 and must have a control box/cabinet fitted to protect against environmental influences in an electrical operating room. To prevent unauthorized access, the doors of control boxes/cabinets must be closed and possibly locked during operation.



HAZARD

TB20 devices can be equipped with modules that can carry dangerously high voltages. The voltages connected to the TB20 devices can result in hazards during work on the TB20 devices.

1.5. Improper use



WARNING

The consequences of improper use may include personal injury to the user or third parties, as well as property damage to the control system, the product, or the environment. Use TB20 devices only as intended!

1.6. Installation

1.6.1. Access restriction

The modules are open operating equipment and must only be installed in electrical equipment rooms, cabinets, or housings.

Access to the electrical equipment rooms, cabinets, or housings must only be possible using a tool or key, and access should only be granted to trained or authorized personnel.

1.6.2. Electrical installation

Observe the regional safety regulations.



TB20 devices can be equipped with modules that can carry dangerously high voltages. The voltages connected to the TB20 devices can result in hazards during work on the TB20 devices.

1.6.3. Protection against electrostatic discharges

To prevent damage through electrostatic discharges, the following safety measures are to be followed during assembly and service work:

- Never place components and modules directly on plastic items (such as polystyrene, PE film) or in their vicinity.
- Before starting work, touch the grounded housing to discharge static electricity.
- Only work with discharged tools.
- Do not touch components and assemblies on contacts.

1.6.4. Overcurrent protection

To protect the TB20 and the supply line, a slow-blowing 8 A line protection fuse is required.

1.6.5. EMC protection

To ensure electromagnetic compatibility (EMC) in your control cabinets in electrically harsh environments, the known rules of EMC-compliant configuration are to be observed in the design and construction.

1.6.6. Operation

Operate the TB20 only in flawless condition. The permissible operating conditions and performance limits must be adhered to. Retrofits, changes, or modifications to the device are strictly forbidden.

The TB20 is a piece of operating equipment intended for use in industrial plants. During operation, the TB20 can carry dangerous voltages. During operation, all covers on the unit and the installation must be closed in order to ensure protection against contact.

1.6.7. Liability

The contents of this manual are subject to technical changes resulting from the continuous development of products of Helmholtz GmbH & Co. KG. In the event that this manual contains technical or clerical errors, we reserve the right to make changes at any time without notice. No claims

for modification of delivered products can be asserted based on the information, illustrations, and descriptions in this documentation. Beyond the instructions contained in the operating manual, the applicable national and international standards and regulations must also be observed in any case.

1.6.8. Disclaimer of liability

Helmholz is not liable for damages if these were caused by use or application of the products that was improper or not as intended.

Helmholz assumes no responsibility for any printing errors or other inaccuracies that may appear in the operating manual, unless there are serious errors about which Helmholz was already demonstrably aware.

Beyond the instructions contained in the operating manual, the applicable national and international standards and regulations must also be observed in any case.

Helmholz GmbH is not liable for damage caused by software running on the user's equipment that compromises, damages, or infects additional equipment or processes through the remote maintenance connection and which triggers or enables unwanted data transfer.

1.6.9. Warranty

Report any defects to the manufacturer immediately upon discovery of the defect.

The warranty is not valid in case of:

- Failure to observe these operating instructions
- Use of the device that is not as intended
- Improper work on and with the device
- Operating errors
- Unauthorized modifications to the device

The agreements met upon contract conclusion under "General Terms and Conditions of Helmholz GmbH & Co. KG" apply.

1.6.10. Recycling / WEEE

The Helmholz GmbH & Co. KG company is registered as a manufacturer with the brand HELMHOLZ and the device type "Small devices of information and telecommunications technology for exclusive usage in other than private households", as well as the following registration data:

Helmholz GmbH & Co. KG,
Seat of the company: 91091 Großenseebach,
Address: Hannberger Weg 2,
Name of the authorized representative: Carsten Bokholt,
Registration number: DE 44315750.

The electrical devices described in this documents are to be recycled. Pursuant to Directive 2012/19/EU on Waste of Electrical and Electronic Equipment (WEEE), they may not be disposed of by way of municipal disposal companies.



2. System overview

2.1. General

The TB20 I/O system is an open, modular, and distributed peripheral system designed to be mounted on a 35mm DIN rail.

It is made up of the following components:

- Bus couplers
- Peripheral modules
- Power and isolation modules
- Power modules

By using these components, you can build a custom automation system that is tailored to your specific needs and that can have up to 64 modules connected in series to a bus coupler. All components have a protection rating of IP20.

2.2. The components that make up the TB20 I/O system

2.2.1. Bus coupler

The system's bus coupler includes a bus interface and a power module. The bus interface is responsible for establishing a connection to the higher-level bus system and is used to exchange I/O signals with the automation system's CPU.

The power module is responsible for powering the coupler's electronics and all connected peripheral modules.

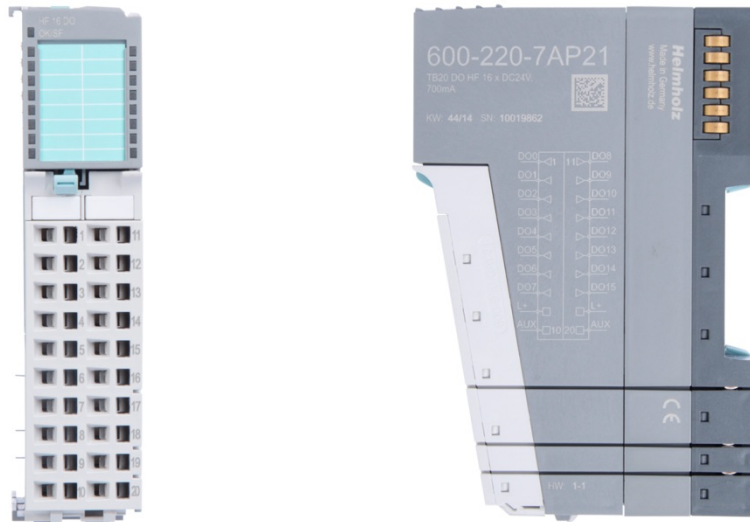
2.2.2. Peripheral modules

The system's peripheral modules are electronic components to which peripheral devices such as sensors and actuators can be connected. A variety of peripheral modules with different tasks and functions are available.

Example: Peripheral module with 10-pin front connector



Example: Peripheral module with 20-pin front connector



2.2.3. Power and isolation module

The system's bus coupler provides the supply voltage for the communications bus (5 V, top) and for external signals (24 V, bottom). These voltages are passed from module to module through the base modules.

Power and isolation modules make it possible to segment the power supply for external signals into individual power supply sections that are powered separately. Meanwhile, the communications bus signals and supply voltage simply continue to be passed through, in contrast to the way they are handled by power modules (see section 2.2.4).



NOTE

Power and insulation modules have a lighter body color.

2.2.4. Power module

The system's bus coupler provides the supply voltage for the communications bus (5 V, top) and for external signals (24 V, bottom). These voltages are passed from module to module through the base modules.

Power modules make it possible to segment the power supply for both external signals and the communication bus into individual power supply sections that are powered separately.

Power modules deliver all necessary power to the peripheral modules connected after them and, if applicable, all the way to the next power module or power and isolation module. A power module is required whenever the power supplied by the coupler alone is not sufficient, e.g., when there are a large number of modules with high power requirements. The "TB20 ToolBox" configuration program can be used to determine whether power modules are needed, as well as how many of them will be needed.

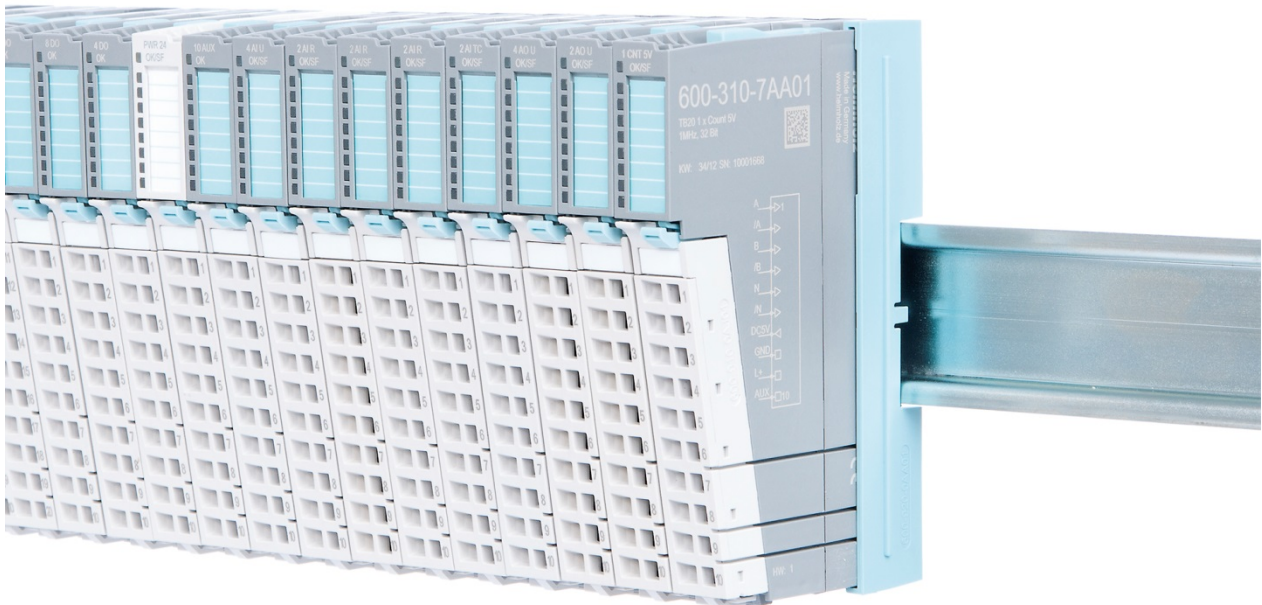


NOTE

Power modules have a lighter body color.

2.2.5. Final cover

The final cover protects the contacts on the last base module from accidental contact by covering its outer right-hand side.



2.2.6. Components in a module

Each module consists of three parts:

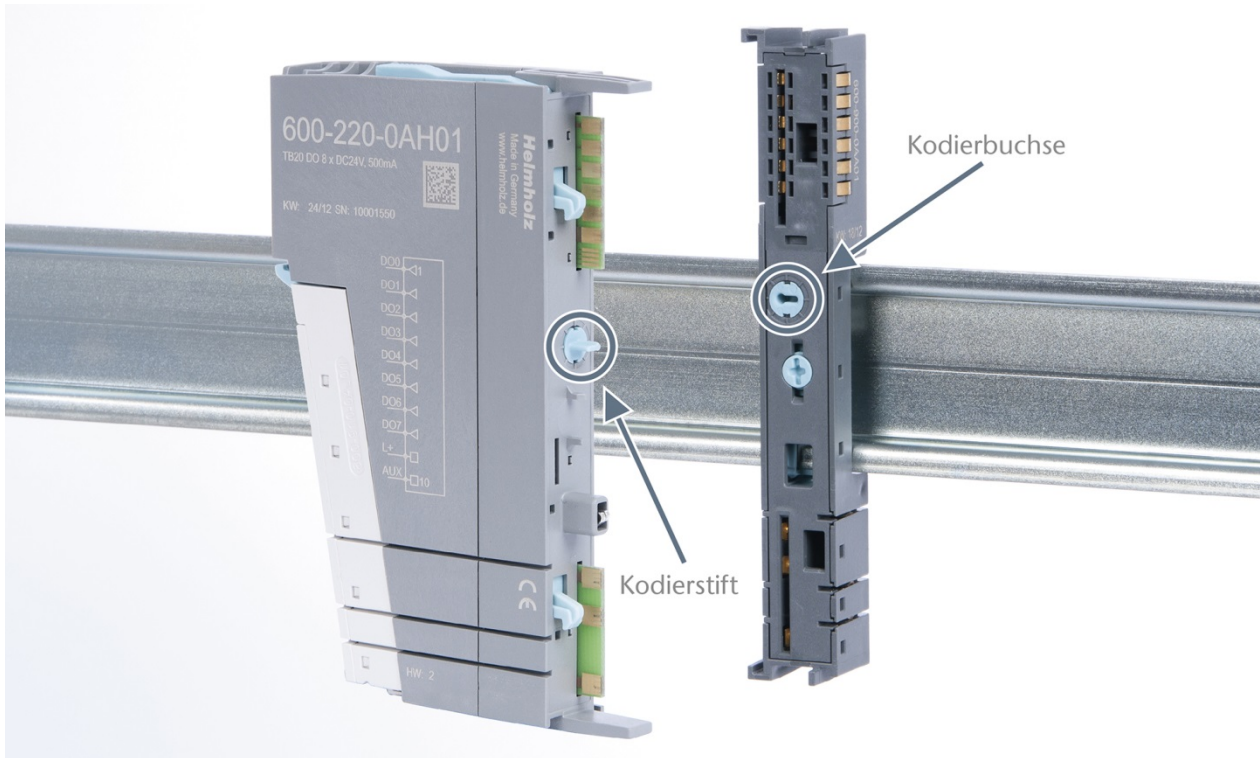
- A base module
- An electronic module
- A front connector



2.2.7. Module Coding

Electronic modules and base modules feature coding elements meant to prevent the wrong spare electronic modules from being plugged in during maintenance and repairs.

These coding elements consist of a coding plug on the electronic module and a coding socket on the base module (see following figure).



The coding plug and coding socket can each be in one of eight different positions. Each of the eight positions can be assigned to a specific type of module (Digital In, Digital Out, Analog In, Analog Out, Power). It will only be possible to plug an electronic module into a base module if the position of the coding plug and the position of the coding socket match. If the positions differ, the electronic module is mechanically blocked.

3. Installation and removal



HAZARD

TB20 modules can carry lethal voltage.

Before starting any work on TB20 system components, make sure to deenergize all components and the cables supplying them with power! During work when the system is live, there is the risk of fatal electrocution!



ATTENTION

Installation must be carried out according to VDE 0100/IEC 364 or in accordance with applicable national standards. The TB20 IO system has protection rating IP20. If a higher protection rating is required, the system must be installed in a housing or control cabinet. In order to ensure safe operation, the ambient temperature must not exceed 60 °C.

3.1. Installation position

The TB20 I/O system can be installed in any position.

In order to achieve optimum ventilation and be able to use the system at the specified maximum ambient temperature, it will, however, be necessary to use a horizontal installation layout.

3.2. Minimum clearance

It is recommended to adhere to the minimum clearances specified when installing the coupler and modules. Adhering to these minimum clearances will ensure that:

- the modules can be installed and removed without having to remove any other system components
- there will be enough space to make connections to all existing terminals and contacts using standard accessories
- there will be enough space for cable management systems (if needed)

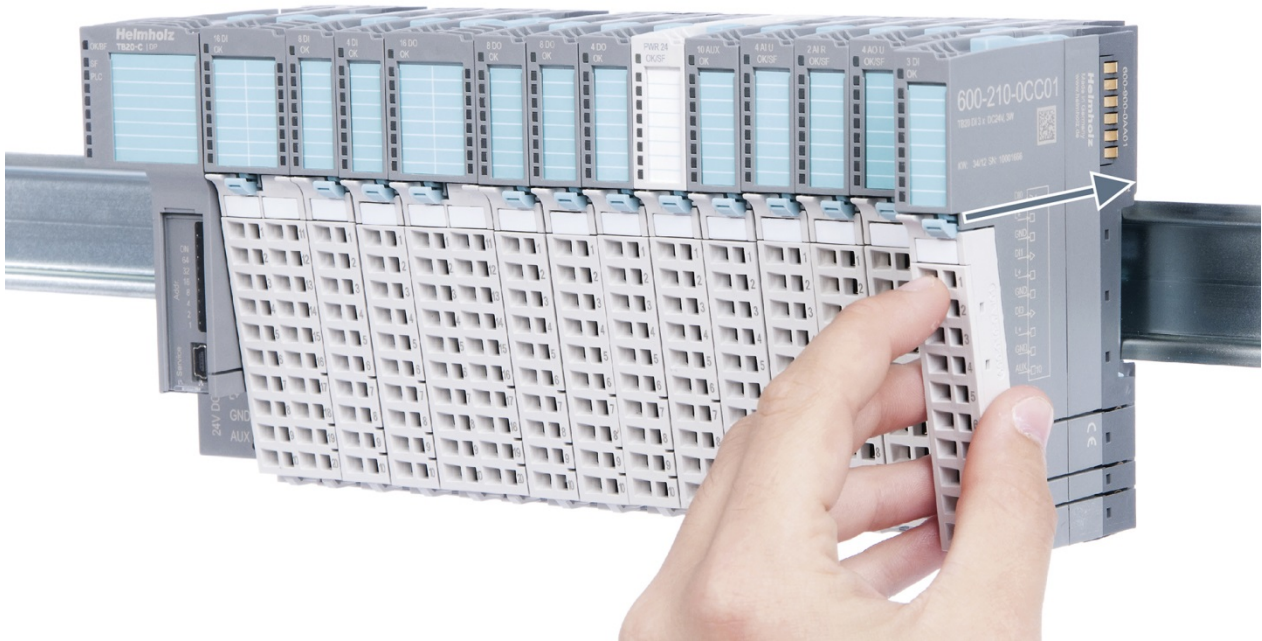
The minimum clearances for mounting TB20 components are: 30 mm on the top and on bottom and 10 mm on each side.

3.3. Installing and removing peripheral modules

3.3.1. Installation

Installing an assembled peripheral module

Place the assembled module on the DIN rail by moving it straight towards the rail. Make sure that the module engages the upper and lower guide elements of the previous module. Then push the upper part of the module towards the DIN rail until the rail fastener inside fastens into place with a soft click.



Installing the individual parts of a peripheral module one after the other:

Place the base module on the DIN rail from below in an inclined position. Then push the upper part of the base module towards the rail until the module is parallel with the rail and the rail fastener on the inside snaps into place with a soft click.

Place an electronic module with matching coding (see the “Module coding” section on page 18) on the base module in a straight line and then gently push it onto the base module until both modules are fully resting on top of one another and the module fastener snaps into place with a soft click.

Finally, place the front connector on the electronic module from below in an inclined position and then gently push it onto the electronic module until the front connector fastener snaps into place with a soft click.

3.3.2. Removal

To remove a peripheral module, follow the four steps below:

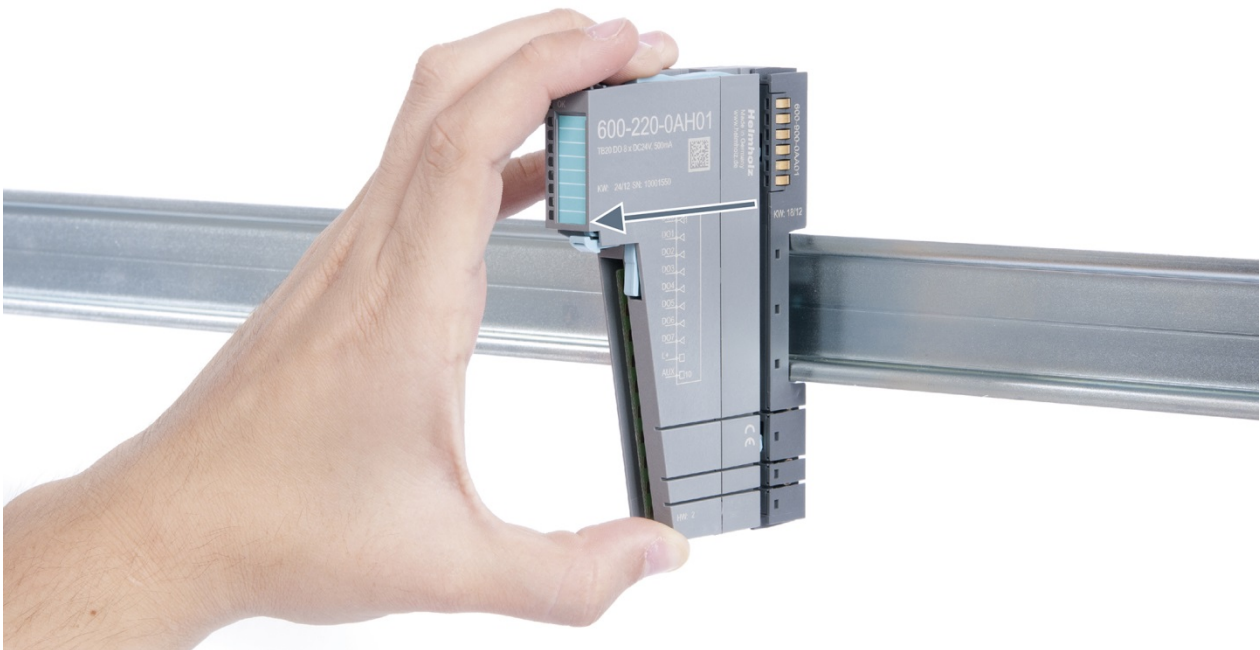
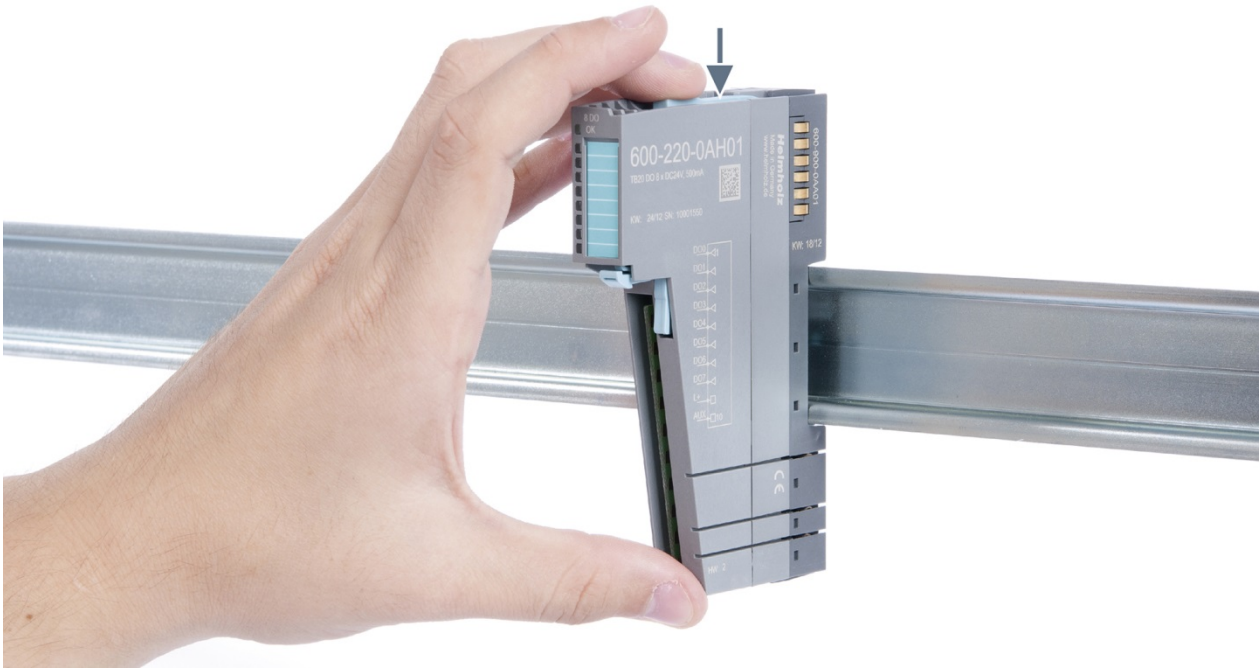
Step 1: Remove the front connector

To remove the front connector, push the tab above the front connector upwards (see the picture below). This will push out the front connector, after which you can pull it out.



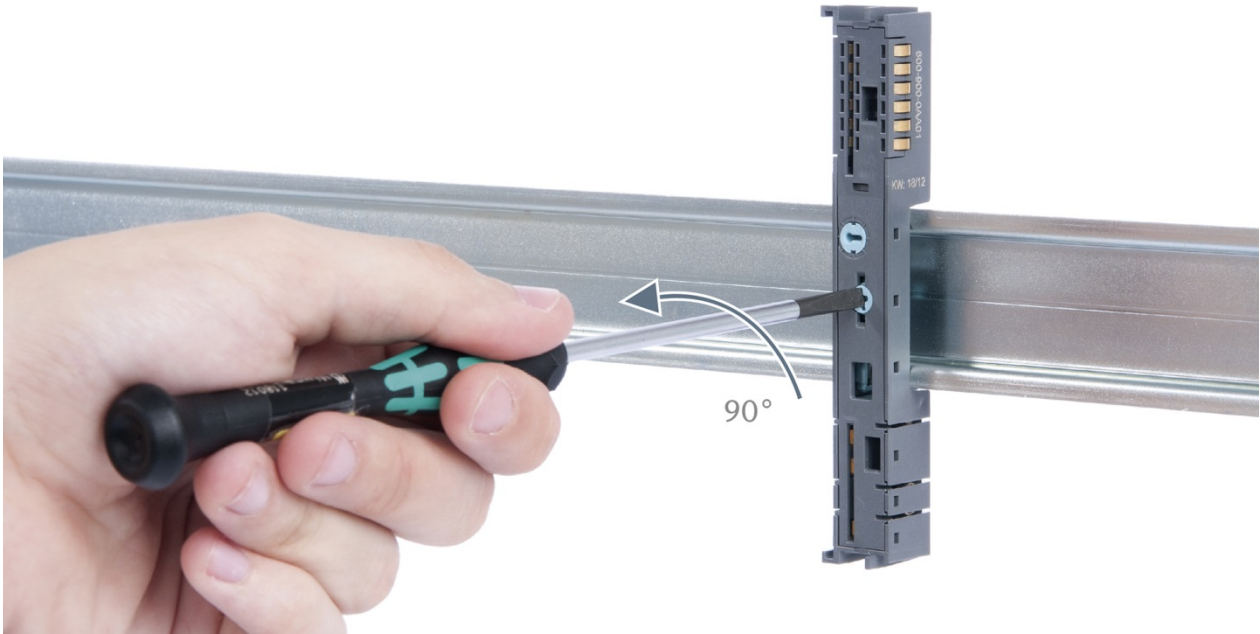
Step 2: Remove the electronic module

To do so, use your middle finger to push on the lever from above and then use your thumb and index finger to pull out the electronic module while holding the lever down (see the picture below).



Step 3: Release the base module

Use a screwdriver to release the base module. Turn the screwdriver 90° counterclockwise to release.



Step 4: Remove the base module

Remove the base module by pulling it towards you.

3.4. Replacing an electronic module

The procedure for replacing the electronic module on a peripheral module consists of four steps.

If you need to replace the electronic module while the system is running, make sure to take into account the general technical specifications for the bus coupler being used.



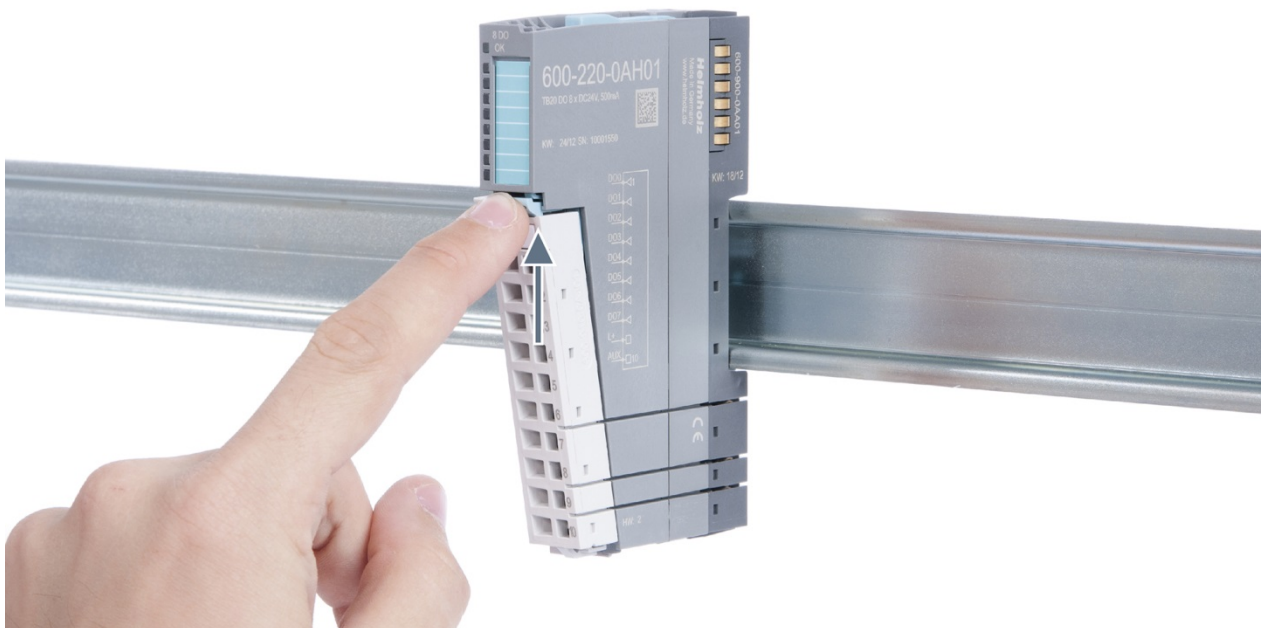
TB20 modules can carry lethal voltage.

Before starting any work on TB20 system components, make sure to deenergize all components and the cables supplying them with power! During work when the system is live, there is the risk of fatal electrocution!

Note the wiring diagram of the system and switch off dangerous voltages before starting work!

Step 1: Remove the front connector

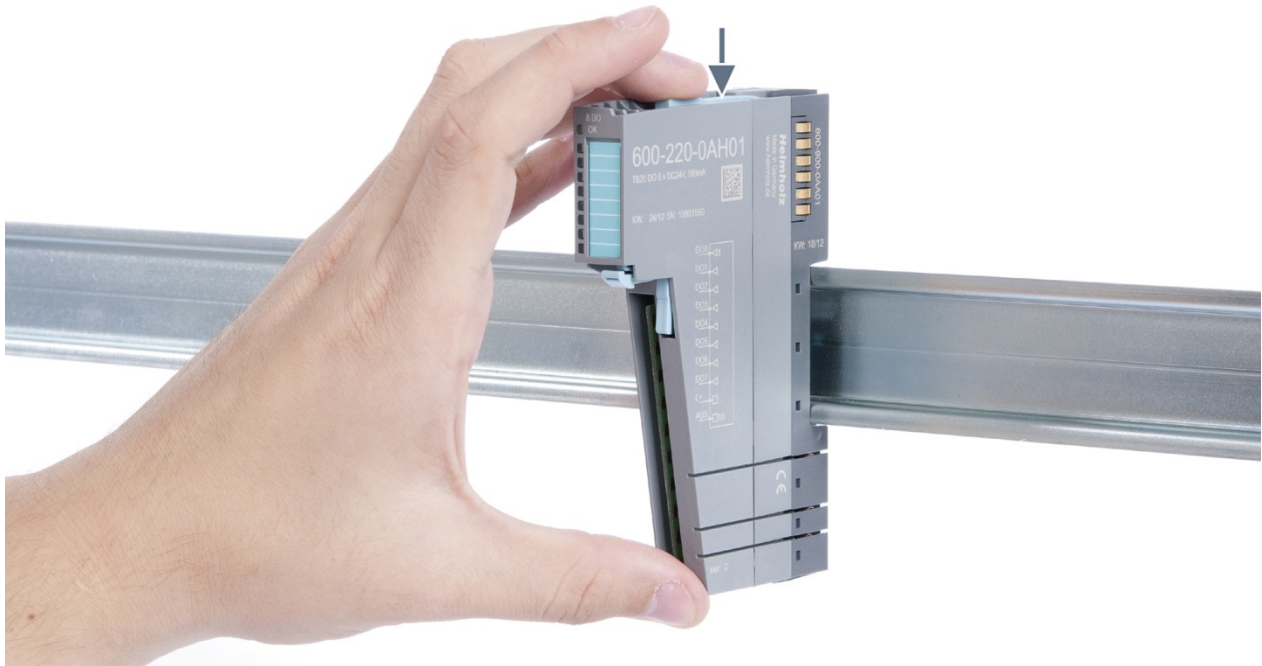
To remove the front connector, push the tab above the front connector upwards (see the picture below). This will push out the front connector, after which you can pull it out.

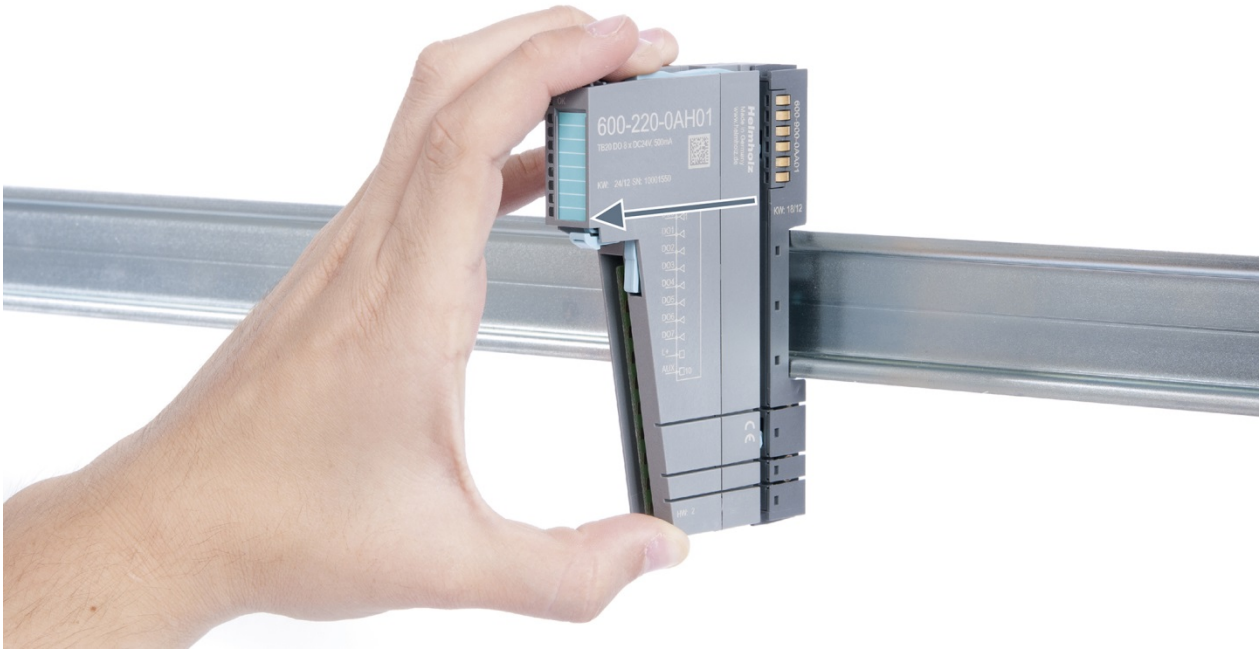




Step 2: Remove the electronic module

To remove the electronic module, use your middle finger to push on the lever from above and then use your thumb and index finger to pull out the electronic module while holding the lever down (see the picture below).





Step 3: Plug in a new electronic module



ATTENTION

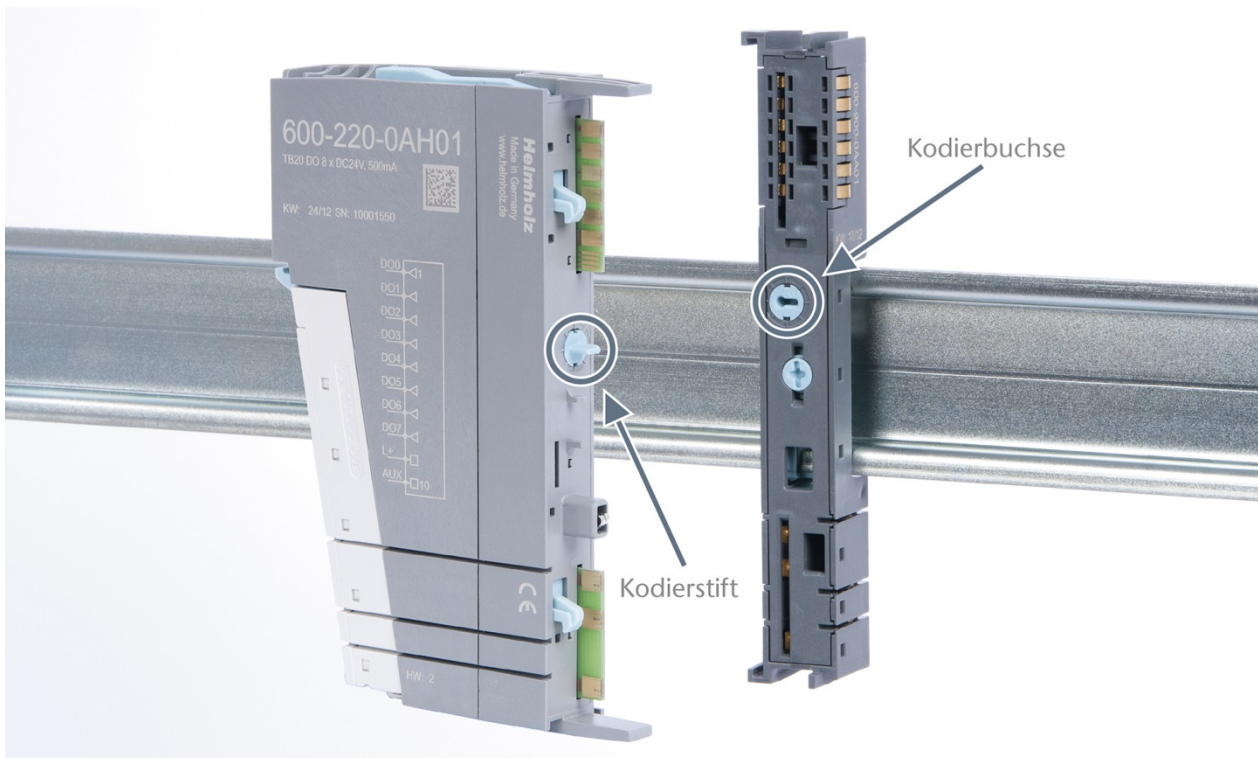
The electronic module must be snapped into place on the base module with a single continuous movement. If the electronic module is not snapped into place firmly and straight on the base module, bus malfunctions may occur.



ATTENTION

If the electronic module cannot be plugged into the base module, check whether the coding elements on the electronic module and base module (see figure below) match. If the coding elements on the electronic module do not match those on the base module, you may be attempting to plug in the wrong electronic module.

For more information on coding elements, please consult section 2.2.7.

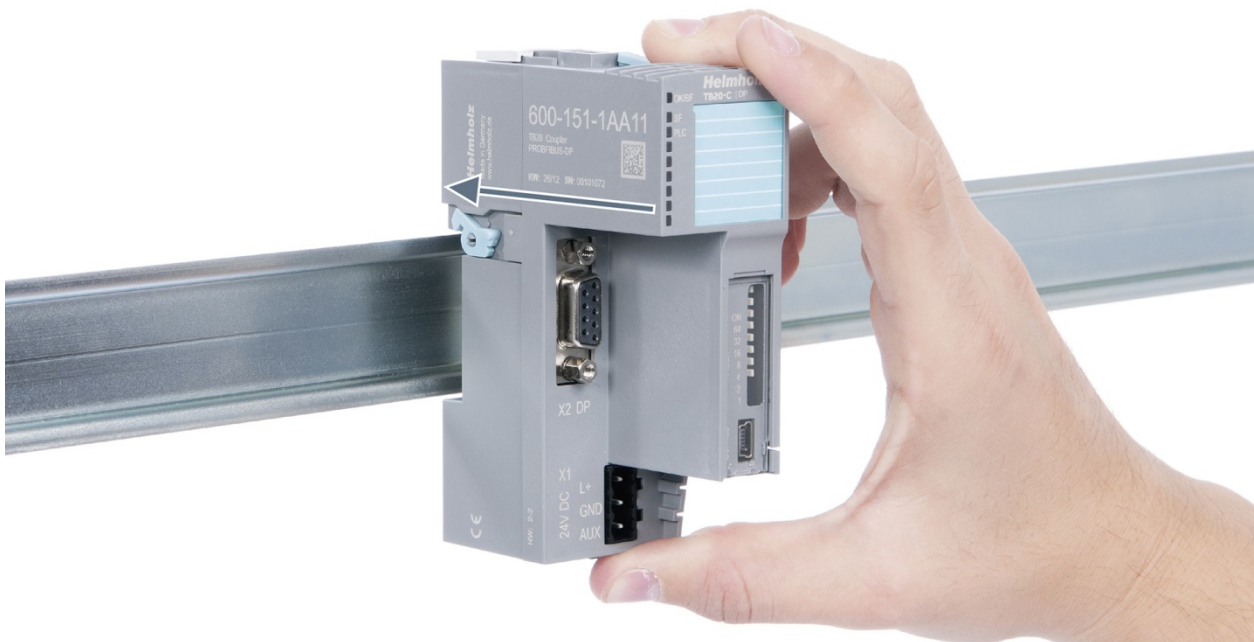


Step 4: Plug in the front connector

3.5. Installing and removing the coupler

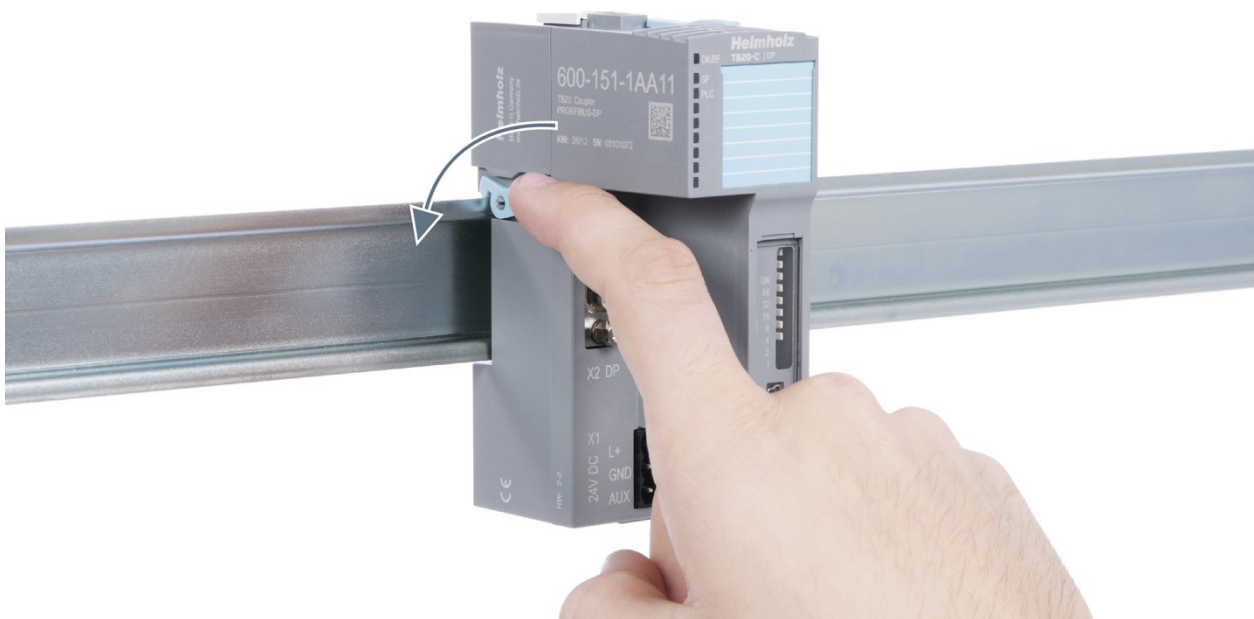
3.5.1. Installation

Place the coupler, together with the attached base module, on the DIN rail by moving it straight towards the rail. Then push the coupler towards the rail until the base module's rail fastener snaps into place with a soft click.



Step 2: Secure the coupler on the DIN rail

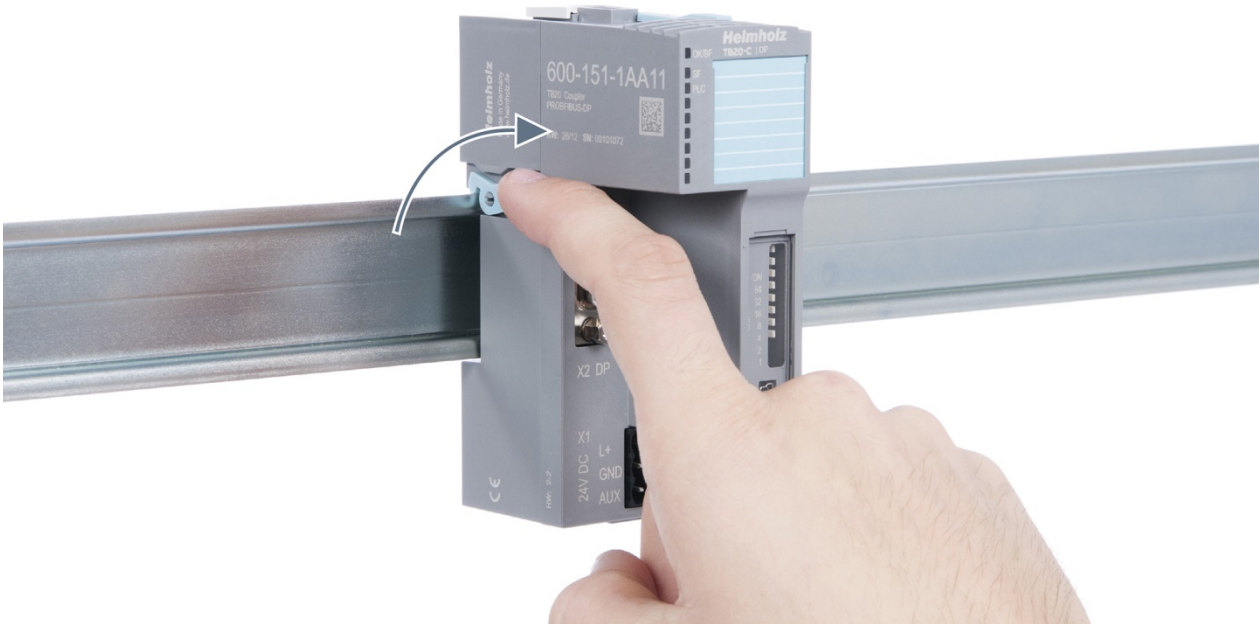
Use the locking lever on the left side of the coupler to lock the coupler into position on the DIN rail.



3.5.2. Removal

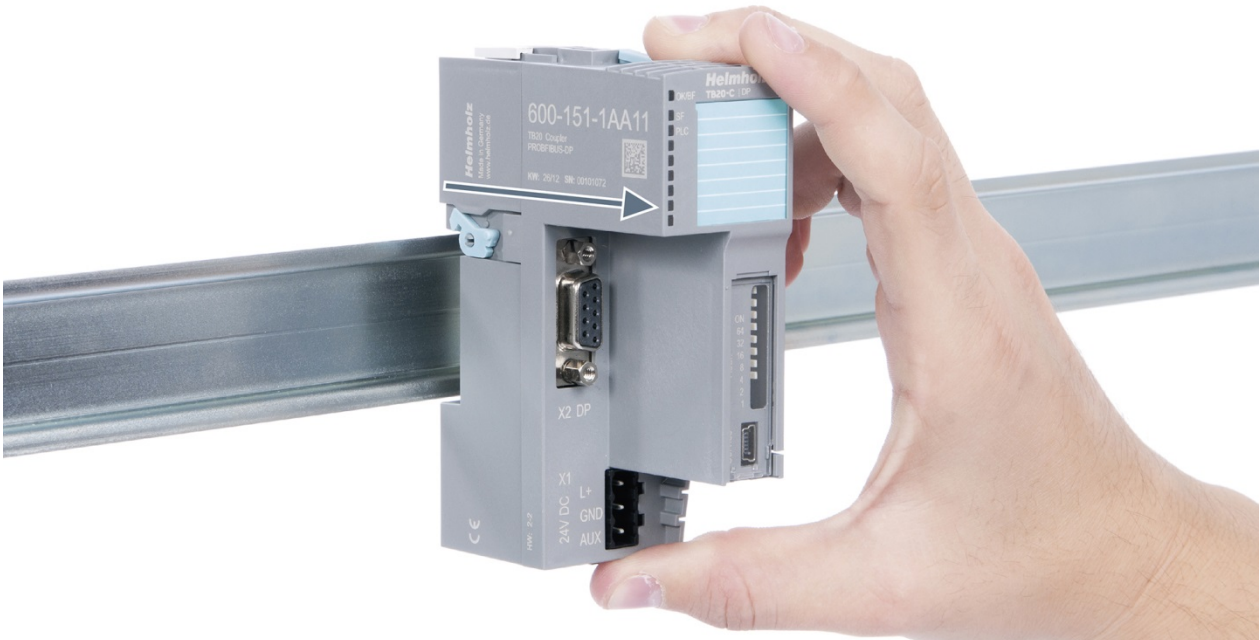
Step 1: Release the locking mechanism

Release the locking lever on the left side of the coupler in order to disengage it from the DIN rail.



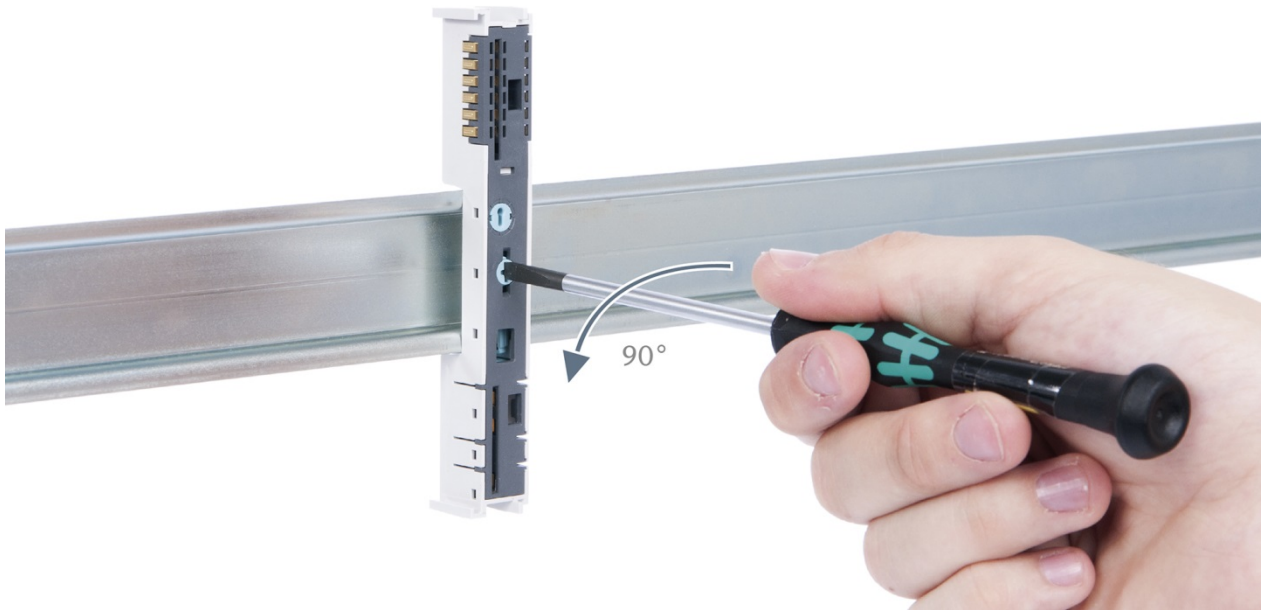
Step 2: Remove the coupler

Use your middle finger to push on the lever from above and use your thumb and index finger to pull out the coupler while holding the lever down.



Step 3: Release the base module

Use a screwdriver to release the base module.



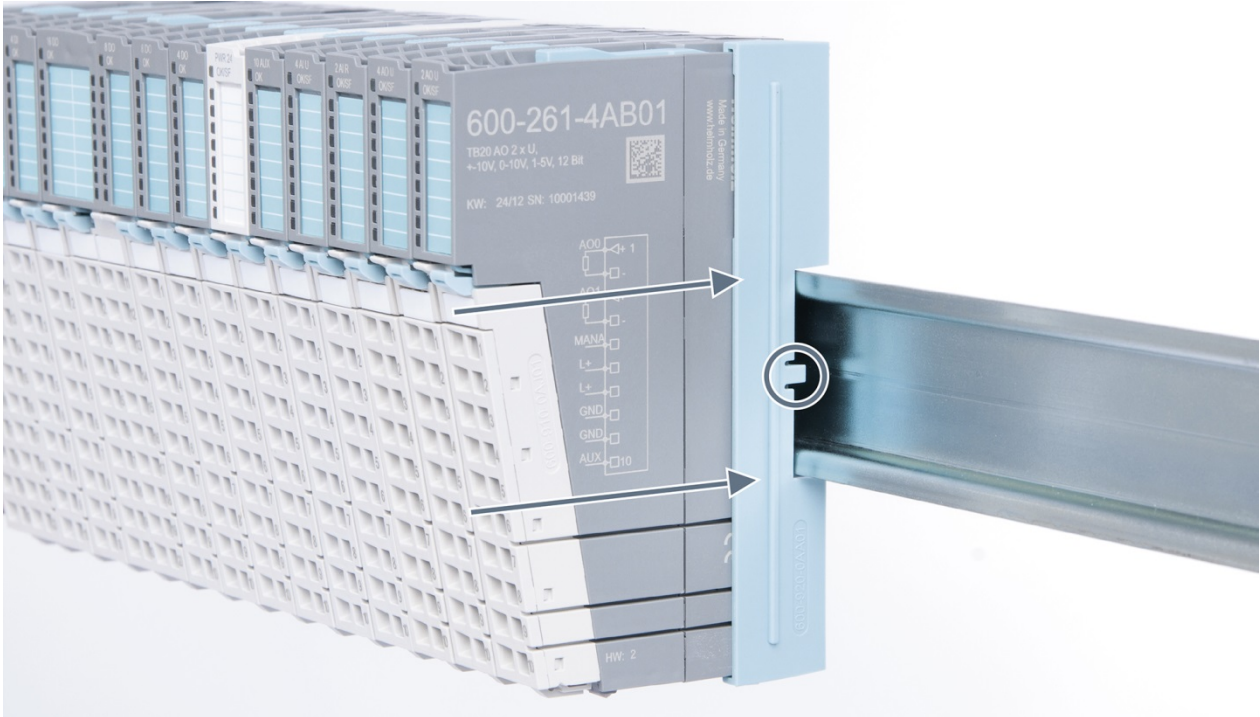
Step 4: Remove the base module

Remove the base module by pulling it towards you.

3.6. Installing and Removing the Final Bus Cover

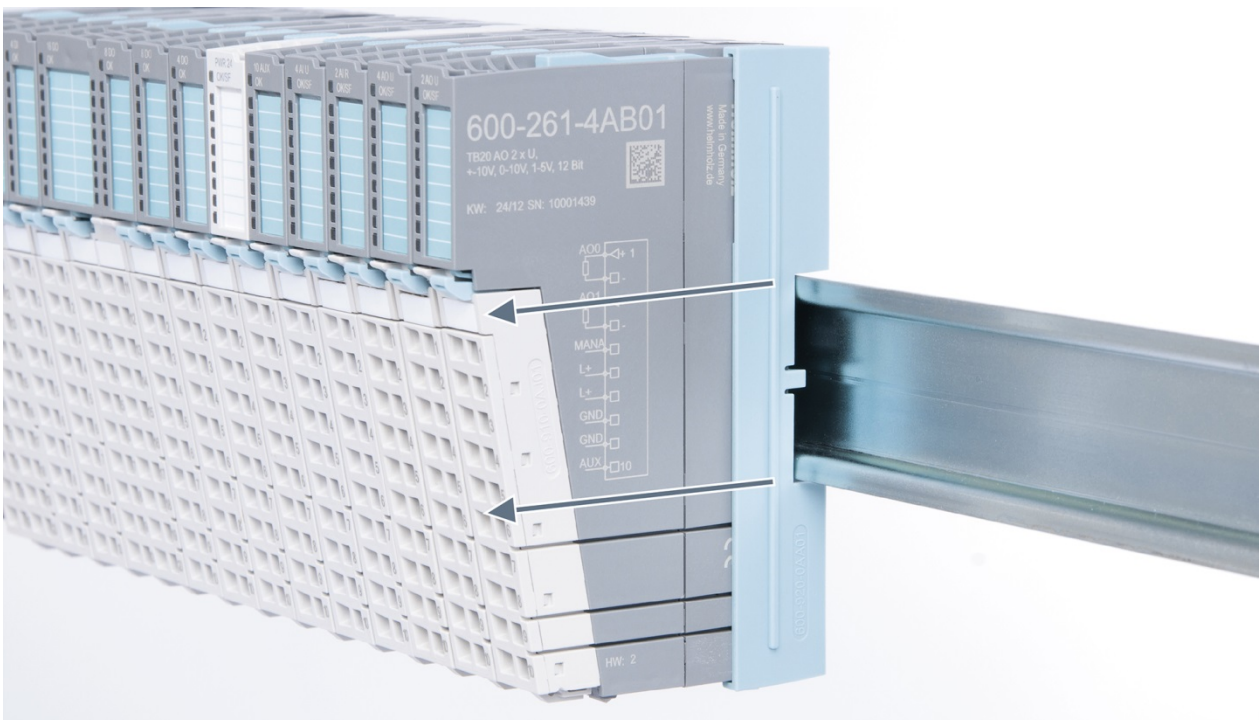
3.6.1. Installation

Slide the final cover onto the last module along the case, starting from the end with the front connector and moving towards the DIN rail, until the cover covers the base module's contacts and the tab snaps into place.



3.6.2. Removal

Pull the final bus cover upward along and off of the module.



4. Setup and wiring

4.1. EMC/safety/shielding

The TB20 IO system complies with EU Directive 2004/108/EC (“Electromagnetic Compatibility”).

One effective way to protect against disturbances caused by electromagnetic interference is to shield electric cables, wires, and components.



ATTENTION

When setting up the system and laying the necessary cables, make sure to fully comply with all standards, regulations, and rules regarding shielding (please also consult the relevant guidelines and documents published by the PROFIBUS User Organization). All work must be done professionally!

Shielding faults can result in serious malfunctions, including the system’s failure.

To ensure electromagnetic compatibility (EMC) in your control cabinets in electrically harsh environments, the following EMC rules are to be observed in the design:

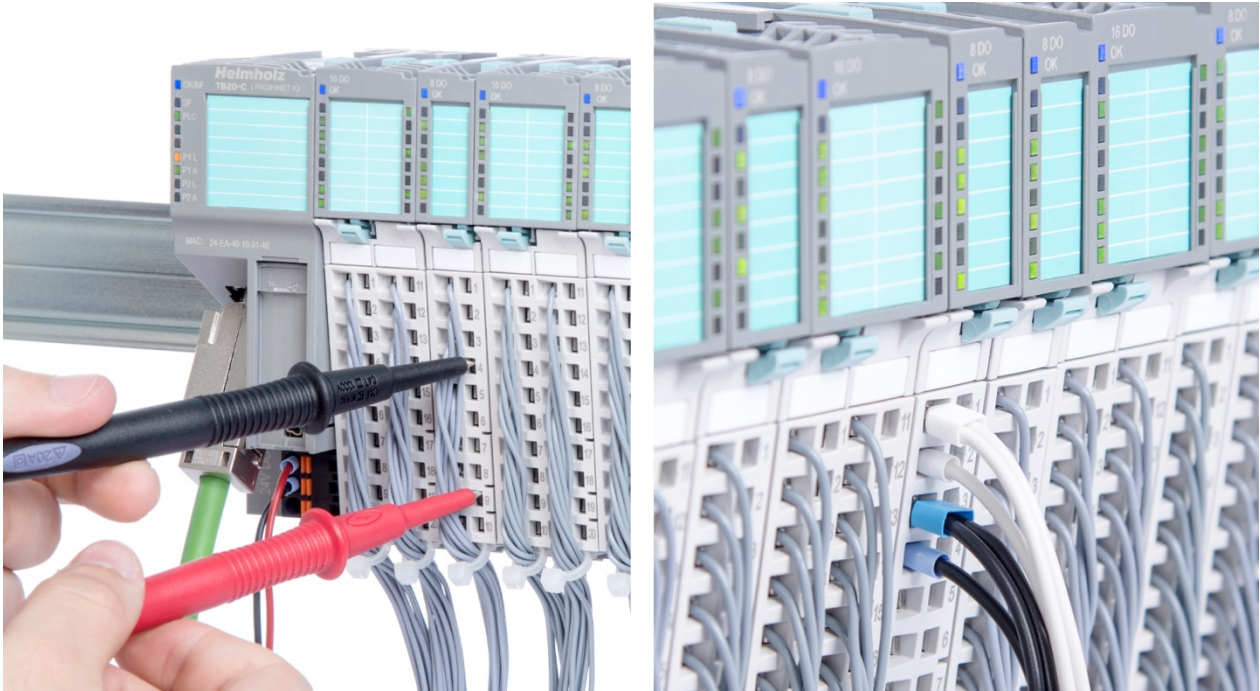
- All metal parts of the cabinet are to be connected with each other over a large area with good conductivity (no paint on paint). Where necessary, use contact washers or serrated washers.
- The cabinet door must be connected to the ground straps (top, middle, bottom) over as short a distance as possible.
- Signal cables and power cables are to be laid separated spatially by a minimum distance of 20 cm from each in order to avoid coupling paths.
- Run signal lines only from one level into the cabinet if possible.
- Unshielded cables in the same circuit (outgoing and incoming conductors) must be twisted if possible.
- Contactors, relays, and solenoid valves in the closet, or in adjacent cabinets if applicable, must be provided with quenching combinations; e.g., with RC elements, varistors, diodes.
- Do not lay wires freely in the closet; instead, run them as closely as possible to the cabinet housing or mounting panels. This also applies to reserve cables. These must be grounded on at least one end, and it is better if they are grounded at both ends (additional shielding effect).
- Unnecessary line lengths should be avoided. Coupling capacitances and inductances are kept low in this way.
- Analog signal lines and data lines must be shielded.

4.2. Front connectors

The front connector's spring-clamp terminals are designed for a cross-sectional cable area of up to 1.5 mm² (16–22 AWG) with or without ferrules.

It is also possible, for example, to connect two 0.75 mm² wires to a single spring-type terminal, provided the maximum cross-sectional cable area of 1.5 mm² per terminal is not exceeded.

The cables can be attached to the underside of the front connector with a cable tie.



4.3. Wiring the coupler

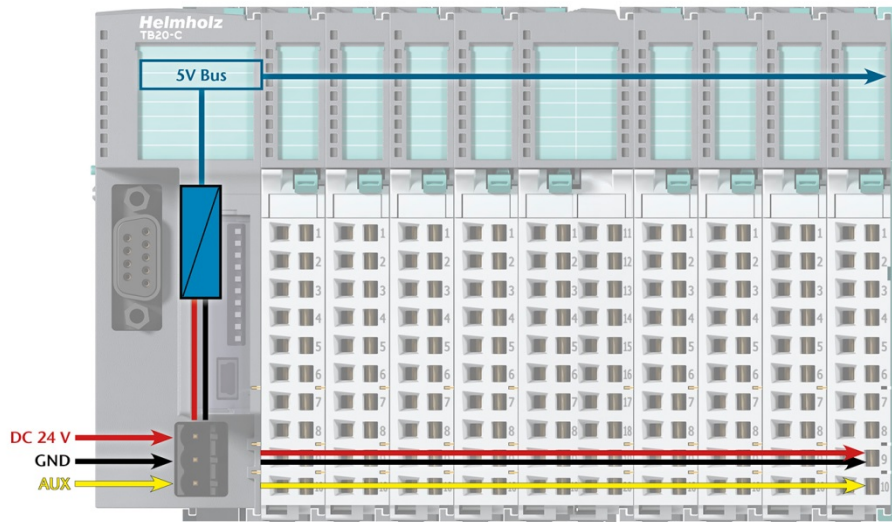
A power supply unit is integrated into the bus coupler. The power supply unit is responsible for powering the peripheral modules connected to the coupler.

In turn, it draws its own power from the three-pin connector on the front (24 VDC, GND, AUX).

The 24 V connector is used to power two buses:

- The power bus used to power the I/O components (24 VDC, GND, AUX)
- The communications bus used to power the electronics in the peripheral modules

The AUX pin can be used to connect and use an additional voltage potential. Every peripheral module has an AUX terminal on its front connector (the bottommost terminal, i.e., terminals 10 and 20).

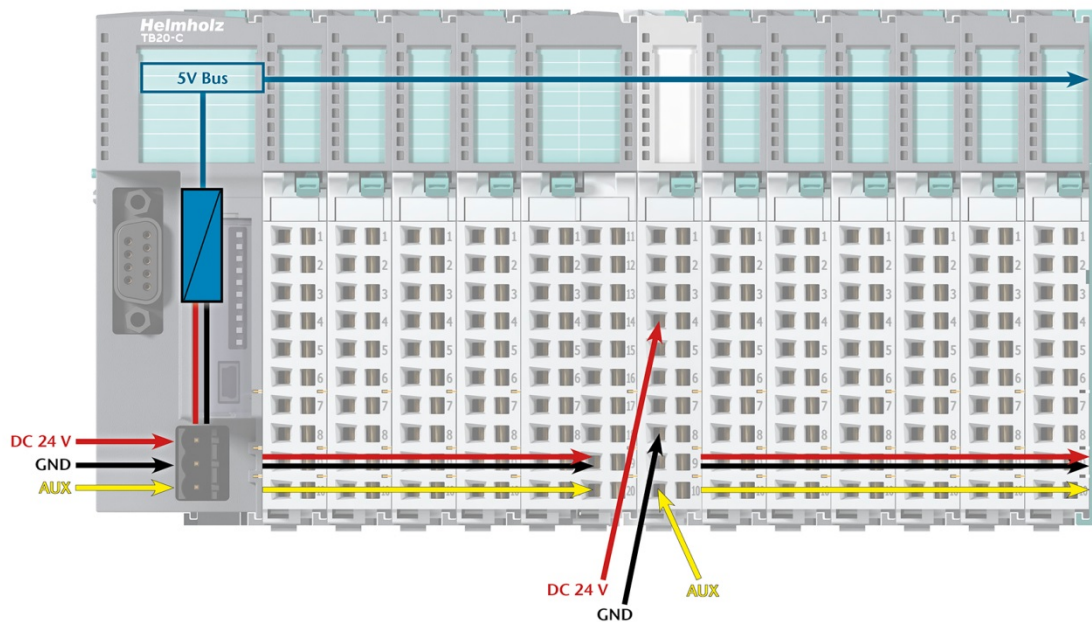


The coupler and the modules are grounded via the shield contact to the DIN rail. The DIN rail must be grounded. The surface of the DIN rail must be clean and conduct electricity well.



4.4. Using power and isolation modules

Power and isolation modules make it possible to segment the power supply for external signals (24 V, GND, AUX) into individual power supply sections that are powered separately.



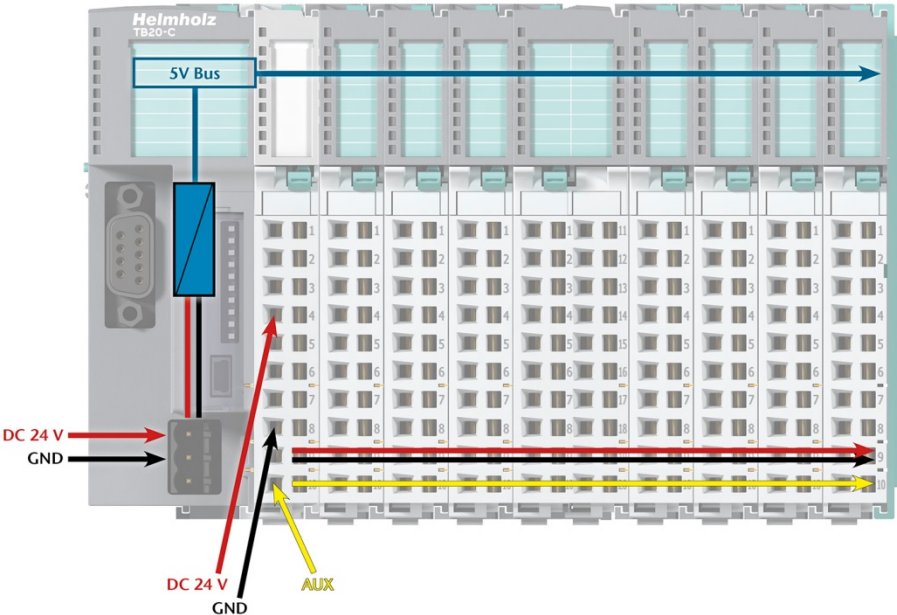
The order no. for the power and isolation module for 24 V signals is 600-710-0AA01.

Its electronic module and base module have the same light gray color as the front connector, ensuring that all power and isolation modules will stand out visually in the system and make it easy to clearly distinguish each individual power supply segment.



4.5. Separate power supply segments for the coupler and the I/O components

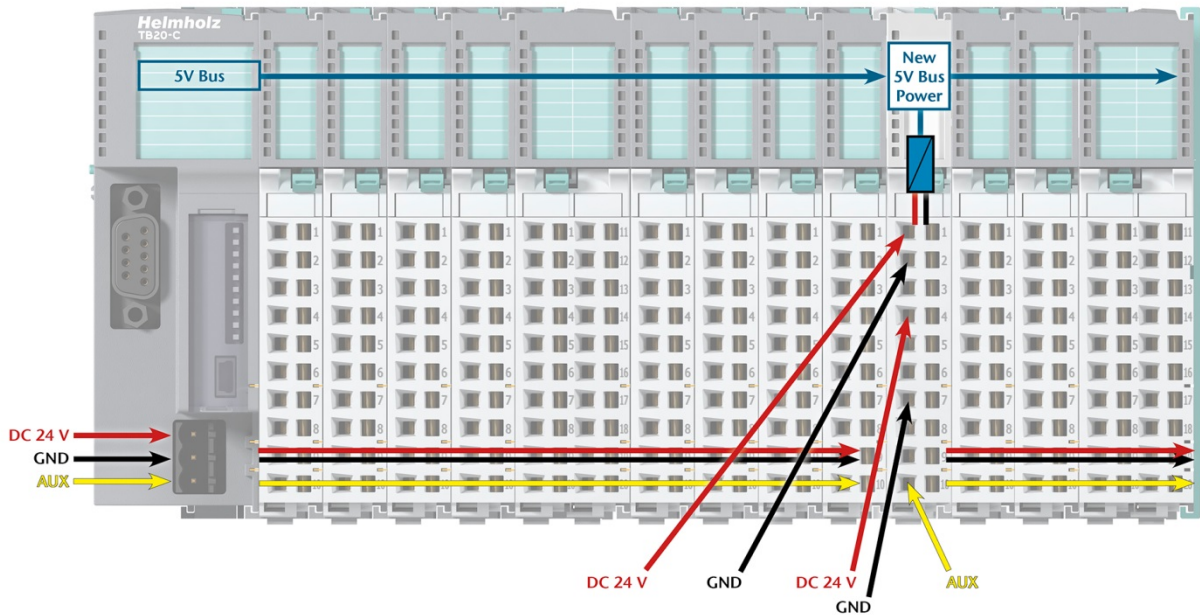
If the power supply for the coupler needs to be separate from the power supply for the I/O modules, a power and isolation module can be used right after the coupler.



4.6. Using power modules

Power modules deliver all necessary power to the connected peripheral modules and, if applicable, all the way to the next power module or power and isolation module. Power modules must be used whenever the power supplied by the coupler alone is not sufficient, that is, when there are a large number of modules on the bus. The “TB20 ToolBox” parameter configuration and diagnosis program can be used to calculate a system’s total current draw.

24 VDC, GND, and AUX are fed into the terminals on the front, while the connected modules are powered through the base modules’ bus system.



The order no. for the power module is 600-700-0AA01. The electronic module of the power module is light gray like the front connector. The base module of the power module is light gray with a dark top part.



4.7. Electronic nameplate

All of a TB20 module's important information can be found on its electronic nameplate. This information includes, for example, the corresponding module ID, module type, order number, unique serial number, hardware version, firmware version, and internal range of functionalities.

This information can be read in a number of ways, one of which is using the "TB20 ToolBox" configuration and diagnostics software. The modules' electronic nameplates not only make it possible to prevent configuration errors (setup), but also make maintenance (servicing) easier.

4.8. Fusing

The TB20 coupler's and power modules' power supply must be externally fused with a slow-blowing fuse, maximum 8 A, appropriate for the required maximum current.

5. CANopen Coupler Characteristics

The TB20 CANopen coupler has the following characteristics:

- CANopen® protocol as per DS 301 and DSP 401
- Bit rates of 50 kbps to 1 Mbps
- 24 TPDOs / 24 RPDOs
- 1 SDO server
- Heartbeat producer
- 2 x heartbeat consumers
- Node guarding
- SYNC object
- Storable parameter configuration
- A maximum of 64 modules
- Hot-swap capability
- 24 V DC power supply
- Integrated power supply unit for powering peripheral modules (2.5 A)
- Supplies the system's I/O voltage (24 VDC)
- DIP switch for setting the node ID and bit rate; can be covered
- 5 LEDs, two of them bi-color
- USB port for configuration and online diagnosis with “TB20-ToolBox”



6. CAN bus

CAN bus (Control Area Network) is an international open fieldbus standard for building, manufacturing, and process automation applications. CAN bus communications are defined in standard ISO 11898.



Comprehensive error detection measures, which result in a residual error probability of less than 4.7×10^{-11} , make the CAN bus the most reliable bus system on the market today.

6.1. CAN bus wiring

A CAN bus cable requires at least three wires: CAN high, CAN low, and CAN ground. Only linear bus topologies are allowed, and a terminating resistor of 120 ohms must be connected between CAN High and CAN Low at both ends of the CAN bus cable. Please note that the TB20 CANopen coupler does not feature an integrated terminating resistor.

The maximum cable lengths will depend mainly on the bit rate being used.

Bit rate	Bus length	Bit time
1 Mbps	30 m	1 μ s
800 kbps	50 m	1.25 μ s
500 kbps	100 m	2 μ s
250 kbps	250 m	4 μ s
125 kbps	500 m	8 μ s
50 kbps	1500 m	20 μ s

The cable lengths specified above should only be used as reference values. The maximum cable length will also depend on the number of connected nodes and on the cable type.

6.2. CAN bus plug

Helmholz GmbH & Co. KG offers a comprehensive range of CAN bus connectors that can be used with the TB20 CANopen coupler. All CAN bus connectors come with a terminating resistor that can be switched on and off.



6.3. The CANopen® protocol

The CANopen protocol is a layer 7 (application layer) protocol based on the CAN bus. CAN bus layers 1 and 2 (physical layer and data link layer) are not affected by the CANopen® protocol.



The service elements provided by the application layer make it possible to implement applications that are distributed throughout the network. CANopen communications profiles and standards are managed by CIA (CAN in Automation e.V.). Out of these standards, DS 301, “CANopen application layer and communication profile”, and DS 401, “Device profile for generic I/O modules”, are relevant to the TB20 CANopen coupler. These are available at CAN in Automation’s website (<http://www.can-cia.org>).

The CANopen standards assign a fixed definition to the 11-bit identifier and the 8 data bytes of a CAN bus frame. Each device in a CANopen network has a configurable node ID (module number, 1–127).

6.3.1. CANopen Objects

The data exchange with a CANopen slave takes place either via service data objects (SDOs) with fixed definitions or via process data objects (PDOs) that can be freely configured.

Every CANopen slave has a fixed directory of SDOs that are addressed with an object number (16 bits) and a subindex (8 bits).

Example: Object 0x1000 / Subindex 0 = Device type, 32-bit unsigned

SDOs with a width of 8/16/32 bits can be read and written with a CANopen frame, while longer SDOs (e.g. strings) are transferred using multiple frames.

SDOs can be processed as soon as a CANopen slave is ready for operation. The “SDO request” and “SDO response” functions are available for SDOs. The object number, access mode, and type are stored in the first four bytes of the CAN frame, The last four bytes of the CAN frame then contain the value for the SDO.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Command specifier	16 SDO index		8 bit SDO sub-index	1-4 bytes of parameter data			

PDOs contain the “operating values” of a CANopen slave for cyclical process operations. Each CANopen slave can administer multiple PDOs (normally up to four for transmission and four for reception).

Each of these PDOs will have its own COB-ID. The frame’s eight data bytes can be used to “map” any CANopen slave information for reading and writing. The values from the object dictionary (SDOs) are always the values mapped.

PDOs are automatically mapped at the startup by most CANopen slaves (default mapping). However, these mapping assignments can be changed by using defined SDOs.

6.3.2. CANopen Functions

The CANopen functions are subdivided into the following basic types:

- SDO read and SDO write operations
- PDO read and PDO write operations
- Network management
- Emergency messages

The function code is stored in the upper four bits of the CAN identifier, which, together with the node ID, forms the communication object identifier, or COB-ID. CANopen normally uses CAN bus frames with 11-bit identifiers (CAN 2.0A).

COB identifier (COB-ID):

10	9	8	7	6	5	4	3	2	1	0
Function				Node ID						

Broadcast functions:

Function	Function code (binary)	Resulting COB ID
NMT	0000	0h
SYNC	0001	80h
TIME STAMP	0010	100h

Node functions:

Function	Function code (binary)	Resulting COB ID
EMERGENCY	0001	81h–FFh
PDO1 (tx)	0011	181h–1FFh
PDO1 (rx)	0100	201h–27Fh
PDO2 (tx)	0101	281h–2FFh
PDO2 (rx)	0110	301h–37Fh
PDO3 (tx)	0111	381h–3FFh
PDO3 (rx)	1000	401h–47Fh
PDO4 (tx)	1001	481h–4FFh
PDO4 (rx)	1010	501h–57Fh
SDO (tx)	1011	581h–5FFh
SDO (rx)	1100	601h–67Fh
NMT error control	1110	701h–77Fh

“Tx” =
sent by the slave

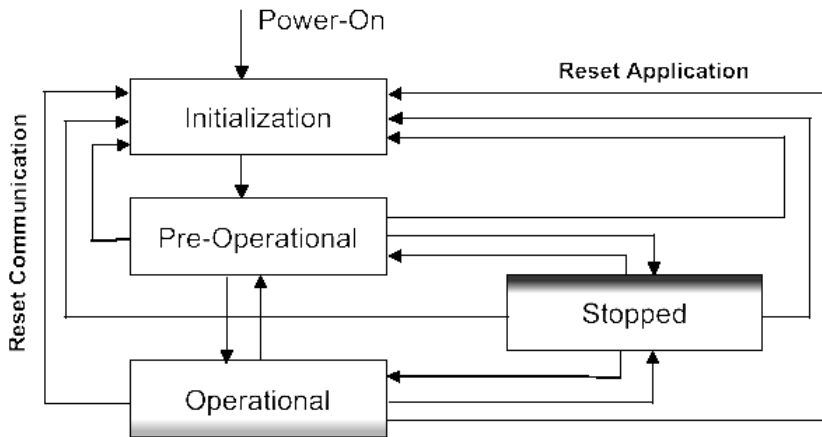
“Rx” =
received by the slave

6.3.3. Network management

Network status (NMT states):

Each CANopen slave can have several different system states. After switching on the device, the slave carries out an internal system initialization (hardware initialization, RAM test, setting up the basic objects) and reports a boot-up frame [COB-ID: 000_{hex} + node ID / data: 1 byte 00_{hex}] following successful initialization.

After this, the slave will be ready for operation and in the **Pre-Operational** state. Although the slave's parameters can be configured in this state (SDOs can be read and written), it will not transfer process data (PDOs).



With the NMT command “Operational” [COB-ID: 000_{hex} / data: 2 byte 01_{hex} node ID], the coupler can be switched to **Operational** status. Once in the Operational state, the coupler's process data will be enabled (RPDO/TPDO communications will be running). However, the options available for making changes to SDOs will be limited.

With the NMT command “Pre-Operational” [COB-ID: 000_{hex} / data: 2 byte 80_{hex} node ID], the coupler can be switched to **Pre-Operational** status.

With the NMT command “Reset (Application)” [COB-ID: 000_{hex} / data: 2 byte 81_{hex} node ID], a restart of the coupler is triggered.

With the NMT command “Reset Communication” [COB-ID: 000_{hex} / data: 2 byte 82_{hex} node ID], the coupler is reset, but the parameterization of the modules is maintained. and the coupler will switch to the **Pre-Operational** state.

SYNC:

The SYNC frame is a “broadcast” frame that is sent periodically in order to provide a bus clock signal for synchronization purposes. The SYNC frame makes it possible to apply input data in a synchronized manner and to activate output data simultaneously throughout the whole system. In order to ensure that the intervals at which the SYNC frames are sent are equally spaced, the SYNC frame has a high priority level. [COB-ID: 80h]

Node guarding:

Node guarding means that the master monitors the CANopen slave modules by cyclically sending frames to each slave. Each CANopen slave must respond to the node guarding frame with a status frame.

Node guarding enables the control system to detect failure of a CANopen slave. [COB-ID: 700h + node ID / data: 1 byte with slave status]

Life guarding:

Life guarding means that each CANopen slave will monitor to make sure that the master continuously carries out the started node guarding within specific time limits. If the node guarding frame is not sent, the distributed peripheral module can detect this with the life guarding function and, for example, switch all its outputs to a safe state.

The node guarding and life guarding functions should always be running at the same time.

Heartbeat:

Heartbeat monitoring works the same way as node guarding, with the exception that the CANopen® master does not generate any request frames. Instead, the heartbeat frame is sent automatically by each node (producer heartbeat) and can be evaluated in the master (consumer heartbeat). [COB-ID: 700h + node ID / data: 1 byte with slave status]

Emergency message:

If one of the CANopen slaves detects that something has gone wrong (e.g., life guarding times out), it will send an emergency message on the bus. [COB-ID: 80h + node ID / data: 8 bytes]

7. Setup and use

7.1. Address switches

The lower 7 switches of the 10-terminal DIP switch define the node ID of the coupler.

Changes to the address switches' positions will take effect only after the coupler is restarted.

When the node ID 0 is set, the CANopen coupler works in a test mode without CAN communication.

If a configuration is saved in the flash of the coupler, the COB IDs of the PDOs and the other messages can change when the node ID is readjusted following the restart.

7.2. Bit rate

The top 3 switches of the 10-terminal DIP switches define the CAN bit rate:

0	1	2	3	4	5	6	7
<i>Reserved</i>	50k	100k	125k	250k	500k	800k	1M

7.3. EDS file-based configuration

A standard EDS file (Electronic Data Sheet) can be downloaded from the Internet site www.helmholz.de.

It is also possible to export a project-specific EDS or DCF file from a TB20 ToolBox configuration. To this purpose read Chapter 0.

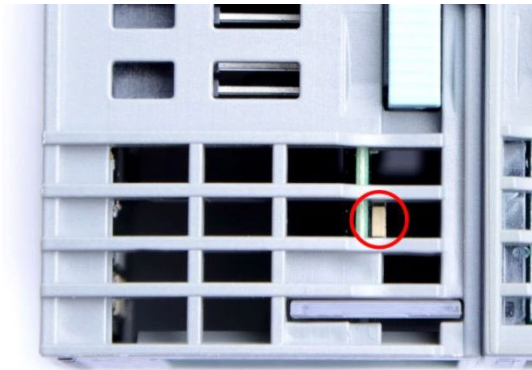
7.4. CAN bus connection

Pin	SubD plug CAN bus
1	-
2	CAN low
3	CAN GND
4	-
5	-
6	-
7	CAN high
8	-
9	-



7.5. Factory Reset (Restoring the Coupler to Its Factory Settings)

The CANopen coupler's factory settings can be restored by pressing a concealed button. All coupler parameters and the project will be deleted and the coupler will be restored to its original delivery condition. The button can be accessed from above through the ventilation opening of the housing (see illustration)



The factory settings are restored when the button is pressed while the device is powered up. As soon as the first three LEDs light up, you can let go of the button. At this point, the coupler will restart with its factory settings.

7.6. LED-based diagnosis

7.6.1. CANopen Coupler LEDs

The blue “OK” LED is used to indicate the coupler's general status:

Solid light: System is ready for operation, module configuration completed

Flashing light: Module configuration prevents the transition to pre-operational or operational (e.g. a missing module)

Quickly flashing light: Module configuration is false, startup cannot be completed (“NOT-Operational”)

The red “BF” LED (bus error) indicated error statuses:

Solid light: A module diagnosis report is available (e.g. wire break)

Flashing light: CAN bus errors have occurred

The yellow “SF” LED is used to indicate system errors and backplane bus errors:

Flashing light: Modules missing (startup) / Module removed (hot swap)

Solid light: Wrong module models plugged in

The green “PLC” LED is used to indicate the CANopen state:

Off: Pre-Operational

Solid light: Operational

Flashing light: Stop

Quickly flashing light: Simulation mode (triggered by TB20 ToolBox)

The yellow “TX” LED is used to indicate sent CAN frames.

The green “RX” LED is used to indicate received CAN frames.



7.6.2. Module LEDs

The topmost “OK/SF” LED indicates the current system status of each module.

Solid blue light: The module is running (RUN)

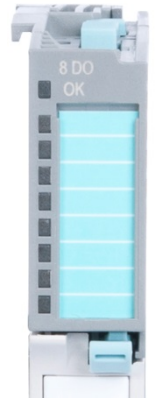
Slowly flashing blue light: The module is stopped (STOP); substitute values (if any) are being applied

Quickly flashing blue light: The module is idle (IDLE); its parameters have not been configured yet

Solid red light: The module is indicating a diagnostic error

Flashing red light: The module is indicating a parameter assignment error

The red “SF” LED lights will only be shown on modules with configurable parameters or diagnostic capabilities.



NOTE

IDLE mode (quickly flashing blue LED) indicates modules that have not been added to ongoing system operation by the coupler. One of the reasons that can cause this is an incorrect configuration (wrong module model on the slot).

7.7. Startup behavior

The startup process for the CANopen coupler is divided into three phases:

1. Performing a basic initialization and creating the basic objects (SDO 10xx)
2. The coupler scans the modules
3. In the case of error-free and correct module configuration, this is followed by the creation of all objects of relevance for the module (SDO 2000, 3000, 6000, PDO mapping) and the setting of these.

Whenever one of the following three events occurs, the CANopen coupler will start / restart:

1. Power on
2. “NMT Reset” command from CAN bus
3. Module configuration change

After the basic initialization process is complete, the coupler sends a bootup message (NMT Bootup) and starts scanning the modules.

The coupler will first wait for a completed module configuration, i.e., starting with a gap in the module configuration (even with hot plugging enabled) is not permitted.

Once the coupler detects a complete module configuration, the remaining process will depend upon whether a module configuration has been predefined, i.e., on whether a parameter configuration has been stored permanently with SDO 1010 (see also Ch. 7.11) or a project has been loaded from the TB20 ToolBox.

If no module configuration has been stored previously (no project in the flash memory), the coupler will create all module-relevant objects with the existing module configuration.

Once the initialization process has been successfully completed, the coupler will send an emergency message with error code 0 and will be ready for operation in the Pre-Operational state.

7.8. SDOs Assigned to I/O Data (SDOs 21xx, 22xx, 6000, 6200, 6401, 6411)

The modules' input data is stored in SDOs 2101 to 2140, while their output data is stored in SDOs 2201 to 2240.

The specific SDO number will depend on the relevant slot number. For example: input module in slot 5 →

SDO 2100 + slot number 5 → SDO 2105.

The configuration of the respective SDOs depends on the amount and type of input data. In the case of digital modules, UNSIGNED8 objects with 8 input bits each will be stored, while a single SIGNED16 object will be stored for each channel in the case of analog modules. In the case of more complex modules (e.g., counter modules), a RECORD with the relevant data elements will be created.

If a module does not have any input or output data, a length of 0 will be entered in subindex 0 in the corresponding SDO.

Digital and analog I/O data is classified in the corresponding SDOs according to the DS-401 profile. In the case of digital modules, only objects for 8-bit access are supported (SDO 6000, SDO 6200), while only objects for 16-bit access are supported in the case of analog modules (SDO 6401, SDO 6411).

7.9. Process Data Objects (PDOs)

7.9.1. PDO Mapping (SDOs 16xx/1Axx)

TPDO mapping SDOs 1A00–1A0F are used to map the modules' input data to TPDOs:

SDO 1A00 → Mapping TPDO 1
 SDO 1A01 → Mapping TPDO 2
 ...
 SDO 1A17 → Mapping TPDO 24

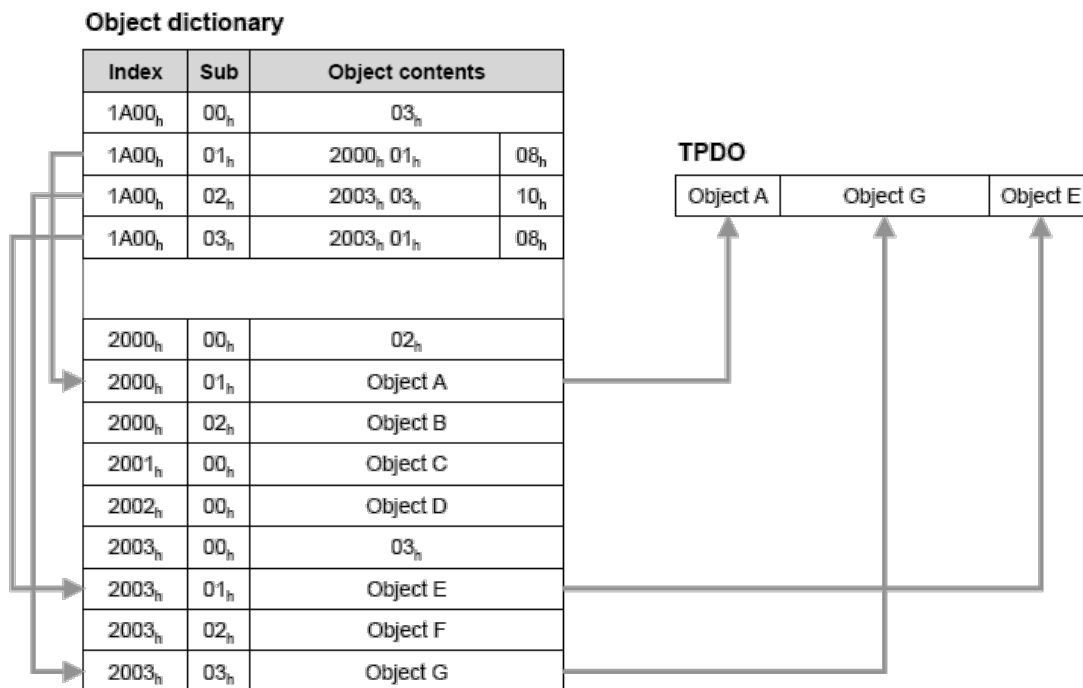
Meanwhile, RPDO mapping SDOs 1600ff. are used to map the modules' output data to RPDOs:

SDO 1600 → Mapping RPDO1
 SDO 1601 → Mapping RPDO2
 ...
 SDO 1617 → Mapping RPDO 24

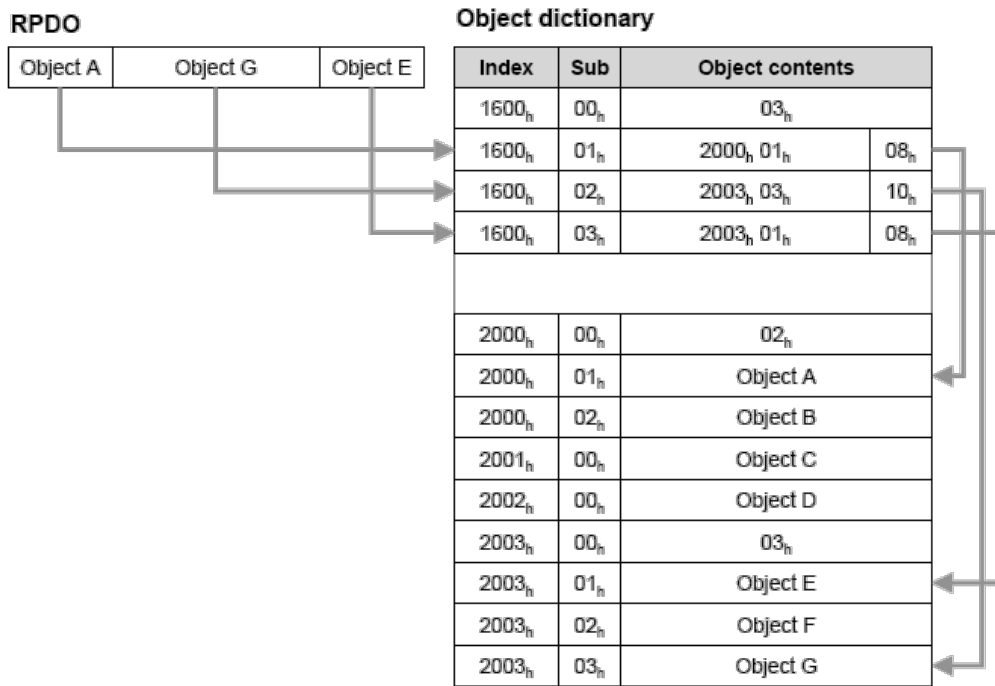
Each mapping object contains the active length of the mapping object (the number of objects actively being mapped) in subindex 0, as well as up to 8 mapping entries in the form of a list. The 8 data bytes of the corresponding PDO are filled from left (byte 0) to right (byte 7) without any gaps based on the data type of the mapped object.

Each mapping entry in subindexes 1–8 is made up of an index, the SDO subindex, and an entry containing the mapping data size (byte, word, double word).

TPDO1 mapping example:



RPDO1 mapping example:



In order to be able to change mapping assignments, the CANopen coupler must be in the Pre-Operational state. The first step is to disable the mapping assignment by writing a value of 0 to the mapping object length in subindex 0. Once you change the mapping entries (write to subindexes 1–8), the length of the mapping object (subindex 0) will be set to the number of objects being mapped and the new mapping assignment will be enabled. If any incorrect entries are detected, an emergency message will be sent as a response when the mapping assignment is enabled.

7.9.2. PDO Default Mapping

Automatic default PDO mapping is exclusive to digital and analog data objects. If other module types are used (e.g., counter module), the mapping assignments must be modified manually.

PDO1 = 8 bytes, digital (as per DS401)
 PDO2 = 4 words, analog (as per DS401)
 PDO3 = 4 words, analog (as per DS401)
 PDO4 = 4 words, analog (as per DS401)

PDO5 = 8 bytes, digital
 PDO6 = 8 bytes, digital
 PDO7 = 8 bytes, digital
 PDO8 = 8 bytes, digital

PDO9 = 4 words, analog
 PDO10 = 4 words, analog
 PDO11 = 4 words, analog
 PDO12 = 4 words, analog

PDO13 = 4 words, analog
 PDO14 = 4 words, analog
 PDO15 = 4 words, analog
 PDO16 = 4 words, analog

PDO17 = 4 words, analog
 PDO18 = 4 words, analog

PDO19 = 4 words, analog
 PDO20 = 4 words, analog
 PDO21 = 8 bytes, digital
 PDO22 = 8 bytes, digital
 PDO23 = 8 bytes, digital
 PDO24 = 8 bytes, digital

7.9.3. PDO Communication Parameters (SDOs 14xx/18xx)

SDOs 1400 (RPDO1) to 1417 (RPDO24) and SDOs 1800 (TPDO1) to 1817 (TPDO24) are used to define the communication properties for the PDOs.

Example using SDO 1800 / TPDO1:

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read-Write	5	Number of entries
1	UNSIGNED32	Read-Write	0x00000183	COB ID of the PDO (Example TPDO1 Node 3) Bit 31: 0 = PDO active / 1 = PDO turned off
2	UNSIGNED8	Read-Write	0xFF	PDO transmission type
3	UNSIGNED16	Read-Write	0x0000	Inhibit time (x * 100 µs)
4	UNSIGNED8	Read-Write	0x00	Not used
5	UNSIGNED16	Read-Write	0x0000	Event timer (x * 1 ms)

Communication properties can only be changed in the Pre-Operational state.

7.9.4. CAN Identifier for PDOs (SDOs 14xx/18xx, Subindex 1)

When the CANopen coupler is initialized, PDOs 1–4 are assigned the following default COB-IDs:

RPDO1 = 0x0200 + node ID
 RPDO2 = 0x0300 + node ID
 RPDO3 = 0x0400 + node ID
 RPDO4 = 0x0500 + node ID
 TPDO1 = 0x0180 + node ID
 TPDO2 = 0x0280 + node ID
 TPDO3 = 0x0380 + node ID
 TPDO4 = 0x0480 + node ID

All other PDOs (5-24) are switched off (Bit 31 is set), even when PDO Mapping is present. If you want to also use these PDOs, you will have to activate and configure them with a CAN identifier.

When the PDOs are activated in the TB20 ToolBox, they automatically receive the COB IDs of the following slave numbers:

RPDO5 = 0x0200 + node ID + 1	TPDO5 = 0x0180 + node ID + 1
RPDO6 = 0x0300 + node ID + 1	TPDO6 = 0x0280 + node ID + 1
RPDO7 = 0x0400 + node ID + 1	TPDO7 = 0x0380 + node ID + 1
RPDO8 = 0x0500 + node ID + 1	TPDO8 = 0x0480 + node ID + 1
RPDO9 = 0x0200 + node ID + 2	TPDO9 = 0x0180 + node ID + 2
RPDO10 = 0x0300 + node ID + 2	TPDO10 = 0x0280 + node ID + 2
RPDO11 = 0x0400 + node ID + 2	TPDO11 = 0x0380 + node ID + 2
RPDO12 = 0x0500 + node ID + 2	TPDO12 = 0x0480 + node ID + 2
RPDO13 = 0x0200 + node ID + 3	TPDO13 = 0x0180 + node ID + 3
RPDO14 = 0x0300 + node ID + 3	TPDO14 = 0x0280 + node ID + 3
RPDO15 = 0x0400 + node ID + 3	TPDO15 = 0x0380 + node ID + 3
RPDO16 = 0x0500 + node ID + 3	TPDO16 = 0x0480 + node-ID + 3

RPDO17 = 0x0200 + node ID + 4
RPDO18 = 0x0300 + node ID + 4
RPDO19 = 0x0400 + node ID + 4
RPDO20 = 0x0500 + node ID + 4

RPDO21 = 0x0200 + node-ID + 5
RPDO22 = 0x0300 + node-ID + 5
RPDO23 = 0x0400 + node-ID + 5
RPDO24 = 0x0500 + node-ID + 5

TPDO17 = 0x0180 + node ID + 4
TPDO18 = 0x0280 + node ID + 4
TPDO19 = 0x0380 + node ID + 4
TPDO20 = 0x0480 + node ID + 4

TPDO21 = 0x0180 + node-ID + 5
TPDO22 = 0x0280 + node-ID + 5
TPDO23 = 0x0380 + node-ID + 5
TPDO24 = 0x0480 + node-ID + 5



ATTENTION

If more than 4 TPDOs or 4 RPDOs are used, the COB IDs of the following node may be occupied by the automatically generated PDOs.



ATTENTION

If a configuration is saved in the flash of the coupler, and the node ID at the DIL switch is then adjusted, all PDO COB IDs will be automatically adapted to the new node ID that previously corresponded to the standard schema named above.

7.9.5. PDO Transmission Type (SDOs 14xx/18xx, Subindex 2)

The PDO transmission type is used to define when, and under which conditions, an RPDO will be processed or a TPDO will be sent.

The following transmission types are supported:

Event (0xFF / 0xFE):

Data is accepted immediately (RPDO) or sent immediately in case of a change (TPDO).

Sync (0–240):

Data is sent / accepted after a SYNC frame is received. If transmission type = 1, every single SYNC frame will be used as a trigger regardless of whether the value has remained constant or has changed.

Transmission type = 2–240 will activate a SYNC counter. Data will only be accepted or sent after the 2nd–240th SYNC frame.

If transmission type = 0 (acyclic SYNC), data will be sent after every SYNC frame, provided the data has changed.

Event-driven RTR (0xFD)

If the transmission type is set to “Event-driven RTR (0xFD)”, it will only be possible to fetch a TPDO’s data by using a request frame with the RTR bit set to 1. The TPDO will never be sent automatically in this mode.

In the case of RPDOs, the “Synchronous RTR (0xFC)” transmission type setting has no function.

Synchronous RTR (0xFC)

If the transmission type is set to “Synchronous RTR (0xFC)”, it will only be possible to fetch a TPDO’s data by using a request frame with the RTR bit set to 1. The input data is always only accepted internally into the TPDO following a SYNC frame. As long as a SYNC frame is not received, the TPDO will continue to retain the previous data. The TPDO will never be sent automatically in this mode.

In the case of RPDOs, the “Synchronous RTR (0xFC)” transmission type setting has no function.

7.9.6. PDO Inhibit Timer (SDOs 14xx/18xx, Subindex 3)

Inhibit timer for TPDOs:

An inhibit timer can be used in order to prevent a TPDO from being sent too often. The inhibit timer is only active during the transmission types 0xFF and 0xFE (event).

When transmission type 0xFF or 0xFE (event) is used, fast-changing input data can quickly flood the CAN bus. In order to prevent this, the inhibit timer can be used to set a minimum amount of time between transmissions.

The inhibit time is of type UNSIGNED16 and is defined in multiples of 100 µs. A value of 0 will disable the inhibit time.

The inhibit timer has no function in the case of RPDOs.

7.9.7. PDO Event Timer (SDOs 14xx/18xx, Subindex 5)

Event timer for TPDOs:

The event timer can be used to enable cyclic TPDO transmissions. As soon as the coupler is in the Operational state and the event timer has a value other than zero, the TPDO will be sent cyclically regardless of whether the value has changed.

Event timer for RPDOs:

In the case of RPDOs, the event timer can be used to define a monitoring timeout period for the reception of an RPDO. As soon as the coupler is in the Operational state and the first RPDO frame is received, the configured monitoring timeout period will be enabled. If the timeout period elapses before the next RPDO frame is received, the coupler will transmit an emergency message (8250) and, if necessary, change its NMT state based on SDO 1029 / subindex 1.

The event timer is of type UNSIGNED16 and is to be used as multiples of 1 ms. A value of 0 will disable the event timer.

The event timer is only active during the transmission types 0xFF and 0xFE (event).

7.10. Module Parameter Configurations (SDO 30xx)

If a module has parameters, these will be stored in SDOs 3001 to 3040.

The specific SDO number will depend on the module's slot number. For example: module in slot 5 → SDO 3000 + slot number 5 → SDO 3005.

Subindex 0 contains the number of parameter values for the module, while subindex 1–n contains the corresponding parameter values.

After the system has restarted, the SDOs will either contain the default parameter sets for the modules or the parameter sets previously stored through object 1010 (see Ch. 7.11).

There are two ways to change module parameters:

1. Directly changing a parameter: Every parameter in SDOs 30xx can be written to directly at any time. When doing so, make sure to keep in mind what the parameter's function is and what kind of dependencies it has. Making the wrong changes to parameters can cause parameter configuration errors and undefined module behavior.
2. Changing multiple parameters together: In order to change a parameter set completely and consistently, you will first have to set the length byte (subindex 0) to 0. After doing so, you can change the parameters in any order. The parameters will not be re-enabled in the module until after the length byte (subindex 0) is set back to the correct number of parameters.

Please refer to the module manuals for module parameter sets.

7.11. Storing a Parameter Configuration (SDO 1010)

SDO 1010 is used to permanently store the current state of the coupler or the parameter configuration of the module in the internal flash memory of the coupler even in the event of a power failure. To this purpose, the UNSIGNED32 value 0x73617665 (ASCII 'save') must be written in the corresponding subindex of the SDO.

Subindex 1 saves:

Combination between subindex 2 and subindex 3 (see below)

Subindex 2 saves:

System SDOs 1005, 100C, 100D, 1014, 1016, 1017, 1029

PDO settings and mapping SDOs 14xx, 16xx, 18xx, and 1Axx

SDO 2020

Subindex 3 saves:

List of module codes of the bus configuration for comparison (SDO 2F01)

Module parameter configurations, SDOs 3001–3040

After the power is turned on or a reset, the stored parameter configuration will be automatically used if the detected bus configuration matches the stored bus configuration. If the bus configurations do not match, at least the stored system SDOs (10xx) will be used.

Settings can only be changed when the coupler is in the Pre-Operational state.

7.12. Restoring Factory Settings (SDO 1011)

SDO 1011 can be used to delete / reset the settings stored with SDO 1010. When writing to one of the subindexes, an UNSIGNED32 value of 0x6C6F6164 (ASCII "load") must be written in order to trigger the reset operation.

Subindex 1 deletes all stored settings, meaning that the coupler is reset to its factory settings.

Subindex 2 deletes:

System SDOs 1005, 100C, 100D, 1014, 1016, 1017, 1029

PDO settings and mapping SDOs 14xx, 16xx, 18xx, 1Axx

SDO 2020

Subindex 3 deletes:

List of module codes of the bus configuration for comparison (SDO 2F01)

Module parameter configurations, SDOs 3001–3040

Factory settings can only be restored when the coupler is in the Pre-Operational state.



ATTENTION

After the stored parameters are reset, the coupler must be reset (power cycled or NMT reset) without fail so that the factory settings will take effect.

7.13. Module Diagnostics (SDO 31xx)

Some modules can send messages to the coupler when they detect internal errors or faults. For instance, an analog input module used to measure 4–20 mA can detect wire breaks.

The messages are stored in SDOs 3101 to 3140.

The specific SDO number will depend on the relevant slot number. For example: input module in slot 5 →

SDO 3100 + slot number 5 → SDO 3105.

If there are no messages, the subindex 0 will have a length of 0. If there is a diagnostic message, subindex 0 will hold the number of diagnostic bytes and subindexes 1–n will hold the module’s and channels’ diagnostic states.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	0 = There are no diagnostic messages n = There is a diagnostic message
1	UNSIGNED8	Read only	x	Diagnostic message for entire module
2	UNSIGNED8	Read only	x	Diagnostic message for channel 0
3	UNSIGNED8	Read only	x	Diagnostic message for channel 1
4	UNSIGNED8	Read only	x	Diagnostic message for channel 2
5	UNSIGNED8	Read only	x	Diagnostic message for channel 3

When the parameter bit “Diagnostic alarm” has been set for the module triggering the diagnostic message, an emergency message will be sent at the CAN bus upon occurrence and end of a diagnostic message. The diagnostic information will otherwise only be entered in the SDO.

Emergency message for an incoming diagnosis:

Emergency error code FFxx xx = Slot number	Error register (SDO 1001)	0 = module- wide / 1 –n =	Diagnostics code	Number of messages with diagnostic messages	<i>Reserved</i>	first module with emergency diagnosis
---	------------------------------	------------------------------------	---------------------	--	-----------------	---

Emergency message for an outgoing diagnosis:

Emergency error code FExx xx = Slot number	Error register (SDO 1001)	0 = module- wide / 1 –n =	0	Number of messages with diagnostic messages	<i>Reserved</i>	first module with emergency diagnosis
---	------------------------------	------------------------------------	---	--	-----------------	---

Diagnostics codes:

0 = There is no error / The error has been fixed

1 = Short circuit (to GND)

2 = Undervoltage

3 = Overvoltage

4 = Overload

5 = Excess temperature

6 = Line break / Wire break

7 = Overflow: Value falls above measuring range

8 = Underflow: Value falls below measuring range

16 = Power module: Bus supply voltage drop

17 = L+ reference voltage missing

18 = Incorrect parameterization

7.14. Behavior in the Event of an Error (SDO 1029)

SDO 1029 defines how the coupler will behave if an error occurs. The following categories of errors apply within this context:

Communications error (subindex 1): Bus error on CAN controller, life guarding monitoring timeout (SDO 100C/100D), heartbeat monitoring timeout (SDO 1016)

In alarm error (subindex 2): The coupler behavior if it receives a diagnostic message from input modules or combined input/output modules (see also Ch. 8.10).

Out alarm error (subindex 3): The coupler behavior if it receives a diagnostic message from output modules (see also Ch. 8.10).

The coupler behavior in response to the errors above can be defined in the corresponding subindex.

00_{hex} = Switch to Pre-Operational NMT state

01_{hex} = No response

02_{hex} = Switch to Stop NMT state



ATTENTION

If a module communication error occurs, the coupler will always switch to the Pre-Operational NMT state and send an emergency message. This may also happen, for example, when a module fails or is removed (or when two modules are removed if hot-swapping is permitted).

7.15. Hot-swap function (SDO 2020)

SDO 2020 is used to define the coupler's behavior when a module is removed while the coupler is in the Operational state.

SDO 2020 = 0 (*default*): Hot-swapping is not allowed! Removing a module in Operational state will cause the system to stop (→ Pre-Operational state)

SDO 2020 = 1: Hot-swapping is allowed! Replacement of a module in Operational state is permitted.

If SDO 2020 = 0, any change made to the bus configuration while the coupler is in the Operational state will trigger an emergency and the coupler will switch to the Pre-Operational state (the yellow "SF-LED" will flash).

If hot-swapping is allowed, an emergency message will be sent when a module is removed, and the fact that the module is missing will be recorded in the list of modules (the yellow "SF-LED" will show a solid light). If the module is plugged back in, an emergency message will be transmitted again. The type of message and response depends on whether a module of the same model is unplugged or a different module has been plugged in.

If the module is of the same model as the previous module, the module's parameters will be configured with the stored data and started accordingly. If the module doesn't match, the module will remain in the IDLE state and will not be used further.



NOTE

Do not remove more than one module at any one time. Removing a second module will cause the system to stop!

The LED indicator of the module makes it possible to easily find out which modules have been incorrectly configured or are of the wrong type (see also Ch. 7.6.2).

8. SDO object table

8.1. System objects (SDO 1000–1200, 2020)

8.1.1. Device Type (1000)

CANopen coupler model.

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	0x000a0191	“0191” = DS401 profile a = Bit coding (DI, DO, AI, AO) as specified in DS401

8.1.2. Error Register (1001)

CANopen coupler error state

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	0x00	Bit 0: Generic error Bit 4: Node guarding or heartbeat timeouts or PDO length errors (Emcy. 8xxx) Bit 5: Diagnostic errors of the modules (Emcy. FFxx) Bit 7: Bus configuration (Emcy. 7xxx)

8.1.3. Pre-Defined Error Field (1003)

List of most recent error messages (emergency history).

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	0x00	Number of entries
1	UNSIGNED32	Read only		The most recent error message
...				...
n	UNSIGNED32	Read only		The oldest error message

8.1.4. Number of PDOs (1004)

Number of existing RPDOs and TPDOs.

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	0x00180018	High word = Number of TPDOs (24) Low word = Number of RPDOs (24)

8.1.5. SYNC identifier (1005)

COB ID of the SYNC message.

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read/write	0x00000080	SYNC identifier (0x0001 – 0x07FF) Bit 31 = 1 → deactivation of the SYNC

8.1.6. Device Name (1008)

Contains the CANopen coupler’s device name in the form of a string.

Subindex	Type	Access	Default	Description
0	VISSTRING	Read only	“TB20 CANopen”	Name of the coupler

8.1.7. Hardware version (1009)

Contains the CANopen coupler's hardware version in the form of a string.

Subindex	Type	Access	Default	Description
0	VISSTRING	Read only	"HW1"	The coupler's hardware version

8.1.8. Firmware version (100A)

Contains the CANopen coupler's device name in the form of a string.

Subindex	Type	Access	Default	Description
0	VISSTRING	Read only	"1.00.000"	Firmware version

8.1.9. Node ID (100B)

The CANopen coupler's currently active node number.

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	0x00000003	Active node number (using node 3 as an example)

8.1.10. Guard Time (100C)

Subindex	Type	Access	Default	Description
0	UNSIGNED16	Read/write	0x0000	Node guarding monitoring time

8.1.11. Life Time Factor (100D)

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read/write	0x00	Node guarding monitoring interval

8.1.12. Store (1010) (Used to Store Parameters)

Used to store the coupler and module settings in non-volatile memory (see also Ch. 7.11).

When writing to one of the subindexes, an UNSIGNED32 value of 0x73617665 (ASCII "save") must be written in order to trigger the storing operation.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	3	Number of entries
1	UNSIGNED32	Read/write	1	Stores the system & module SDOs
2	UNSIGNED32	Read/write	1	Saves system SDOs 1005, 100C, 100D, 1014, 1016, 1017, 1029
3	UNSIGNED32	Read/write	1	Stores the module configuration (SDO 1027) and the module parameters (SDO 30xx)

8.1.13. Restore (1011) (Used to Load Parameters)

Resetting the coupler and module settings to factory settings.

When writing to one of the subindexes, an UNSIGNED32 value of 0x6C6F6164 (ASCII "load") must be written in order to trigger the reset operation. After a positive response to the SDO write operation, a reset needs to take place in order to activate the restored parameters.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	3	Number of entries
1	UNSIGNED32	Read/write	1	Resets all SDOs

2	UNSIGNED32	Read/write	1	Reset system SDOs 1005, 100C, 100D, 1014, 1016, 1017, 1029
3	UNSIGNED32	Read/write	1	Resets the module configuration (SDO 1027) and the module parameters (SDO 30xx)

8.1.14. Emergency identifier (1014)

COB ID of the emergency message.

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read/write	0x00000083	Emergency Identifier (using node 3 as an example)

8.1.15. Consumer Heartbeat (1016)

The CANopen coupler can monitor two external heartbeats. In order to enable heartbeat monitoring, an UNSIGNED32 value that specifies both the node ID to be monitored and the monitoring timeout must be written to subindex 1 or 2:

0x00	Node ID	Monitoring timeout, in ms
------	---------	---------------------------

Monitoring will be activated by the first heartbeat frame received from the slave.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	2	Number of entries
1	UNSIGNED32	Read/write	0x00000000	Consumer heartbeat 1
2	UNSIGNED32	Read/write	0x00000000	Consumer heartbeat 2

8.1.16. Producer Heartbeat (1017)

SDO 1017 is used to define the interval at which the coupler will send a heartbeat telegram, containing its current NMT state, on the bus. Once this interval is defined, the coupler will immediately begin transmitting the frame, regardless of its NMT state.

Subindex	Type	Access	Default	Description
0	UNSIGNED16	Read/write	0x00000000	Producer heartbeat interval, in ms

8.1.17. Identity Object (1018)

Detailed device code and device information.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	4	Number of entries
1	UNSIGNED32	Read only	0x00000223	Vendor ID
2	UNSIGNED32	Read only	0x000003E8	Product code 1000
3	UNSIGNED32	Read only	0x00010000	Version number: 1.00.000
4	UNSIGNED32	Read only	0x00018ED0	Serial number: 102096 (example)

8.1.18. Module List (1027)

SDO 1027 contains the TB20 modules' models.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	UNSIGNED16	Read only	0xYYYY	Module model on slot 1
...				...
n	UNSIGNED16	Read only	0xYYYY	Module model on slot n

Any gaps in the bus configuration will be flagged with a module model with a value of 0.

Module type 0xFFFF is used to flag a module with the wrong module model (i.e., the last time there was a valid system startup, there was a different module in the slot, or SDO 1010 has been used to store a specific module configuration and the current module does not match it).

SDO 2F01 (see Ch. 8.7.2) can be used to read the anticipated module model at the corresponding slot.

The module types and codes can be found in the module manuals.

8.1.19. Object Behavior in Case of Error (1029)

Behavior of the CANopen coupler when errors occur and when modules send diagnostic messages.

The following options can be used:

00_{hex} = Switch to Pre-Operational NMT state

01_{hex} = No status change

02_{hex} = Switch to Stop NMT state

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	3	Number of entries
1	UNSIGNED8	Read/write	0x00	Response to communication errors
2	UNSIGNED8	Read/write	0x00	Response to diagnostic messages from input modules
3	UNSIGNED8	Read/write	0x00	Response to diagnostic messages from output modules

8.1.20. SDO server parameters (1200)

COB IDs of the SDO CAN frame.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	2	Number of entries
1	UNSIGNED32	Read/write	0x00000603	COB-ID for SDO request (using node 3 as an example)
2	UNSIGNED32	Read/write	0x00000583	COB-ID for SDO confirm (using node 3 as an example)

8.1.21. Hot-swap behavior (2020)

Defining the hot-swap behavior (see Ch. 7.15).

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read/write	0x00	0: No hot-swap allowed. 1: Hot-swap is allowed.

8.2. RPDO Communication Parameters (SDO 14xx)

SDOs 1400 (RPDO1) to 1417 (RPDO24) are used to define the communication properties for the RPDOs. For more details, please refer to Ch. 7.9.3ff.

Communication properties can only be changed when the coupler is in the Pre-Operational state.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read-Write	5	Number of entries
1	UNSIGNED32	Read-Write	0x00000203	COB ID of the RPDO (Example RPDO1 Node 3) Bit 31: 0 = RPDO active / 1 = RPDO turned off
2	UNSIGNED8	Read-Write	0xFF	RPDO transmission type
3	UNSIGNED16	Read-Write	0x0000	Inhibit time (no function)
4	UNSIGNED8	Read-Write	0x00	Not used
5	UNSIGNED16	Read-Write	0x0000	Event time (x * 1 ms)

8.3. RPDO Mapping (SDO 16xx)

SDOs 1600 (RPDO1) to 1617 (RPDO24) are used to define the mapping for the RPDOs. After the coupler is restarted or reset, the mapping configuration will return to the default mapping configuration (please refer to Ch. 7.9.2 as well). In order to be able to change the mapping configuration, the coupler must be in the Pre-Operational state and the length byte (subindex 0) has to be set to 0 first.

The configuration of SDOs 1600–1617 is identical. Example using SDO 1600 (RPDO1):

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read-Write	8	Number of entries
1	UNSIGNED32	Read-Write	0x62000108	1. mapping entry for RPDO x
2	UNSIGNED32	Read-Write	0x62000208	2. mapping entry for RPDO x
3	UNSIGNED32	Read-Write	0x62000308	3. mapping entry for RPDO x
4	UNSIGNED32	Read-Write	0x62000408	4. mapping entry for RPDO x
5	UNSIGNED32	Read-Write	0x62000508	5. mapping entry for RPDO x
6	UNSIGNED32	Read-Write	0x62000608	6. mapping entry for RPDO x
7	UNSIGNED32	Read-Write	0x62000708	7. mapping entry for RPDO x
8	UNSIGNED32	Read-Write	0x62000808	8. mapping entry for RPDO x

8.4. TPDO Communication Parameters (SDOs 18xx)

SDOs 1800 (TPDO1) to 1817 (TPDO24) are used to define the communication properties for the TPDOs. For more details, please refer to Ch. 7.9.3ff.

Communication properties can only be changed when the coupler is in the Pre-Operational state.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read-Write	5	Number of entries
1	UNSIGNED32	Read-Write	0x00000183	COB ID of the TPDO (Example RPDO1 Node 3) Bit 31: 0 = TPDO active / 1 = TPDO turned off
2	UNSIGNED8	Read-Write	0xFF	TPDO transmission type
3	UNSIGNED16	Read-Write	0x0000	Inhibit time (x * 100 µs)
4	UNSIGNED8	Read-Write	0x00	Not used
5	UNSIGNED16	Read-Write	0x0000	Event timer (x * 1 ms)

8.5. TPDO Mapping (SDO 1Axx)

SDOs 1A00 (TPDO1) to 1A17 (TPDO24) are used to define the mapping for the TPDOs. After the coupler is restarted or reset, the mapping configuration will return to the default mapping

configuration (please refer to Ch. 7.9.2 as well). In order to be able to change the mapping configuration, the coupler must be in the Pre-Operational state and the length byte (subindex 0) has to be set to 0 first.

The configuration for SDOs 1A00–1A17 is identical. Example using SDO 1A00 (TPDO1):

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read-Write	8	Number of entries
1	UNSIGNED32	Read-Write	0x60000108	1. mapping entry for TPDO x
2	UNSIGNED32	Read-Write	0x60000208	2. mapping entry for TPDO x
3	UNSIGNED32	Read-Write	0x60000308	3. mapping entry for TPDO x
4	UNSIGNED32	Read-Write	0x60000408	4. mapping entry for TPDO x
5	UNSIGNED32	Read-Write	0x60000508	5. mapping entry for TPDO x
6	UNSIGNED32	Read-Write	0x60000608	6. mapping entry for TPDO x
7	UNSIGNED32	Read-Write	0x60000708	7. mapping entry for TPDO x
8	UNSIGNED32	Read-Write	0x60000808	8. mapping entry for TPDO x

8.6. Coupler identification (SDO 200x)

8.6.1. Module ID of the coupler (2001)

Subindex	Type	Access	Default	Description
0	UNSIGNED16	Read only	3	TB20 module ID of the CANopen [®] coupler

8.6.2. Order number of the coupler (2002)

Subindex	Type	Access	Default	Description
0	VISSTRING	Read only	'600-160-1AA11'	The coupler's order no.

8.6.3. Series number of the coupler (2003)

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	x	The coupler's serial number

8.6.4. Firmware version of the coupler (2004)

SDO 2004 contains the firmware versions of the coupler. The firmware versions are of type UNSIGNED32. Syntax: 0X00aabbcc → V a.bb.ccc (decimal representation)

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	0x00aabbcc	Firmware version of the coupler

8.6.5. Hardware version of the coupler (2006)

Subindex	Type	Access	Default	Description
0	VISSTRING	Read only	'HW1-1'	The coupler's hardware version

8.6.6. Active bit rate (2010)

Subindex	Type	Access	Default	Description
0	UNSIGNED32	Read only	1000000	Bit rate in bit/s

8.7. Module Configuration (SDO 2F0x)

8.7.1. Module Models (2F00)

SDO 2F00 contains the TB20 modules' models. For more information on module models, please refer to the module manuals. Any gaps in the bus configuration will be flagged with a module model with a value of 0.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	UNSIGNED16	Read only	0xYYYY	Module model on slot 1
...				...
n	UNSIGNED16	Read only	0xYYYY	Module model on slot n

8.7.2. Module Codes (2F01)

SDO 2F01 contains the TB20 modules' module codes. For more information on module codes, please refer to the module manuals. The module codes are coded as decimal numbers. A gap in the bus configuration is marked with the module ID 0 when no module has been plugged in here previously and no module is saved in the flash. The module ID of the expected module is otherwise displayed.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	UNSIGNED16	Read only	YYYYY	Module detection slot 1
...				...
n	UNSIGNED16	Read only	YYYYY	Module detection slot n

8.7.3. Module Order Numbers (2F02)

SDO 2F02 contains the order numbers for the TB20 modules, in VISSTRING format. Any gaps in the bus configuration will be flagged with a 0.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	VISSTRING	Read only	'600-xxx-xYYzz'	Order number for module in slot 1
...				...
n	VISSTRING	Read only	'600-xxx-xYYzz'	Order number for module in slot n

8.7.4. Module Serial Numbers (2F03)

SDO 2F03 contains the TB20 modules' serial numbers. Serial numbers are coded in decimal and are of type UNSIGNED32. Any gaps in the bus configuration will be flagged with a serial number with a value of 0.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	UNSIGNED32	Read only	YYYYYYYYY	Serial number of module in slot 1
...				...
n	UNSIGNED32	Read only	YYYYYYYYY	Serial number of module in slot n

8.7.5. Firmware versions of the modules (2F04)

SDO 2F04 contains the firmware versions of the TB20 modules. The firmware versions are of type UNSIGNED32. Syntax: 0X00aabbcc → V a.bb.ccc (decimal representation)

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	UNSIGNED32	Read only	0x00aabbcc	Firmware version of module in slot 1
...				...
n	UNSIGNED32	Read only	0x00aabbcc	Firmware version of module in slot n

8.7.6. Module Hardware Versions (2F06)

SDO 2F06 contains the hardware versions corresponding to the TB20 modules, in VISSTRING format.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of detected/active slots/modules
1	VISSTRING	Read only	"HW1"	Hardware version of module in slot 1
...				...
n	VISSTRING	Read only	"HW1"	Hardware version of module in slot n

8.8. Module Data (SDO 21xx, 22xx, 6xxx)

8.8.1. Input Data (21xx)

SDOs 2102 to 2140 hold the modules' input data. This input data can be read as soon as the module is in the STOP or RUN state.

The specific SDO number will depend on the relevant slot number. For example: input module in slot 5 →

SDO 2100 + slot number 5 → SDO 2105.

If a module does not have any input data, a length of 0 will be written to subindex 0. The data types depend upon the module type.

Input data from digital input modules:

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	1 / 2	Amount of input data 1 / 2 bytes
1	UNSIGNED8	Read only	0x00	Input bits 0-7 (2 DI, 3 DI, 4 DI, 6 DI, 8 DI, 16 DI)
2	UNSIGNED8	Read only	0x00	Input bits 8–15 (16 DI)

Input data from analog input modules:

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	1–n	Number of input channels
1	INTEGER16	Read only	0	Analog input value on channel 0
2	INTEGER16	Read only	0	Analog input value on channel 1
3	INTEGER16	Read only	0	Analog input value on channel 2
4	INTEGER16	Read only	0	Analog input value on channel 3

In the case of more complex modules, such as counter modules, the SDO contains the input data in the form of a RECORD with all elements. Please refer to the module's manual for the corresponding structure.

8.8.2. Output Data (22xx)

SDOs 2201 to 2240 hold the modules' output data. Writing to these SDOs will automatically activate the output data in the relevant module, provided the module is in the RUN state (and the coupler is in the Operational state). If the module is still in the STOP state (and the coupler is in the Pre-Operational state), the data will be stored and will be actively applied as soon as there is a transition to the RUN (Operational) state.

The specific SDO number will depend on the relevant slot number. For example: output module in slot 5 →

SDO 2200 + slot number 5 → SDO 2205.

If a module does not have any output data, a length of 0 will be written to subindex 0. The data types depend upon the module type.

Output data from digital output modules:

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	1 / 2	Amount of output data 1 / 2 bytes
1	UNSIGNED8	Read/write	0x00	Output bits 0–7 (2 DO, 4 DO, 8 DO, 16 DO)
2	UNSIGNED8	Read/write	0x00	Output bits 8–15 (16 DO)

Output data for analog output modules:

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	1–n	Number of output channels
1	INTEGER16	Read/write	0	Analog output value on channel 0
2	INTEGER16	Read/write	0	Analog output value on channel 1
3	INTEGER16	Read/write	0	Analog output value on channel 2
4	INTEGER16	Read/write	0	Analog output value on channel 3

In the case of more complex modules, such as a counter module, the SDO contains the output data in the form of a RECORD with all elements. Please refer to the module's manual for the corresponding structure.

8.8.3. Digital Inputs (6000)

Digital input data is also stored in SDO 6000, as specified in CANopen profile DS 401. All input data is stored byte-wise in the subindexes in SDO 6000. Writing starts with the first byte of the first digital input module (from the left) and keeps filling the subindexes all the way to the last byte of the last digital input module.

This input data can be read as soon as the module is in the STOP or RUN state.

Only modules that also have digital input data will be taken into account. There is no relation whatsoever between the subindex in the SDO and the module's slot number.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Amount of input data 0–n bytes
1	UNSIGNED8	Read only	0x00	The first module's first input bits
...				
n	UNSIGNED8	Read only	0x00	The last module's last input bits

8.8.4. Digital Outputs (6200)

Digital output data is also made available in SDO 6200, as specified in CANopen profile DS 401. All output data is stored byte-wise in the subindexes in SDO 6200. Writing starts with the first byte of the first digital output module (from the left) and keeps filling the subindexes all the way to the last byte of the last digital output module.

Writing to these SDOs will automatically activate the output data in the relevant module, provided the module is in the RUN state (and the coupler is in the Operational state). If the module is still in the STOP state (and the coupler is in the Pre-Operational state), the data will be stored and will be actively applied as soon as there is a transition to the RUN (Operational) state.

Only modules that also have digital output data will be taken into account. There is no relation whatsoever between the subindex in the SDO and the module's slot number.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Amount of output data 0–n bytes
1	UNSIGNED8	Read/write	0x00	The first module's first output bits
...				
n	UNSIGNED8	Read/write	0x00	The last module's last output bits

8.8.5. Analog Inputs (6401)

Analog input data is also stored in SDO 6401, as specified in CANopen profile DS 401. All analog input data is stored channel-wise, in INTEGER16 format, in the subindexes in SDO 6401. Writing starts with the first channel of the first analog input module (from the left) and keeps filling the subindexes all the way to the last channel of the last analog input module.

This input data can be read as soon as the module is in the STOP or RUN state.

Only modules that also have analog input data will be taken into account. There is no relation whatsoever between the subindex in the SDO and the module's slot number.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of input channels
1	INTEGER16	Read only	0	The first module's first channel
...				
n	INTEGER16	Read only	0	The last module's last channel

8.8.6. Analog Outputs (6411)

Access to analog output data is also provided in SDO 6411, as specified in CANopen profile DS 401. All output data is stored channel-wise in the subindexes in SDO 6411. Writing starts with the first channel of the first analog output module (from the left) and keeps filling the subindexes all the way to the last channel of the last analog output module.

Writing to these SDOs will automatically activate the output data in the relevant module, provided the module is in the RUN state (and the coupler is in the Operational state). If the module is still in the STOP state (and the coupler is in the Pre-Operational state), the data will be stored and will be actively applied as soon as there is a transition to the RUN (Operational) state.

Only modules that also have analog output data will be taken into account. There is no relation whatsoever between the subindex in the SDO and the module's slot number.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	Number of output channels
1	INTEGER16	Read/write	0x00	The first module's first channel
...				
n	INTEGER16	Read/write	0x00	The last module's last channel

8.9. Module Parameters (SDO 30xx)

SDOs 3001 to 3040 hold the modules' parameter sets. This input data can be read as soon as the module is in the STOP or RUN state.

The specific SDO number will depend on the relevant slot number. For example: module in slot 5 → SDO 3000 + slot number 5 → SDO 3005.

Subindex 0 contains the number of parameter values for the module, while subindex 1–n contains the corresponding parameter values.

After the system is restarted, the SDOs will either contain the default parameter sets for the modules or the parameter sets previously stored with object 1010 (see Ch. 7.11).

Please refer to the module manuals for module parameter sets.

8.10. Module diagnostic states (SDO 31xx)

The SDOs 3101 to 3140 contain the diagnostic states of modules that support diagnostic modules (e.g. analog modules).

The specific SDO number will depend on the relevant slot number. For example: input module in slot 5 →

SDO 3100 + slot number 5 → SDO 3105.

If there are no diagnostic messages, the subindex will have a length of 0. If there is a diagnostic message, subindex 0 will hold the number of diagnostic bytes and subindexes 1–n will hold the module's and channels' diagnostic states.

Subindex	Type	Access	Default	Description
0	UNSIGNED8	Read only	x	0 = There are no diagnostic messages n = There is a diagnostic message
1	UNSIGNED8	Read only	x	module-wide diagnosis
2	UNSIGNED8	Read only	x	Diagnosis for channel 0
3	UNSIGNED8	Read only	x	Diagnosis for channel 1
4	UNSIGNED8	Read only	x	Diagnosis for channel 2
5	UNSIGNED8	Read only	x	Diagnosis for channel 3
...				
34	UNSIGNED8	Read only	x	Diagnosis for channel 32

9. Emergency messages

The CANopen coupler supports the standard CANopen emergency protocol. Emergency messages are generated by the coupler itself (internal errors, hot-swap), or are triggered by diagnostic messages from the modules (see also Ch. 8.10).

The 8 data bytes in an emergency message have the following structure:

Emergency error code XXXX	Error register (SDO 1001)	Byte 3 Manufacturer	Byte 4 Manufacturer	Byte 5 Manufacturer	Byte 6 Manufacturer	Byte 7 Manufacturer
------------------------------	------------------------------	------------------------	------------------------	------------------------	------------------------	------------------------

The following emergency messages are generated (emergency error code):

0000 OK

Is sent when no more errors are present (Error directory = 00).

7001 OPERATIONAL_INHIBITED

This message is transmitted when a change in state to operational cannot take place after a “Switch to Operational” NMT command because the bus configuration contains a gap or incorrect modules. These modules may include a module for which a different type has been stored in the flash or a module that has been replaced with a non-matching module during hot swapping.

Byte 3: Number of the first slot with errors.

Byte 4: Number of the first slot that is not in agreement with the configuration in the flash.

Byte 5: Number of the first slot with a false module type.

Byte 6: Number of the first slot representing a gap.

Byte 7: Number of the last slot taken into account.

7002 PREOPERATIONAL_INHIBITED

Is transmitted when a change to Pre-Operational cannot be made after starting up because the bus configuration presents a gap or contains modules other than those configured in the flash project.

Byte 3: Number of the first slot with errors.

Byte 4: Number of the first slot that is not in agreement with the configuration in the flash.

Byte 5: Number of the first slot with a false module type.

Byte 6: Number of the first slot representing a gap.

Byte 7: Number of the last slot taken into account.

7010 MODULE_INSERTED

A module has been plugged in. If hot-swap is allowed for the coupler in the Operational state and the module is plugged in behind the end, the coupler changes to the Pre-Operational state.

Byte 3: Number of the slot of the plugged module.

Byte 4: Number of the first slot that is not in agreement with the configuration in the flash.

Byte 5: Number of the first slot with a false module type.

Byte 6: Number of the first slot representing a gap.

Byte 7: Number of the last slot taken into account.



NOTE

It may be necessary following the correction of the module configuration to transmit an NMT reset in order to be able to reestablish the SDO list accordingly with the changed module configuration. Correct configuration is reported with the Emergency 0000.

7011 MODULE_REMOVED

A module has been removed. If the coupler is in the Operational state and hot-swap is not allowed, the coupler changes to the Pre-Operational state.

Byte 3 contains the numbers of the slot of the removed module.

Byte 4: Number of the first slot that is not in agreement with the configuration in the flash.

Byte 5: Number of the first slot with a false module type.

Byte 6: Number of the first slot representing a gap.

Byte 7: Number of the last slot taken into account.

7020 BUS_UNRECOVERABLE

The state of the bus configuration is no longer defined, for example, two gaps have resulted due to unplugging/plugging in or failure of a module, or another error has occurred that results in the bus configuration to appear as being no longer valid. An NMT reset of the coupler is always triggered following this message.

Byte 3 contains more detailed information about the cause:

24: A module no longer responds

25: Communication with a module has been disrupted

26: A new module couldn't be initialized

27: More than 64 modules were detected in the context of module detection

28: An unclassifiable response occurred in the context of module detection

29: Bus configuration disrupted

7030 FLASH_WRONG_MODULE

Is transmitted when the module configuration doesn't match the planned module configuration (in the flash).

Byte 3: Number of the detected defective slot

Byte 4: Number of the first slot that is not in agreement with the configuration in the flash.

Byte 5: Number of the first slot with a false module type.

Byte 6: Number of the first slot representing a gap.

Byte 7: Number of the last slot taken into account.

8130 CAN_HB_CONS / CAN_NG_EXP

Node guarding lifetime has expired or one of the two consumer heartbeat times has been exceeded.

Byte 3: 0 = Node guarding error

1 = Consumer heartbeat 1 error

2 = Consumer heartbeat 2 error

8210 CAN_WRONG_RPDO_LEN

Reception of an RPDO with wrong length. Byte 3 contains the RPDO numbers, byte 4 contains the expected length.

8211 CAN_WRONG_TPDO_LEN

Reception of an TPDO RTR polling with wrong length. Byte 3 contains the TPDO numbers, byte 4 contains the expected length.

8250 CAN_RPDO_TIMEOUT

The maximum time between two RPDOs has been exceeded, provided a maximum time has been defined for this RPDO. Manufacturer bytes 1+2 contain the COB-ID of the RPDO.

FFxx MODULE_DIAG_OCCUR

A diagnosis of a module has occurred.

Emergency error code FFxx xx = Slot number	Error register (SDO 1001)	0 = module- wide /	Diagnostics code	Number of diagnostic messages	First module with diagnostic	First module with active diagnosis
--	------------------------------	--------------------------	---------------------	-------------------------------------	------------------------------------	--

FExx MODULE_DIAG_END

A diagnosis of a module has finished.

Emergency error code FExx xx = Slot number	Error register (SDO 1001)	0 = module- wide /	0	Number of diagnostic messages	First module with diagnostic	First module with active diagnosis
--	------------------------------	--------------------------	---	-------------------------------------	------------------------------------	--



ATTENTION

The content of the diagnosis emergency messages has changed with the coupler firmware V1.08. If you use an older firmware version, please read edition 4 of the manual.

10. Configuration with the TB20 ToolBox

The TB20 ToolBox is free software for the configuration, testing and diagnostics of the TB20 system. The TB20 ToolBox is not absolutely necessary to configure and operate the CANopen coupler, but it can be easily used as a free alternative to a CANopen Master or other external CAN tools.

A USB cable is required for communication with the TB20 CANopen coupler.

10.1. Installation of the TB20 ToolBox

The software can be downloaded at no charge at the website www.helmholz.de in the download area of the TB20 system. The software currently runs with Windows 7 and 10.

Carry out the setup and follow the instructions of the program.

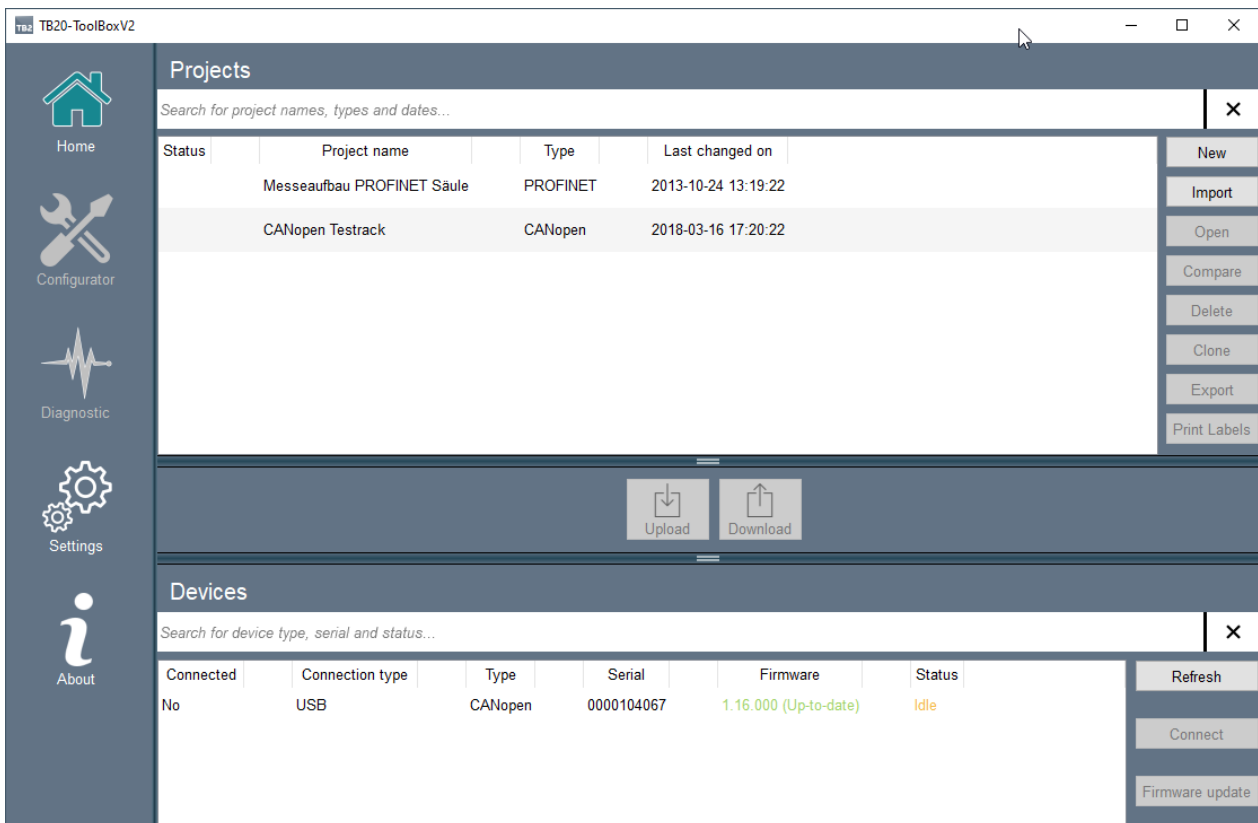
The USB driver is copied onto the hard drive together with the TB20 ToolBox setup and is available for installation.

You can find the USB driver in the TB20 ToolBox under “Settings”.



10.2. Start and create a new project (“Welcome”)

In the TB20 ToolBox “Welcome” view, you can create a new project, import a project or open an existing project. Select the “New” button here.



The screenshot shows the TB20-ToolBoxV2 software interface. On the left is a navigation sidebar with icons for Home, Configurator, Diagnostic, Settings, and About. The main area is divided into two sections: 'Projects' and 'Devices'.

Projects Section:

Status	Project name	Type	Last changed on
	Messeaufbau PROFINET Säule	PROFINET	2013-10-24 13:19:22
	CANopen Testrack	CANopen	2018-03-16 17:20:22

Buttons on the right of the Projects table: New, Import, Open, Compare, Delete, Clone, Export, Print Labels.

Buttons below the Projects table: Upload, Download.

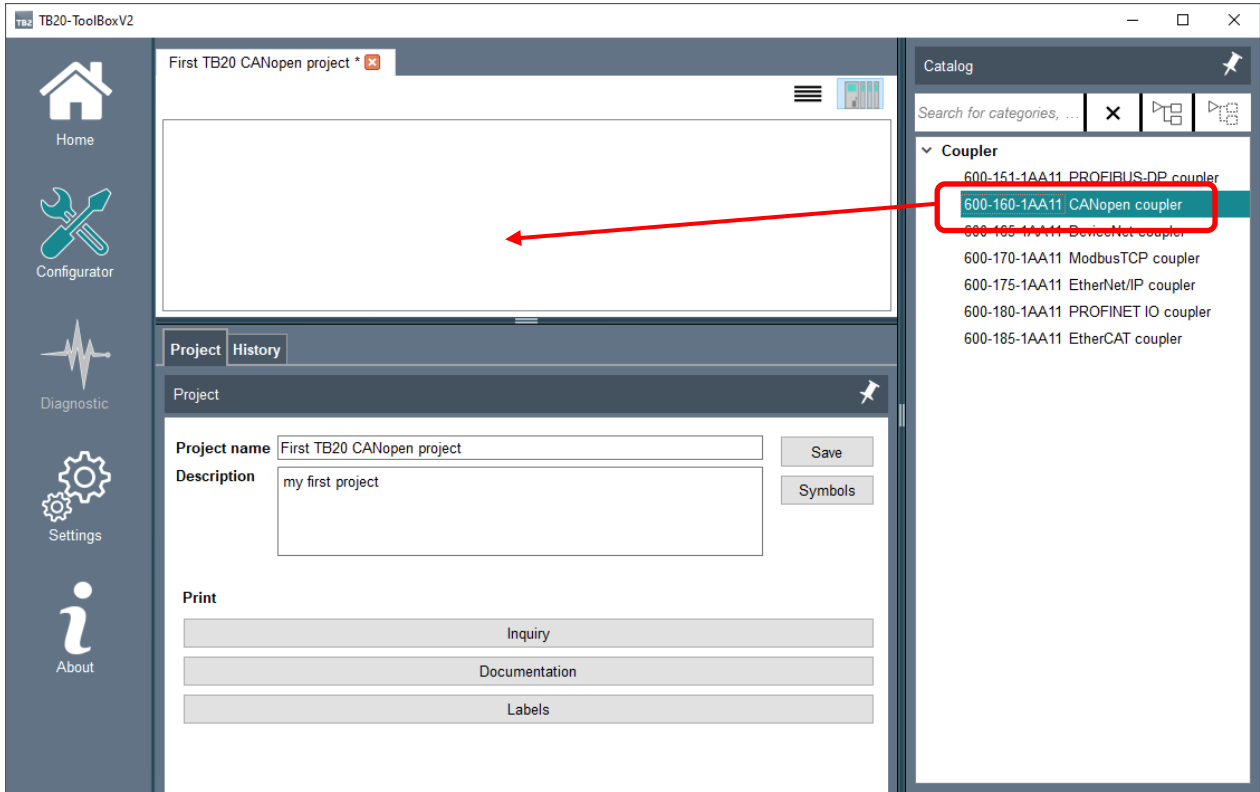
Devices Section:

Connected	Connection type	Type	Serial	Firmware	Status
No	USB	CANopen	0000104067	1.16.000 (Up-to-date)	Idle

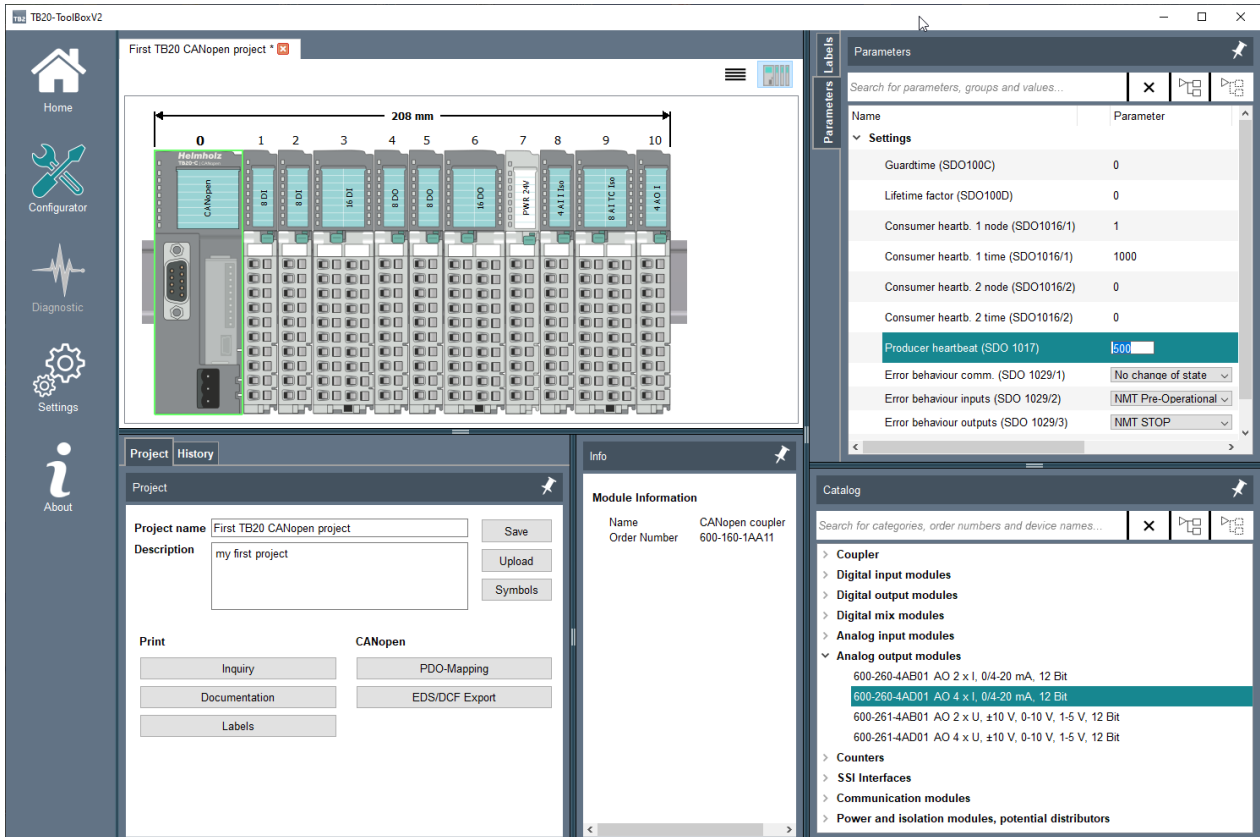
Buttons on the right of the Devices table: Refresh, Connect, Firmware update.

10.3. Configurator

Following the entry of the project name and the pasting of the CANopen coupler into the top left window, the “Configurator” view opens.

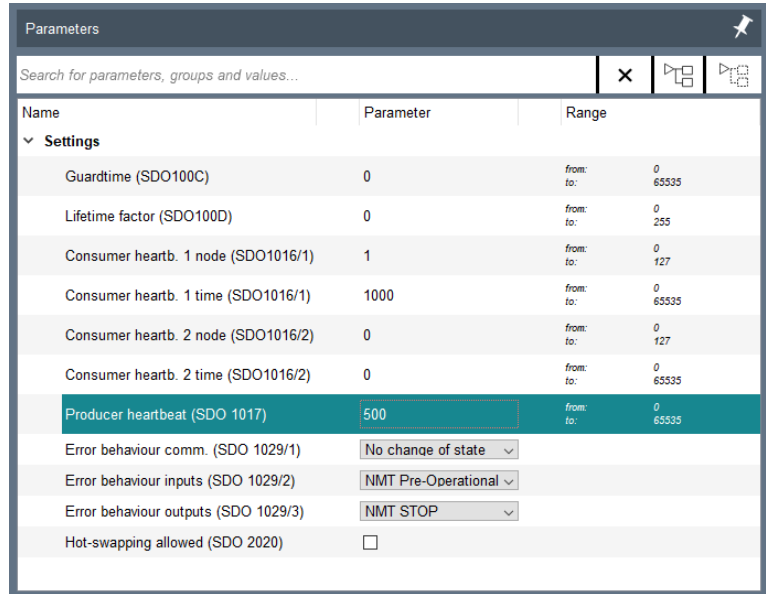


In this view, you can add new modules to the TB20 system and set module and coupler parameters.



10.3.1. Set coupler parameters

When the coupler is selected in the configurator (green frame), fundamental parameters of the CANopen slave behavior can then be set in the dialog below in the “Settings” tab.

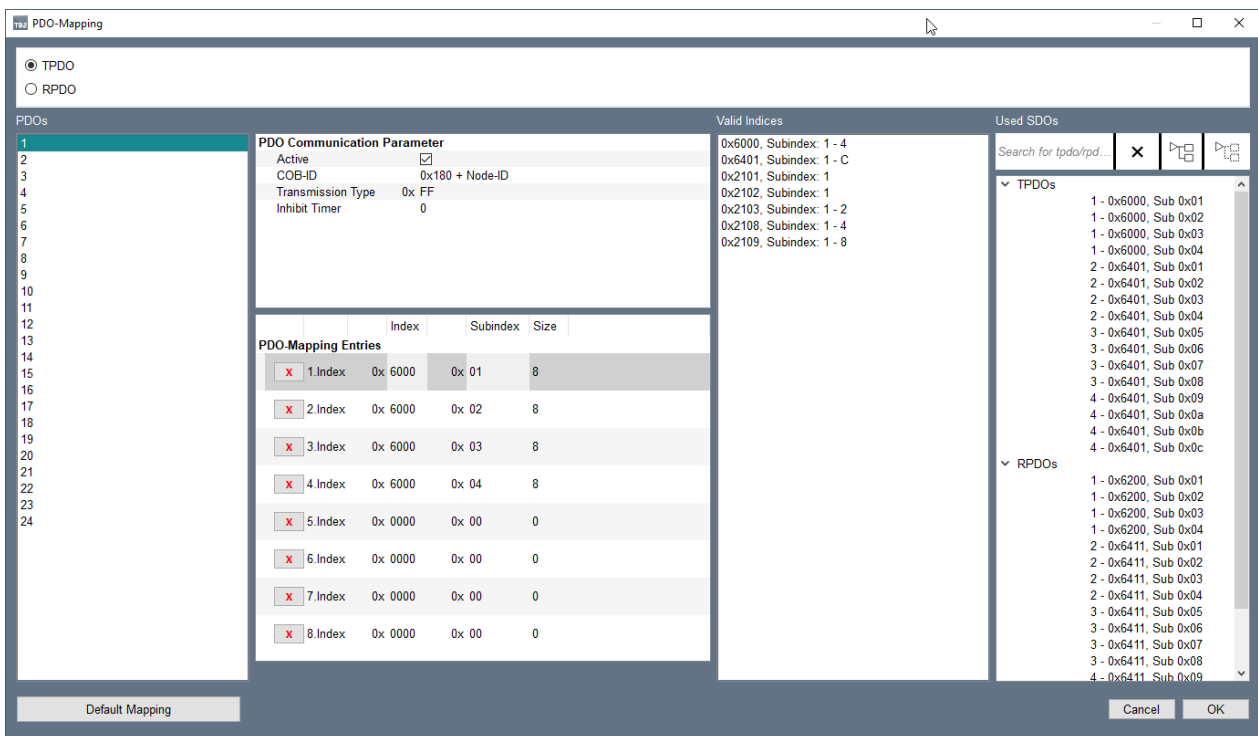


10.3.2. Set PDO mapping

The PDO mapping can be set when all modules have been positioned and configured in the project (optional). If no mapping is created in the project, the CANopen coupler works with default mapping, as described in Ch. 7.9.2. The CANopen coupler is thus functional when only the PDOs 1-4 are used.

When the coupler has been selected in the configurator, the PDO mapping dialog can then be displayed in the dialog below in the “Info” tab by clicking on the “PDO mapping” button of the PDO mapping dialog. The TPDOs or RPDOs 1 - 24 can be activated, deactivated, and set with their communication parameters in this dialog.

When the dialog is called up for the first time, the default mapping is created as it would also be created by the CANopen coupler (see Ch. 7.9.2). The mapping is saved with a click on “OK”. The last changes are rejected with “Cancel”.



If no PDO mapping existed prior to pressing “Cancel”, no mapping will be created. The CANopen coupler then works with the default PDO mapping (see Ch. 7.9.2).

The COB IDs of the PDOs are generated in accordance with the automatism in the Ch. 7.9.4.



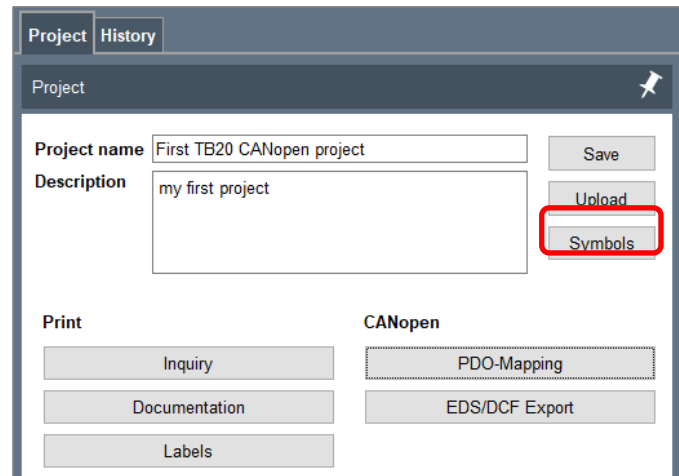
ATTENTION

If more than 4 TPDOs or 4 RPDOs are used, the COB IDs of the following nodes may be used by the automatically generated PDOs. (see Ch. 7.9.4).

10.3.3. Load project into the coupler (“Upload”)

If the project has been completely configured, it can be uploaded in the “Configurator” view to the CANopen coupler, “Upload” button.

The part of the project required for CANopen operation is transmitted via the USB connection of the coupler and filed in the flash. This means that a previously existing project is deleted.



ATTENTION

Please note that the coupler logs out of the bus during the upload procedure.



NOTE

Only the part of the project necessary for CANopen operation is downloaded into the coupler. Supplementary information, such as comments and text labels are not transmitted. You should always export the original project from the TB20 ToolBox as a file and store it safely as a backup.

10.3.4. EDS/DCF export

When the coupler has been selected in the configurator, the “EDS/DCF export” function can then be called up in the dialog in the “Info” tab.

The EDS file (Electronic Data Sheet) contains all SDOs of the configured CANopen coupler and can be loaded into the engineering tool of the CANopen master device.

The DCF file is a concrete configuration of a CANopen slave with node ID and bit rate and otherwise contains, like the EDS file, all SDOs.

The screenshot shows a software interface with two tabs: "Project" and "History". The "Project" tab is active. It contains a "Project name" field with the text "First TB20 CANopen project" and a "Save" button. Below it is a "Description" field with the text "my first project" and buttons for "Upload" and "Symbols". At the bottom, there are two sections: "Print" with buttons for "Inquiry", "Documentation", and "Labels"; and "CANopen" with buttons for "PDO-Mapping" and "EDS/DCF Export". The "EDS/DCF Export" button is highlighted with a red rectangular border.

The screenshot shows a dialog box titled "EDS/DCF Export". It has two main sections: "EDS" and "DCF". The "EDS" section contains a text box with the following text: "The EDS/DCF-Exporter creates a file, which is used to Configure a CAN-Slave over a CAN-Master. The EDS-File is a overall basis for this Configuration. The DCF has all parameter values saved in itself and configures the CAN-Slave even more precise." Below this text is an "Export EDS" button. The "DCF" section contains a "Node-ID:" label followed by a text box containing the number "1", and a dropdown menu showing "500 kbit/s". Below these is an "Export DCF" button. At the bottom right of the dialog is a "Close" button.

10.4. Online diagnostics

In the TB20-ToolBox “Welcome” view, you can establish a connection with a TB20 coupler under “Devices”.

The screenshot shows the TB20-ToolBox V2 interface. The 'Projects' section contains a table with the following data:

Status	Project name	Type	Last changed on
	Messeaufbau PROFINET Säule	PROFINET	2013-10-24 13:19:22
	CANopen Testrack	CANopen	2018-03-16 17:20:22
Changed	First TB20 CANopen project	CANopen	2020-03-06 12:43:20

The 'Devices' section contains a table with the following data:

Connected	Connection type	Type	Serial	Firmware	Status
No	USB	CANopen	0000104067	1.16.000 (Up-to-date)	Idle

A red box highlights the 'Refresh' and 'Connect' buttons in the 'Devices' table.

The screenshot shows the TB20-ToolBox V2 interface in the 'Online Diagnostics' view. The main area displays a rack of 15 modules, with the first module highlighted as 'CANopen coupler: 104067 @ USB'. The 'Parameters' panel on the right shows the following settings:

Name	Active Parameter
Guardtime (SDO100C)	0
Lifetime factor (SDO100D)	0
Consumer heartb. 1 node (SDO1016/1)	0
Consumer heartb. 1 time (SDO1016/1)	0
Consumer heartb. 2 node (SDO1016/2)	0
Consumer heartb. 2 time (SDO1016/2)	0
Producer heartbeat (SDO 1017)	0
Error behaviour comm. (SDO 1029/1)	NMT Pre-Operational
Error behaviour inputs (SDO 1029/2)	NMT Pre-Operational
Error behaviour outputs (SDO 1029/3)	NMT Pre-Operational

The 'Info' panel on the right shows the following module information:

Module Information	Value
Name	CANopen coupler
Order Number	600-160-1AA11
Status	Idle
App State	None
Diagnostic	None
Hardware	
Serial	104067
HW revision	HW1-1
Software	
FW-Version	1.16.000
CI-Version	1.16.000
Coupler Parameters	
Node-ID	3

In the Diagnosis view, click on the coupler or one of the modules (green frame) and receive information concerning the status (“Info” window), the currently set parameters (“Parameters” window) and see the current content of the PDOs (“Inputs/Outputs“ window).

In the “Project” window, you can start (“Operational”) and stop (“Pre-operational”) the TB20 CANopen coupler, carry out a restart and reset the coupler to factory settings.

10.4.1. Load project from the coupler (“Download”)

If you have not created a project from an existing TB20 system, or if the project file has been lost, you can load the current status of the coupler with all modules and module parameters into the TB20 ToolBox.

Use the “Download” function in the “Diagnosis” view when you are connected with a coupler via USB. The function loads all relevant information from the coupler and creates a new project file.

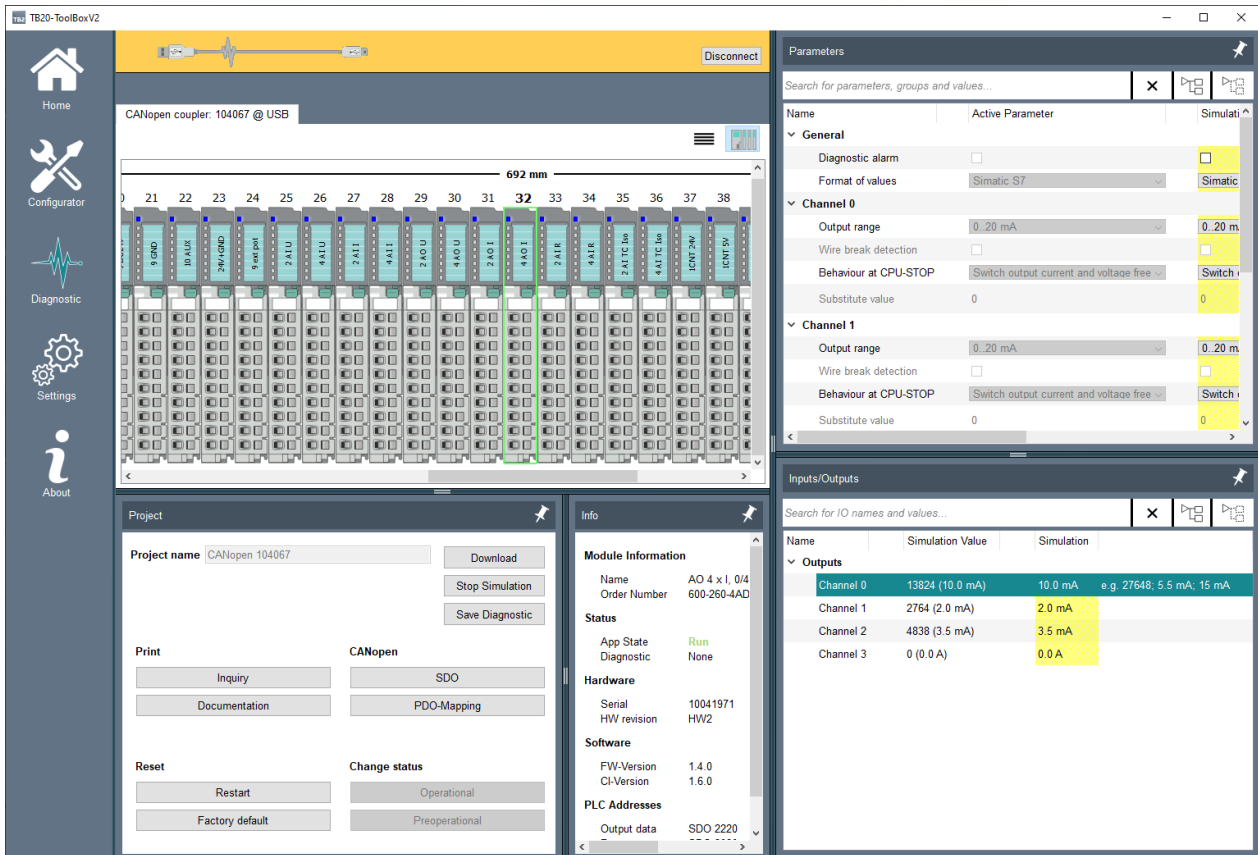


ATTENTION

If PDO COB IDs have been falsely adjusted in the CANopen coupler by SDO accesses, so that they no longer follow the automatically generated PDO COB-IDs, these PDO COB-IDs are replaced by the automatically generated PDO COB IDs following the loading of the project from the coupler.

10.4.2. Simulation

In the “Diagnosis” view, you can switch the CANopen coupler to a simulation mode (“Start simulation”). You thus assume control over the behavior of the CANopen coupler in the current status and can display the current input data, activate the outputs and change the configuration of modules.



The coupler displays the simulation mode with a quickly flashing, green “PLC” LED.



ATTENTION

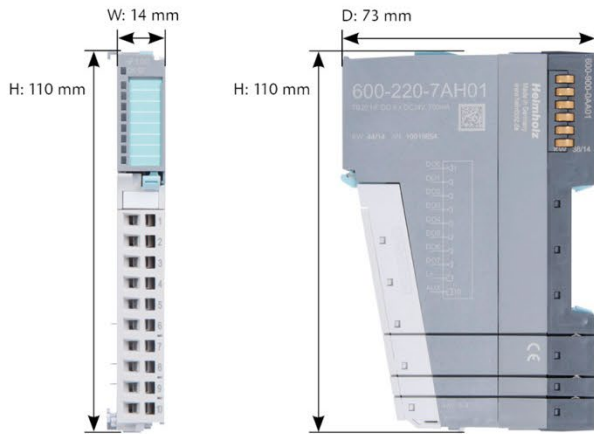
The CANopen coupler is completely separated from the CAN bus in the simulation mode. When the simulation mode ends, the CANopen coupler restarts and logs back onto the CAN bus. When leaving simulation operation, all of the changes carried out in simulation mode are discarded.

11. Technical data

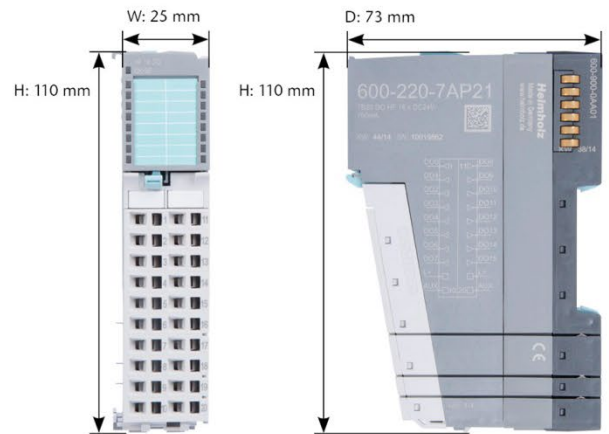
Order no.	600-160-1AA11
Module type	CANopen Coupler
Electrically isolated from backplane bus	No
Electrically isolated from USB	Yes
CAN interface	
Protocol	CANopen as per DSP-301 V4.2 and DS-401 V3.0.0
Bit rate	50, 100, 125, 250, 500, 800, 1000 kbps
TPDOs	24
RPDOs	24
Interface	ISO 11898-2
Connection	9-pin D-sub male connector
USB interface	
Protocol	Full-speed USB 2.0 device
Connection	Mini-USB
Isolation voltage	1.5 kV
Voltage supply	24 V DC, 18–28 V DC
Current draw without modules (internal)	75 mA
Power dissipation	Max. 8 W
Power supply for modules	5 V DC, max. 2.5 A
Dimensions (H x W x D)	110 mm x 35 mm x 73 mm
Weight	115 g
Certifications	CE, UL
Noise immunity	DIN EN 61000-6-2 “EMC Immunity”
Interference emission	DIN EN 61000-6-4 “EMC Emission”
Vibration and shock resistance	DIN EN 60068-2-6:2008 “Vibration” DIN EN 60068-2-27:2010 “Shock”
Protection rating	IP 20
Relative humidity	95% without condensation
Installation position	Any
Permissible ambient temperature	0 °C to 60 °C For UL applications: 0 °C to 50 °C
Transport and storage temperature	-20 °C to 80 °C
Pollution degree	2

12. TB20 System Dimensions

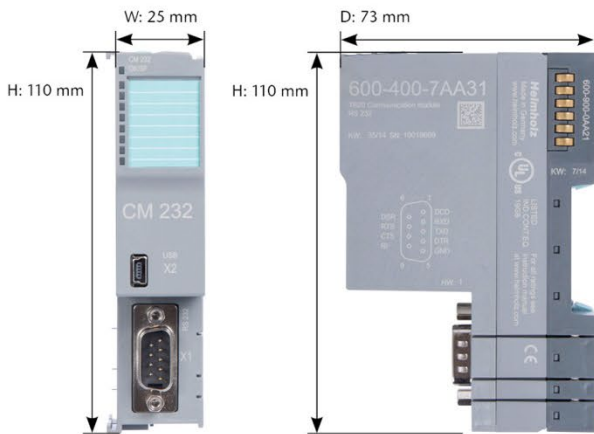
Module with standard width



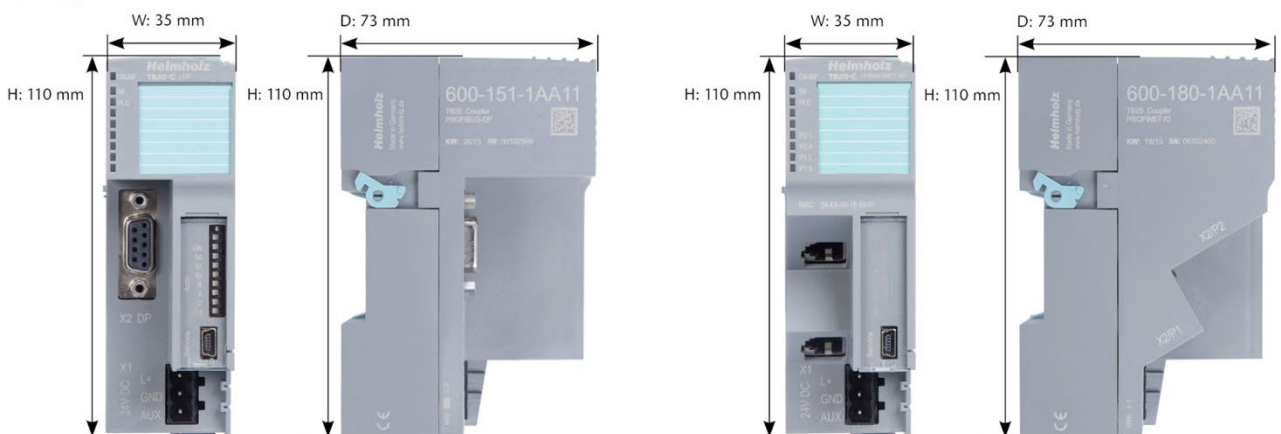
Module with double width



Communication Module



Bus Coupler



13. Spare parts

13.1. Base modules

13.1.1. 14 mm width standard base module

The 14 mm standard base module is available in sets of five with order no. 600-900-9AA01.



13.1.2. 25 mm width base module

The 25 mm standard base module is available in sets of five with order no. 600-900-9AA21.



13.1.3. Power and isolation base module

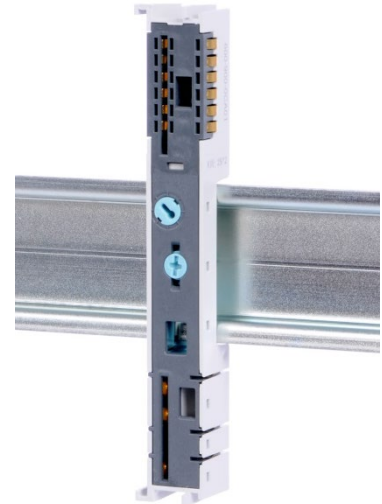
The power and isolation base module is available in sets of five with order no. 600-900-9BA01.



13.1.4. Power base module

The power base module is available in sets of five with order no. 600-900-9CA01.

It can be used with the power module (600-700-0AA01) and with all bus couplers.



13.2. Front connectors

13.2.1. 10-terminal front connector

The 10-terminal front connector is available in sets of five with order no. 600-910-9AJ01.



13.2.2. 20-terminal front connector

The 20-pin front connector is available in sets of five with order no. 600-910-9AT21.



13.3. Electronic modules

Electronic modules can be ordered as spare parts with the order number of the original product. Electronic modules are always sent as a complete assembly, including the corresponding base module and front connector.

13.4. Final cover

The final cover is available in sets of five with order no. 600-920-9AA01.

