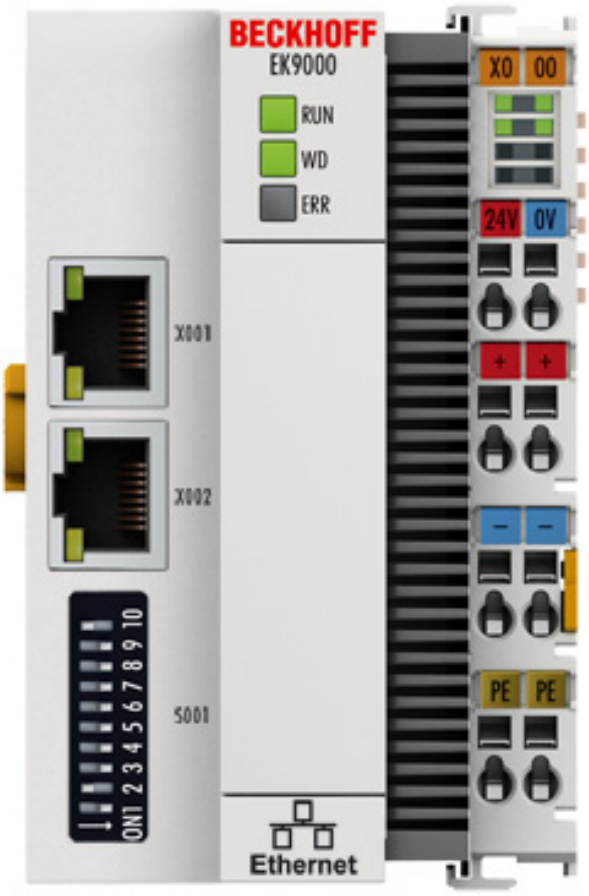


Documentation

EK9000

Modbus TCP/UDP Bus Coupler for EtherCAT Terminals

Version: 2.2.0
Date: 2019-06-03

BECKHOFF

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
2.2.0	• Update chapter “UL notice”
2.1.0	• Update chapter ”Modbus interfaces” • Update chapter ”Technical data”
2.0.2	• Update chapter ”Update Bus Coupler image”
2.0.1	• Update chapter ”Technical data”
2.0.0	• Migration
1.1.0	• HTML configuration description updated
1.0.0	• First published

Image Version EK9000

Firmware	Hardware version	Description
1 (V0.03)	3.9	• First version

2 Product overview

2.1 EKxxxx - System overview

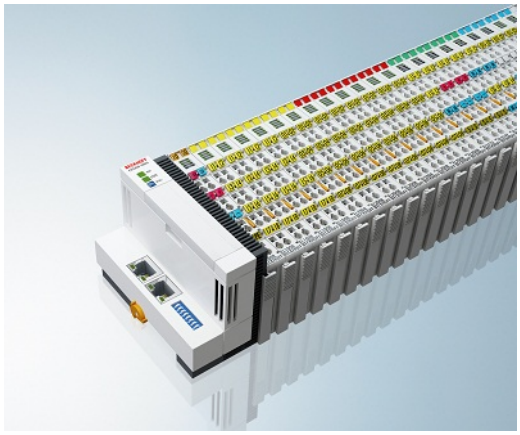


Fig. 1: EtherCAT Terminals at an EKxxxx series Bus Coupler

The Bus Couplers from the EKxxxx series allow EtherCAT Terminals to be operated on conventional fieldbus systems. The ultra-fast, high-performance EtherCAT Terminals with their large range of signal types are thus also available for other fieldbus and Industrial Ethernet systems.

The EKxxxx Bus Couplers are fieldbus slaves and contain an EtherCAT master for the EtherCAT terminals. They convert the telegrams from the higher-level fieldbus systems into the E-bus signal representation. A station consists of an EKxxxx and a number of EtherCAT Terminals.

The EKxxxx is integrated in exactly the same way as the Bus Couplers from the BKxxxx series via the corresponding fieldbus system configuration tools and the associated configuration files, such as GSD, ESD or GSDML.

EtherCAT makes a very flexible topology configuration possible. Thanks to the Ethernet physics, long distances can also be bridged without the bus speed being affected. When changing to the field level – without a control cabinet – the EtherCAT Box modules (EPxxxx) in protection class IP65 can also be connected to the EK9xxx.

Bus Couplers for various fieldbus systems

The variants from the EKxxxx series differ from one another by the interface for the higher-level fieldbus system.
An overview of the various Beckhoff Bus Couplers covering the most important fieldbus systems can be found on the Beckhoff Website.

Embedded PCs with fieldbus interface and decentralized control

The TwinCAT-programmable variant is the CX80xx Embedded PC series.

The variants from the CX80xx series differ from one another by the interface for the higher-level fieldbus system and the possibility to program it.
An overview of the various Beckhoff Embedded PCs covering the most important fieldbus systems can be found on the Beckhoff Website.

2.2 EK9000 - Introduction

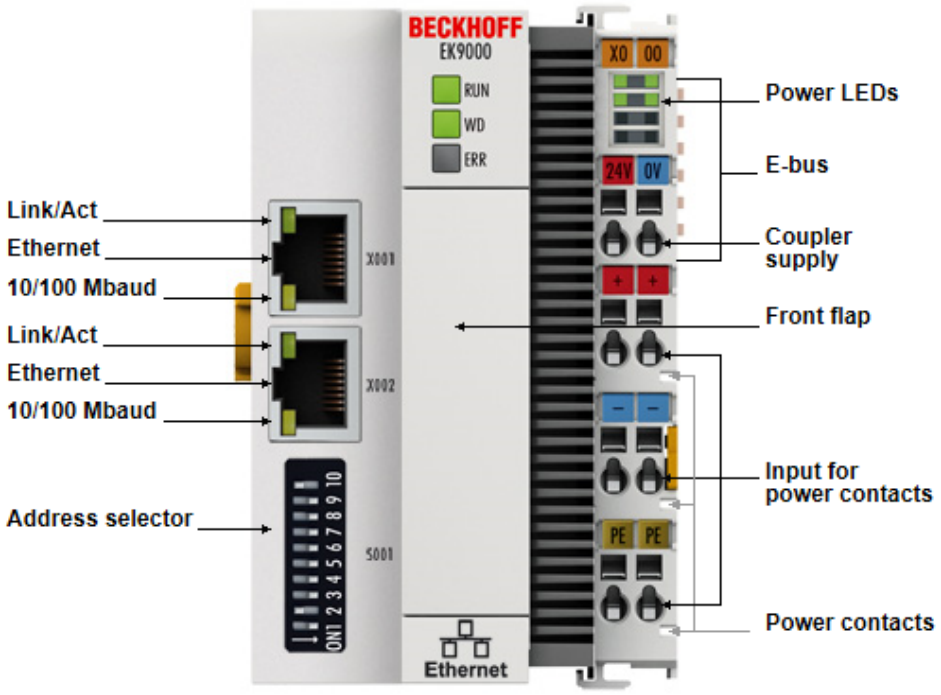


Fig. 2: EK9000

The EK9000 Bus Coupler connects Ethernet networks with the EtherCAT Terminals (ELxxxx) and EtherCAT Box modules (EPxxxx) and converts the telegrams from Ethernet to E-bus signal representation.

One station consists of an EK9000 and any number of EtherCAT Terminals. RJ45 is used for the Ethernet connection. In EtherCAT, the Ethernet coupler has at its disposal a lower-level, powerful and ultra-fast I/O system with a large selection of terminals. The coupler supports the Modbus-TCP protocol and therefore fits seamlessly into Ethernet networks.

Configuration

The EK9000 is configured based on HTML pages provided by the Bus Coupler or via the Modbus interface.

2.3 Technical data - EK9000

Technical data	EK9000
Protocol	ModbusTCP/UDP
Interfaces	2 x Ethernet 100 Mbit/s, 1 x USB device (behind the front flap)
Bus interface	2 x RJ 45 (switched)
I/O connection	E-Bus (EtherCAT terminals)
Web-based Management	yes
I/O terminals	E-bus (EL, ES, EP), standard digital signals, standard analog signals No gateway EC terminals, no EC terminals with XFC or DC function, no general EtherCAT devices
Number of EC terminals	max. 255
Max. size of process data	max. 1440 bytes In- and Output data
Supply voltage	24 V DC (-15 %/+20 %)
Power supply I/O terminals	2 A
Max. power loss	3 W
Dimensions (W x H x L)	65 mm x 100 mm x 80 mm
Approvals	CE, UL, ATEX

System data	Modbus (EK9000)
Number of I/O modules	depending on controller
Number of I/O points	depending on controller
Transmission medium	4 x 2 twisted pair copper cable category 5 (100 Mbaud)
Cable length	100 m
Data transfer rate	100 MBaud
Topology	Star-form cabling, line topology

2.4 Technical data - Modbus

Technical data Ethernet	EK9000
Number of ports	2
integrated switch	2 x Ethernet 100 Mbit/s, 1 x USB device (behind the front flap)
Bus interface	2 x RJ 45 (switched)
100 Mbit/s	Yes, full duplex
Autocrossing	Yes
Protocol	
ModbusTCP	Yes
ModbusUDP	Yes
Functions/service	
Read Coils 1	Yes
Read Discrete Inputs 2	Yes
Read Holding Register 3	Yes
Read Input Register 4	Yes
Write Single Coil 5	Yes
Diagnostics 8	Yes
Write Single Register 6	Yes
Write Multiple Coils 15	Yes
Write Multiple Register 16	Yes
Read/Write Multiple Register 23	Yes
TCP/IP ADS	Yes
ARP	Yes
Web services	Yes
DHCP	Yes
Diagnosis/Status/Alarm	
RUN LED	Yes, green/red
WD LED	Yes, green/red
ERR LED	Yes, green/red
A station consists of an EKxxxx and a number of EtherCAT Terminals.	Yes
Diagnostic messages	Yes

3 Mounting and wiring

3.1 Mounting

3.1.1 Dimensions

The following illustrations show the dimensions of the Bus Couplers.

Drawings in DWF and STEP format can be found in the Download section of the Beckhoff website.

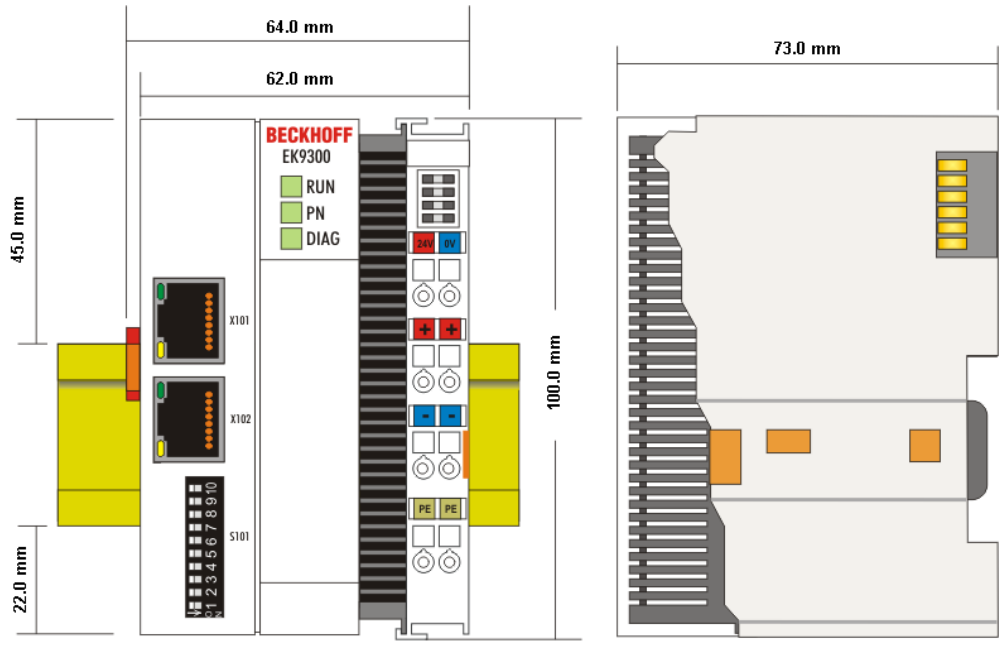


Fig. 3: EK9xxx – dimensions taking the EK9300 as an example

3.1.2 Installation on mounting rails – Bus Coupler

Snapping onto the mounting rail

The Bus Coupler can simply be snapped onto the mounting rail. To this end position the block on the mounting rail and push it slightly until it engages on the right-hand side. This is indicated by a distinct click. Use a screwdriver to push up the lock on the left-hand side, thereby turning it and causing it to engage audibly.

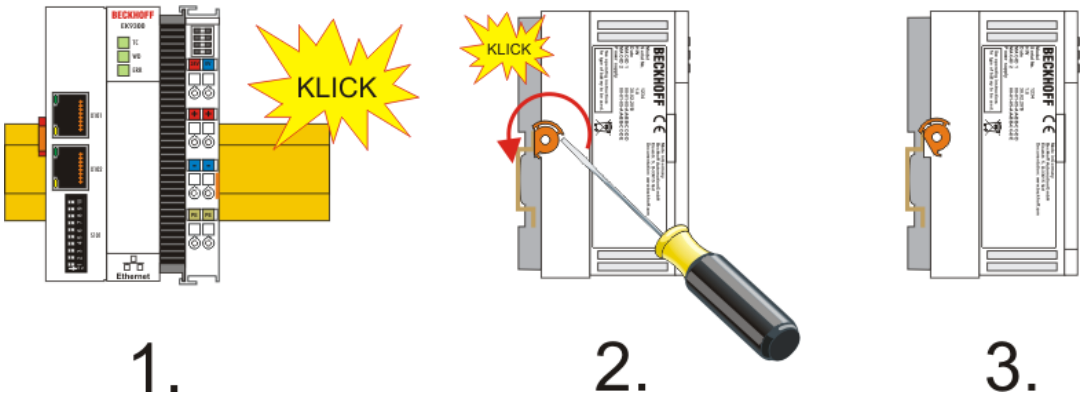


Fig. 4: EK9300 - Snapping onto the mounting rail

NOTE

Avoid damage!

Do not force the module or apply excessive pressure!

Installation positions

The installation position of the Bus Coupler is arbitrary.

NOTE

Installation position of EtherCAT terminals

Observe the installation position of the EtherCAT terminals used – not all of them have an arbitrary installation position. Pay attention to the respective EtherCAT infrastructure components and installation instructions.

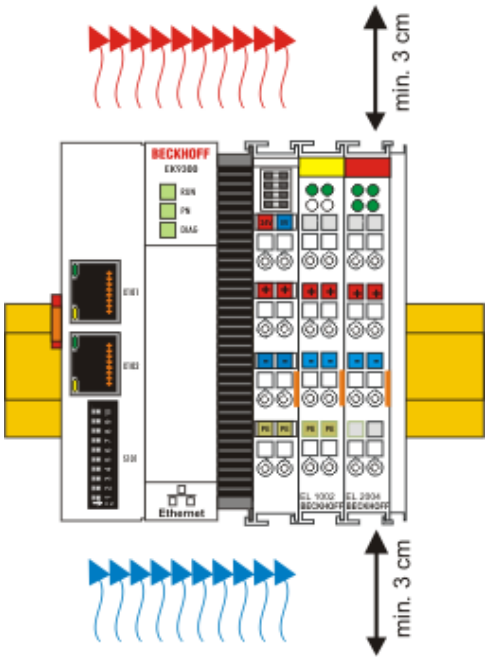


Fig. 5: Recommended distances for standard installation position

NOTE

Comply with the permitted installation position and minimum distances!

We recommend the installation in the horizontal position for optimum ventilation. Furthermore, it is not necessary with this installation position to check whether there are terminals present that may only be installed horizontally.

Other installation positions are allowed, but not recommended.

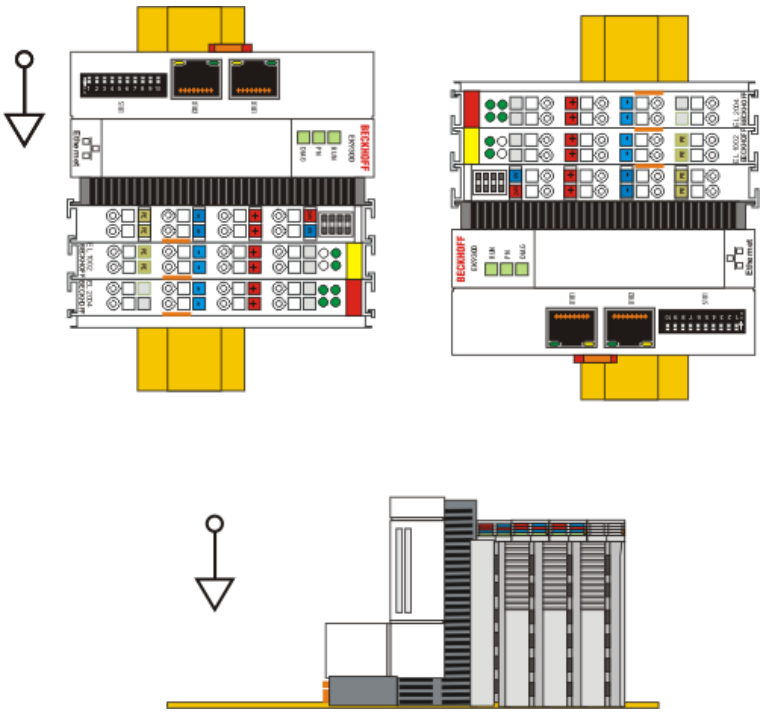


Fig. 6: Other installation positions

3.2 Wiring

3.2.1 Power supply

Supply of Bus Couplers and Bus Terminals (Us)

The Bus Coupler requires a supply voltage of 24 V_{DC} (-15 %/+20 %) for operation.

The connection is made via the two upper terminal points labelled **24 V** and **0 V**. This power input supplies the Bus Coupler electronics and, via the E-Bus, the electronics of the EtherCAT Terminals. It is electrically isolated from the peripheral supply (Up) of the power contacts.

In order to guarantee the operation of the Bus Coupler and the terminal segment in all cases, the power supply unit must supply 2 A at 24 V.

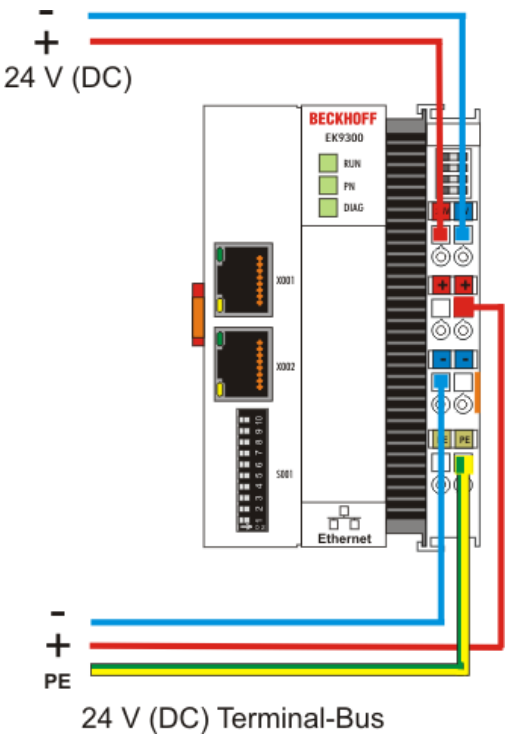


Fig. 7: Bus Coupler power supply

Power contacts supply (Up)

The three power contacts (spring contacts) are located on the right-hand lateral surface of the Bus Coupler. The spring contacts are hidden in slots so that they can not be accidentally touched. When a Bus Terminal is added, its blade contacts are connected to the spring contacts of the Bus Coupler. The tongue and groove guides on the top and bottom of the Bus Terminal Controllers and of the Bus Terminals guarantees that the power contacts mate securely.

The lower six terminal points labelled **+**, **-** and **PE** are for supplying power to the peripheral supply (Up). These terminal points are connected in pairs to a power contact. This power input allows voltages of up to 24 V and is electrically isolated from the Bus Coupler supply voltage (Us).

The connection of these terminal points in pairs enables the supply to be forwarded. The current load of the power contacts may not permanently exceed 10 A. The current carrying capacity between two terminal points is identical to the current carrying capacity of the connecting wires.

The spring-loaded terminal points are designed for wires with cross-sections from 0.08 mm² to 2.5 mm².

LED

If the power supply unit is connected correctly and the power supply is switched on, the two upper LEDs in the terminal prism are green. The left-hand LED (Us) indicates the supply for the Bus Coupler electronics. The right-hand LED (Up) indicates the supply for the power contacts. The other LEDs indicate the state of the E-bus. The detailed meanings of the LED displays are described in the chapter *LED displays*.

PE power contacts

NOTE

Power contact PE

The PE power contact must not be used for other potentials.

3.2.2 Ethernet

3.2.2.1 Ethernet connections

Fig. 8: RJ45 interface

Assignment of the RJ45 interface, port 1

X001

PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserved
5		
6	RD -	Receive -
7	connected	reserved
8		

Assignment of the RJ45 interface, port 2 (switched)

CX8010, CX809x: X101/102
EK9xxx: X001 / X002

PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserved
5		
6	RD -	Receive -
7	connected	reserved
8		

3.2.2.2 Ethernet cable

Transmission standards

10Base5

The transmission medium for 10Base5 consists of a thick coaxial cable ("yellow cable") with a max. transmission speed of 10 Mbaud arranged in a line topology with branches (drops) each of which is connected to one network device. Because all the devices are in this case connected to a common transmission medium, it is inevitable that collisions occur often in 10Base5.

10Base2

10Base2 (Cheaper net) is a further development of 10Base5, and has the advantage that the coaxial cable is cheaper and, being more flexible, is easier to lay. It is possible for several devices to be connected to one 10Base2 cable. It is frequent for branches from a 10Base5 backbone to be implemented in 10Base2.

10BaseT

Describes a twisted pair cable for 10 Mbaud. The network here is constructed as a star. It is no longer the case that every device is attached to the same medium. This means that a broken cable no longer results in failure of the entire network. The use of switches as star couplers enables collisions to be reduced. Using full-duplex connections they can even be entirely avoided.

100BaseT

Twisted pair cable for 100 Mbaud. It is necessary to use a higher cable quality and to employ appropriate hubs or switches in order to achieve the higher data rate.

10BaseF

The 10BaseF standard describes several optical fiber versions.

Short description of the 10BaseT and 100BaseT cable types

Twisted-pair copper cable for star topologies, where the distance between two devices may not exceed 100 meters.

UTP

Unshielded twisted pair
This type of cable belongs to category 3, and is not recommended for use in an industrial environment.

S/UTP

Screened/unshielded twisted pair (screened with copper braid)
Has an overall shield of copper braid to reduce influence of external interference. This cable is recommended for use with Bus Couplers.

FTP

Foiled shielded twisted pair (screened with aluminum foil)
This cable has an overall shield of laminated aluminum and plastic foil.

S/FTP

Screened/foiled-shielded twisted pair (screened with copper braid and aluminum foil)
Has a laminated aluminum screen with a copper braid on top. Such cables can provide up to 70 dB reduction in interference power.

STP

Shielded twisted pair
Describes a cable with an outer screen, without defining the nature of the screen any more closely.

S/STP

Screened/shielded twisted pair (wires are individually screened)
This identification refers to a cable with a shield for each of the two wires as well as an overall shield.

ITP

Industrial Twisted-Pair
The structure is similar to that of S/STP, but, in contrast to S/STP, it has only one pair of conductors.

3.2.2.3 Ethernet topology

EK9000

The construction of the EK9000 can take place in a line, with adherence to the following points:

- Maximum 20 Bus Couplers one behind the other
- No switches should be used in the line

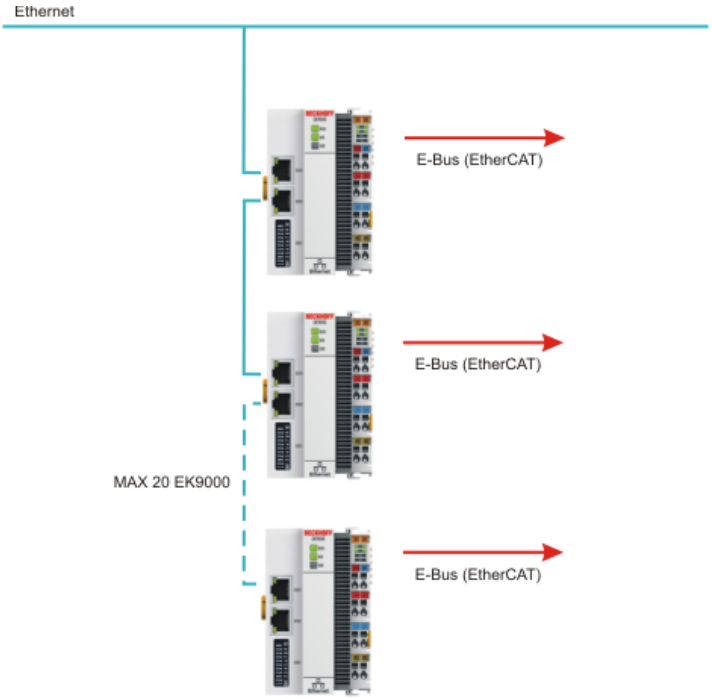


Fig. 9: Ethernet topology

3.3 UL notice

c UL us	**Application** Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.
c UL us	**Examination** For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).
c UL us	**For devices with Ethernet connectors** Not for connection to telecommunication circuits.

Basic principles

UL certification according to UL508 with limited power consumption. The current consumed by the device is limited to a max. possible current consumption of 4 A. Devices with this kind of certification are marked by this sign:

Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V_{DC} must be limited accordingly by means of supply

- from an isolated source protected by a fuse of max. 4 A (according to UL248) or
- from a voltage supply complying with *NEC class 2*.
  A voltage source complying with *NEC class 2* may not be connected in series or parallel with another *NEC class 2*compliant voltage supply!

These requirements apply to the supply of all EtherCAT bus couplers, power adaptor terminals, Bus Terminals and their power contacts.

4 Parameterization and commissioning

4.1 DIP switch

Ten-pole DIP switch S001

The DIP switch has the following meaning for the Ethernet interfaces X001 and X002, which are switched:

Fig. 10: DIP switch S001, left off "0", right on "1"

DIP 9	DIP 10	Function of DIP 1 to DIP 8	Restart behaviour	Behaviour with factory settings
0	0	Last byte of the IP address via DIP switches 1 to 8	• IP address via DIP switch (byte 4) Bytes 1..3 from the setting (Web page)	• IP address via DIP switches 192.168.1.xxx (xxx DIP switch) SNM 255.255.255.0
0	1	DHCP DIP switch 1 to 8 set to OFF	• IP address via DHCP	• IP address via DHCP
1	0	Reserved	-	-
1	1	Reserved	-	-

Two-pole DIP switch

(under the flap between the battery and the SD card slot)

DIP switch (red)	Meaning
1 off and 2 off	normal mode, coupler is started
1 on and 2 off	The EK starts in Config Mode; the internal Flash memory can be accessed via the USB interface (for example for an image update).
1 off and 2 on	Factory setting
1 on and 2 on	No function so far

4.2 Further interfaces

Additional interfaces are located under the flap of the EK9xx0.

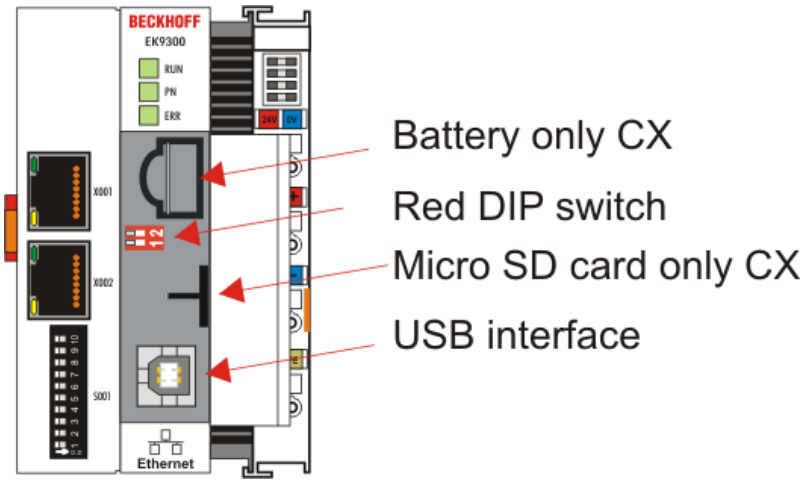


Fig. 11: Additional interfaces of the EK9xx0

Battery

No battery is required for the EK9xx0, therefore this option is not included.

Red DIP switch

Default setting is OFF/OFF.
In order, for example, to load new firmware to the EK via USB, the first DIP switch must be set to “1” before switching on. If the RUN LED lights up blue, the EK can be connected to the PC by a USB cable. The PC then finds the internal Flash as the storage medium. The storage medium may not be formatted!

Micro SD card

Alternatively the firmware can also be loaded to an SD card. Booting always takes place from the SD card if there is one in the slot. This can be used, for example, to test a firmware before copying it to the EK’s internal Flash.

USB interface

The USB interface can only be used if the “red” DIP switch has been set accordingly. See “Red DIP switch”.

4.3 IP address

The IP address or the mode (e.g. DHCP) can be set using the DIP switch [▶ 21]. Furthermore, an HTML page [▶ 23] is available for the configuration.

5 Configuration

5.1 Configuration via the HTML pages of the Bus Coupler

An HTML page is available for the configuration. This can be reached via the IP address/Config (e.g. 192.168.1.3/config). We recommend the use of Chrome or Firefox as browser.

If DHCP is used, enter the name of the Bus Coupler instead of the IP address. The default name of the Bus Coupler starts with the string "EK-", followed by the last 3 bytes of its MAC address (MAC ID). The MAC address can be found on the sticker on the left of the Bus Coupler.
Sample: The MAC address is 00-01-05-02-03-04. The resulting default name is "EK-020304". Now enter "EK-020304/Config" in your browser. The login name is "guest", the default password is "1" (without quotes).

The firmware and hardware versions as well as the serial number can be read on the information diagnostic page. The diagnosis history can be read if problems occur. The diagnosis history is not saved and is cleared in the case of a restart.

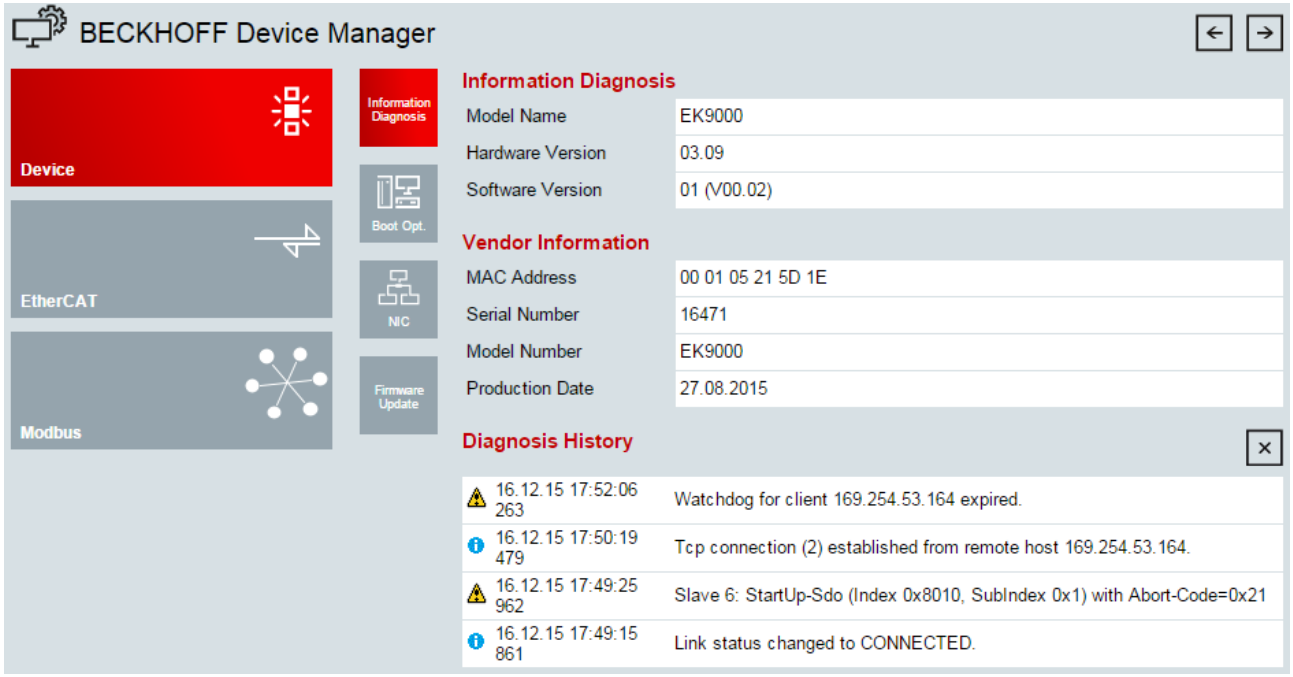


Fig. 12: Configuration via HTML pages - information diagnostic page

Network-Interface

The network interface enables you to set the IP address. Please note that the DIP switch of the EK9000 takes precedence and its setting applies regardless of what you set in the dialog.

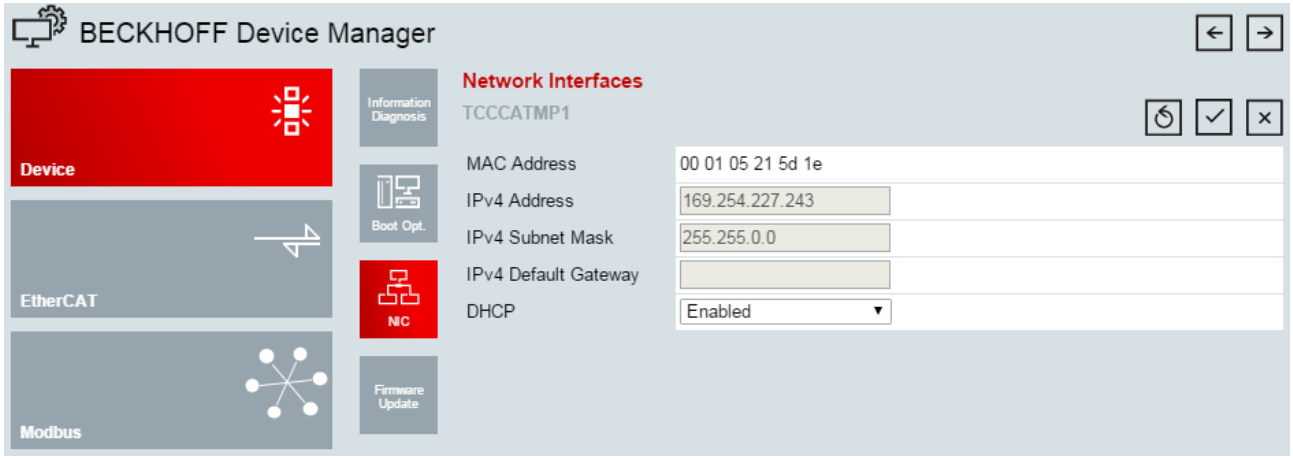


Fig. 13: Configuration via HTML pages - network interface

Sample

DIP switch DIP 1 = on; DIP 2..10 off, setting in the dialog 10.1.2.3 -> genuine IP address = 10.1.2.1
(the DIP switch overwrites the last byte of the IP address).

Set the desired IP address and then click on the checkmark.
Note: the old IP address is displayed again in the dialog field since it is still the valid address.

A software reboot is necessary after changing the IP address. To do this, go onto "Boot Opt." and click on "Reboot…".

With the DIP switch setting

- DIP 1 to DIP 8 = on and
- DIP 9 and DIP 10 = off

the complete 4 bytes of the IP address are accepted from the dialog field.

5.2 EtherCAT configuration

EtherCAT Terminals can be configured and parameterized via the HTML page *Beckhoff Device Manager*.

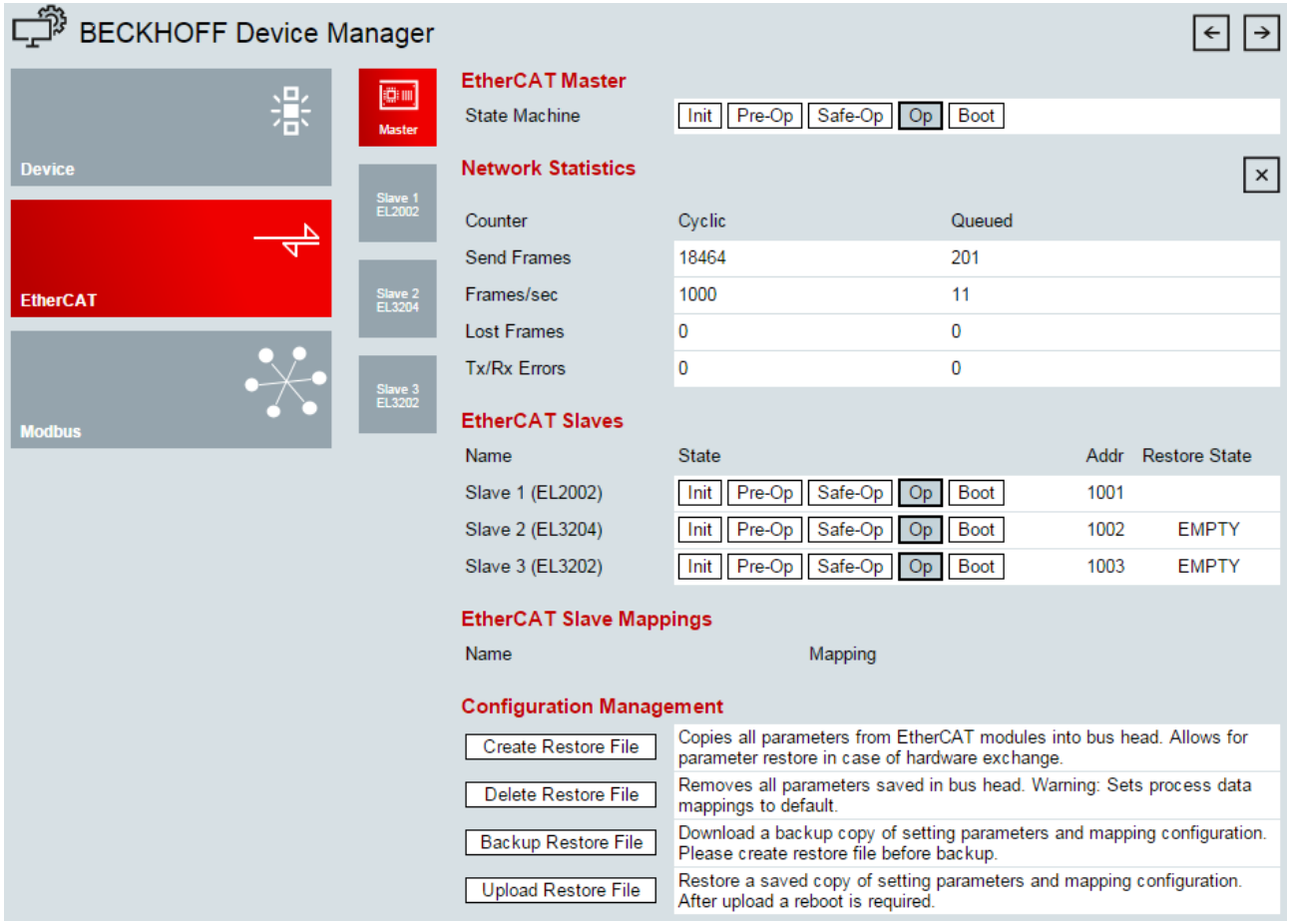


Fig. 14: Configuration via HTML pages - EtherCAT configuration

EtherCAT Master

The current state of the EtherCAT Master on the EK coupler is displayed here. It should usually be in the OP state.

Network Statistics

The EtherCAT statistics are output here.

EtherCAT Slaves

Display of the EtherCAT slaves and their states. The Restore State indicates whether a Restore File has been created for the terminals.

Restore File

The Restore File is required in order to be able to parameterize EtherCAT Terminals again. If EtherCAT Terminals are exchanged and have been parameterized, this information is usually lost when the EtherCAT Terminal is exchanged. The Restore File loads the parameters to the new terminal when the coupler is started. The Restore File has to be created if you want to change the default mapping of the terminals, or if you have to.

- EMPTY
  means there is no Restore File for the terminal
- VALID
  a valid Restore File has been created

- MAPPING
  The terminal mapping has been changed, but has not yet been stored in a Restore File.

EtherCAT Slaves Mappings

In some EtherCAT Terminals the process image can be changed; it must be stored in the EtherCAT master. The terminals that can be changed are displayed under "EtherCAT Slaves Mapping"; the corresponding mapping must be set and stored in the Restore file. The coupler is then restarted so that it can activate the mapping (attention: the process image is changed as a result)

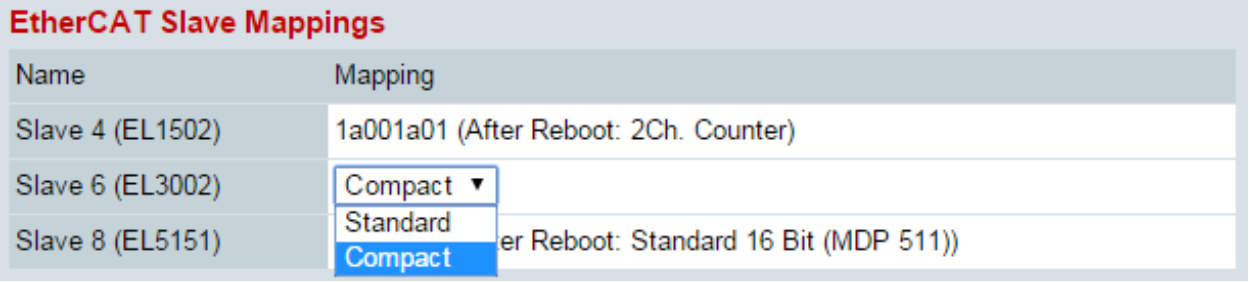


Fig. 15: Configuration via HTML pages - EtherCAT slave mappings

Parameterization of the EtherCAT Terminals

To parameterize an EtherCAT Terminal, select the required terminal. Its objects are then displayed and can be edited if necessary. The settings are then stored in the terminal. Note that any modifications are lost if the terminal is replaced. In this case use the restore file, which contains your modifications.

Restore file overwrites Modbus modifications

If the Restore File is used, the object parameters are always loaded into the terminal on starting the coupler. This will overwrite changes that you have made by Modbus Interface or web page.

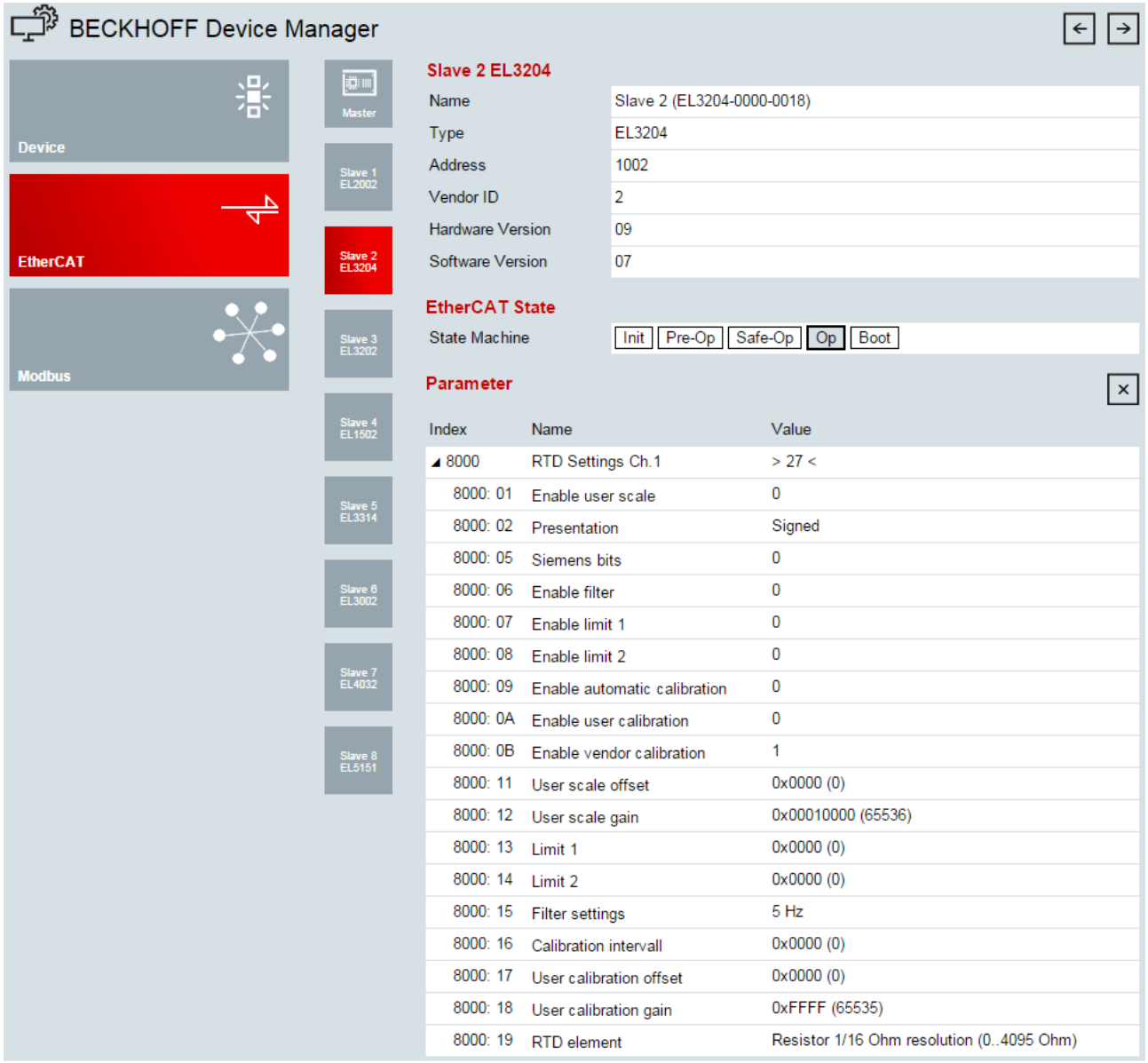


Fig. 16: Configuration via HTML pages - parameterizing the EtherCAT Terminals

5.3 Modbus configuration

Modbus configuration settings.

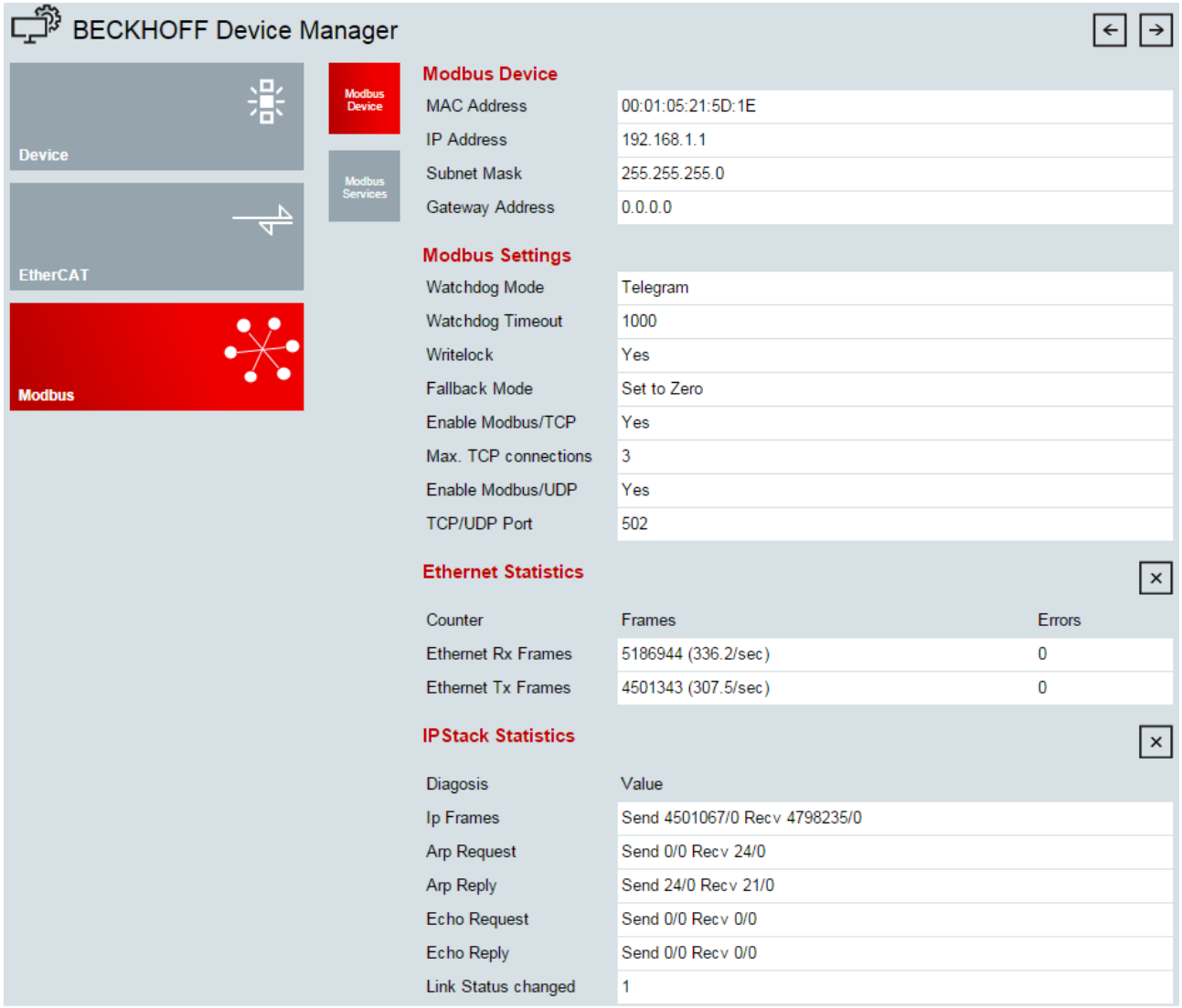


Fig. 17: Configuration via HTML pages - Modbus configuration

Modbus Device

All parameters are "read only" and are for the purposes of diagnosis only.

Modbus Settings

Settings can be made here for the Modbus interface.

Watchdog Mode (0x1122)

You can choose between Disabled, Telegram and Write here. The watchdog is activated with the first write telegram to the process data; write access is given to the master that was the first to execute a write access to the process data.

- Disabled
  the watchdog is switched off; outputs that are set remain set.
- Telegram
  the watchdog is also retriggered by read telegrams to the process data.
- Write
  the watchdog is retriggered only by write telegrams to the process data.

The setting is also to be written via the Modbus Interface Offset 0x1122.

Watchdog Timeout (0x1120)

The timeout in [ms] is to be entered here, recommended values 500 ms - 5000 ms; smaller values are not recommended because this can very quickly lead to a watchdog error. The maximum value is one minute.

Writelock (0x1124)

Enables writing from a second master (client) also. Note that only the first master triggers the watchdog when writing from a second master. Furthermore, also pay attention to overlapping write commands. Sample: if a master sets the first digital output to TRUE and the second Modbus client sets the first output to FALSE, the output will be switched on and off continuously.

Fallback Mode (0x1123)

The fallback mode is only active if watchdog mode is also activated. You can choose here between "Freeze" (outputs are retained) and "Set to Zero" (outputs are written to zero).

Enable ModbusTCP

Activate or deactivate the ModbusTCP protocol.

Max. TCP connection

Number of simultaneously used TCP connections. It is recommended not to allow more than 3. The more TCP connections, the worse the performance of the coupler.

Enable ModbusUDP

Activate or deactivate the ModbusUDP protocol.

TCP/UDP-Port

TCP or UDP port number of the Modbus communication.

Ethernet statistics

Displays the statistics for the incoming and transmitted Ethernet frames.

IP Stack statistics

Displays the IP statistics.

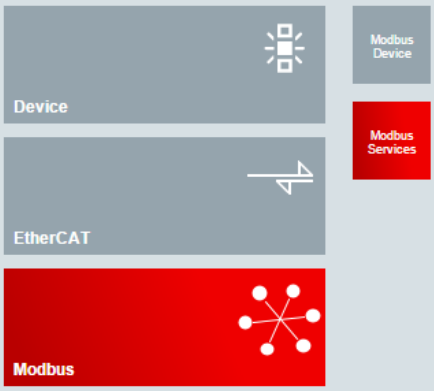
Modbus Service statistics

Displays the Modbus statistics (see also Diagnosis Function 8 [▶ 43]).

5.4 EtherCAT mapping in the Modbus process image

The Modbus service page shows in which Modbus process image (offset) the EtherCAT Terminals that are used are located. The page also indicates where the terminals are located in the Modbus process image and which Modbus function can be used to access them.

Information on how to interpret the data of the individual EtherCAT Terminals can be found in the terminal-specific documentation.



Modbus Register Interface

Address	Function	Value
0x0000..0x07FF	Process data interface - Inputs	
0x0800..0x0FFF	Process data interface - Outputs	
0x1000	Coupler Id	EK9000Modbus
0x1010	Holding register area byte size	23
0x1011	Input register area byte size	70
0x1012	Coil area bit size	2
0x1013	Discrete input area bit size	0
0x1020	Watchdog expiration time in ms	997
0x1120	Watchdog Timeout in ms	1000
0x1121	Watchdog Reset (write 1 to reset)	0
0x1122	Watchdog Mode	Telegram
0x1123	Fallback Mode	Set to Zero
0x1124	Use write lock	0

Modbus Mapping - Input Register (Read 4)

Address	Slave	Bit Offset	Data Size
0x0000..0x0007	Slave 2 (EL3204-0000-0018)	0	16 Bytes
0x0008..0x000B	Slave 3 (EL3202-0000-0016)	0	8 Bytes
0x000C..0x0011	Slave 4 (EL1502-0000-0020)	0	12 Bytes
0x0012..0x0019	Slave 5 (EL3314-0000-0017)	0	16 Bytes
0x001A..0x001B	Slave 6 (EL3002-0000-0018)	0	4 Bytes
0x001C..0x0022	Slave 8 (EL5151-0000-0024)	0	14 Bytes

Modbus Mapping - Holding Register (Read 3,23 Write 6,16,23)

Address	Slave	Bit Offset	Data Size
0x0800..0x0805	Slave 4 (EL1502-0000-0020)	0	12 Bytes
0x0806..0x0807	Slave 7 (EL4032-0000-0016)	0	4 Bytes
0x0808..0x080A	Slave 8 (EL5151-0000-0024)	0	6 Bytes
0x080B..0x080B	Slave 1 (EL2002-0000-0017)	0	2 Bits

Modbus Mapping - Input Status (Read 2)

Bit Address	Slave	Data Size

Modbus Mapping - Coils (Read 1 Write 5)

Bit Address	Slave	Data Size
1	Slave 1 (EL2002-0000-0017)	2 Bits

Fig. 18: EtherCAT mapping in the Modbus process image

5.5 EK9000 - EtherCAT configurations

The EK9000 is an EtherCAT master with automatic configuration, i.e. all EtherCAT Terminals must always be present when switching on the system. Since the boot-up of the EK9000 generally takes considerably longer than the start-up of the EtherCAT slave devices, the latter can be operated on the same power supply. With decentralised EtherCAT slaves, care must be taken that they are switched on earlier or at the same time as the supply voltage.

Switching EtherCAT devices on or off during the runtime

If one or more EtherCAT devices should fail during the operating phase, an error response is generated on the Modbus. The input data of all EtherCAT devices are then invalid and the output data are no longer accepted. This also applies to the devices that are still in operation on the EK9000. If you wish to use the option to plug in or unplug devices during the runtime, a further “Sync Unit” must be configured. This is not possible with an EK9000. In this case use a CX8090.

EtherCAT topology

All EtherCAT devices must be entered in the order in which they map themselves on the EK9000 and thus on the EtherCAT master. EtherCAT devices are automatically addressed; with a few exceptions all EtherCAT Bus Terminals are equipped with an EtherCAT ASIC. EtherCAT Terminals without an ASIC are, for example, EL9400, EL9070 and other EL9xxx. You can identify these EtherCAT Terminals using the technical data "Message to E-bus". If there is a "-" here, this terminal need not be taken into account for the mapping.

EtherCAT devices are registered in the direction of the EtherCAT telegram.

Sample configuration with EK1100 EtherCAT coupler

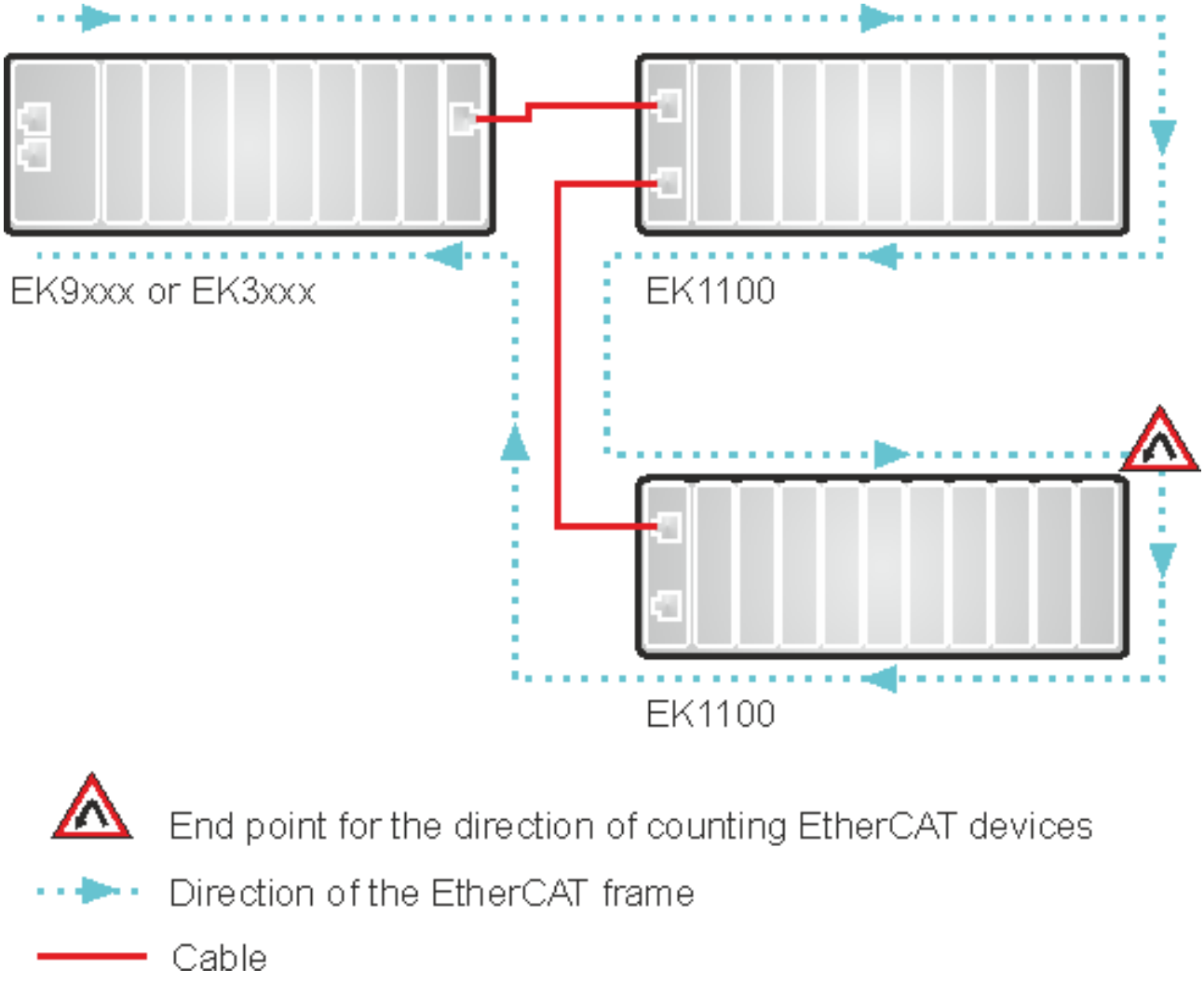


Fig. 19: Sample configuration with EK1100 EtherCAT coupler

Sample configuration with EPxxxx EtherCAT Box

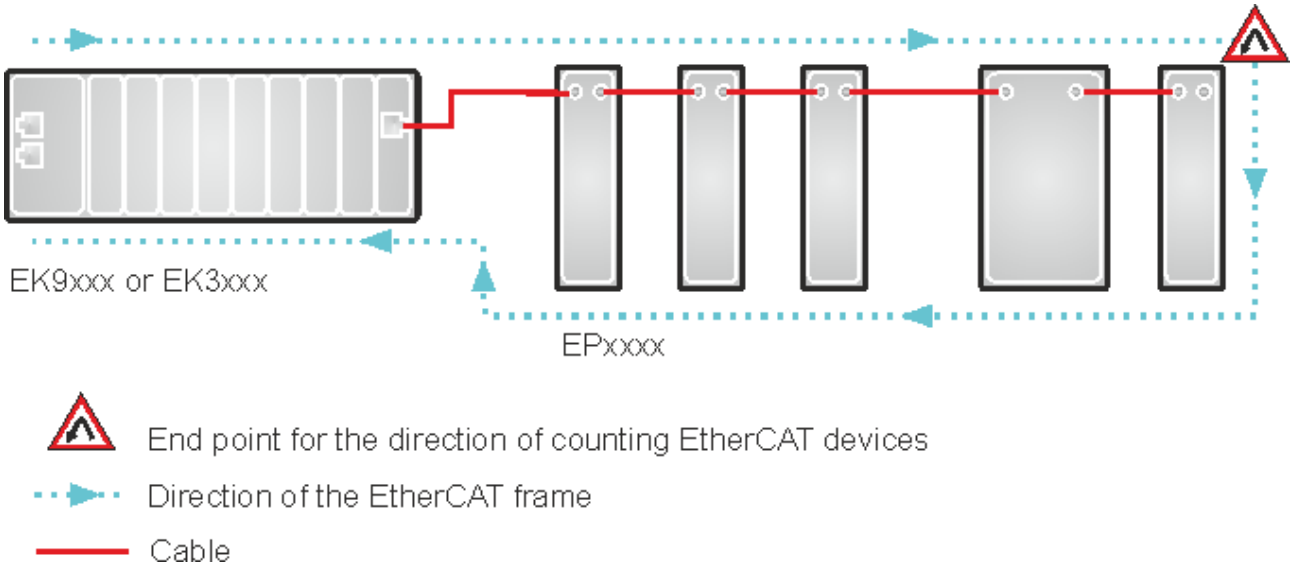


Fig. 20: Sample configuration with EPxxxx EtherCAT Box

Example configuration with EK1122 (2-port EtherCAT junction in protection class IP20)

The counting direction is to be observed when using an EK1122!

If EtherCAT junction 1 on the EK1122 is connected, then the EtherCAT frame is forwarded here first (1); if junction 1 is not connected the frame on junction 2 is sent (2), only after that does the sequence continue with the E-bus on the right-hand side (3).

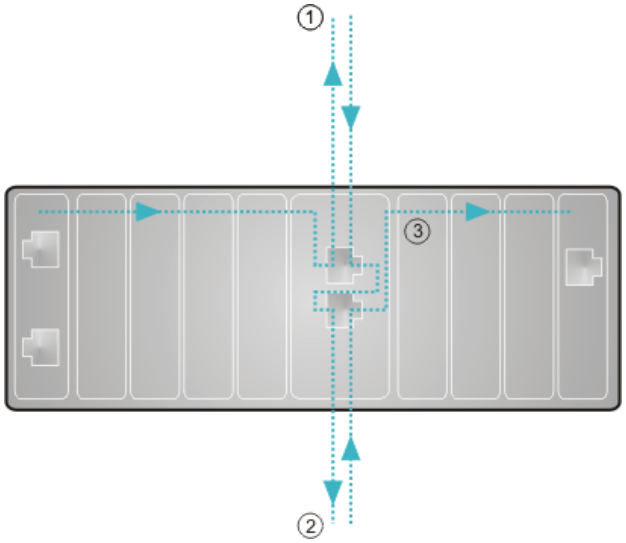


Fig. 21: Example configuration with EK1122 (2-port EtherCAT junction in protection class IP20)

If neither junction is used, then junctions 1 and 2 are bridged, so to speak, and the EtherCAT frame goes directly to the E-Bus on the right-hand side.

Example configuration with EP1122 (2-port EtherCAT junction in protection class IP65)

The counting direction is to be observed when using an EP1122! It is comparable with the EK1122.

If EtherCAT junction 1 on the EP1122 is connected, then the EtherCAT frame is forwarded here first (1); if junction 1 is not connected the frame on junction 2 is sent (2), only after that does the sequence continue with the EtherCAT connection on the right-hand side (3).

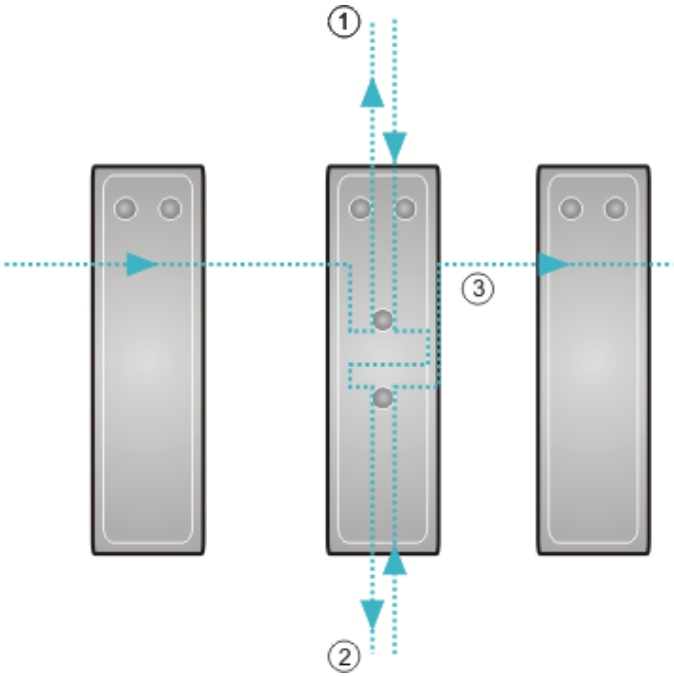


Fig. 22: Example configuration with EP1122 (2-port EtherCAT junction in protection class IP65)

If neither junction is used, then junctions 1 and 2 are bridged, so to speak, and the EtherCAT frame goes directly to the EtherCAT connection on the right-hand side.

No Hot Swap during operation

You cannot use EP1122 and EK1122 on an EKxxxx for Hot Swap and also not for connection and disconnection during operation. EP1122 and EK1122 are suitable only for topology extension (star) on an EKxxxx.

6 Ethernet

6.1 ModbusTCP/UDP

6.1.1 ModbusTCP/UDP protocol

The Ethernet protocol is addressed by means of the MAC-ID. The user does not normally need to be concerned about this address. The IP number has a length of 4 bytes, and must be parameterized by the user on the Bus Coupler and in the application. In ModbusTCP, the TCP port is set to 502. The UNIT can be freely selected under ModbusTCP, and does not have to be configured by the user.

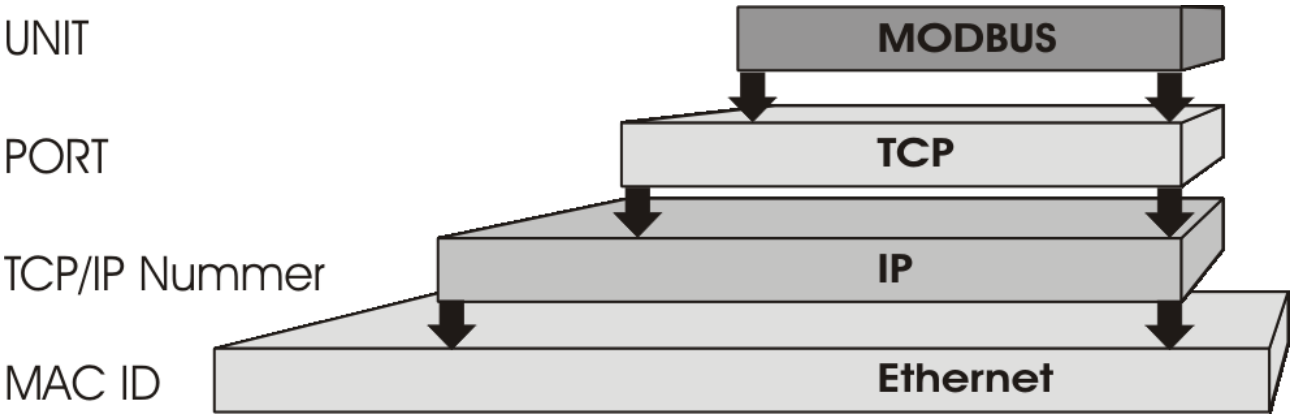


Fig. 23: ModbusTCP/UDP protocol stack

ModbusUDP

Modbus UDP can also be used instead of ModbusTCP with some products. The port number usually remains the same. Very little changes for the user.

The advantages of ModbusUDP are that it is in most cases simpler to use and usually somewhat faster than a TCP/IP connection.

The advantages of ModbusTCP are the more secure data transmission with repetitions already built in as well as further typical advantages that a TCP/IP connection features.

TCP port number

The TCP port number for ModbusTCP has been standardised to 502.

Modbus-Unit

The unit is returned by the slave.

ModbusTCP Protocol

Byte	Name	Description
0	Transaction identifier	Is returned by the slave
1	Transaction identifier	Is returned by the slave
2	Protocol identifier	always 0
3	Protocol identifier	always 0
4	Length field	0 (if the message is less than 256 bytes in length)
5	Length field	Number of following bytes
6	UNIT identifier	returned by the slave
7	Modbus	Modbus protocol with the function follows

6.1.2 Modbus-Interfaces

Address		Description		
0x0000 0x00FF		Process data interface Inputs		
0x0800 0x08FF		Process data interface Outputs		
0x1000 0x1006	Read only	Bus Coupler identification		
0x1010		Process image length in bits, analog outputs		
0x1011		Process image length in bits, analog inputs		
0x1012		Process image length in bits, digital outputs		
0x1013		Process image length in bits, digital inputs		
0x1020		Watchdog, current time in [ms]		
0x1021		Number of fallbacks triggered (watchdog timeout or bus error)		
0x1022		Number of active TCP connections		
0x1030		Hardware version		
0x1031		Software Version Main		
0x1032		Software Version Sub-Main		
0x1033		Software Version Beta		
0x1034		Serial number		
0x1035		Date of manufacture - day		
0x1036		Date of manufacture - month		
0x1037		Date of manufacture - year		
0x1040		E-bus status 1 OK, 0 not OK		
0x1120	Read/Write	Watchdog, pre-defined time in [ms] (Default value: 1000)		
0x1121		Watchdog Reset Register		
0x1122		Type of watchdog	2	Disable
			1	Telegram watchdog (default)
			0	Write telegram watchdog
0x1123		Fallback Mode Behaviour in case of a watchdog or E-bus error.	2	Stop E-bus; E-bus is stopped and can be reactivated with 0x1140
			1	Freeze
			0	Set to Zero (Default)
0x1124		Writelock Locks writing from a second Modbus client	1	Yes (Default)
			0	No
0x1140		Write 0, E-bus is switched to INUIT, value > 0 EtherCAT master is switched to OP (attention: the last output values will be overwritten again), 5-sec timeout in which no fallback is taken into account.		

Extended Modbus interface

Address		Description
0x6000	Read only	Bus Coupler ID (9000dec)
0x6001		1st terminal
0x60xx		xxth terminal

Address		Register communication via Modbus interface - description	
0x1400	Read/Write	READ	WRITE
		STATUS	CONTROL
		0x010x - Error	0x0001 execute
		0x020x - Busy	-
		0x040x - done	-
0x1401		Bit 15 0 - Read/ 1 - Write | Bit 0..14 terminal number (position in the EtherCAT Terminal System), starts with 1	
0x1402		CoE index number	
0x1403		Register bits 0..7 sub-index, bits 8 - 15 not used	
0x1404		Length in bytes	
0x1405		Error Code: ADS error code	
0x1406..14FF		Data[1]...Data[FF]	

Sample: Read the 2nd EtherCAT Terminal (EL3204) object 0x1008 sub-index 0x00 (name of the terminal)

1st step

ModbusRegWrite

MB.Addr:=0x1400
MB.Len:= 5 (words or register)
0x1400:=0x0001
0x1401:=0x0002 ; 2nd terminal
0x1402:=0x1008 ; read object 0x1008
0x1403:=0x0000; SubIndex 0
0x1404:=0x0000 ; length (only necessary for writing)

2nd step

ModbusRegRead

MB.Addr:=0x1400
MB.Len:= 12 (words or register)
0x1400:=0x0200 ; SDO communication not yet completed; polling must continue; step 2 must be repeated until the value is 0x0400 (where 0x0500 means an error).
0x1401:=0x0002;2nd terminal
0x1402:=0x1008;
0x1403:=0x0000;

3rd step

ModbusRegRead

MB.Addr:=0x1400
MB.Len:= 12 (words or register)
0x1400:=0x0400 ; no error, data ready
0x1401:=0x0002 ; 2nd terminal
0x1402:=0x1008 ; object 0x1008
0x1403:=0x0000 ;SubIndex
0x1404:=0x000B ; length
0x1405:=0x0000; Error Code
0x1406:=0x4C45; 'EL'
0x1407:=0x3233; '32'
0x1408:=0x3430; '02'
0x1409:=0x30D2; '-0'
0x140A:=0x3030; '00'
0x140B:=0x0030; '0'

Sample: Write the 5th EtherCAT Terminal (EL3318), object 0x8000sSub-index 0x19, value for 75µV measurement = 102 dec

Example of a write to an EL3318 (5th terminal) at which the first channel of the terminal is to be set to the type 75µV measurement.

1st step

Write Multiple Register (Modbus Function Code 16)
MB.Addr:=0x1400
MB.Len:= 7 (words or register)
0x1400:=0x0001
0x1401:=0x8005; write (bit 15 = TRUE), 5th terminal
0x1402:=0x8000; object 0x8000
0x1403:=0x0019; sub-index 0x19
0x1404:=0x0002; length
0x1405:=0; Error Code
0x1406:=102; value for 0x8000, SubIdx 0x19

2nd step

Read Multiple Register (Modbus Function Code 3), at least 6 words should be read here.
If an error occurs, the error code is from offset 0x1405.
Query until bit 10 in offset 0x1400, if this is "done" (0x04xx), the writing was successful.
If the value 0x02xx is the answer in offset 0x1400, writing is not yet completed.
The reading should then be repeated.
If the value 0x01xx is the answer in offset 0x1400, an error has occurred.
The error code is then found in offset 0x1405, it is an ADS error code.

Modbus UDP

The communication can also optionally be used via Modbus UDP.

6.1.3 ModbusTCP slave error answer (BK9000, BX/BC9000, IP/ILxxxx-B/C900, EK9000)

When the user sends the slave either a request or information that the coupler does not understand, the slave responds with an error report. This answer contains the function and the error code. 0x80 is added to the value returned by the function.

Code	Name	Meaning
1	ILLEGAL FUNCTION	Modbus function not implemented
2	ILLEGAL DATA ADDRESS	Invalid address or length
3	ILLEGAL DATA VALUE	Invalid parameter - Diagnostic functions - Incorrect register
4	SLAVE DEVICE ERROR	Watchdog or K-Bus error EK9000: E-bus error
6	SLAVE DEVICE BUSY	Output data is already been received from another IP device

6.1.4 Access via a second client

The EK9000 can also be addressed via a further client. This can access the coupler for reading. However, a write procedure is rejected and acknowledged with a Modbus error as long as the first client is still triggering the watchdog.

If the first client fails, the second client takes over after half the watchdog time and can write the outputs itself after half the watchdog time. If the watchdog time is set accordingly high, this can take place virtually non-reactively, i.e. the set outputs remain set. If the first client becomes active again, it then receives a Modbus error message when writing.

Function

The manufacturer of the application is solely responsible for the use and method of this type of communication.

When this method doesn't take effect

The first client's application is frozen, but Modbus telegrams are still being sent. In this case the second Modbus client would have no chance to take over the EK9000 because Modbus telegrams are still being sent.

6.1.5 Modbus functions

6.1.5.1 ModbusTCP functions

In the Modbus protocol, the functions determine whether data is to be read or written, and what kind of data is involved.

Function	Code	Description
Read coil status [▶ 40]	1	Read digital outputs
Read input status [▶ 41]	2	Read digital inputs
Read holding register [▶ 41]	3	Read analog outputs and inputs / GPR
Read input register [▶ 42]	4	Reading the inputs / GPR
Force single coil [▶ 42]	5	Writing a digital output
Preset single register [▶ 43]	6	Writing an output / GPR
Diagnostics [▶ 43]	8	Diagnostics
Force multiple coils [▶ 47]	15	Write a number of digital outputs
Preset multiple register [▶ 47]	16	Writing several outputs / GPRs
Read / write registers [▶ 48]	23	Write and read a number of process data outputs / GPRs

GPR (General Preset Register) - register structure of the Modbus interface (see Appendix)

6.1.5.2 Read coil status (Function 1)

The *Read coil status* function can be used to read the digital outputs that have been set.

The first 10 digital outputs are read in this example. The start address is zero. An offset can be entered in the *Start address* field

Query

Byte Name	Sample
Function code	1
Start address high	0
Start address low	0
Count high	0
Count low	10

The fieldbus coupler answers with *byte count* 2, i.e. 2 bytes of data are returned. The query was for 10 bits, and these are now distributed over 2 bytes. The third bit is set in the output process image, and the fieldbus coupler shows the value 4 in the first data byte.

Response

Byte Name	Sample
Function code	1
Byte Count	2
Data bits 0...7	4
Data bits 8..0.18	0

6.1.5.3 Read input status (Function 2)

The *Read input status* function can be used to read the digital input data. The first 10 digital inputs are read in this example. The start address is zero. An offset can be entered in the *Start address* field

Query

Byte Name	Sample
Function code	2
Start address high	0
Start address low	0
Count high	0
Count low	10

The fieldbus coupler answers with *byte count* 2, i.e. two bytes of data are returned. The query was for 10 bits, and these are now distributed over 2 bytes. The third bit is set in the output process image, and the fieldbus coupler shows the value 4 in the first data byte.

Response

Byte Name	Sample
Function code	2
Byte Count	2
Data bits 0..0.7	1
Data bits 8...18	0

6.1.5.4 Read holding register (Function 3)

The *Read holding register* function can be used to read the input and output words and the registers. Inputs from offset 0 - 0xFF and outputs from offset 0x800 - 0x8FF, and for controllers (BC, BX) the flag area from offset 0x4000.

In this example the first two analog outputs (or two output words) are read. The analog outputs (or output words) start at offset 0x800. The length indicates the number of channels (or words) to be read.

Query

Byte Name	Sample
Function code	3
Start address high	8
Start address low	0
Count high	0
Count low	2

The fieldbus coupler answers with byte count 4, i.e. 4 bytes of data are returned. The query was for two analog channels, and these are distributed over two words. In the analog output process image, the first channel has the value 0x3FFF, while the second channel has the value 0x0.

Response

Byte Name	Sample
Function code	3
Byte Count	4
Data 1 high byte	63
Data 1 low byte	255
Data 2 high byte	0
Data 2 low byte	0

6.1.5.5 Read input register (Function 4)

The function *Read input register reads the inputs on a word basis.*

In this example the first two analog inputs (or the first two input words) are read. The analog inputs (or input words) start at an offset of 0x0000. The length indicates the number of words to be read. A KL3002, for example, has two words of input data. Therefore, the length to be entered at *Number low* is two.

Query

Byte Name	Sample
Function code	4
Start address high	0
Start address low	0
Count high	0
Count low	2

The fieldbus coupler answers with byte count 4, i.e. four bytes of data are returned. The query was for two analog channels, and these are now distributed over 2 words. In the analog input process image, the first channel has the value 0x0038, while the second channel has the value 0x3F1B.

Response

Byte Name	Sample
Function code	4
Byte Count	4
Data 1 high byte	0
Data 1 low byte	56
Data 2 high byte	63
Data 2 low byte	11

6.1.5.6 Force single coil (Function 5)

The *Force single coil* function can be used to write a digital output. The third digital output is written in this example. The digital outputs start at address 0x0000. The digital value is located in *Data high*. To switch the output on, *Data high* must contain the value 0xFF, while 0x00 is used to switch the output off again. *Data low* must contain the value 0x00.

Query

Byte Name	Sample
Function code	5
Start address high	0
Start address low	2
Data high	255
Data low	0

The coupler answers with the same telegram.

Response

Byte Name	Sample
Function code	5
Start address high	0
Start address low	2
Data high	255
Data low	0

6.1.5.7 Preset single register (Function 6)

The function *Preset singles register* can be used to access the output or flag process image (only for controllers) and the Modbus TCP interface [▶ 36].

Function 6 writes the first output word. The outputs start at an offset of 0x0800. Here again the offset always describes a word. This means offset 0x0803 refers to the fourth word in the output process image.

Query

Byte Name	Sample
Function code	6
Start address high	8
Start address low	0
Data high	63
Data low	255

The Fieldbus Coupler replies with the same telegram and confirmation of the received value.

Response

Byte Name	Sample
Function code	6
Start address high	8
Start address low	0
Data high	63
Data low	255

6.1.5.8 Diagnosis (Function 8)

The diagnosis function provides a series of tests for checking the communication system between the master and the slave and for examining a variety of internal error states within the slave.

The function uses two bytes in the query to specify a sub-function code defining the test that is to be carried out. The slave returns the function code and the sub-function code in the response.

The diagnostic queries use a two-byte data field to send diagnostics data or control information to the slave.

Query

Byte Name	Sample
Function code	8
Sub-function high	0
Sub-function low	0
Data high	2
Data low	3

Response

Byte Name	Sample
Function code	8
Sub-function high	0
Sub-function low	0
Data high	2
Data low	3

Coupler reset (Sub-function 1)

The Bus Coupler is reset with sub-function 1, error counters are reset and the Bus Coupler performs a self-test. No telegrams are either received or sent while the Bus Coupler is being reset. The IP socket is closed.

Tip or pointer

Before the Coupler restarts it sends a reply with sub-function 1, after which the IP socket is closed.

Sub-function	Data field (query)	Data field (response)
0x0001	0x0000	0x0000

Delete all counter contents (sub-function 10)

When this sub-function is called the controller clears all error counters.

Sub-function	Data field (query)	Data field (response)
0x000A	0x0000	Echo query data

Read Coils (sub-function 32)

Indicates the number of Read Coils commands.

Sub-function	Data field (query)	Data field (response)
0x0020	0x0000	Counter value

Read Coils Error (sub-function 33)

Indicates the number of Read Coils commands with error.

Sub-function	Data field (query)	Data field (response)
0x0021	0x0000	Counter value

Read Discrete Inputs (sub-function 34)

Indicates the number of Read Discrete Inputs commands.

Sub-function	Data field (query)	Data field (response)
0x0022	0x0000	Counter value

Read Discrete Inputs Error (sub-function 35)

Indicates the number of Read Discrete Inputs commands with error.

Sub-function	Data field (query)	Data field (response)
0x0023	0x0000	Counter value

Read Holding Register (sub-function 36)

Indicates the number of Read Holding Register commands.

Sub-function	Data field (query)	Data field (response)
0x0024	0x0000	Counter value

Read Holding Register Error (sub-function 37)

Indicates the number of Read Holding Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x0025	0x0000	Counter value

Read Input Register (sub-function 38)

Indicates the number of Read Input Register commands.

Sub-function	Data field (query)	Data field (response)
0x0026	0x0000	Counter value

Read Input Register Error (sub-function 39)

Indicates the number of Input Holding Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x0027	0x0000	Counter value

Write Single Coil (sub-function 40)

Indicates the number of Write Single Coil commands.

Sub-function	Data field (query)	Data field (response)
0x0028	0x0000	Counter value

Write Single Coil Error (sub-function 41)

Indicates the number of Write Single Coil commands with error.

Sub-function	Data field (query)	Data field (response)
0x0029	0x0000	Counter value

Write Single Register (sub-function 42)

Indicates the number of Write Single Register commands.

Sub-function	Data field (query)	Data field (response)
0x002A	0x0000	Counter value

Write Single Register Error (sub-function 43)

Indicates the number of Write Single Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x002B	0x0000	Counter value

Diagnostic Register (sub-function 44)

Indicates the number of Diagnostic Register commands.

Sub-function	Data field (query)	Data field (response)
0x002C	0x0000	Counter value

Diagnostic Register Error (sub-function 45)

Indicates the number of Diagnostic Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x002D	0x0000	Counter value

Write Multiple Coils (sub-function 46)

Indicates the number of Write Multiple Coils commands.

Sub-function	Data field (query)	Data field (response)
0x002E	0x0000	Counter value

Write Multiple Coils Error (sub-function 47)

Indicates the number of Write Multiple Coils commands with error.

Sub-function	Data field (query)	Data field (response)
0x002F	0x0000	Counter value

Write Multiple Register (sub-function 48)

Indicates the number of Write Multiple Register commands.

Sub-function	Data field (query)	Data field (response)
0x0030	0x0000	Counter value

Write Multiple Register Error (sub-function 49)

Indicates the number of Write Multiple Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x0031	0x0000	Counter value

R/W Multiple Register (sub-function 50)

Indicates the number of R/W Multiple Register commands.

Sub-function	Data field (query)	Data field (response)
0x0032	0x0000	Counter value

R/W Multiple Register Error (sub-function 51)

Indicates the number of R/W Multiple Register commands with error.

Sub-function	Data field (query)	Data field (response)
0x0033	0x0000	Counter value

6.1.5.9 Force multiple coils (Function 15)

The *Force multiple coils* function can be used to set or reset a number of digital outputs at the same time.

The first 20 digital outputs are written in this example. The digital outputs start at an offset of 0x0000. Here the offset always describes a bit. Offset 0x0003 writes to the fourth bit in the output process image. The length indicates the number of bits, and the *Byte count* is formed from the combination all the bytes that are to be written.

Sample: 20 bits yield a byte count of 3 (rounded up to a byte boundary).

The data bytes contain the values for the individual bits. In this example, bits 0 to 15 are set to TRUE, while bits 16 to 23 are FALSE.

Query

Byte Name	Sample
Function code	15
Start address high	0
Start address low	0
Length high	0
Length low	20
Byte Count	3
Data 1 bit 0..0.7	255
Data 2 bit 8..0.15	255
Data 3 bit 16...23	0

Response

The Bus Coupler answers with the same telegram.

Byte Name	Sample
Function code	15
Start address high	0
Start address low	0
Length high	0
Length low	20

6.1.5.10 Preset single register (Function 16)

The *Preset multiple register* function can be used to write a number of outputs. The first two analog output words are written in this example. The outputs start at an offset of 0x0800. Here the offset always describes a word. Offset 0x0003 writes to the fourth word in the output process image. The length indicates the number of words, and the *Byte count* is formed from the combination of all the bytes that are to be written.

Sample: 4 words – correspond to a byte count of 8

The data bytes contain the values for the analog outputs. In this example, two words are to be written. The first word is to receive the value 0x7FFF, and the second word is to receive the value 0x3FFF.

Query

Byte Name	Sample
Function code	16
Start address high	8
Start address low	0
Length high	0
Length low	2
Byte Count	4
Data 1 byte 1	127
Data 1 byte 2	255
Date 2 byte 1	63
Data 2 byte 2	255

Response

The coupler replies with the start address and the length of the transmitted words.

Byte Name	Sample
Function code	16
Start address high	8
Start address low	0
Length high	0
Length low	2

6.1.5.11 Read / write registers (Function 23)

A number of analog outputs can be written and a number of analog inputs read with one telegram using the *Read / write registers* function. In this example the first two analog output words are written, and the first two analog inputs are read. The analog outputs start at offset 0x0800, while the inputs start at offset 0x0000. Here the offset always describes a word. Offset 0x0003 writes to the fourth word in the output process image. The length indicates the number of words, and the *Byte count* is formed from the combination of all the bytes that are to be written.

Sample: 4 words – correspond to a byte count of 8

The data bytes contain the values for the analog outputs. In this example, two words are to be written. The first word is to receive the value 0x3FFF, and the second word is to receive the value 0x7FFF.

Query

Byte Name	Sample
Function code	23
Read start address high	0
Read start address low	0
Read length high	0
Read length low	2
Write start address high	8
Write start address low	0
Write length high	0
Write length low	2
Byte Count	4
Data 1 high	63
Data 1 low	255
Data 2 high	127
Data 2 low	255

Response

The coupler replies with the start address and the length of the bytes to be transferred in *Byte count*. The data information follows. In this example the first word contains 0x0038 while the second word contains 0x3F0B.

Byte Name	Sample
Function code	23
Byte Count	4
Data 1 high	0
Data 1 low	56
Data 2 high	63
Data 2 low	11

7 Error handling and diagnosis

7.1 LED indicators

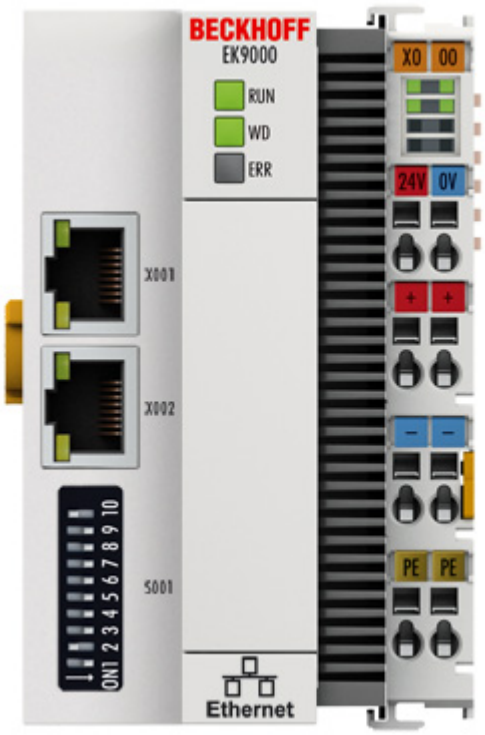


Fig. 24: EK9000 LEDs

Ethernet interface X001

Interface X001/X002	Ethernet	Meaning
LED green	on	Link available/activity
LED yellow	is not used	-

LEDs on the coupler

Labelling	Meaning	Colours	Meaning
RUN	Indicates the status of the coupler	red	May only light up during the start-up phase
		green	Coupler is ready
		blue (If red DIP switch 1 is set to on when starting the coupler)	The internal Flash can be reached via USB (firmware update)

LED WD	Modbus Status	Meaning
	green/red/yellow	
EtherCAT bootup	yellow 200 ms flashing	EC master booting up, Modbus communication not yet possible
Parameterization of the terminals	yellow 400 ms flashing	EC master parameterizing the EC Terminals, Modbus communication not yet possible
Run	green on	OK
Watchdog	red 400 ms flashing	Watchdog error

LED ERR	Modbus diagnosis		Meaning
	green	red	
Modbus error	off	briefly lights up	Erroneous Modbus telegram received (see Diag History)
Coupler IP setting	off	flashing 400 ms	Coupler has no IP address (DHCP active)
Device is in IO exchange	off	off	OK
2nd Client	on	off	2nd client is active, but write telegrams are responded to with an error

LEDs on the power supply terminal

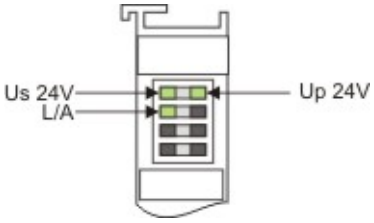


Fig. 25: LEDs on the power supply terminal

Operation with E-bus terminals

Display LED	Description	Meaning
1 Us 24 V (top left, 1st row)	Coupler supply voltage	connected to -24 V
2 Up 24 V (top right, 1st row)	Power contacts supply voltage	connected to -24 V
3 L/A (left center, 2nd row)	EtherCAT LED	flashing green: EtherCAT communication active connected to E-bus / no data traffic not connected to E-bus

8 Appendix

8.1 Update Bus Coupler image

Loss of data

The data in the internal flash memory are deleted.
Save your data before you update the Bus Coupler image.

The Bus Coupler image can be updated via the USB interface. To this end the Bus Coupler is connected with a host PC via a USB cable. Windows then shows the Bus Coupler as a removable data storage device, and the files can be copied.

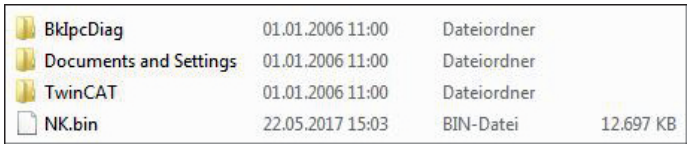
The Bus Coupler should only be updated after consultation with the Beckhoff Service. The Beckhoff Service will provide all the required files.

Requirements

- First, check whether the Bus Coupler supports the image.
- The Bus Coupler is connected with the host PC via a USB cable.

Update the image as follows:

1. Switch off the Bus Coupler.
2. Switch the red 2-pin **DIP switch 1** to “on” (to the right) and switch on the Bus Coupler.
   The Bus Coupler appears as a removable data storage device on the host PC.
3. Select and delete all files. Do not format.



4. Remove the USB cable, once all files have been copied, and switch the 2-pin DIP switch to “off” (to the left).
5. Restart the Bus Coupler.

⇨ The image has been updated successfully. After the update, the Bus Coupler may take a little longer to start up.

8.2 FAQ

How can I leave the outputs in the current state in case of a Modbus error?

Set the watchdog module to disable [Web page] or the Modbus interface 0x1122 to "2".

How can I change the mapping of an EtherCAT Terminal?

Use the Web configuration for this and generate a Restore File.

How do I know what the MAC address of the Bus Coupler is?

The MAC address is printed on the label on the side of the Bus Coupler.

What is the USB interface for and what can I do with it?

The USB interface is to be used at present only for firmware updates.

What is the purpose of the DIP switch behind the flap?

It is necessary, for example, for the use of the firmware update (see chapter entitled "DIP switch").

Can I also operate K-bus terminals?

No, only EtherCAT terminals or EtherCAT boxes can be connected. You can use the BK9050 or BK9100 for K-bus terminals. The use of EtherCAT couplers for K-bus such as the BK1120 or BK1250 is not possible.

I have an EtherCAT slave from a third-party vendor, can I also connect it?

No, devices from other vendors can only be used with a CX (see CX8090 or similar products).

I would like to operate the drive terminals/drives on the EK9000. Is that possible?

No, use a CX with a suitable performance for this, e.g. CX9020 or higher.

I would like to operate TwinSAFE terminals on the EK9000. Is that possible?

No, the TwinSAFE terminals require a TwinCAT system for configuration; use the CX8090 for this.

How do I see that there is an EtherCAT error?

In this case the ERR LED lights up red and Modbus telegrams are responded to with error code 4.

8.3 List of Abbreviations

ADS

Automation Device Specification (disclosed protocol for the communication of all Beckhoff controllers)

DAP

Device Access Point

E/A

Inputs and outputs

E-Bus

This means EtherCAT in the terminal group (ELxxxx, ESxxxx, or EMxxxx terminals)

EtherCAT

EtherCAT (Ethernet for Control Automation Technology) is the Ethernet solution for industrial automation, characterized by outstanding performance and particularly simple handling.

Fast Ethernet

This is taken to mean the data rate 100 Mbits/s to be used according to the standard 100 Base-T.

IP20

Protection class of the Bus Terminals, EtherCAT Terminals

IPC

Industrial PC

K-Bus

Terminal bus (KLxxxx, KMxxxx or KSxxxx terminals)

KS2000

Configuration software for Bus Terminals, Bus Couplers, Bus Terminal Controllers, fieldbus box modules, etc.

ModbusTCP

Modbus telegram based on TCP/IP.

ModbusRTU

Modbus telegram based on RS232/422/485.

ModbusUDP

Modbus telegram based on UDP/IP.

PE

The PE power contact can be used as a protective earth.

TwinCAT

The Windows Control and Automation Technology, programmer and configuration tool from the Beckhoff company.

8.4 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

List of illustrations