BECKHOFF New Automation Technology

Documentation | EN

EP6001-0002

1-channel serial interface (RS232 / RS422 / RS485)

2021-06-07 | Version: 1.2

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®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® 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, EP1456722, EP2137893, DE102015105702 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
1.2	• Dimensions updated • UL requirements updated • Technical data updated
1.1	• Front page updated
1.0	• First release, adapted from the documentation EP600x V2.1.0

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH	Example with D no. 29 10 02 01:
WW - week of production (calendar week) YY - year of production FF - firmware version HH - hardware version	29 - week of production 29 10 - year of production 2010 02 - firmware version 02 01 - hardware version 01

Further information on this topic: Version identification of EtherCAT devices [▶ 58].

2 Product group: EtherCAT Box Modules

EtherCAT Box modules are I/O modules for industrial controllers.

They comply with protection class IP67 and are intended for use outside the control cabinet in wet, dirty or dusty industrial environments.

EtherCAT Box modules communicate with the controller via the EtherCAT fieldbus. They each have two connections for EtherCAT communication and for the power supply:

- Feed
- Downstream connection

This enables the cabling of EtherCAT Box modules in a line structure:

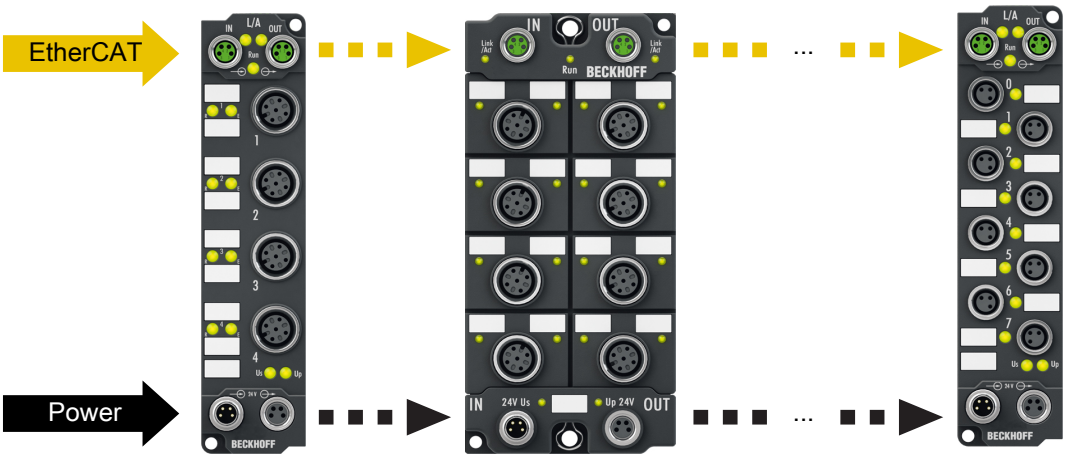


Fig. 1: EtherCAT Box modules: Example of cabling in a line structure

3 Product overview

3.1 Introduction

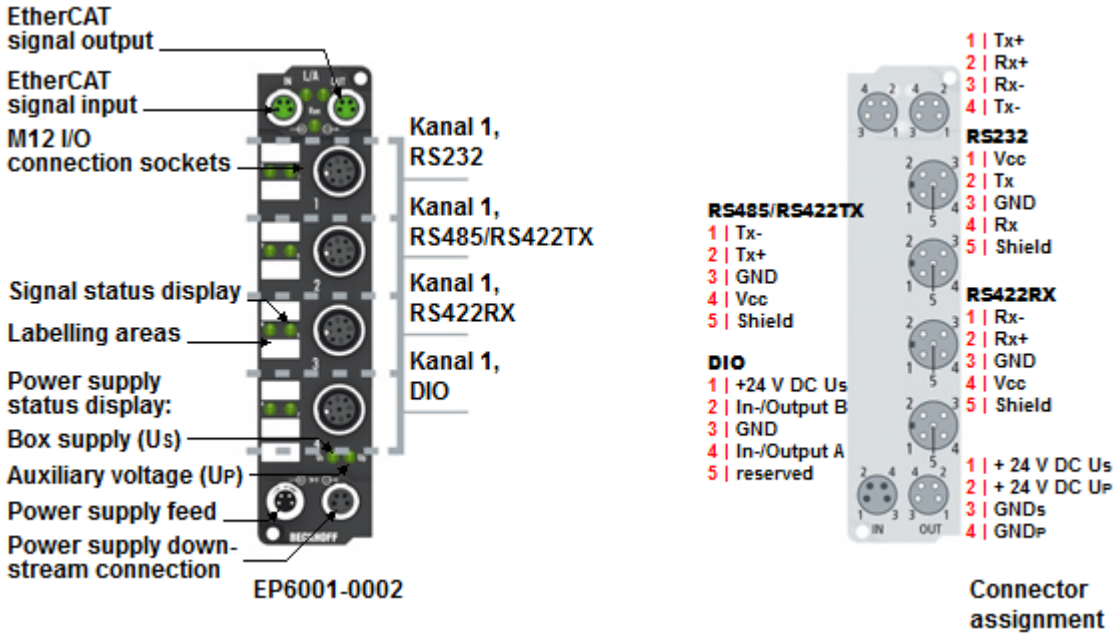


Fig. 2: EP6001-0002

EP6001-0002 | 1-channel serial interface, RS232, RS422/RS485

The EP6001-0002 serial interface module allows the connection of devices with an RS232 or an RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The active serial communication channel functions independently of the higher-level bus system in full duplex mode at up to 115,200 baud, while a 864 byte receive buffer and a 128 byte send buffer are available. The 1-channel version has an increased end device power supply of up to 1 A; the connector assignment depends on the interface selected. The two integrated digital inputs/outputs allow the connection of additional sensors/actuators in order, for example, to trigger the reading process of the barcode reader or, depending on the result, to initiate an action. In conjunction with the TwinCAT Virtual Serial COM Driver the EP6001-0002 can be used as a normal Windows COM interface.

Quick links

Technical data [▶ 10]
Process image [▶ 13]
Dimensions [▶ 17]
RS232 Connection [▶ 23]
RS422 connection [▶ 24]
RS485 connection [▶ 25]
Commissioning [▶ 29]

3.2 Technical data

All values are typical values over the entire temperature range, unless stated otherwise.

EtherCAT	
Connection	2 x M8 socket, 4-pin, green
Electrical isolation	500 V

Supply voltages	
Connection	Input: M8 connector, 4-pin Downstream connection: M8 socket, 4-pin, black
U_S nominal voltage	24 V_{DC} (-15 % / +20 %)
U_S sum current: $I_{S,sum}$	max. 4 A
Current consumption from U_S	130 mA + Current consumption of connected end devices + Loads at digital outputs
Rated voltage U_P	24 V_{DC} (-15 % / +20 %)
U_P sum current: $I_{P,sum}$	max. 4 A
Current consumption from U_P	None. U_P is only forwarded.

Serial interface	
Number of channels	1
Interface type	ajustable: • RS232 • RS422 • RS485
Connection	RS232: 1 x M12 socket RS422: 2 x M12 socket RS485: 1 x M12 socket
End device power supply	Output voltage: 5 V_{DC} Output current: max. 1 A, short-circuit proof The end device power supply is branched off from the supply voltage U_S.
Cable length	RS232: max. 15 m RS422: max. 1000 m RS485: max. 1000 m
Data transfer rate	Adjustable: 300 … 115,200 baud (bit/s)
Data format	Adjustable: 8N1, 7E1, 7O1, 8N1, 8E1, 8O1, 7E2, 7O2, 8N2, 8E2, 8O2
Flow control	Software flow control "XON/XOFF"
Bit distortion	< 3 %
Receive buffer	846 bytes
Send buffer	128 bytes

Digital inputs	
Number	0 to 2 Each digital input can alternatively be used as a digital output.
Connection	1 x M12 socket
Sensor supply	Output voltage: 24 V_{DC} Output current: max. 0.5 A, short-circuit proof The sensor supply is branched off from the supply voltage U_S.
Input filter	10 µs
Characteristics	Type 3 according to EN61131-2, compatible with type 1
Signal voltage "0"	-3…+5 V
Signal voltage "1"	+11…+30 V
Input current	3 mA

Digital outputs	
Number	0 to 2 Each digital output can alternatively be used as a digital input.
Connection	1 x M12 socket
Load type	Ohmic, inductive, lamp load
Nominal output voltage	24 V_{DC}
Output current per channel	max. 0.5 A. Each output is independently short-circuit proof.
Short circuit current	max. 1.5 A
Output driver supply	From the supply voltage U_S.

Housing data	
Dimensions W x H x D	30 mm x 126 mm x 26.5 mm (without connectors)
Weight	approx. 165 g
Installation position	variable
Material	PA6 (polyamide)

Environmental conditions	
Ambient temperature during operation	-25…+60 °C -25...+55 °C according to cURus
Ambient temperature during storage	-40…+85 °C
Vibration resistance, shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 Additional checks [▶ 11]
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 (conforms to EN 60529)

Approvals	
Approvals	CE, cURus [▶ 28]

Additional checks

The boxes have been subjected to the following checks:

Verification	Explanation
Vibration	10 frequency sweeps in 3 axes
	5 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 g, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	35 g, 11 ms

3.3 Scope of supply

Make sure that the following components are included in the scope of delivery:

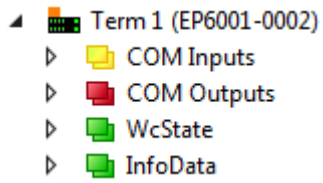
- 1x EtherCAT Box EP6001-0002
- 2x protective cap for EtherCAT socket, M8, green (pre-assembled)
- 1x protective cap for supply voltage input, M8, transparent (pre-assembled)
- 1x protective cap for supply voltage output, M8, black (pre-assembled)
- 10x labels, blank (1 strip of 10)

Pre-assembled protective caps do not ensure IP67 protection

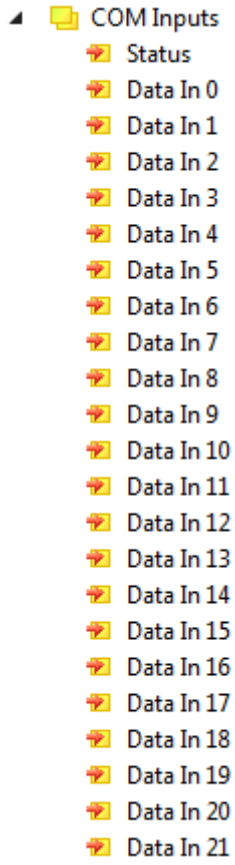
Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

Ensure that the protective caps are correctly seated to ensure IP67 protection.

3.4 Process image



COM Inputs



Status

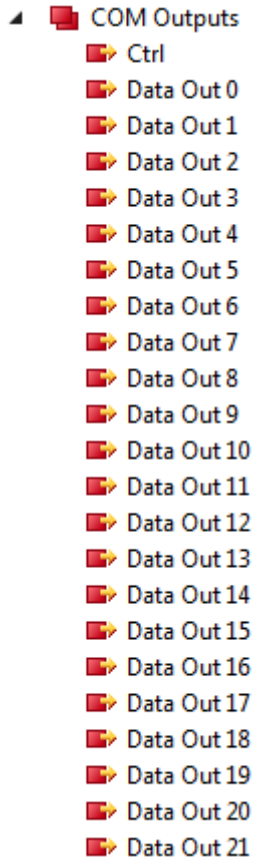
Status word for receive data.

Data In [n]

The input variables "Data In 0" .. "Data In 22" each contain one byte of receive data (USINT).

"Data In 0" contains the first-received byte.

COM outputs



Ctrl

Control word for transmit data.

Data Out [n]

The output variables "Data Out 0" .. "Data Out 22" can each be filled with one byte of send data.

The content of "Data Out 0" is transmitted first.

3.4.1 Control word

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	-	-	-	-	SC	IR	RA	TR

Bit no.	Name	Description	
15 .. 8	OL7…OL0 (OutLength)	$1_{dec}...22_{dec}$	The number of output bytes available for the transmission from the controller to the box.
7 .. 4	reserved		
3	SC (SendContinuous)	rise	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit SC. Once the data has been transferred, this is acknowledged by the box to the controller by setting the SW.2 bit. SW.2 is cancelled with CW.3.
2	IR (InitRequest)	1_{bin}	The controller requests the box to initialize. The send and receive functions are blocked, the FIFO pointers are reset, and the interface is initialized with the values of the responsible objects (baud rate 4073, data frame 4074, feature bits 4075). The execution of the initialization is acknowledged by the box with the SW.2 (IA) bit.
		0_{bin}	The controller once again requests the box to prepare for serial data exchange.
1	RA (ReceiveAccepted)	toggle	The controller acknowledges receipt of data by changing the state of this bit. Only then can new data be transferred from the box to the controller.
0	TR (TransmitRequest)	toggle	Via a change of state of this bit the controller notifies the box that the DataOut bytes contain the number of bytes indicated via the OL bits. The box acknowledges receipt of the data in the status byte by changing the state of the SW.0 (TA) bit. Only then can new data be transferred from the controller to the box.

3.4.2 Status word

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0	-	OVERRUN ERR	FRAMING ERR	PARITY ERR	BUF_F	IA	RR	TA

Key

Bit no.	Name	Description	
15 ... 8	IL7 ... IL0 (InLength)	1_{dec} .. 22_{dec}	The number of input bytes available for the transmission from the box to the controller.
7	reserved		
6	OVERRUN ERR	0 .. 1	An overrun error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost.
5	FRAMING ERR	0 .. 1	A framing error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost.
4	PARITY ERR	0 .. 1	A parity error has occurred. The data concerned is not loaded to the receive FIFO of the box and is lost.
3	BUF_F	1	The number of bytes in the receive buffer exceeds the value of parameter 8010:1A "Rx buffer full notification" (factory setting: 864 bytes).
2	IA (InitAccepted)	1	The initialization has been executed by the box.
		0	The box is ready again for serial data exchange.
1	RR (ReceiveRequest)	toggle	Via a change of state of this bit the box notifies the controller that the DataIn bytes contain the number of bytes indicated via the IL bits. The controller has to acknowledge receipt of the data in the control byte via a change of state of bit CW.1 (RA). Only then can new data be transferred from the box to the controller.
0	TA (TransmitAccepted)	toggle	The box acknowledges the receipt of data by changing the state of this bit. Only then can new data be transferred from the controller to the box.

3.5 Technology

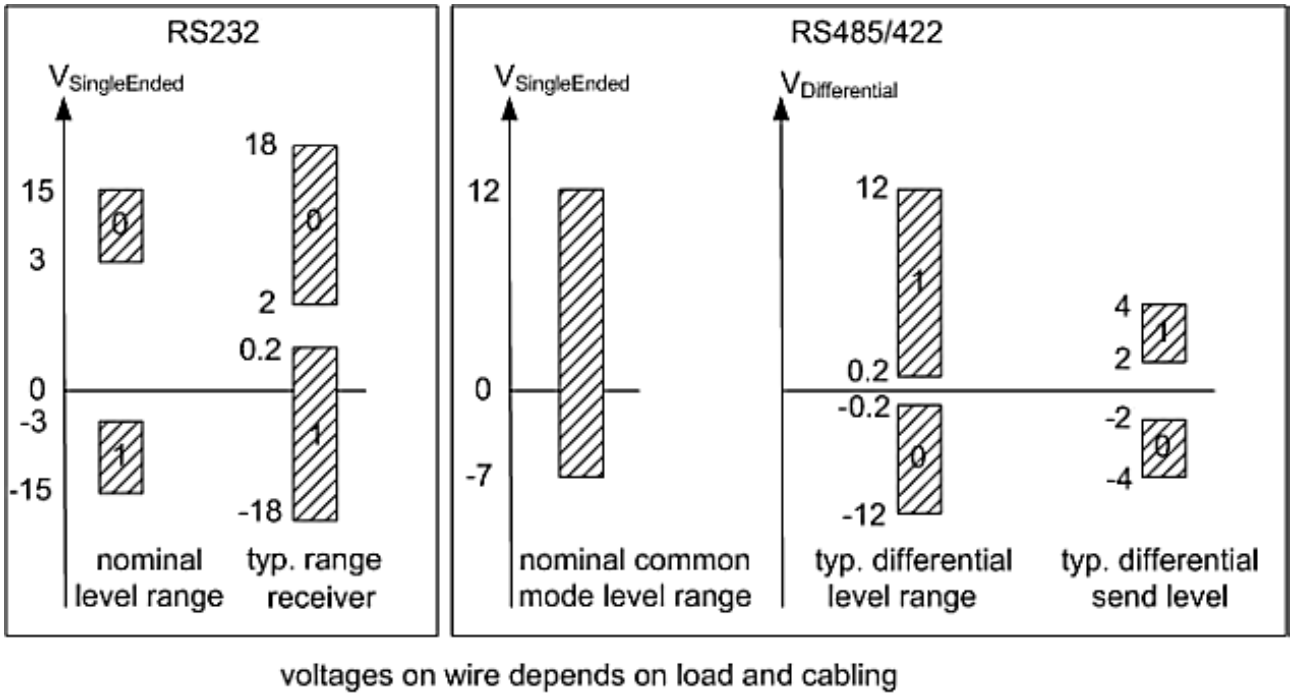


Fig. 3: Level of RS232, RS422, RS485 interfaces

Data transfer rate

The process image contains 22 bytes of user data. It is possible to transmit or receive these 22 bytes every second PLC cycle at the most:

- The data is transferred from the box to the controller in the first PLC cycle.
- In the second PLC cycle, the controller must acknowledge that it has accepted the data.

Therefore, if the cycle time is 10 ms, 50 times 22 bytes can be transmitted per second.

If the data format is set to 8N1, each transmitted byte is made up of a start bit, eight data bits and a stop bit. This is equivalent to 10 bits per byte of user data.

With the above-mentioned settings, a **continuous** data transfer rate of:

- 50[1/s] x 22[bytes] x 10[bits] = 11000 baud (bit/s)

can be achieved.

The next lower standard data transfer rate is 9600 baud. Accordingly, continuous transfer at a maximum baud rate of 9600 can be secured with a cycle time of 10 ms.

If only low quantities of data are transmitted or received sporadically (e.g. barcode scanner), the data transfer rate can also be set higher, or the cycle time can be enlarged.

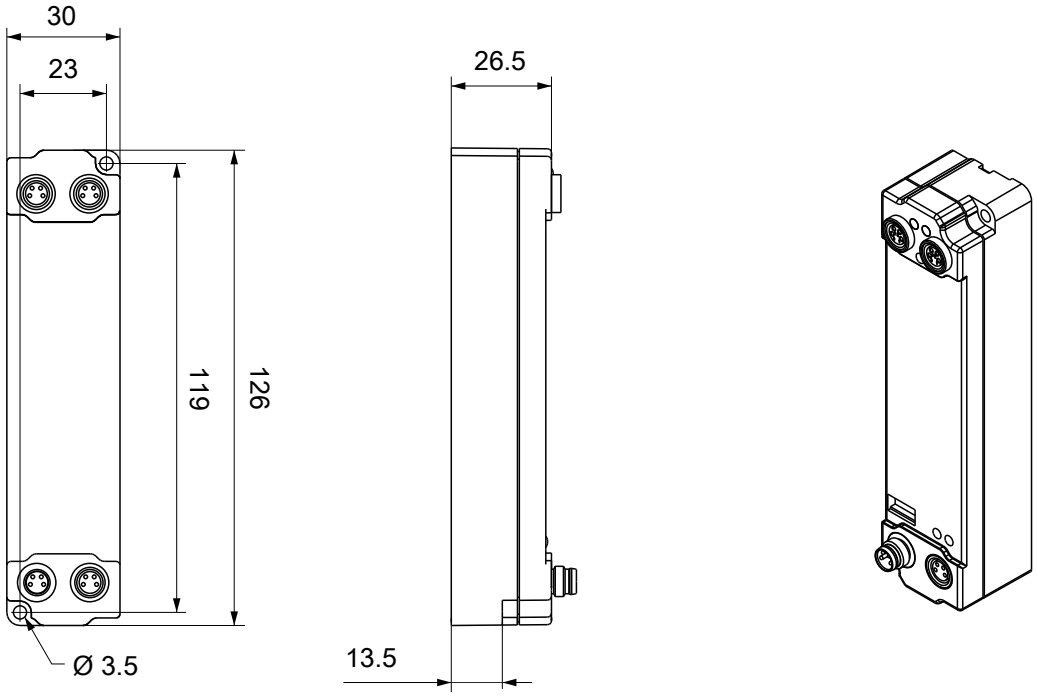
If the controller cannot fetch the data quickly enough from the box, they are buffered in the box's receive buffer. All further data are lost if the receive buffer is full.

A buffer is also available for the send data. With a baud rate of 300 and a data format of 8N1, the box can only transmit 30 bytes per second. However, if more than 30 byte come in per second, the send buffer is written to first in this case also. Once this is full, all further data will be lost.

4 Mounting and connection

4.1 Mounting

4.1.1 Dimensions



All dimensions are given in millimeters.
The drawing is not true to scale.

Housing features

Housing material	PA6 (polyamide)
Sealing compound	polyurethane
Mounting	two fastening holes Ø 3.5 mm for M3
Metal parts	brass, nickel-plated
Contacts	CuZn, gold-plated
Power feed through	max. 4 A
Installation position	variable
Protection class	IP65, IP66, IP67 (conforms to EN 60529) when screwed together
Dimensions (H x W x D)	approx. 126 x 30 x 26.5 mm (without connectors)

4.1.2 Fixing

NOTE

Dirt during assembly

Dirty connectors can lead to malfunctions. Protection class IP67 can only be guaranteed if all cables and connectors are connected.

- Protect the plug connectors against dirt during the assembly.

Mount the module with two M3 screws on the fastening holes in the corners of the module. The fastening holes have no thread.

4.1.3 Tightening torques for plug connectors

Screw connectors tight with a torque wrench. (e.g. ZB8801 from Beckhoff)

Connector diameter	Tightening torque
M8	0.4 Nm
M12	0.6 Nm

4.2 Supply voltages

The EtherCAT Box is supplied with two supply voltages. The supply voltages are electrically isolated in the EtherCAT Box.

- Control voltage U_S
- Peripheral voltage U_P

Redirection of the supply voltages

The IN and OUT power connections are bridged in the module (not IP204x-Bxxx and IE204x). The supply voltages U_S and U_P can thus easily be transferred from EtherCAT Box to EtherCAT Box.

NOTE

Pay attention to the maximum permissible current!

Pay attention also for the redirection of the supply voltages U_S and U_P, the maximum permissible current for M8 connectors of 4 A must not be exceeded!

4.2.1 Connectors

NOTE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

- Observe the color coding of the connectors:
  black: Supply voltages
  green: EtherCAT

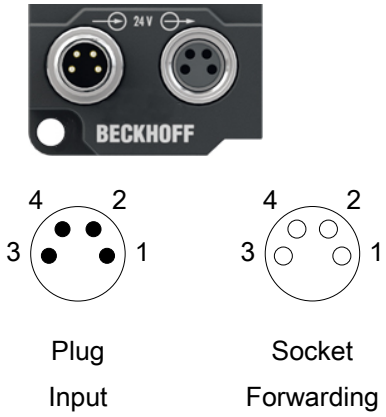


Fig. 4: M8 connector

Contact	Function	Description	Core color [1)]
1	U_S	Control voltage	Brown
2	U_P	Peripheral voltage	White
3	GND_S	GND to U_S	Blue
4	GND_P	GND to U_P	Black

[1)] The core colors apply to cables of the type: Beckhoff ZK2020-3xxx-xxxx

4.2.2 Status LEDs

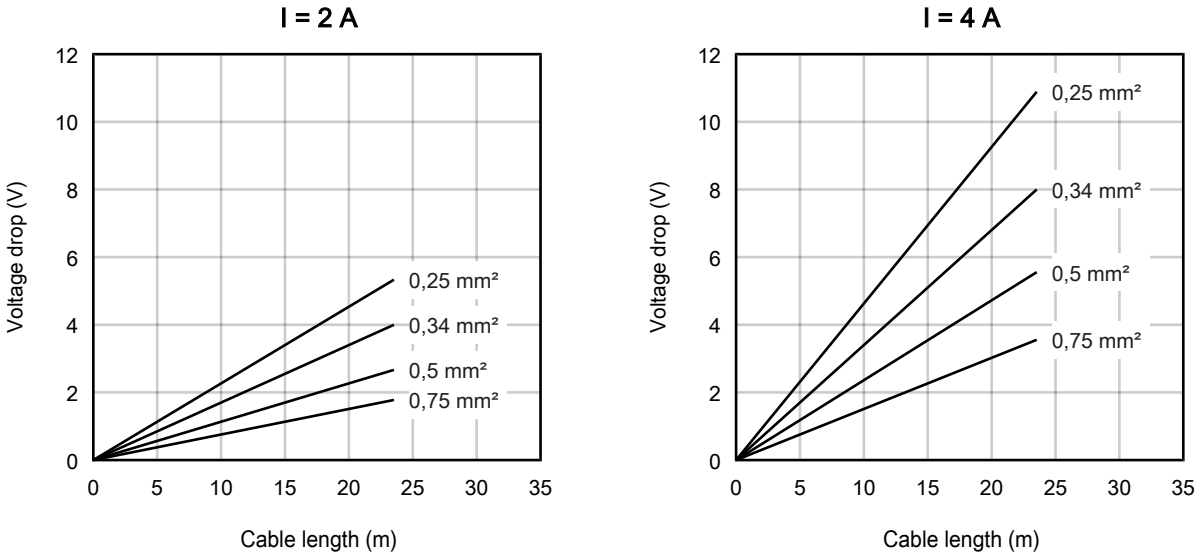
LED	Display	Meaning
U_S (control voltage)	off	Supply voltage U_S is not present
	green illuminated	Supply voltage U_S is present
U_P (peripheral voltage)	off	Supply voltage U_P is not present
	green illuminated	Supply voltage U_P is present

4.2.3 Conductor losses

Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage.

Variations in the voltage of the power supply unit must also be taken into account.

Voltage drop on the supply line



4.3 EtherCAT

4.3.1 Connectors

NOTE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

- Observe the color coding of the connectors:
  black: Supply voltages
  green: EtherCAT

EtherCAT Box Modules have two green M8 sockets for the incoming and downstream EtherCAT connections.

Connection

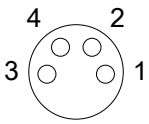


Fig. 5: M8 socket

EtherCAT	M8 connector	Core colors		
Signal	**Contact**	**ZB9010, ZB9020, ZB9030, ZB9032, ZK1090-6292, ZK1090-3xxx-xxxx**	**ZB9031 and old versions of ZB9030, ZB9032, ZK1090-3xxx-xxxx**	**TIA-568B**
Tx +	1	yellow[1)]	orange/white	white/orange
Tx -	4	orange[1)]	orange	orange
Rx +	2	white[1)]	blue/white	white/green
Rx -	3	blue[1)]	blue	green
Shield	Housing	Shield	Shield	Shield

[1)] Core colors according to EN 61918

Adaptation of core colors for cables ZB9030, ZB9032 and ZK1090-3xxxx-xxxx

For standardization, the core colors of the ZB9030, ZB9032 and ZK1090-3xxx-xxxx cables have been changed to the EN61918 core colors: yellow, orange, white, blue. So there are different color codes in circulation. The electrical properties of the cables have been retained when the core colors were changed.

4.3.2 Status LEDs

L/A (Link/Act)

A green LED labelled "L/A" is located next to each EtherCAT socket. The LED indicates the communication state of the respective socket:

LED	Meaning
off	no connection to the connected EtherCAT device
lit	LINK: connection to the connected EtherCAT device
flashes	ACT: communication with the connected EtherCAT device

Run

Each EtherCAT slave has a green LED labelled "Run". The LED signals the status of the slave in the EtherCAT network:

LED	Meaning
off	Slave is in "Init" state
flashes uniformly	Slave is in "Pre-Operational“ state
flashes sporadically	Slave is in "Safe-Operational" state
lit	Slave is in "Operational" state

Description of the EtherCAT slave states

4.3.3 Cables

For connecting EtherCAT devices only shielded Ethernet cables that meet the requirements of at least category 5 (CAT5) according to EN 50173 or ISO/IEC 11801 should be used.

EtherCAT uses four wires for signal transmission.
Thanks to automatic line detection ("Auto MDI-X"), both symmetrical (1:1) or cross-over cables can be used between Beckhoff EtherCAT.

Detailed recommendations for the cabling of EtherCAT devices

4.4 RS232

4.4.1 Connector

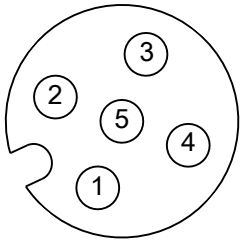


Fig. 6: M12 socket

M12 socket X01

Pin	Wire color	Signal	Description
1	brown	5 V_{DC}	End device supply voltage
2	white	TxD	send data
3	blue	GND	Ground
4	black	RxD	receive data
5	grey	Shield	Shield

4.4.2 Status LEDs

Fig. 7: RS232 Status LEDs

LED	Display	Meaning
R left	green illuminated	The serial port is ready to receive data.
	orange illuminated	The serial port is receiving data.
T right	green illuminated	The serial port is ready to transmit data.
	orange illuminated	The serial port is transmitting data.

4.5 RS422

4.5.1 Connectors

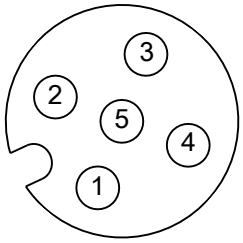


Fig. 8: M12 socket

M12 socket X02

Pin	Wire color	Signal	Description
1	brown	Tx -	send data
2	white	Tx +	send data
3	blue	GND	Ground
4	black	Vcc	End device supply voltage
5	grey	Shield	Shield

M12 socket X03

Pin	Wire color	Signal	Description
1	brown	Rx -	receive data
2	white	Rx +	receive data
3	blue	GND	Ground
4	black	Vcc	End device supply voltage
5	grey	Shield	Shield

4.5.2 Status LEDs

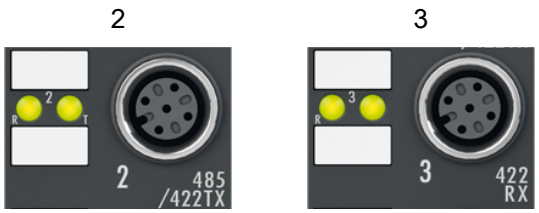


Fig. 9: RS422 Status LEDs

LED	Display	Meaning
R left	green illuminated	The serial port is ready to receive data.
	orange illuminated	The serial port is receiving data.
T right	green illuminated	The serial port is ready to transmit data.
	orange illuminated	The serial port is transmitting data.

4.6 RS485

4.6.1 Connector

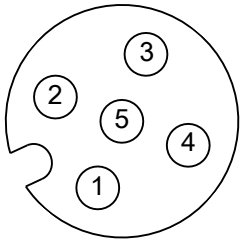


Fig. 10: M12 socket

M12 socket X02

Pin	Wire color	Signal	Description
1	brown	- / A	Inverted data line
2	white	+ / B	Non-inverted data line
3	blue	GND	Ground
4	black	Vcc	End device supply voltage
5	grey	Shield	Shield

The signal designations "A" and "B" are not clear

With some devices the designations of the data lines "A" and "B" are swapped.

RS485 bus structure - use termination resistors

A linear bus with more than two devices can be set-up in RS485 mode. To prevent reflections during the data transmission, it is necessary to terminate the line ends of the bus cable with resistors (120 Ω).

4.6.2 Status LEDs

Fig. 11: RS485 Status LEDs

LED	Display	Meaning
R left	green illuminated	The serial port is ready to receive data.
	orange illuminated	The serial port is receiving data.
T right	green illuminated	The serial port is ready to transmit data.
	orange illuminated	The serial port is transmitting data.

4.7 Digital inputs/outputs

4.7.1 Connector

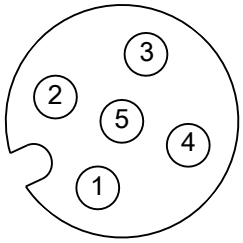


Fig. 12: M12 socket

M12 socket X04

Pin	Wire color	Signal	Description
1	brown	24 V_{DC}	Supply voltage
2	white	Input/output B	Digital input/output B
3	blue	GND	Ground
4	black	Input/output A	Digital input/output A
5	grey	reserved	-

4.7.2 Status LEDs

Fig. 13: Status LEDs for digital inputs/outputs

LED	Display	Meaning
A	off	Digital input/output A: Low level
left	green illuminated	Digital input/output A: High level
B	off	Digital input/output B: Low level
right	green illuminated	Digital input/output B: High level

4.7.3 Examples of external connections

M12 socket "4"

The digital input modules acquire the binary control signals from the process level and transmit them to the higher-level automation device.

The signals are connected via M8 connectors (EPxxxx-0001) or M12 connectors (EPxxxx-0002).

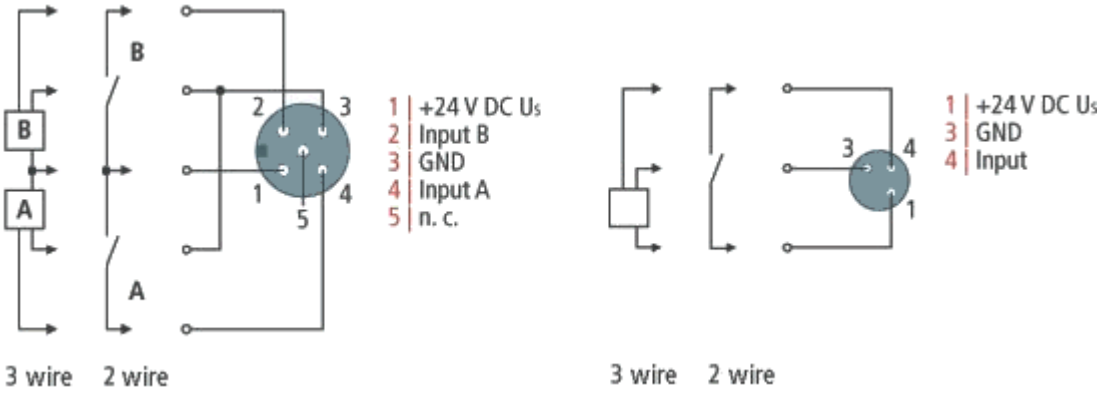


Fig. 14: Digital inputs M8 and M12

The sensors are supplied with a common maximum current of 0.5 A from the control voltage Us.

Light emitting diodes indicate the signal state of the inputs.

Digital outputs M8 and M12

The digital output modules forward the binary control signals of the automation device to the actuators at the process level.

The signals are connected via M8 connectors (EP2xxx-0001) or M12 connectors (EP2xxx-0002).

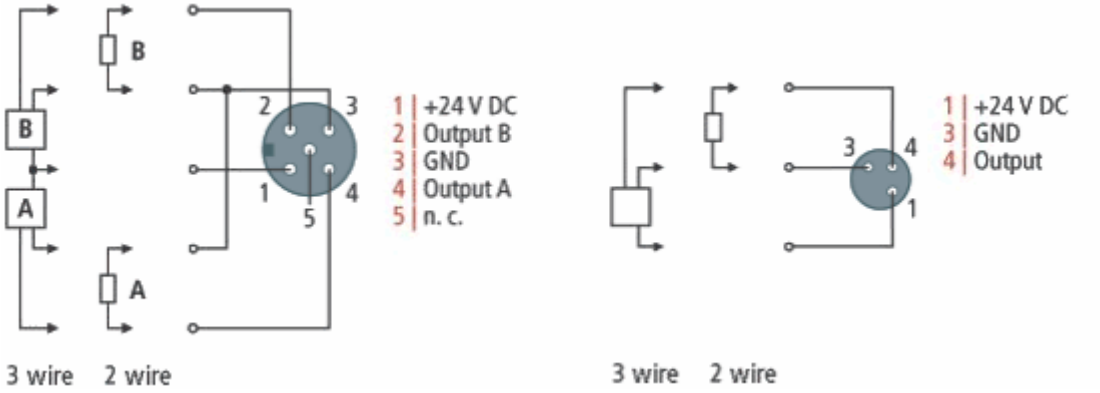


Fig. 15: Digital outputs M8 and M12

The outputs are short-circuit proof and protected against inverse polarity.

LEDs indicate the signal state of the outputs.

4.8 UL Requirements

The installation of the EtherCAT Box Modules certified by UL has to meet the following requirements.

Supply voltage

⚠ CAUTION

CAUTION!

This UL requirements are valid for all supply voltages of all marked EtherCAT Box Modules!
For the compliance of the UL requirements the EtherCAT Box Modules should only be supplied

- by a 24 V_{DC} supply voltage, supplied by an isolating source and protected by means of a fuse (in accordance with UL248), rated maximum 4 Amp, or
- by a 24 V_{DC} power source, that has to satisfy *NEC class 2*.
  A *NEC class 2* power supply shall not be connected in series or parallel with another (class 2) power source!

⚠ CAUTION

CAUTION!

To meet the UL requirements, the EtherCAT Box Modules must not be connected to unlimited power sources!

Networks

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules must not be connected to telecommunication networks!

Ambient temperature range

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules has to be operated only at an ambient temperature range of -25 °C to +55 °C!

Marking for UL

All EtherCAT Box Modules certified by UL (Underwriters Laboratories) are marked with the following label.

Fig. 16: UL label

5 Commissioning/Configuration

5.1 Integration in TwinCAT

The procedure for integration in TwinCAT is described in this Quick start guide.

5.2 Serial interface

5.2.1 Setting the interface type

Via CoE objects the following settings can be done for the interfaces:

Parameterization

The module is parameterized via the "CoE online" tab (with a double-click on the corresponding object). Only the mandatory parameters for the respective interface mode are specified here. Further settings may be possible.

5.2.1.1 RS232

RS232: point-to-point connection to an RS232 device

Direct connection to an RS232 end device, full duplex data transmission (default setting).

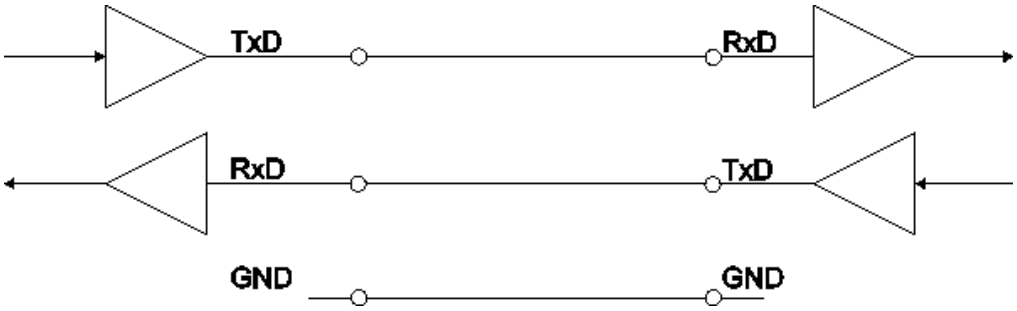


Fig. 17: Point-to-point connection to an RS232 device

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:01	Interface Type Ch 1	0x00	RS232	BIT1	RW	0x00 (0_{dec}) (default)
		0x01	RS485/422			

5.2.1.2 RS422

RS422: 4-wire point-to-point connection to an RS422 device

Direct connection to an RS422 end device, full duplex data transmission.
Data can be transmitted in full duplex in RS422 mode. Only point-to-point connections can be established.

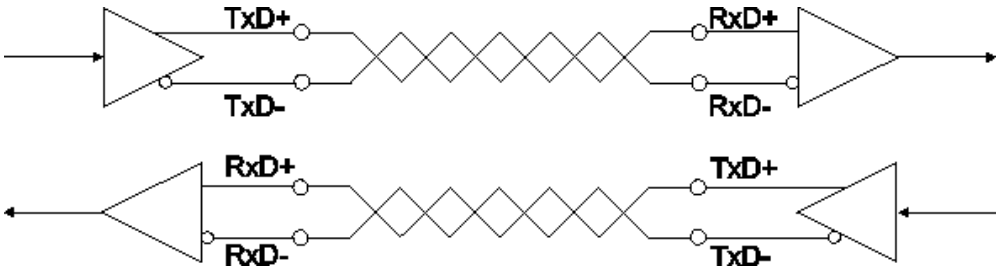


Fig. 18: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:01	Interface type Ch 1	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
8000:07	Enable point-to-point connection (RS422)	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	1_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

5.2.1.3 RS485

You can operate the RS845 communication in two variants:

- Without diagnosis
- With diagnosis of the transmit data [▶ 33]

RS485: 2-wire connection in bus structure to RS485 device(s)

Bus structure, half duplex data transmission

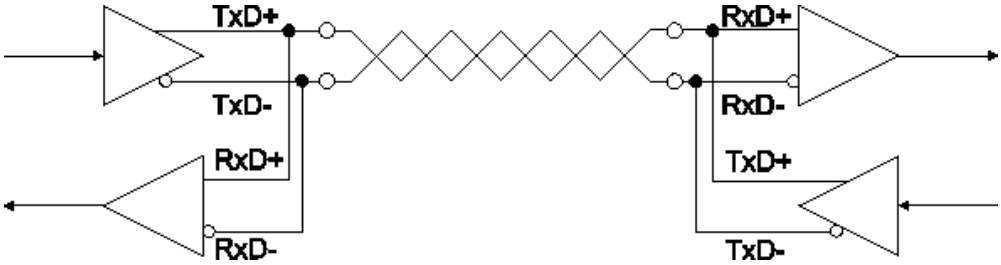


Fig. 19: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:01	Interface type Ch 1	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
8000:06	Enable half duplex	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	1_{bin} (default)
		1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
8000:07	Enable point-to-point connection (RS422)	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

Deactivated receive driver

The receive driver is deactivated during the transmission procedure. The transmitted data are not monitored!

RS485: 2-wire connection with external bridge in bus structure to RS485 device(s)

Bus structure, half duplex data transmission with diagnosis of the transmitted data

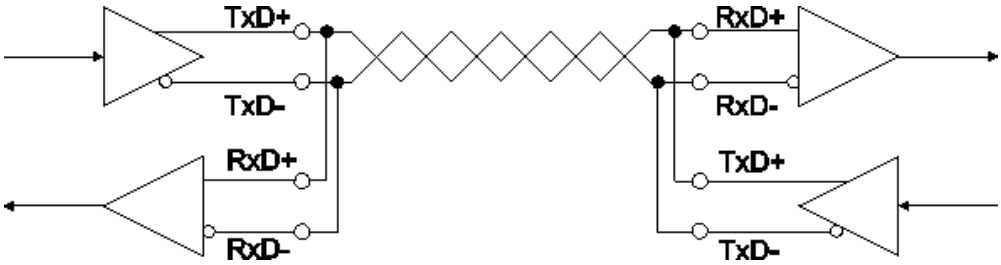


Fig. 20: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

Index	Name	Meaning		Data type	Flags	Setting
F800:01	Interface type Ch 1	0x00	RS232	BIT1	RW	0x01 (1_{dec})
		0x01	RS485/422			

Index	Name	Meaning		Data type	Flags	Setting
8000:06	Enable half duplex	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	0_{bin}
		1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
8000:07	Enable point-to-point connection (RS422)	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

Activated receive driver (from firmware version 03)

The receive driver remains activated during the transmission procedure. The transmitted data are monitored! A conditional diagnosis of the line is thus possible. If there is a discrepancy between the transmitted data and the monitored data, it may be assumed that a further receiver also cannot receive these data flawlessly. In this case, check the bus line!

5.2.2 Setting the interface parameters

The parameterization of the module can be set in the CoE (CAN over EtherCAT) list.

Parameterization via the CoE list (CAN over EtherCAT)

Please note the following general CoE notes when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - Use "CoE reload" for resetting changes

The following CoE settings are possible from object 0x8000 and are shown below in their default settings:

Index	Name	Flags	Value
8000:0	COM Settings Ch.1	RW	> 27 <
8000:02	Enable XON/XOFF supported tx data	RW	FALSE
8000:03	Enable XON/XOFF supported rx data	RW	FALSE
8000:04	Enable send FIFO data continuous	RW	FALSE
8000:05	Enable transfer rate optimization	RW	TRUE
8000:06	Enable half duplex	RW	TRUE
8000:07	Enable point to point connection (RS422)	RW	FALSE
8000:11	Baudrate	RW	9600 Baud (6)
8000:15	Data frame	RW	8N1 (3)
8000:19	Sensor Power Supply Output	RW	5V (1)
8000:1A	Rx buffer full notification	RW	0x0360 (864)
8000:1B	Explicit Baudrate	RW	0x00002580 (9600)

Fig. 21: CoE settings on object 0x8000 (default)

Continuous transmission of data

A continuous data stream is indispensable for many applications. For this purpose, the Beckhoff modules feature the "Enable send FIFO data continuous" setting in the Settings object. The internal transmit buffer of the box can be filled first by setting this switch. After that the entire contents of the buffer can be transmitted without interruption. To this end, data will be sent from the controller to the box as in a normal transmission. The data from the buffer is only sent with a rising edge of the "Send continuous" bit. If the data has been transferred, the box informs the controller by setting the "Init accepted" bit. "Init accepted" is cleared with "SendContinuous".

Optimization of transfer rates

In normal operating mode the data received will be adopted immediately into the process image. In order to enable a contiguous data stream, the "Enable transfer rate optimization" option in the Settings object is activated by default. Due to this switch, the data will first be stored intermediately in the receive buffer (864 bytes).

The data will only be copied into the process image if no further character is received for 16 bit times or if the buffer is full.

5.2.3 Communication by PLC program

Initialization

Initialization is performed prior to the first transmission/reception. The module is thereby parameterized with the data from the corresponding Settings object.

Procedure:

1. Set "Init request" to 1
   ⇨ The module confirms successful initialization by setting "Init accepted".
2. Reset "Init request"
   ⇨ The module sets "Init accepted" to 0.

⇨ The module is ready for data exchange.

Sending data

1. Write the data to be sent in the output variables Data Out [n] [▶ 14].
2. Set the *Output Length* parameter in the Control word to the number of bytes to be transmitted.
3. Toggle the *Transmit Request* bit in the Control word.

⇨ The module acknowledges the data transmission in the Status word via the *Transmit Accepted* parameter.

Receiving data

If the module in the Status word toggles the *Receive Request* bit, there are new receive data in the process data.

1. Read the *Input Length* parameter from the Status word. It contains the number of bytes to be received.
   ⇨ The data are located in the input variable Data In [n]. The first-received data is located in Data In 0.
2. After reading the data, acknowledge this by toggling the *Receive Accepted* bit in the Control word. Only after that does the module transfer new data from the receive buffer to the process data.

Prioritization

Since received data normally cannot be repeated from the other transmitter, they have a higher priority in the module than data to be transmitted.
Furthermore, the priority decreases as the channel number increases. Hence, the reception of data on channel 1 has the highest priority.

5.2.3.1 Samples

Data transmission from the controller to the module (send 2 characters)

1. Set "Output length" to 2
2. Fill "Data Out 0" and "Data Out 1" with user data
3. Change the state of "Transmit request"
   ⇨ The module acknowledges receipt by changing the state of the "Transmit accepted" bit.

Data transmission from the module to the controller (receive characters)

1. The module indicates that there is new data in the process image by changing the state of the "Receive request" bit.
2. The number of bytes received is written in "Input length"
3. The controller acknowledges acceptance of the bytes by changing the state of “Receive request”.

5.2.4 Communication via a virtual COM port

Application Note DK9322-0411-0041 describes the communication via a virtual COM port, taking the EP6002-0002 as an example.

5.3 Digital inputs/outputs

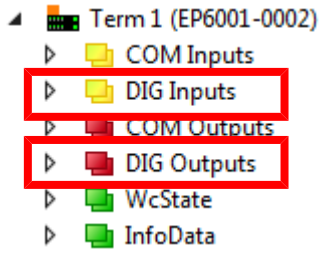
5.3.1 Activating process data

In order to be able to use the digital inputs/outputs, you have to activate the input and output variables in the process image:

✓ Requirement: An EP6001-0002 has been added in the Solution Explorer under the "I/O" entry.

1. Double-click on the EP6001-0002 IO module.
2. Click on the **Process Data** tab.
3. Click on the "Outputs" entry in the **Sync Manager** field.
4. Activate the **PDO Assignment (0x1C12)** checkbox next to the entry "0x1608".
   ⇨ The process data object "DIG Outputs", appears in the process data.
5. Click on the "Inputs" entry in the **Sync Manager** field.
6. Activate the **PDO Assignment (0x1C13)** checkbox next to the entry "0x1A08".
   ⇨ The process data object "DIG Inputs", appears in the process data.

⇨ Result: The process data objects for the digital inputs/outputs are activated:



5.3.2 Assignment of connector pins to process data

Connector	Pin	Channel designation	Input variable	Output variable
X04 "DIO"	2	B	DIG Inputs Digital Input 2	DIG Outputs Digital Output 2
	4	A	DIG Inputs Digital Input 1	DIG Outputs Digital Output 1

5.4 CoE objects

5.4.1 Directory

Index (hex)	Name
1000	Device type [▶ 42]
1008	Device name [▶ 42]
1009	Hardware version [▶ 42]
100A	Software version [▶ 42]
1011	Restore default parameters [▶ 40]
1018	Identity [▶ 42]
10F0	Backup parameter handling [▶ 42]
1400	COM RxPDO-Par Outputs [▶ 42]
1404	COM RxPDO-Par Outputs [▶ 42]
1600	COM RxPDO-Map Outputs [▶ 43]
1604	COM RxPDO-Map Outputs [▶ 44]
1608	DIG RxPDO-Map Outputs [▶ 44]
1800	COM TxPDO-Par Inputs [▶ 45]
1804	COM TxPDO-Par Inputs [▶ 45]
1A00	COM TxPDO-Map Inputs [▶ 46]
1A04	COM TxPDO-Map Inputs [▶ 47]
1A08	DIG TxPDO-Map Inputs [▶ 47]
1C00	Sync manager type [▶ 48]
1C12	RxPDO assign [▶ 48]
1C13	TxPDO assign [▶ 48]
1C32	SM output parameter [▶ 49]
1C33	SM input parameter [▶ 50]
6000	COM Inputs Ch. 1 [▶ 51]
6001	Status Ch. 1 [▶ 51]
6010	DIG Inputs [▶ 51]
7000	COM Outputs Ch. 1 [▶ 52]
7001	Ctrl Ch. 1 [▶ 52]
7010	DIG Outputs [▶ 52]
8000	COM Settings Ch. 1 [▶ 41]
A000	COM Diag data Ch. 1 [▶ 53]
F000	Modular device profile [▶ 53]
F008	Code word [▶ 53]
F010	Module list [▶ 53]
F800	COM Settings [▶ 41]

5.4.2 Object description and parameterization

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▶ 40] during commissioning
- Objects required for the selection of the interface type [▶ 41]
- Objects intended for regular operation, e.g. through ADS access
- Objects for indicating internal settings [▶ 41] (may be fixed)
- Further profile-specific objects [▶ 50] indicating inputs, outputs and status information

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

Objects to be parameterized during commissioning

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default settings	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

Index 8000 COM Settings Ch.1

Index (hex)	Name	Meaning		Data type	Flags	Default
8000:0	COM Settings Ch.1			UINT8	RO	0x1A (26_{dec})
8000:02	Enable XON/XOFF supported tx data	XON/XOFF is supported for send data		BOOLEAN	RW	0x00 (0_{dec})
8000:03	Enable XON/XOFF supported rx data	XON/XOFF is supported for receive data		BOOLEAN	RW	0x00 (0_{dec})
8000:04	Enable send FIFO data continuous	Continuous sending of data from the FIFO enabled		BOOLEAN	RW	0x00 (0_{dec})
8000:05	Enable transfer rate optimization	Switch on the transfer rate optimization		BOOLEAN	RW	0x01 (1_{dec})
8000:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0_{dec})
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8000:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0_{dec})
		1	The module is used as a point-to-point connection (RS422)			
8000:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6_{dec})
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8000:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3_{dec})
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8000:19	Sensor Power Supply Output	0: Off 1: 5 V		BIT4	RW	5 V (1)
8000:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864_{dec})
8000:1B	Explicit baudrate	In this object the desired baud rate can be entered directly as a number.		UINT32	RW	0x00002580 (9600_{dec})

Index F800 COM Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
F800:0	COM Settings			UINT8	RO	0x01 (1_{dec})
F800:01	Interface Type Ch 1	0x00	RS232	BIT1	RW	0x00 (0_{dec})

Additional objects

Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EP6001-0002

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software Version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x17714052 (393298002_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00100002 (1048578_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1400 COM RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1400:0	COM RxPDO-Par Outputs	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[2]	RO	04 16

Index 1404 COM RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	COM RxPDO-Par Outputs	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[2]	RO	00 16

Index 1600 COM RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x1C (28_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1600:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1600:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1600:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1600:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1600:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1600:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1600:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1600:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1600:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1600:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1600:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1600:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1600:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1600:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1600:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1600:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1600:18	SubIndex 024	24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1600:19	SubIndex 025	25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1600:1A	SubIndex 026	26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1600:1B	SubIndex 027	27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1600:1C	SubIndex 028	28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1604 COM RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 2	UINT8	RO	0x17 (23_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl))	UINT32	RO	0x7001:01, 16
1604:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0A	SubIndex 010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:0C	SubIndex 012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:0E	SubIndex 014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:0F	SubIndex 015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:10	SubIndex 016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:11	SubIndex 017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:12	SubIndex 018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:13	SubIndex 019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:14	SubIndex 020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:15	SubIndex 021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:16	SubIndex 022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:17	SubIndex 023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

Index 1608 DIG RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	DIG RxPDO-Map Outputs	PDO Mapping RxPDO 3	UINT8	RO	0x03 (3_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x01 (Digital Output 1))	UINT32	RO	0x7001:01, 1
1608:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (DIG Outputs), entry 0x02 (Digital Output 2))	UINT32	RO	0x7010:02, 1
1608:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1800 COM TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	COM TxPDO-Par Inputs	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[2]	RO	04 1A

Index 1804 COM TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[2]	RO	00 1A

Index 1A00 COM TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x1F (31_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A00:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A00:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A00:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A00:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A00:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A00:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A00:18	SubIndex 024	24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A00:19	SubIndex 025	25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A00:1A	SubIndex 026	26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A00:1B	SubIndex 027	27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A00:1C	SubIndex 028	28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A00:1D	SubIndex 029	29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A00:1E	SubIndex 030	30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A00:1F	SubIndex 031	31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A04 COM TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 2	UINT8	RO	0x17 (23_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status))	UINT32	RO	0x6001:01, 16
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:12	SubIndex 018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:13	SubIndex 019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:14	SubIndex 020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:15	SubIndex 021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:16	SubIndex 022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:17	SubIndex 023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 1A08 DIG TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 6	UINT8	RO	0x03 (3_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x01 (Digital Input 1))	UINT32	RO	0x6010:01, 1
1A08:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (DIG Inputs), entry 0x02 (Digital Input 2))	UINT32	RO	0x6010:02, 1
1A08:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4_{dec})
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})
1C12:02	Subindex 002	2. reserved	UINT16	RW	--

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A04 (6660_{dec})
1C13:02	Subindex 002	2. reserved	UINT16	RW	--

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0003D090 (250000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 49])	UINT16	RO	0x0001 (1_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03 [▶ 49], 1C32:05 [▶ 49], 1C32:06 [▶ 49], 1C32:09 [▶ 49], 1C33:03 [▶ 50], 1C33:06 [▶ 49], 1C33:09 [▶ 50] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 1C32:02 [▶ 49]	UINT32	RW	0x0003D090 (250000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 49] or 1C33:08 [▶ 50])	UINT16	RO	0x0001 (1_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 49]	UINT32	RO	0x0003D090 (250000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C33:08	Command	as 1C32:08 [▶ 49]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:0B	SM event missed counter	as 1C32:11 [▶ 49]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 49]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 49]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 49]	BOOLEAN	RO	0x00 (0_{dec})

Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 COM Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	COM Inputs Ch.1		UINT8	RO	0x26 (38_{dec})
6000:01	Transmit accepted	The module acknowledges receipt of data by changing the state of this bit	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Receive request	By changing the state of this bit, the module informs the controller that the DataIn bytes contain the number of bytes displayed in "Input length"	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Buffer full	The receive FIFO is full	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Parity error	A parity error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:06	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Overrun error	An overrun error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Input length	Number of input bytes available for transfer from the terminal to the controller	UINT8	RO	0x00 (0_{dec})
6000:11	Data In 0	Input data	UINT8	RO	0x00 (0_{dec})
6000:12	Data In 1	Input data	UINT8	RO	0x00 (0_{dec})
6000:13	Data In 2	Input data	UINT8	RO	0x00 (0_{dec})
6000:14	Data In 3	Input data	UINT8	RO	0x00 (0_{dec})
6000:15	Data In 4	Input data	UINT8	RO	0x00 (0_{dec})
6000:16	Data In 5	Input data	UINT8	RO	0x00 (0_{dec})
6000:17	Data In 6	Input data	UINT8	RO	0x00 (0_{dec})
6000:18	Data In 7	Input data	UINT8	RO	0x00 (0_{dec})
6000:19	Data In 8	Input data	UINT8	RO	0x00 (0_{dec})
6000:1A	Data In 9	Input data	UINT8	RO	0x00 (0_{dec})
6000:1B	Data In 10	Input data	UINT8	RO	0x00 (0_{dec})
6000:1C	Data In 11	Input data	UINT8	RO	0x00 (0_{dec})
6000:1D	Data In 12	Input data	UINT8	RO	0x00 (0_{dec})
6000:1E	Data In 13	Input data	UINT8	RO	0x00 (0_{dec})
6000:1F	Data In 14	Input data	UINT8	RO	0x00 (0_{dec})
6000:20	Data In 15	Input data	UINT8	RO	0x00 (0_{dec})
6000:21	Data In 16	Input data	UINT8	RO	0x00 (0_{dec})
6000:22	Data In 17	Input data	UINT8	RO	0x00 (0_{dec})
6000:23	Data In 18	Input data	UINT8	RO	0x00 (0_{dec})
6000:24	Data In 19	Input data	UINT8	RO	0x00 (0_{dec})
6000:25	Data In 20	Input data	UINT8	RO	0x00 (0_{dec})
6000:26	Data In 21	Input data	UINT8	RO	0x00 (0_{dec})

Index 6001 Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	Status Ch.1		UINT8	RO	0x01 (1_{dec})
6001:01	Status	Status word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 6010 DIG Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	DIG Inputs		UINT8	RO	0x02 (2_{dec})
6010:01	Digital Input 1		BOOLEAN	RO	0x00 (0_{dec})
6010:02	Digital Input 1		BOOLEAN	RO	0x00 (0_{dec})
6010:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})

Index 7000 COM Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	COM Outputs Ch.1		UINT8	RO	0x26 (38_{dec})
7000:01	Transmit request	By changing the state of this bit, the controller informs the terminal that the DataOut bytes contain the number of bytes displayed in "Output length".	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Init request	The controller requests the module to initialize.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Send continuous	Continuous sending of data from the FIFO.	BOOLEAN	RO	0x00 (0_{dec})
7000:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
7000:11	Data Out 0	Output data	UINT8	RO	0x00 (0_{dec})
7000:12	Data Out 1	Output data	UINT8	RO	0x00 (0_{dec})
7000:13	Data Out 2	Output data	UINT8	RO	0x00 (0_{dec})
7000:14	Data Out 3	Output data	UINT8	RO	0x00 (0_{dec})
7000:15	Data Out 4	Output data	UINT8	RO	0x00 (0_{dec})
7000:16	Data Out 5	Output data	UINT8	RO	0x00 (0_{dec})
7000:17	Data Out 6	Output data	UINT8	RO	0x00 (0_{dec})
7000:18	Data Out 7	Output data	UINT8	RO	0x00 (0_{dec})
7000:19	Data Out 8	Output data	UINT8	RO	0x00 (0_{dec})
7000:1A	Data Out 9	Output data	UINT8	RO	0x00 (0_{dec})
7000:1B	Data Out 10	Output data	UINT8	RO	0x00 (0_{dec})
7000:1C	Data Out 11	Output data	UINT8	RO	0x00 (0_{dec})
7000:1D	Data Out 12	Output data	UINT8	RO	0x00 (0_{dec})
7000:1E	Data Out 13	Output data	UINT8	RO	0x00 (0_{dec})
7000:1F	Data Out 14	Output data	UINT8	RO	0x00 (0_{dec})
7000:20	Data Out 15	Output data	UINT8	RO	0x00 (0_{dec})
7000:21	Data Out 16	Output data	UINT8	RO	0x00 (0_{dec})
7000:22	Data Out 17	Output data	UINT8	RO	0x00 (0_{dec})
7000:23	Data Out 18	Output data	UINT8	RO	0x00 (0_{dec})
7000:24	Data Out 19	Output data	UINT8	RO	0x00 (0_{dec})
7000:25	Data Out 20	Output data	UINT8	RO	0x00 (0_{dec})
7000:26	Data Out 21	Output data	UINT8	RO	0x00 (0_{dec})

Index 7001 Ctrl Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	Ctrl Ch.1		UINT8	RO	0x01 (1_{dec})
7001:01	Ctrl	Control word for compatible process image	UINT16	RO	0x0000 (0_{dec})

Index 7010 DIG Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	DIG Outputs		UINT8	RO	0x26 (38_{dec})
7010:01	Digital Output 1		BOOLEAN	RO	0x00 (0_{dec})
7010:02	Digital Output 2		BOOLEAN	RO	0x00 (0_{dec})

Index A000 COM Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	COM Diag data Ch.1		UINT8	RO	0x21 (33_{dec})
A000:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A000:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A000:21	Data bytes in receive buffer	Number of data bytes in the receive FIFO	UINT16	RO	0x0000 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0002 (2_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word		UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list		UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 001		UINT32	RW	0x00000258 (600_{dec})
F010:02	SubIndex 002		UINT32	RW	0x00000118 (280_{dec})

5.5 Restoring the delivery state

To restore the delivery state for backup objects in ELxxxx terminals / EPxxxx- and EPPxxxx boxes, the CoE object *Restore default parameters, SubIndex 001* can be selected in the TwinCAT System Manager (Config mode).

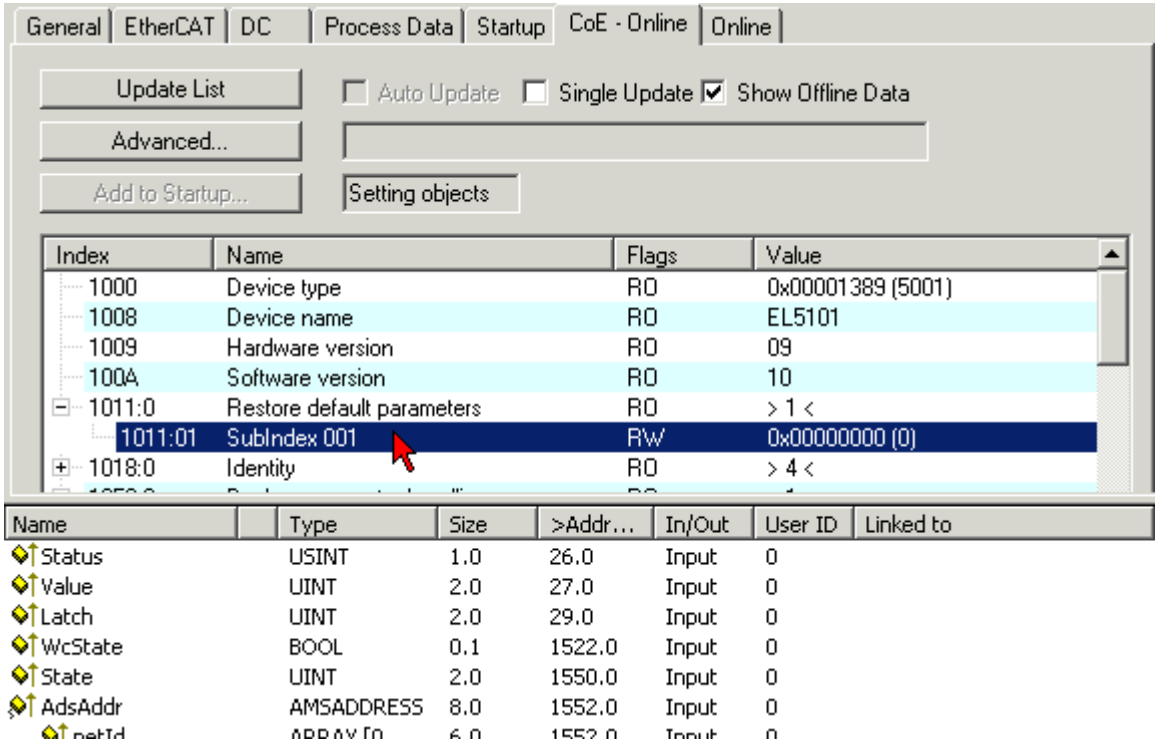


Fig. 22: Selecting the Restore default parameters PDO

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* and confirm with OK.

All backup objects are reset to the delivery state.

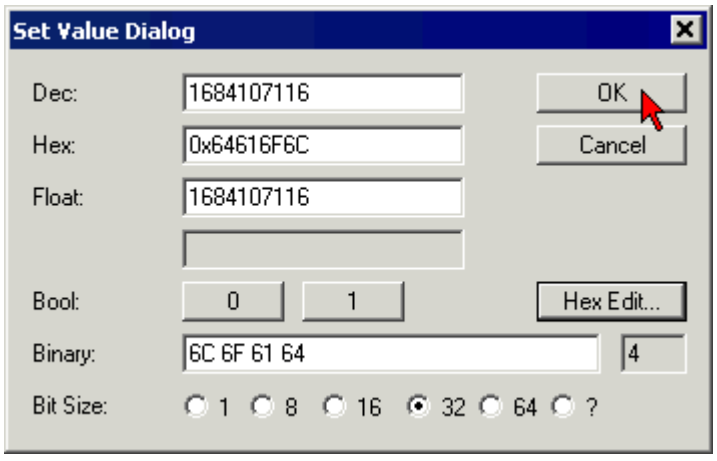


Fig. 23: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals / boxes the backup objects can be switched with an alternative restore value:
Decimal value: 1819238756
Hexadecimal value: 0x6C6F6164

An incorrect entry for the restore value has no effect.

5.6 Decommissioning

⚠ WARNING

Risk of electric shock!

Bring the bus system into a safe, de-energized state before starting disassembly of the devices!

Disposal

In order to dispose of the device, it must be removed.

In accordance with the WEEE Directive 2012/19/EU, Beckhoff takes back old devices and accessories in Germany for proper disposal. Transport costs will be borne by the sender.

Return the old devices with the note "for disposal" to:

Beckhoff Automation GmbH & Co. KG
Service Department
Stahlstraße 31
D-33415 Verl

6 Appendix

6.1 General operating conditions

Protection degrees (IP-Code)

The standard IEC 60529 (DIN EN 60529) defines the degrees of protection in different classes.

1. Number: dust protection and touch guard	Definition
0	Non-protected
1	Protected against access to hazardous parts with the back of a hand. Protected against solid foreign objects of Ø 50 mm
2	Protected against access to hazardous parts with a finger. Protected against solid foreign objects of Ø 12.5 mm.
3	Protected against access to hazardous parts with a tool. Protected against solid foreign objects Ø 2.5 mm.
4	Protected against access to hazardous parts with a wire. Protected against solid foreign objects Ø 1 mm.
5	Protected against access to hazardous parts with a wire. Dust-protected. Intrusion of dust is not totally prevented, but dust shall not penetrate in a quantity to interfere with satisfactory operation of the device or to impair safety.
6	Protected against access to hazardous parts with a wire. Dust-tight. No intrusion of dust.

2. Number: water* protection	Definition
0	Non-protected
1	Protected against water drops
2	Protected against water drops when enclosure tilted up to 15°.
3	Protected against spraying water. Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects.
4	Protected against splashing water. Water splashed against the disclosure from any direction shall have no harmful effects
5	Protected against water jets
6	Protected against powerful water jets
7	Protected against the effects of temporary immersion in water. Intrusion of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water for 30 min. in 1 m depth.

*) These protection classes define only protection against water!

Chemical Resistance

The Resistance relates to the Housing of the IP 67 modules and the used metal parts. In the table below you will find some typical resistance.

Character	Resistance
Steam	at temperatures >100°C: not resistant
Sodium base liquor (ph-Value > 12)	at room temperature: resistant > 40°C: not resistant
Acetic acid	not resistant
Argon (technical clean)	resistant

Key

- resistant: Lifetime several months
- non inherently resistant: Lifetime several weeks
- not resistant: Lifetime several hours resp. early decomposition

6.2 Accessories

Mounting

Ordering information	Description
ZS5300-0011	Mounting rail

Cables

A complete overview of pre-assembled cables for fieldbus components can be found here.

Ordering information	Description	Link
ZK1090-3xxx-xxxx	EtherCAT cable M8, green	Website
ZK1093-3xxx-xxxx	EtherCAT cable M8, yellow	Website
ZK2000-6xxx-xxxx	Sensor cable M12, 4-pin	Website
ZK2000-7xxx-0xxx	Sensor cable M12, 4-pin + shield	Website
ZK2020-3xxx-xxxx	Power cable M8, 4-pin	Website

Labeling material, protective caps

Ordering information	Description
ZS5000-0010	Protective cap for M8 sockets, IP67 (50 pieces)
ZS5000-0020	Protective cap for M12 sockets, IP67 (50 pcs.)
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10
ZS5000-xxxx	Printed inscription labels on enquiry

Tools

Ordering information	Description
ZB8801-0000	Torque wrench for plugs, 0.4…1.0 Nm
ZB8801-0001	Torque cable key for M8 / wrench size 9 for ZB8801-0000
ZB8801-0002	Torque cable key for M12 / wrench size 13 for ZB8801-0000
ZB8801-0003	Torque cable key for M12 field assembly / wrench size 18 for ZB8801-0000

Further accessories

Further accessories can be found in the price list for fieldbus components from Beckhoff and online at https://www.beckhoff.com.

6.3 Version identification of EtherCAT devices

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

Example	Family	Type	Version	Revision
EL3314-0000-0016	EL terminal (12 mm, non-pluggable connection level)	3314 (4-channel thermocouple terminal)	0000 (basic type)	0016
ES3602-0010-0017	ES terminal (12 mm, pluggable connection level)	3602 (2-channel voltage measurement)	0010 (high-precision version)	0017
CU2008-0000-0000	CU device	2008 (8-port fast ethernet switch)	0000 (basic type)	0000

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification number

Beckhoff EtherCAT devices from the different lines have different kinds of identification numbers:

Production lot/batch number/serial number/date code/D number

The serial number for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**

KK - week of production (CW, calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Example with
Ser. no.: 12063A02: 12 - production week 12 06 - production year 2006 3A - firmware version 3A 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation
ww - calendar week
yy - year
x - firmware version of the bus PCB
y - hardware version of the bus PCB
z - firmware version of the I/O PCB
u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

Unique serial number/ID, ID number

In addition, in some series each individual module has its own unique serial number.

See also the further documentation in the area

- IP67: EtherCAT Box
- Safety: TwinSafe
- Terminals with factory calibration certificate and other measuring terminals

Examples of markings

Fig. 24: EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01)

Fig. 25: EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number

Fig. 26: CU2016 switch with serial/ batch number

Fig. 27: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

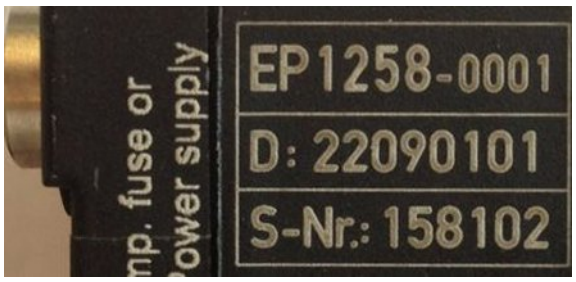


Fig. 28: EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

Fig. 29: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070

Fig. 30: EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701

Fig. 31: ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201

6.3.1 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

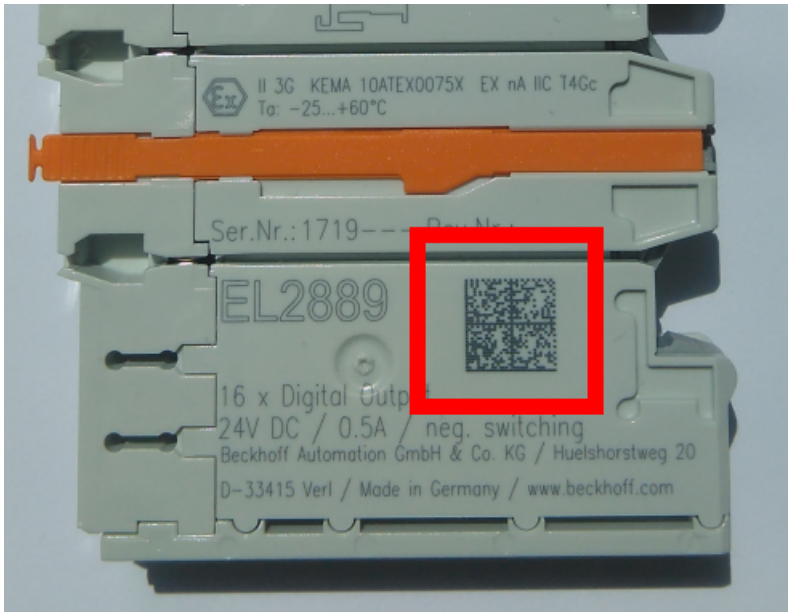


Fig. 32: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it. The data under positions 1 to 4 are always available.

The following information is contained:

Item no.	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	S	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products or calibrated terminals	51S	12	51S678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from item 1 to 4 and 6. The data identifiers are marked in red for better display:

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this information.

6.4 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: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

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

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
web:	https://www.beckhoff.com

More Information:
www.beckhoff.com/ep6001/

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com