ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

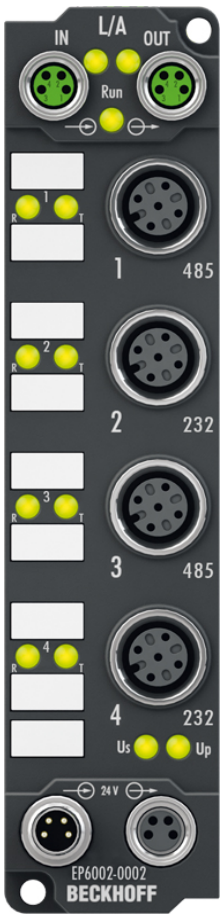
Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Beckhoff EP6002-0002* at our website: *Click HERE*



Documentation

EP6002-0002

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

Version: 1.0
Date: 2019-12-11

BECKHOFF

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 Safety instructions 6
1.3 Documentation Issue Status 7
2 Product overview 8
2.1 Product group: EtherCAT Box Modules 8
2.2 Introduction 9
2.3 Technical data 10
2.4 Scope of supply 11
2.5 Process image 12
2.5.1 Assignment of connectors to process data 13
2.5.2 Control word 14
2.5.3 Status word 14
2.6 Technology 15
3 Mounting and connection 16
3.1 Mounting 16
3.1.1 Dimensions 16
3.1.2 Fixing 17
3.2 Supply voltages 18
3.2.1 Connectors 18
3.2.2 Status LEDs 19
3.2.3 Conductor losses 19
3.3 EtherCAT 20
3.3.1 Connectors 20
3.3.2 Status LEDs 21
3.3.3 Cables 21
3.4 RS232 22
3.4.1 Connectors 22
3.4.2 Status LEDs 22
3.5 RS422 / RS485 23
3.5.1 Connectors 23
3.5.2 Status LEDs 23
3.6 UL Requirements 24
4 Commissioning/Configuration 25
4.1 Configuration in TwinCAT 25
4.2 Serial interfaces 26
4.2.1 Parameterization of the interface type 26
4.2.2 Setting the interface parameters 30
4.2.3 Communication by PLC program 31
4.2.4 Communication via a virtual COM port 33
4.3 CoE objects 34
4.3.1 Directory 34
4.3.2 Object description and parameterization 35

4.4 Restoring the delivery state 57
4.5 Decommissioning 58

5 Appendix 59
5.1 General operating conditions 59
5.2 EtherCAT Box- / EtherCAT P Box - Accessories 60
5.3 General note on the introduction of the Beckhoff Identification Code (BIC) 61
5.4 Support and Service 63

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 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

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

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	Modifications
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)	29 - week of production 29
YY - year of production	10 - year of production 2010
FF - firmware version	02 - firmware version 02
HH - hardware version	01 - hardware version 01

Beckhoff Identification Code (BIC)

The Beckhoff Identification Code contains additional information about the delivery state of the module: General note on the introduction of the Beckhoff Identification Code (BIC) [▶ 61].

2 Product overview

2.1 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:

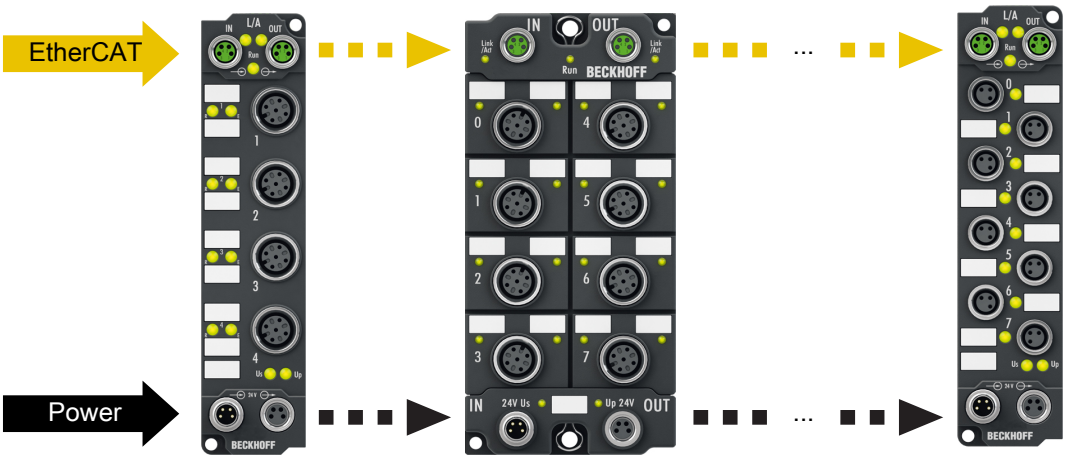


Fig. 1: EtherCAT Box modules: Example of cabling in a line structure

2.2 Introduction

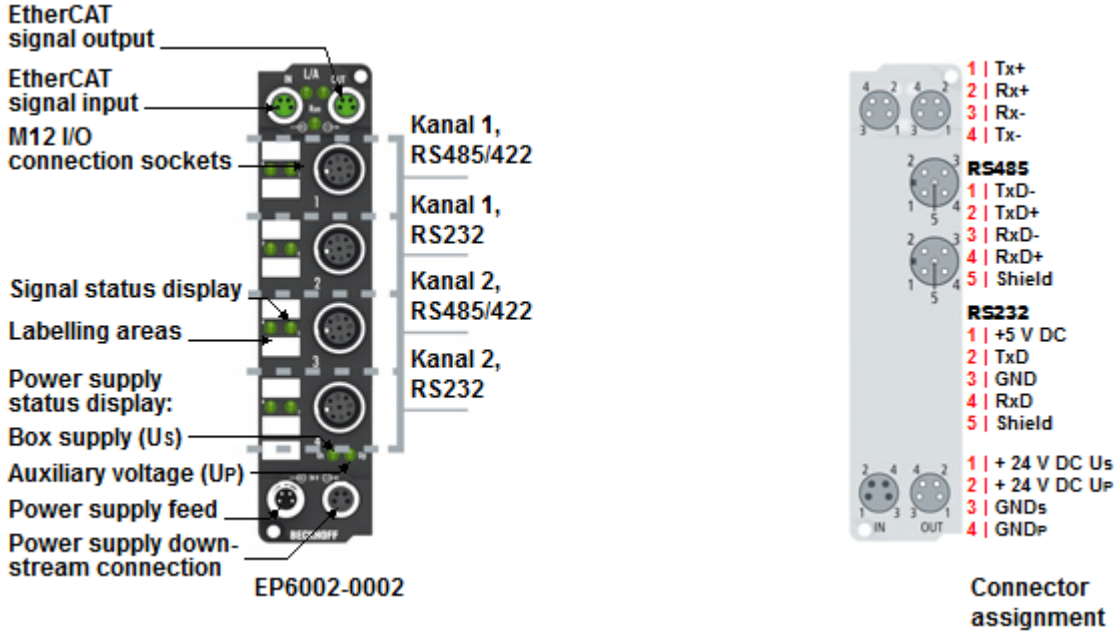


Fig. 2: EP6002-0002

EP6002-0002 | 2-channel serial interface, RS232, RS422/RS485

The EP6002 serial interface module allows the connection of devices with an RS232 or RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The data is transferred via the fieldbus using a simple handshake protocol. This does not have any effect on the protocol of the serial interface. 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 connector assignment depends on the interface. For each channel, RS232 or RS422/RS485 can be selected.

In conjunction with the TwinCAT Virtual Serial COM Driver (see TwinCAT Supplements – Communication) the EP6002 can be used as a normal Windows COM interface.

2.3 Technical data

All values are typical values at 25 °C, unless otherwise stated.

Technical data	EP6002-0002
Fieldbus	
Fieldbus	EtherCAT
Connection	2 x M8 socket, green
Electrical isolation	500 V (fieldbus / IO)
Supply voltages	
Connection	Feed: 1 x M8 plug, 4-pin, black Downstream connection: 1 x M8 socket, 4-pin, black
Control voltage U_S	
Nominal voltage	24 V_{DC} (-15 % / +20 %)
Sum current	max. 4 A [1]
Consumers	• Module electronics: 130 mA • Remote communication stations (e.g. barcode scanner)
Peripheral voltage U_P	
Nominal voltage	24 V_{DC} (-15 % / +20 %)
Sum current	max. 4 A [1]
Consumers	-
Serial interface	
Number of channels	2
Interface type	Individually parameterizable for each channel: • RS232 • RS422 • RS485
Connection	1x M12 socket per channel
Remote station supply voltage [2]	5 V_{DC}, isolated potential max. 20 mA
Cable length	RS232: max. 15 m RS422/RS485: max. 1,000 m
Data transfer rate	Parameterizable: 300 .. 115,200 baud (bit/s)
Data format	Parameterizable: 8N1, 7E1, 7O1, 8N1, 8E1, 8O1, 7E2, 7O2, 8N2, 8E2, 8O2
Receive buffer	864 bytes
Transmit buffer	128 bytes
Bit distortion	< 3 %

[1] Sum current of consumers and power transmission. This value corresponds to the current carrying capacity of the connections for the supply voltages.

[2] Supply voltage available at the connections of the serial interface.

Technical data	EP6002-0002
Environmental conditions	
Ambient temperature during operation	-25 .. +60 °C -25 .. +55 °C according to cURus
Ambient temperature during storage	-40 .. +85 °C
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 conforms to EN 60529
Mechanics	
Weight	approx. 165 g
Installation position	variable
Approvals and conformity	
Approvals	CE, cURus

2.4 Scope of supply

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

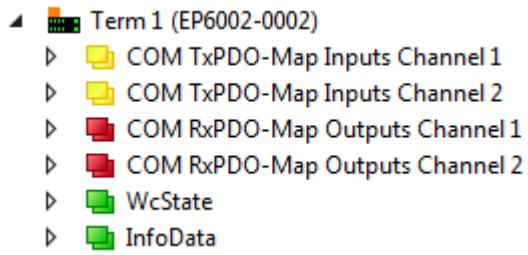
- 1x EtherCAT Box EP6002-0002
- 2x protective cap for EtherCAT socket, M8, green (pre-assembled)
- 1x protective cap for supply voltage input, M8, transparent (pre-assembled)
- 1x protective cap for supply voltage output, M8, black (pre-assembled)
- 10x labels, blank (1 strip of 10)

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.

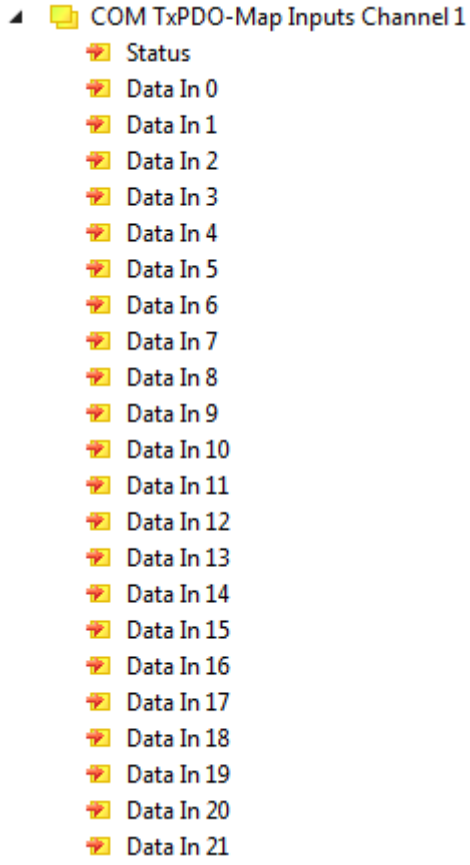
Ensure that the protective caps are correctly seated to ensure IP67 protection.

2.5 Process image



COM TxPDO-Map Inputs

The following figure shows an example of the process data object for serial channel 1. The process data object for channel 2 is structured in exactly the same way.



Status

Status word [▶ 14] for receive data.

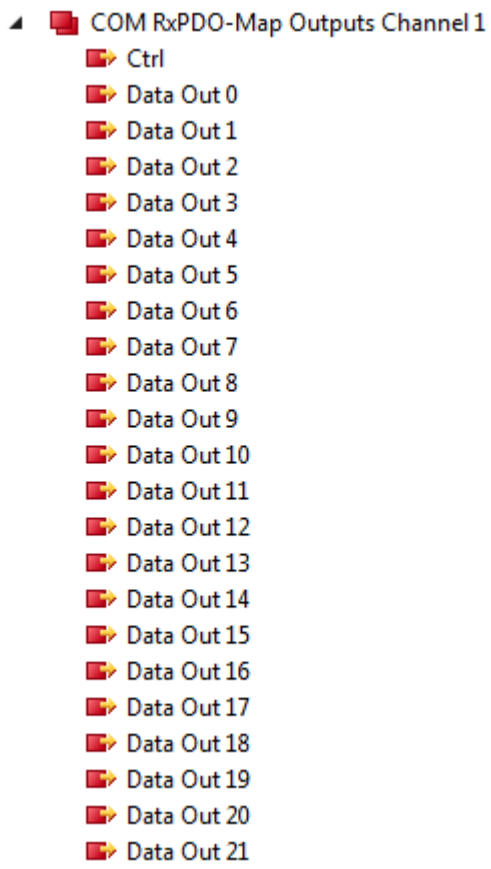
Data In [n]

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 RxPDO-Map Outputs

The following figure shows an example of the process data object for serial channel 1. The process data object for channel 2 is structured in exactly the same way.



Ctrl

Control word [▶ 14] 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.

2.5.1 Assignment of connectors to process data

Connector	Channel	receive data	send data
X01 X02	1	COM TxPDO-Map Inputs Channel 1	COM RxPDO-Map Outputs Channel 1
X03 X04	2	COM TxPDO-Map Inputs Channel 2	COM RxPDO-Map Outputs Channel 2

2.5.2 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}	Number of output bytes available for transfer 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.

2.5.3 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}	Number of input bytes available for transfer 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 receive FIFO is full. All further incoming data will be lost!
2	IA (InitAccepted bit)	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.

2.6 Technology

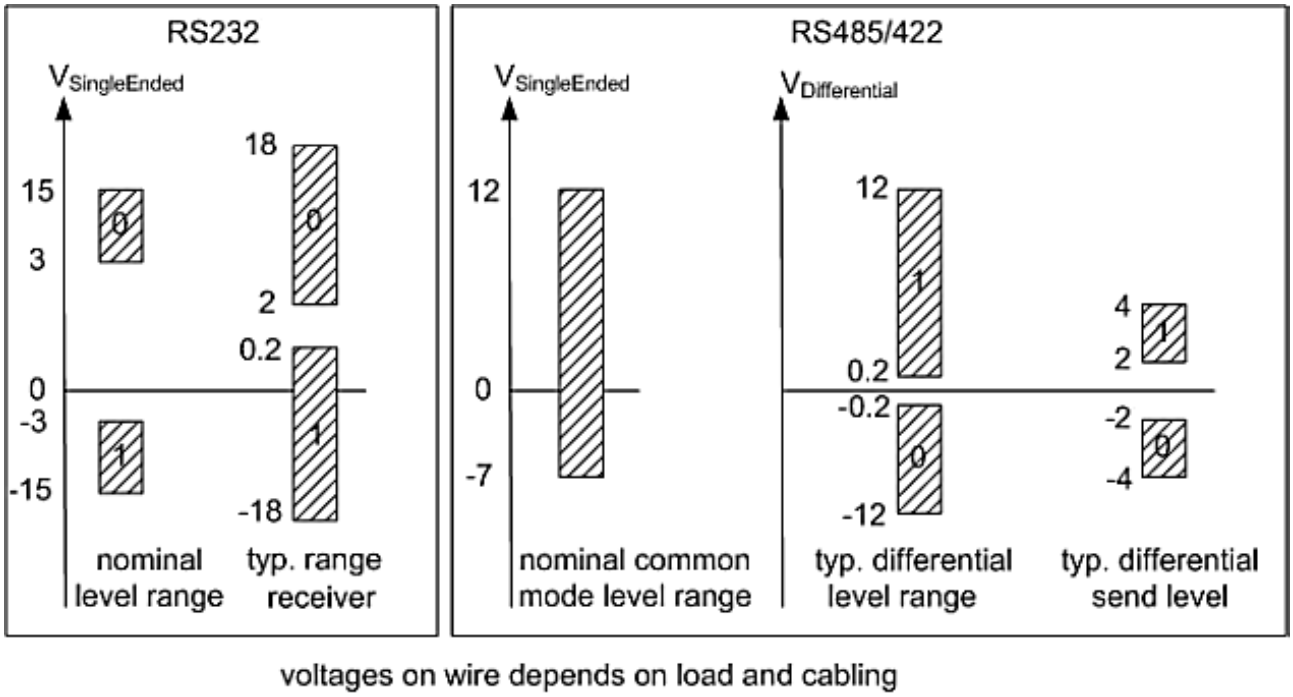


Fig. 3: Level of RS232, RS422, RS485 interfaces

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 baud rate can also be set higher, or the cycle time can be enlarged.

If the controller cannot collect the data quickly enough from the box, the data will be stored intermediately in the internal buffer of the box. The buffer for received data has a size of 864 bytes. If this is exhausted, all further data will be lost.

A buffer is also available for the transmit 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 these 30 bytes per second are received, a 128 bytes transmit buffer will be written to first in this case also. Once this is full, all further data will be lost.

3 Mounting and connection

3.1 Mounting

3.1.1 Dimensions

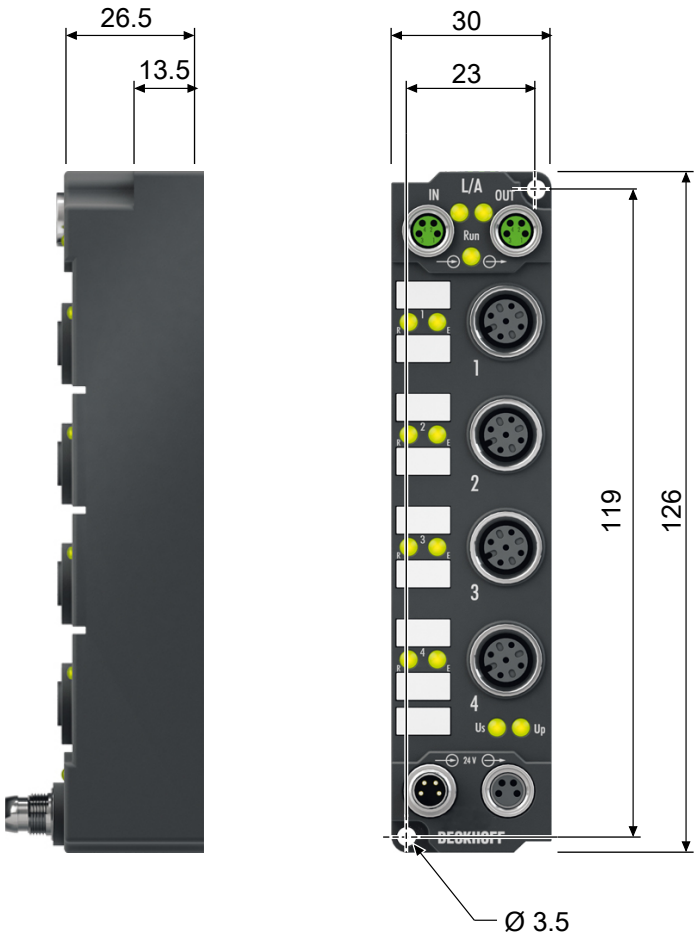


Fig. 4: Dimensions

All dimensions are given in millimeters.

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)

3.1.2 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.

3.2 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!

3.2.1 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. 5: Connectors for supply voltages

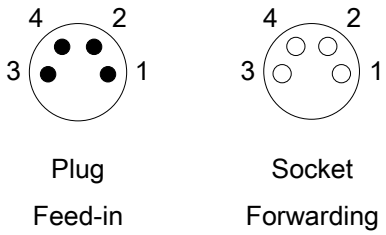


Fig. 6: M8 connector

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-xxxx-xxxx

3.2.2 Status LEDs



Fig. 7: Status LEDs for the power supply

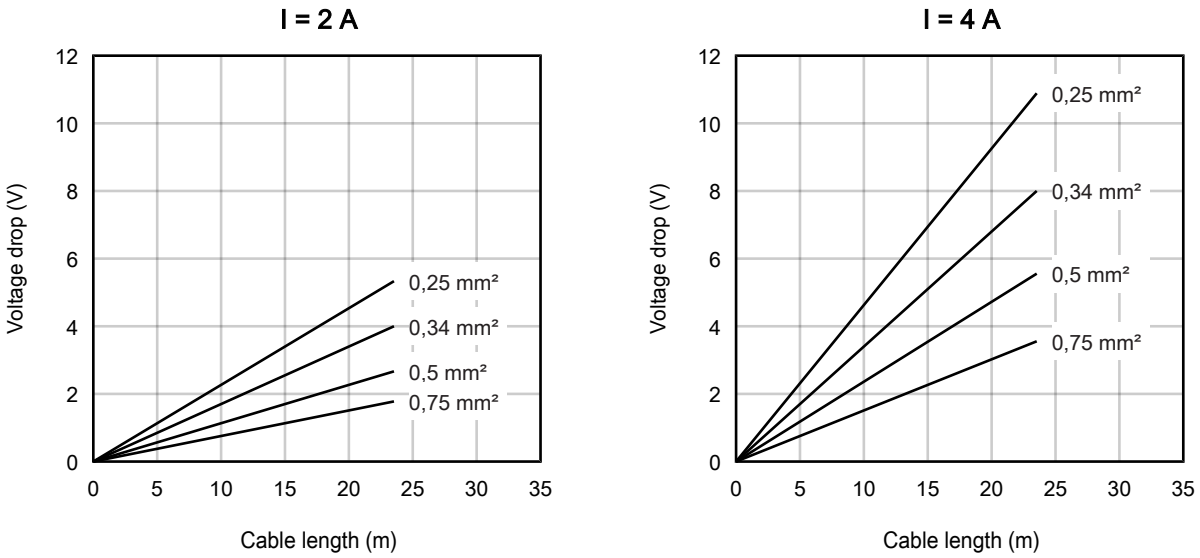
LED	Display	Meaning
U_S (control voltage)	off	Supply voltage, U_S, is not present
	green illuminated	Supply voltage, U_S, is present
U_P (peripheral voltage)	off	Supply voltage, U_P, is not present
	green illuminated	Supply voltage, U_P, is present

3.2.3 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



3.3 EtherCAT

3.3.1 Connectors

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



Fig. 8: EtherCAT connector

Connection

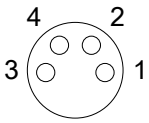


Fig. 9: M8 socket

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.

3.3.2 Status LEDs



Fig. 10: EtherCAT 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

3.3.3 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

3.4 RS232

3.4.1 Connectors

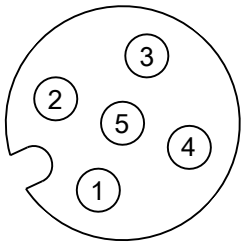


Fig. 11: M12 socket

M12 sockets X02 and X04

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

3.4.2 Status LEDs

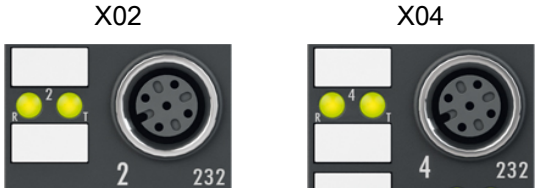


Fig. 12: RS232 Status LEDs

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.

3.5 RS422 / RS485

3.5.1 Connectors

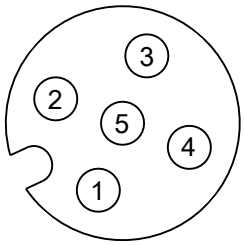


Fig. 13: M12 socket

M12 sockets X01 and X03

Pin	Core color [1)]	RS422		RS485	
		Signal	Description	Signal	Description
1	brown	Tx -	send data	- / A	Inverted data line
2	white	Tx +	send data	+ / B	Non-inverted data line
3	blue	Rx -	receive data	do not connect	
4	black	Rx +	receive data	do not connect	
5	grey	Shield	Shield	Shield	Shield

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).

Signal designations "A" and "B" are not clear

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

3.5.2 Status LEDs

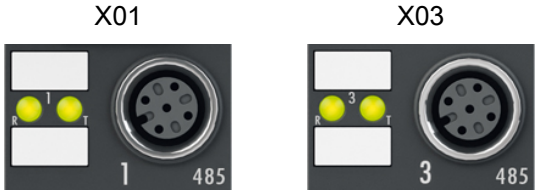


Fig. 14: RS422 / RS485 Status LEDs

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.

3.6 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 0 to 55°C!

Marking for UL

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



Fig. 15: UL label

4 Commissioning/Configuration

4.1 Configuration in TwinCAT

An EtherCAT Box must be configured in TwinCAT so that its functions can be used in a PLC program.

The following link will take you to a quick start guide describing the configuration of an EtherCAT Box in TwinCAT:

https://infosys.beckhoff.com/content/1033/epioconfiguration/index.html?id=6991403443235907429

4.2 Serial interfaces

4.2.1 Parameterization of 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.

4.2.1.1 RS232

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

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

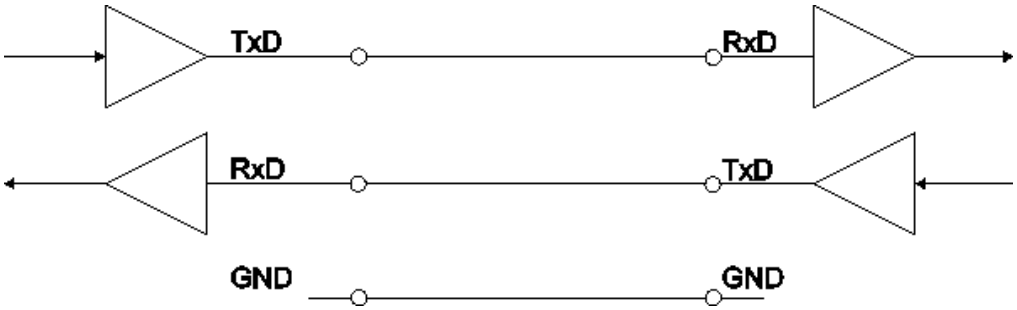


Fig. 16: Point-to-point connection to an RS232 device

The following CoE objects must be set

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

4.2.1.2 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.

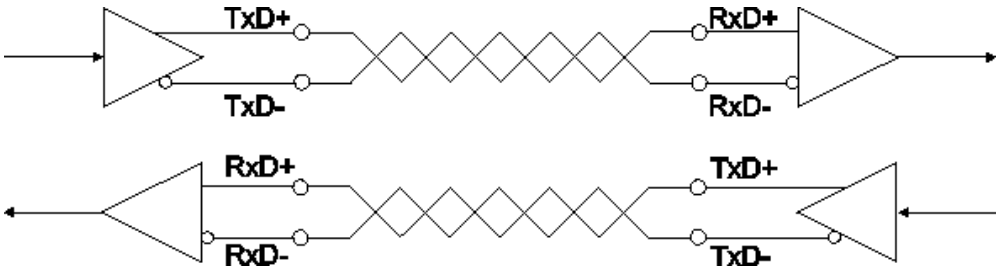


Fig. 17: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	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).			

4.2.1.3 RS485

You can operate the RS845 communication in two variants:

- Without diagnosis
- With diagnosis of the transmit data [▶ 29]

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

Bus structure, half duplex data transmission

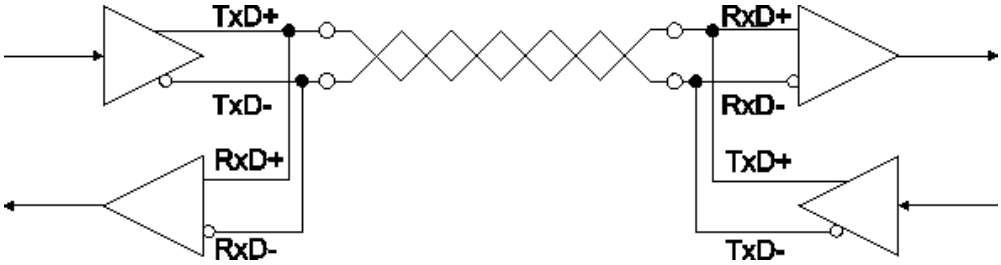


Fig. 18: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex channel n	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
80n0:07	Enable point-to-point connection (RS422) Channel n	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).			

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

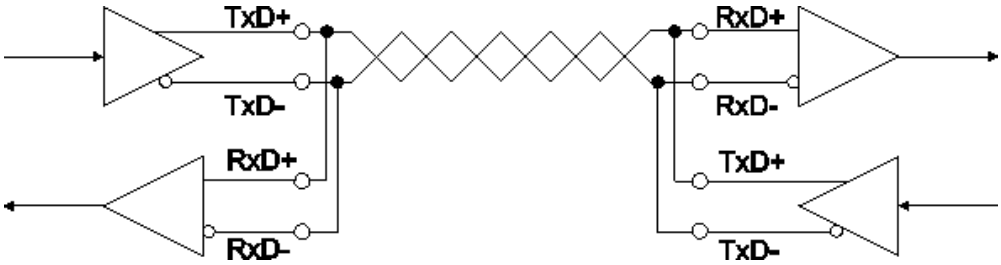


Fig. 19: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex channel n	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
80n0:07	Enable point-to-point connection (RS422) Channel n	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).			

4.2.2 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 from objects 0x8000 and 0x8010 are possible and are shown below in their default settings:

Index	Name	Flags	Value
8000:0	COM Settings Ch.1	RW	> 26 <
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)
8000:1A	Rx buffer full notification	RW	0x0360 (864)
8010:0	COM Settings Ch.2	RW	> 26 <
8010:02	Enable XON/XOFF supported tx data	RW	FALSE
8010:03	Enable XON/XOFF supported rx data	RW	FALSE
8010:04	Enable send FIFO data continuous	RW	FALSE
8010:05	Enable transfer rate optimization	RW	TRUE
8010:06	Enable half duplex	RW	TRUE
8010:07	Enable point to point connection (RS422)	RW	FALSE
8010:11	Baudrate	RW	9600 Baud (6)
8010:15	Data frame	RW	8N1 (3)
8010:1A	Rx buffer full notification	RW	0x0360 (864)

Fig. 20: CoE settings on objects 0x8000 and 0x8010 (default)

4.2.3 Communication by PLC program

Sending data

You can transmit up to 22 bytes of data to the module in one PLC cycle.

1. Write the data to be transmitted in the output variables DataOut 0 ... DataOut 21.
2. Set the *Output Length* parameter in the Control word [▶ 14] to the number of bytes to be transmitted.
3. Toggle the *Transmit Request* bit in the Control word [▶ 14].

⇨ The module acknowledges the data transmission in the Status word [▶ 14] via the *Transmit Accepted* parameter.

Receiving data

If the module toggles the *Receive Request* bit in the Status word [▶ 14], this means that there are new receive data.

1. Read the *Input Length* parameter from the Status word [▶ 14]. It contains the number of bytes to be received.

   ⇨ The data are provided in DataOut 0 ... DataOut 21. The first datum is contained in DataIn 0.
2. After reading the data, acknowledge this by toggling the *Receive Request* bit in the Control word [▶ 14]. The module only makes new data available after that.

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.

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 (128 bytes) 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".

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.

4.2.3.1 Samples

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
2. The module confirms successful initialization by setting "Init accepted".
3. Reset "Init request"
4. The module sets "Init accepted" to 0.

The module is now ready for data exchange.

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"
4. 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”.

4.2.4 Communication via a virtual COM port

Application Note DK9322-0411-0041 describes the communication via a virtual COM port.

4.3 CoE objects

4.3.1 Directory

Index (hex)	Name
1000	Device type [▶ 38]
1008	Device name [▶ 38]
1009	Hardware version [▶ 38]
100A	Software version [▶ 38]
1011	Restore default parameters [▶ 35]
1018	Identity [▶ 38]
10F0	Backup parameter handling [▶ 38]
1400	COM RxPDO-Par Outputs Ch. 1 [▶ 38]
1401	COM RxPDO-Par Outputs Ch. 2 [▶ 38]
1404	COM RxPDO-Par Outputs Ch. 1 [▶ 39]
1405	COM RxPDO-Par Outputs Ch. 2 [▶ 39]
1600	COM RxPDO-Map Outputs Ch. 1 [▶ 40]
1601	COM RxPDO-Map Outputs Ch. 2 [▶ 41]
1604	COM RxPDO-Map Outputs Ch. 1 [▶ 42]
1605	COM RxPDO-Map Outputs Ch. 2 [▶ 43]
1800	COM TxPDO-Par Inputs Ch. 1 [▶ 43]
1801	COM TxPDO-Par Inputs Ch. 2 [▶ 44]
1804	COM TxPDO-Par Inputs Ch. 1 [▶ 44]
1805	COM TxPDO-Par Inputs Ch. 2 [▶ 44]
1A00	COM TxPDO-Map Inputs Ch. 1 [▶ 45]
1A01	COM TxPDO-Map Inputs Ch. 2 [▶ 46]
1A04	COM TxPDO-Map Inputs Ch. 1 [▶ 47]
1A05	DIG TxPDO-Map Inputs Ch. 2 [▶ 48]
1C00	Sync manager type [▶ 48]
1C12	RxPDO assign [▶ 49]
1C13	TxPDO assign [▶ 49]
1C32	SM output parameter [▶ 50]
1C33	SM input parameter [▶ 51]
6000	COM Inputs Ch. 1 [▶ 52]
6001	Status Ch. 1 [▶ 52]
6010	COM Inputs Ch. 2 [▶ 53]
6011	Status Ch. 2 [▶ 53]
7000	COM Outputs Ch. 1 [▶ 54]
7001	Ctrl Ch. 1 [▶ 54]
7010	COM Outputs Ch. 2 [▶ 55]
7011	Ctrl Ch. 2 [▶ 55]
8000	COM Settings Ch. 1 [▶ 36]
8010	COM Settings Ch. 2 [▶ 37]
A000	COM Diag data Ch. 1 [▶ 55]
A010	COM Diag data Ch. 2 [▶ 56]
F000	Modular device profile [▶ 56]
F008	Code word [▶ 56]
F010	Module list [▶ 56]
F800	COM Settings [▶ 37]

4.3.2 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 [▶ 35] during commissioning
- Objects required for the selection of the interface type [▶ 37]
- Objects intended for regular operation, e.g. through ADS access
- Objects for indicating internal settings [▶ 37] (may be fixed)
- Further profile-specific objects [▶ 51] 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			
8000:1A	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})

Index 8010 COM Settings Ch.2

Index (hex)	Name	Meaning		Data type	Flags	Default
8010:0	COM Settings Ch.2			UINT8	RO	0x1A (26_{dec})
8010:02	Enable XON/XOFF supported tx data	XON/XOFF is supported for send data		BOOLEAN	RW	0x00 (0_{dec})
8010:03	Enable XON/XOFF supported rx data	XON/XOFF is supported for receive data		BOOLEAN	RW	0x00 (0_{dec})
8010:04	Enable send FIFO data continuous	Continuous sending of data from the FIFO enabled		BOOLEAN	RW	0x00 (0_{dec})
8010:05	Enable transfer rate optimization	Switch on the transfer rate optimization		BOOLEAN	RW	0x01 (1_{dec})
8010:06	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.			
8010:07	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)			
8010:11	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			
8010:15	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			
8010:1A	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})

Index F800 COM Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
F800:0	COM Settings			UINT8	RO	0x03 (3_{dec})
F800:01	Interface Type Ch 1	0x00	RS232	BIT1	RW	0x00 (0_{dec})
		0x01	RS485/422			
F800:02	Interface Type Ch 2	0x00	RS232	BIT1	RW	0x00 (0_{dec})
		0x01	RS485/422			

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	0x02581389 (39326601_{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	EP6002-0002

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	0x17724052 (393363538_{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 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1400:0	COM RxPDO-Par Outputs Ch.1	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 1401 COM RxPDO-Par Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1401:0	COM RxPDO-Par Outputs Ch.2	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	05 16

Index 1404 COM RxPDO-Par Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	COM RxPDO-Par Outputs Ch.1	PDO Parameter RxPDO 5	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 1405 COM RxPDO-Par Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1405:0	COM RxPDO-Par Outputs Ch.2	PDO Parameter RxPDO 6	UINT8	RO	0x06 (6_{dec})
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6	OCTET-STRING[2]	RO	01 16

Index 1600 COM RxPDO-Map Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	COM RxPDO-Map Outputs Ch.1	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 1601 COM RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	COM RxPDO-Map Outputs Ch.2	PDO Mapping RxPDO 2	UINT8	RO	0x1C (28_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x01 (Transmit request))	UINT32	RO	0x7010:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x02 (Receive accepted))	UINT32	RO	0x7010:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x03 (Init request))	UINT32	RO	0x7010:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x04 (Send continuous))	UINT32	RO	0x7010:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x09 (Output length))	UINT32	RO	0x7010:09, 8
1601:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT32	RO	0x7010:11, 8
1601:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT32	RO	0x7010:12, 8
1601:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT32	RO	0x7010:13, 8
1601:0A	SubIndex 010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT32	RO	0x7010:14, 8
1601:0B	SubIndex 011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT32	RO	0x7010:15, 8
1601:0C	SubIndex 012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT32	RO	0x7010:16, 8
1601:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT32	RO	0x7010:17, 8
1601:0E	SubIndex 014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT32	RO	0x7010:18, 8
1601:0F	SubIndex 015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT32	RO	0x7010:19, 8
1601:10	SubIndex 016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT32	RO	0x7010:1A, 8
1601:11	SubIndex 017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT32	RO	0x7010:1B, 8
1601:12	SubIndex 018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT32	RO	0x7010:1C, 8
1601:13	SubIndex 019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT32	RO	0x7010:1D, 8
1601:14	SubIndex 020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT32	RO	0x7010:1E, 8
1601:15	SubIndex 021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT32	RO	0x7010:1F, 8
1601:16	SubIndex 022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT32	RO	0x7010:20, 8
1601:17	SubIndex 023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT32	RO	0x7010:21, 8
1601:18	SubIndex 024	24. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT32	RO	0x7010:22, 8
1601:19	SubIndex 025	25. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT32	RO	0x7010:23, 8
1601:1A	SubIndex 026	26. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT32	RO	0x7010:24, 8
1601:1B	SubIndex 027	27. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT32	RO	0x7010:25, 8
1601:1C	SubIndex 028	28. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT32	RO	0x7010:26, 8

Index 1604 COM RxPDO-Map Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 5	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 1605 COM RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	COM RxPDO-Map Outputs Ch.2	PDO Mapping RxPDO 6	UINT8	RO	0x17 (23_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7011 (Ctrl Ch.2), entry 0x01 (Ctrl))	UINT32	RO	0x7011:01, 16
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT32	RO	0x7010:11, 8
1605:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT32	RO	0x7010:12, 8
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT32	RO	0x7010:13, 8
1605:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT32	RO	0x7010:14, 8
1605:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT32	RO	0x7010:15, 8
1605:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT32	RO	0x7010:16, 8
1605:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT32	RO	0x7010:17, 8
1605:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT32	RO	0x7010:18, 8
1605:0A	SubIndex 010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT32	RO	0x7010:19, 8
1605:0B	SubIndex 011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT32	RO	0x7010:1A, 8
1605:0C	SubIndex 012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT32	RO	0x7010:1B, 8
1605:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT32	RO	0x7010:1C, 8
1605:0E	SubIndex 014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT32	RO	0x7010:1D, 8
1605:0F	SubIndex 015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT32	RO	0x7010:1E, 8
1605:10	SubIndex 016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT32	RO	0x7010:1F, 8
1605:11	SubIndex 017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT32	RO	0x7010:20, 8
1605:12	SubIndex 018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT32	RO	0x7010:21, 8
1605:13	SubIndex 019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT32	RO	0x7010:22, 8
1605:14	SubIndex 020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT32	RO	0x7010:23, 8
1605:15	SubIndex 021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT32	RO	0x7010:24, 8
1605:16	SubIndex 022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT32	RO	0x7010:25, 8
1605:17	SubIndex 023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT32	RO	0x7010:26, 8

Index 1800 COM TxPDO-Par Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	COM TxPDO-Par Inputs Ch.1	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 1801 COM TxPDO-Par Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1801:0	COM TxPDO-Par Inputs Ch.2	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[2]	RO	05 1A

Index 1804 COM TxPDO-Par Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs Ch.1	PDO parameter TxPDO 5	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 5	OCTET-STRING[2]	RO	00 1A

Index 1805 COM TxPDO-Par Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1805:0	COM TxPDO-Par Inputs Ch.2	PDO parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 6	OCTET-STRING[2]	RO	01 1A

Index 1A00 COM TxPDO-Map Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	COM TxPDO-Map Inputs Ch.1	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 1A01 COM TxPDO-Map Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 2	UINT8	RO	0x1F (31_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x01 (Transmit accepted))	UINT32	RO	0x6010:01, 1
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x02 (Receive request))	UINT32	RO	0x6010:02, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x03 (Init accepted))	UINT32	RO	0x6010:03, 1
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x04 (Buffer full))	UINT32	RO	0x6010:04, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x05 (Parity error))	UINT32	RO	0x6010:05, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x06 (Framing error))	UINT32	RO	0x6010:06, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x07 (Overrun error))	UINT32	RO	0x6010:07, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x09 (Input length))	UINT32	RO	0x6010:09, 8
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT32	RO	0x6010:11, 8
1A01:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT32	RO	0x6010:12, 8
1A01:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT32	RO	0x6010:13, 8
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT32	RO	0x6010:14, 8
1A01:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT32	RO	0x6010:15, 8
1A01:0F	SubIndex 015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT32	RO	0x6010:16, 8
1A01:10	SubIndex 016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT32	RO	0x6010:17, 8
1A01:11	SubIndex 017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT32	RO	0x6010:18, 8
1A01:12	SubIndex 018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT32	RO	0x6010:19, 8
1A01:13	SubIndex 019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT32	RO	0x6010:1A, 8
1A01:14	SubIndex 020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT32	RO	0x6010:1B, 8
1A01:15	SubIndex 021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT32	RO	0x6010:1C, 8
1A01:16	SubIndex 022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT32	RO	0x6010:1D, 8
1A01:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT32	RO	0x6010:1E, 8
1A01:18	SubIndex 024	24. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT32	RO	0x6010:1F, 8
1A01:19	SubIndex 025	25. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT32	RO	0x6010:20, 8
1A01:1A	SubIndex 026	26. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT32	RO	0x6010:21, 8
1A01:1B	SubIndex 027	27. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT32	RO	0x6010:22, 8
1A01:1C	SubIndex 028	28. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT32	RO	0x6010:23, 8
1A01:1D	SubIndex 029	29. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT32	RO	0x6010:24, 8
1A01:1E	SubIndex 030	30. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT32	RO	0x6010:25, 8
1A01:1F	SubIndex 031	31. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT32	RO	0x6010:26, 8

Index 1A04 COM TxPDO-Map Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	COM TxPDO-Map Inputs Ch.1	PDO Mapping TxPDO 5	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 1A05 COM TxPDO-Map Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 6	UINT8	RO	0x17 (23_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6011 (Status Ch.2), entry 0x01 (Status))	UINT32	RO	0x6011:01, 16
1A05:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT32	RO	0x6010:11, 8
1A05:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT32	RO	0x6010:12, 8
1A05:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT32	RO	0x6010:13, 8
1A05:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT32	RO	0x6010:14, 8
1A05:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT32	RO	0x6010:15, 8
1A05:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT32	RO	0x6010:16, 8
1A05:08	SubIndex 008	8. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT32	RO	0x6010:17, 8
1A05:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT32	RO	0x6010:18, 8
1A05:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT32	RO	0x6010:19, 8
1A05:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT32	RO	0x6010:1A, 8
1A05:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT32	RO	0x6010:1B, 8
1A05:0D	SubIndex 013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT32	RO	0x6010:1C, 8
1A05:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT32	RO	0x6010:1D, 8
1A05:0F	SubIndex 015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT32	RO	0x6010:1E, 8
1A05:10	SubIndex 016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT32	RO	0x6010:1F, 8
1A05:11	SubIndex 017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT32	RO	0x6010:20, 8
1A05:12	SubIndex 018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT32	RO	0x6010:21, 8
1A05:13	SubIndex 019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT32	RO	0x6010:22, 8
1A05:14	SubIndex 020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT32	RO	0x6010:23, 8
1A05:15	SubIndex 021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT32	RO	0x6010:24, 8
1A05:16	SubIndex 022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT32	RO	0x6010:25, 8
1A05:17	SubIndex 023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT32	RO	0x6010:26, 8

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. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1605 (5637_{dec})

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. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A05 (6661_{dec})

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	0x000F4240 (1000000_{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 [▶ 50])	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00002710 (10000_{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 [▶ 50], 1C32:05 [▶ 50], 1C32:06 [▶ 50], 1C32:09 [▶ 50], 1C33:03 [▶ 51], 1C33:06 [▶ 50], 1C33:09 [▶ 51] 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 [▶ 50]	UINT32	RW	0x000F4240 (1000000_{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 [▶ 50] or 1C33:08 [▶ 51])	UINT16	RO	0xC007 (49159_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 50]	UINT32	RO	0x00002710 (10000_{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 [▶ 50]	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 [▶ 50]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 50]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 50]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 50]	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 COM Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	COM Inputs Ch.2		UINT8	RO	0x26 (38_{dec})
6010:01	Transmit accepted	The module acknowledges receipt of data by changing the state of this bit	BOOLEAN	RO	0x00 (0_{dec})
6010:02	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})
6010:03	Init Accepted	The initialization is carried out from the terminal	BOOLEAN	RO	0x00 (0_{dec})
6010:04	Buffer full	The receive FIFO is full	BOOLEAN	RO	0x00 (0_{dec})
6010:05	Parity error	A parity error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:06	Framing error	A framing error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:07	Overrun error	An overrun error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:09	Input length	Number of input bytes available for transfer from the terminal to the controller	UINT8	RO	0x00 (0_{dec})
6010:11	Data In 0	Input data	UINT8	RO	0x00 (0_{dec})
6010:12	Data In 1	Input data	UINT8	RO	0x00 (0_{dec})
6010:13	Data In 2	Input data	UINT8	RO	0x00 (0_{dec})
6010:14	Data In 3	Input data	UINT8	RO	0x00 (0_{dec})
6010:15	Data In 4	Input data	UINT8	RO	0x00 (0_{dec})
6010:16	Data In 5	Input data	UINT8	RO	0x00 (0_{dec})
6010:17	Data In 6	Input data	UINT8	RO	0x00 (0_{dec})
6010:18	Data In 7	Input data	UINT8	RO	0x00 (0_{dec})
6010:19	Data In 8	Input data	UINT8	RO	0x00 (0_{dec})
6010:1A	Data In 9	Input data	UINT8	RO	0x00 (0_{dec})
6010:1B	Data In 10	Input data	UINT8	RO	0x00 (0_{dec})
6010:1C	Data In 11	Input data	UINT8	RO	0x00 (0_{dec})
6010:1D	Data In 12	Input data	UINT8	RO	0x00 (0_{dec})
6010:1E	Data In 13	Input data	UINT8	RO	0x00 (0_{dec})
6010:1F	Data In 14	Input data	UINT8	RO	0x00 (0_{dec})
6010:20	Data In 15	Input data	UINT8	RO	0x00 (0_{dec})
6010:21	Data In 16	Input data	UINT8	RO	0x00 (0_{dec})
6010:22	Data In 17	Input data	UINT8	RO	0x00 (0_{dec})
6010:23	Data In 18	Input data	UINT8	RO	0x00 (0_{dec})
6010:24	Data In 19	Input data	UINT8	RO	0x00 (0_{dec})
6010:25	Data In 20	Input data	UINT8	RO	0x00 (0_{dec})
6010:26	Data In 21	Input data	UINT8	RO	0x00 (0_{dec})

Index 6011 Status Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6011:0	Status Ch.2		UINT8	RO	0x01 (1_{dec})
6011:01	Status	Status word for compatible process image	UINT16	RO	0x0000 (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 COM Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	COM Outputs Ch.2		UINT8	RO	0x26 (38_{dec})
7010:01	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})
7010:02	Receive accepted	The controller acknowledges receipt of data by changing the state of this bit.	BOOLEAN	RO	0x00 (0_{dec})
7010:03	Init request	The controller requests the module to initialize.	BOOLEAN	RO	0x00 (0_{dec})
7010:04	Send continuous	Continuous sending of data from the FIFO.	BOOLEAN	RO	0x00 (0_{dec})
7010:09	Output length	Number of output bytes available for transfer from the controller to the terminal.	UINT8	RO	0x00 (0_{dec})
7010:11	Data Out 0	Output data	UINT8	RO	0x00 (0_{dec})
7010:12	Data Out 1	Output data	UINT8	RO	0x00 (0_{dec})
7010:13	Data Out 2	Output data	UINT8	RO	0x00 (0_{dec})
7010:14	Data Out 3	Output data	UINT8	RO	0x00 (0_{dec})
7010:15	Data Out 4	Output data	UINT8	RO	0x00 (0_{dec})
7010:16	Data Out 5	Output data	UINT8	RO	0x00 (0_{dec})
7010:17	Data Out 6	Output data	UINT8	RO	0x00 (0_{dec})
7010:18	Data Out 7	Output data	UINT8	RO	0x00 (0_{dec})
7010:19	Data Out 8	Output data	UINT8	RO	0x00 (0_{dec})
7010:1A	Data Out 9	Output data	UINT8	RO	0x00 (0_{dec})
7010:1B	Data Out 10	Output data	UINT8	RO	0x00 (0_{dec})
7010:1C	Data Out 11	Output data	UINT8	RO	0x00 (0_{dec})
7010:1D	Data Out 12	Output data	UINT8	RO	0x00 (0_{dec})
7010:1E	Data Out 13	Output data	UINT8	RO	0x00 (0_{dec})
7010:1F	Data Out 14	Output data	UINT8	RO	0x00 (0_{dec})
7010:20	Data Out 15	Output data	UINT8	RO	0x00 (0_{dec})
7010:21	Data Out 16	Output data	UINT8	RO	0x00 (0_{dec})
7010:22	Data Out 17	Output data	UINT8	RO	0x00 (0_{dec})
7010:23	Data Out 18	Output data	UINT8	RO	0x00 (0_{dec})
7010:24	Data Out 19	Output data	UINT8	RO	0x00 (0_{dec})
7010:25	Data Out 20	Output data	UINT8	RO	0x00 (0_{dec})
7010:26	Data Out 21	Output data	UINT8	RO	0x00 (0_{dec})

Index 7011 Ctrl Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7011:0	Ctrl Ch.2		UINT8	RO	0x01 (1_{dec})
7011:01	Ctrl	Control word for compatible process image	UINT16	RO	0x0000 (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 A010 COM Diag data Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	COM Diag data Ch.2		UINT8	RO	0x21 (33_{dec})
A010:01	Buffer overflow	A buffer overflow has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:02	Parity error	A parity error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:03	Framing error	A framing error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:04	Overrun error	An overrun error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:05	Buffer full	The receive FIFO is full.	BOOLEAN	RO	0x00 (0_{dec})
A010:11	Data bytes in send buffer	Number of data bytes in the send FIFO	UINT16	RO	0x0000 (0_{dec})
A010:21	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	0x00000258 (600_{dec})

4.4 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).

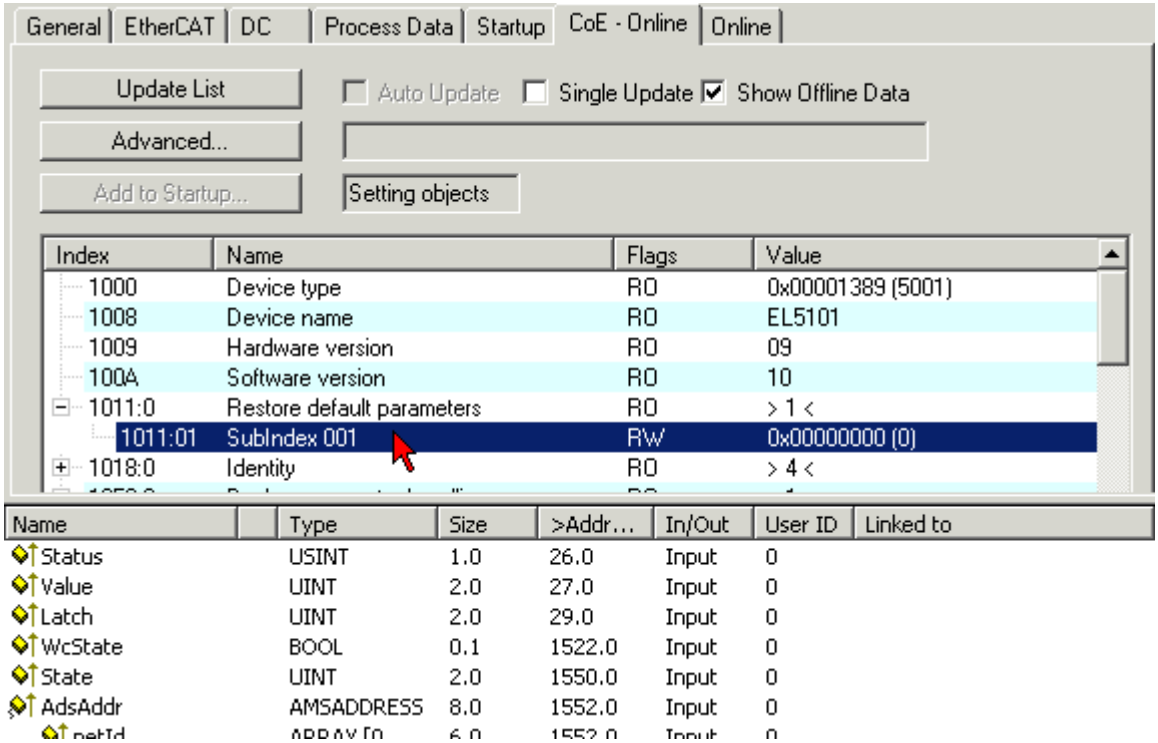


Fig. 21: Selecting the Restore default parameters PDO

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.

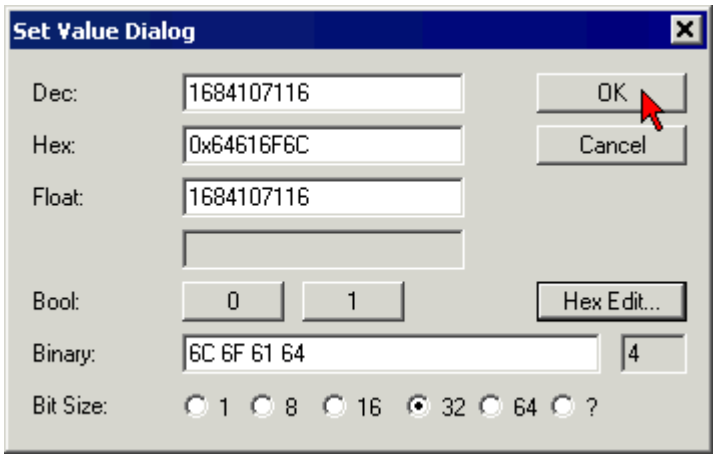


Fig. 22: Entering a restore value in the Set Value dialog

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.

4.5 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

5 Appendix

5.1 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

5.2 EtherCAT Box- / EtherCAT P Box - Accessories

Fixing

Ordering information	Description
ZS5300-0001	Mounting rail (500 mm x 129 mm)

Marking material, plugs

Ordering information	Description
ZS5000-0000	Fieldbus Box set M8 (contact labels, plugs)
ZS5000-0002	Fieldbus Box set M12 (contact labels, plugs)
ZS5000-0010	plugs M8, IP67 (50 pieces)
ZS5000-0020	plugs M12, IP67 (50 pieces)
ZS5100-0000	marking labels, not printed, 4 stripes at 10 pieces
ZS5100-xxxx	printed marking labels, on request

Tools

Ordering information	Description
ZB8800	torque wrench for M8 cables with knurl, incl. ratchet
ZB8800-0001	M12 ratchet for torque wrench ZB8800
ZB8800-0002	M8 ratchet (field assembly) for torque wrench ZB8800
ZB8801-0000	torque wrench for hexagonal plugs, adjustable
ZB8801-0001	torque cable key, M8/wrench size 9, for torque wrench ZB8801-0000
ZB8801-0002	torque cable key, M12/wrench size 13, for torque wrench ZB8801-0000
ZB8801-0003	torque cable key, M12 field assembly/wrench size 13, for torque wrench ZB8801-0000

Further accessories

Further accessories may be found at the price list for Beckhoff fieldbus components and at the internet under https://www.beckhoff.com

5.3 General note on the introduction of the Beckhoff Identification Code (BIC)

General

In future you will increasingly find machine-readable information on Beckhoff products in the form of a Data Matrix Code (DMC, ECC200). This helps us to improve the quality assurance process, beyond which you can use it for better identification of our products.

The introduction of the Data Matrix Code (called BIC [**B**eckhoff **I**dentification **C**ode] at Beckhoff) is taking place gradually across all product groups.

The information in the BICs is oriented to the ANSI standard MH10.8.2-2016

Representation and contents of the BIC

The BIC can be found in the following places, depending on the product:

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

The BIC is readable and contains information that you can use for your internal handling and administration of the products. When scanning the BICs you will find the following information:

Coded information on the BIC

Item no.	Type of information	Explanation
1	**Beckhoff order number**	**Beckhoff order number**
2	**Beckhoff Traceability Number (BTN)**	**Unique serial number, see note below**
3	**Article description**	**Beckhoff article description, e.g. EL1008**
4	**Quantity**	**Quantity in packaging unit, e.g. 1, 10, etc.**
5	Batch number	Optional: Year and week of production
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products
7	Variant number	Optional: Product variant number on the basis of standard products
8	Date code	Internal
9	Job/batch number	Internal
10	Serial number	Internal
...		

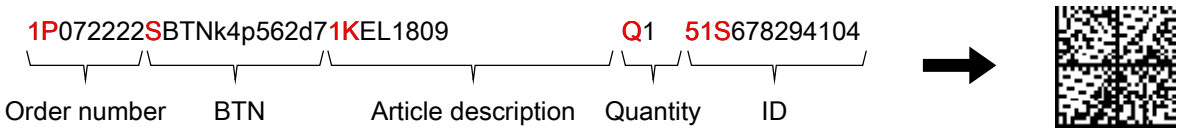
Each item of information is clearly identifiable on the basis of the data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the following table. If the items of information are shorter, they are replaced by spaces. The data under positions 1-4 always exist (temporary restriction, see BTN below).

Overview of the defined data identifiers with examples

Item no.	Type of information	Data identifier	Number of characters	Sample
1	Beckhoff order number	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	S	12	SBTNk4p562d7
3	Article description	1K	32	1KEL1809
4	Quantity	Q	6	Q1
5	Batch number	2P	14	2P401503180016
6	ID/serial number	51S	12	51S678294104
7	Version	30P	32	30PF971 , 2*K183
8	Date code	9D	8	9DG0118
9	Job/batch number	1T	14	1TFA12345678
10	Serial number	52S	23	52S2304853-1-004
...				

Structure of the BIC

Example of an item of information made up of positions 1 – 4 and 6. The data identifiers are each marked in red for clearer illustration:



BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 2). The BTN is a unique 8-character serial number that in future will replace all other serial number systems at Beckhoff (e.g. batch designations on IO components, hitherto serial number circle for safety products, etc.). The BTN is likewise being introduced gradually, so it may be the case that the BTN is not yet coded in the BIC.

5.4 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

Table of Figures

Fig. 1 EtherCAT Box modules: Example of cabling in a line structure 8
Fig. 2 EP6002-0002 9
Fig. 3 Level of RS232, RS422, RS485 interfaces 15
Fig. 4 Dimensions 16
Fig. 5 Connectors for supply voltages 18
Fig. 6 M8 connector 18
Fig. 7 Status LEDs for the power supply 19
Fig. 8 EtherCAT connector 20
Fig. 9 M8 socket 20
Fig. 10 EtherCAT status LEDs 21
Fig. 11 M12 socket 22
Fig. 12 RS232 Status LEDs 22
Fig. 13 M12 socket 23
Fig. 14 RS422 / RS485 Status LEDs 23
Fig. 15 UL label 24
Fig. 16 Point-to-point connection to an RS232 device 26
Fig. 17 4-wire point-to-point connection to an RS422 device 27
Fig. 18 2-wire connection in bus structure to RS485 device(s) 28
Fig. 19 2-wire connection with external bridge in bus structure to RS485 device(s) 29
Fig. 20 CoE settings on objects 0x8000 and 0x8010 (default) 30
Fig. 21 Selecting the Restore default parameters PDO 57
Fig. 22 Entering a restore value in the Set Value dialog 57

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP