BECKHOFF New Automation Technology

Documentation | EN

EP6001-0002

1-channel serial interface, RS232/RS422/RS485



2026-10-29 | Version: 1.3

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 EtherCAT Box - Introduction 8
3 Product overview 10
3.1 Technical data 11
3.1.1 Interface levels 13
3.2 Scope of supply 14
3.3 Process image 15
3.3.1 Status word 17
3.3.2 Control word 17
3.4 Technology 18
4 Mounting and connection 19
4.1 Mounting 19
4.1.1 Dimensions 19
4.1.2 Fixing 20
4.1.3 Tightening torques for plug connectors 20
4.2 Supply voltages 21
4.2.1 Connectors 22
4.2.2 Status LEDs 22
4.2.3 Conductor losses 23
4.3 EtherCAT 24
4.3.1 Connectors 24
4.3.2 Status LEDs 25
4.3.3 Cables 25
4.4 RS232 26
4.4.1 Connector 26
4.4.2 Status LEDs 26
4.5 RS422 27
4.5.1 Connectors 27
4.5.2 Status LEDs 27
4.6 RS485 28
4.6.1 Connector 28
4.6.2 Status LEDs 28
4.7 Digital combi channels 29
4.7.1 Connector 29
4.7.2 Status LEDs 29
4.8 UL Requirements 30
4.9 Disposal 31
5 Commissioning and configuration 32
5.1 Integrating into a TwinCAT project 32
5.2 Serial interface 33

5.2.1 Setting the interface type ... 33
5.2.2 Setting the interface parameters ... 37
5.2.3 Communication by PLC program ... 38
5.2.4 Communication via a virtual COM port ... 40
5.3 Digital combi channels ... 41
5.3.1 Activating process data ... 41
5.3.2 Assignment of connector pins to process data ... 41
5.4 Restore the delivery state ... 42
5.5 Decommissioning ... 43
6 CoE parameters ... 44
6.1 Configuration objects ... 45
6.2 Standard objects ... 47
6.3 Profile-specific objects ... 56
7 Appendix ... 59
7.1 General operating conditions ... 59
7.2 Accessories ... 60
7.3 Version identification of EtherCAT devices ... 61
7.3.1 General notes on marking ... 61
7.3.2 Version identification of IP67 modules ... 62
7.3.3 Beckhoff Identification Code (BIC) ... 63
7.3.4 Electronic access to the BIC (eBIC) ... 65
7.4 Support and Service ... 67

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.
The qualified personnel is obliged to always use the currently valid documentation.

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.





EtherCAT® and EtherCAT P® are registered trademarks and patented technologies, 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.

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

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.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Documentation issue status

Version	Comment
1.3	• Technical data updated
1.2	• Dimensions updated • UL requirements updated • Technical data updated
1.1	• Front page updated
1.0	• First release, adapted from the documentation EP600x V2.1.0

Firmware and hardware versions

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

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

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

Syntax of the batch number (D-number)

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

Further information on this topic: Version identification of EtherCAT devices [▶ 61].

2 EtherCAT Box - Introduction

The EtherCAT system has been extended with EtherCAT Box modules with protection rating IP67. Through the integrated EtherCAT interface the modules can be connected directly to an EtherCAT network without an additional Coupler Box. The high-performance of EtherCAT is thus maintained into each module.

The extremely low dimensions of only 126 x 30 x 26.5 mm (h x w x d) are identical to those of the Fieldbus Box extension modules. They are thus particularly suitable for use where space is at a premium. The small mass of the EtherCAT modules facilitates applications with mobile I/O interface (e.g. on a robot arm). The EtherCAT connection is established via screened M8 connectors.

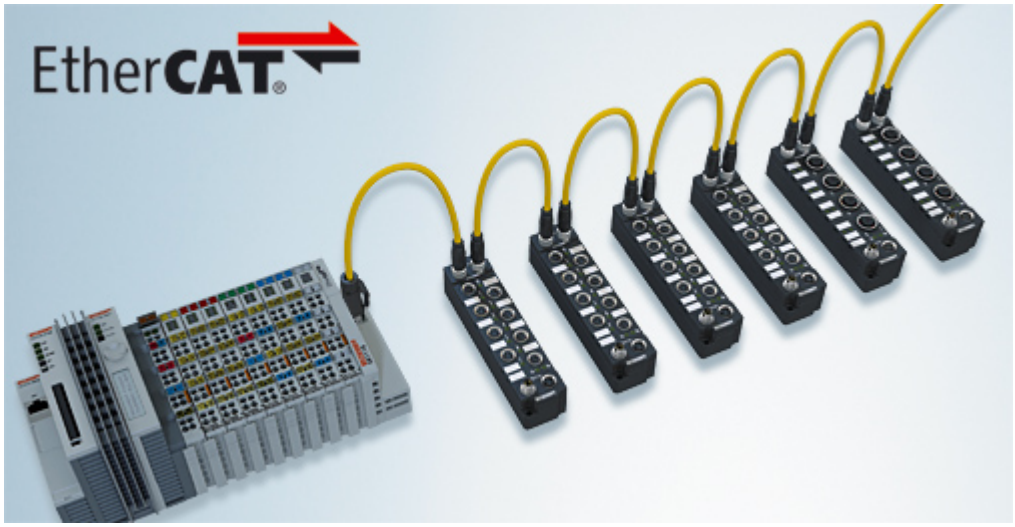


Fig. 1: EtherCAT Box Modules within an EtherCAT network

The robust design of the EtherCAT Box modules enables them to be used directly at the machine. Control cabinets and terminal boxes are now no longer required. The modules are fully sealed and therefore ideally prepared for wet, dirty or dusty conditions.

Pre-assembled cables significantly simplify EtherCAT and signal wiring. Very few wiring errors are made, so that commissioning is optimized. In addition to pre-assembled EtherCAT, power and sensor cables, field-configurable connectors and cables are available for maximum flexibility. Depending on the application, the sensors and actuators are connected through M8 or M12 connectors.

The EtherCAT modules cover the typical range of requirements for I/O signals with protection rating IP67:

- digital inputs with different filters (3.0 ms or 10 µs)
- digital outputs with 0.5 or 2 A output current
- analog inputs and outputs with 16 bit resolution
- Thermocouple and RTD inputs
- Stepper motor modules

XFC (eXtreme Fast Control Technology) modules, including inputs with time stamp, are also available.



Fig. 2: EtherCAT Box with M8 connections for sensors/actuators

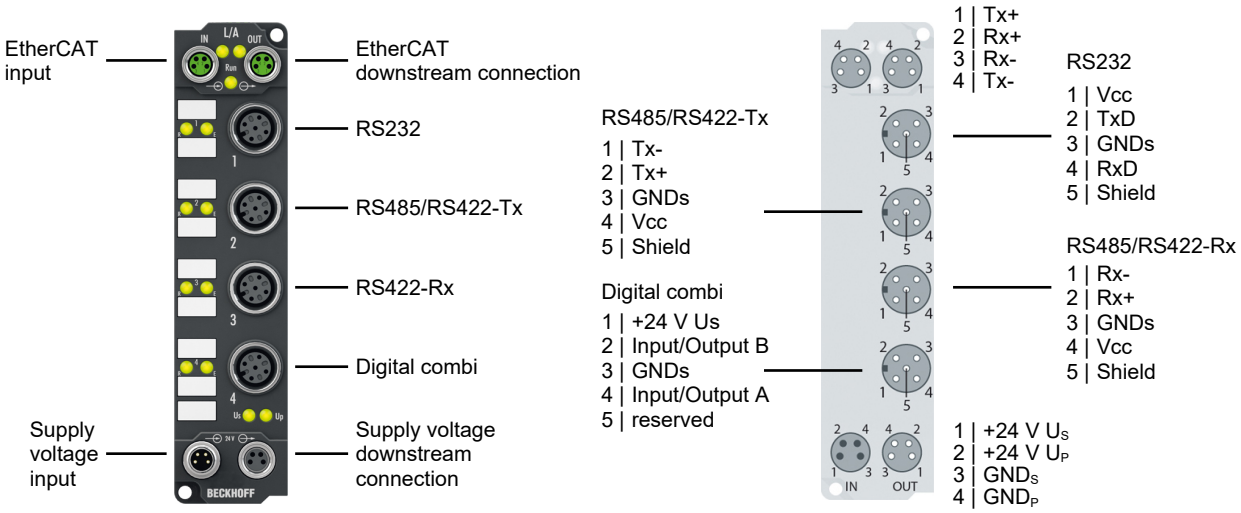


Fig. 3: EtherCAT Box with M12 connections for sensors/actuators

Basic EtherCAT documentation

You will find a detailed description of the EtherCAT system in the Basic System Documentation for EtherCAT, which is available for download from our website (www.beckhoff.com) under Downloads.

3 Product overview



1-channel communication interface, RS232/RS422/RS485

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

Quick links

Technical data [▶ 11]
Process image [▶ 15]
Dimensions [▶ 19]
RS232 connection [▶ 26]
RS422 connection [▶ 27]
RS485 connection [▶ 28]
Commissioning [▶ 32]

3.1 Technical data

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

EtherCAT	
Connection	2 x M8 socket, 4-pin, A-coded, shielded
Electrical isolation	500 V

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

Serial interface	
Number of channels	1
Interface type	Adjustable: • RS232 (default) • RS422 • RS485
Connection	RS232: 1 x M12 socket RS422: 2 x M12 socket RS485: 1 x M12 socket
End device power supply	5 V DC from U_S, output current 1 A max. short-circuit proof
Cable length	RS232: max. 15 m RS422: max. 1000 m RS485: max. 1000 m
Transmission rate	Adjustable: 300…115,200 baud (bit/s)
Data format	Adjustable: 8N1, 7E1, 7O1, 8N1, 8E1, 8O1, 7E2, 7O2, 8N2, 8E2, 8O2
Flow control	Software flow control “XON/XOFF”
Bit distortion	< 3%
Receive buffer	846 bytes
Transmit buffer	128 bytes

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

Digital outputs	
Number	0 to 2 Each digital output can alternatively be used as a digital input.
Connection	1 x M12 socket
Load type	Ohmic, inductive, lamp load
Rated output voltage	24 V DC from U_S
Output current per channel	max. 0.5 A. Each output is independently short-circuit proof.
Short circuit current	max. 1.5 A
Output driver supply	From the supply voltage U_S
Switching times	T_{ON}: 50 µs typ., T_{OFF}: 100 µs typ.

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

Environmental conditions	
Ambient temperature during operation	-25…+60 °C -25…+55 °C according to cURus
Ambient temperature during storage	-40…+85 °C
Vibration resistance, shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 Additional tests [▶ 12]
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection rating	IP65, IP66, IP67 (conforms to EN 60529)

Approvals/markings	
Approvals/markings *)	CE, cURus [▶ 30]

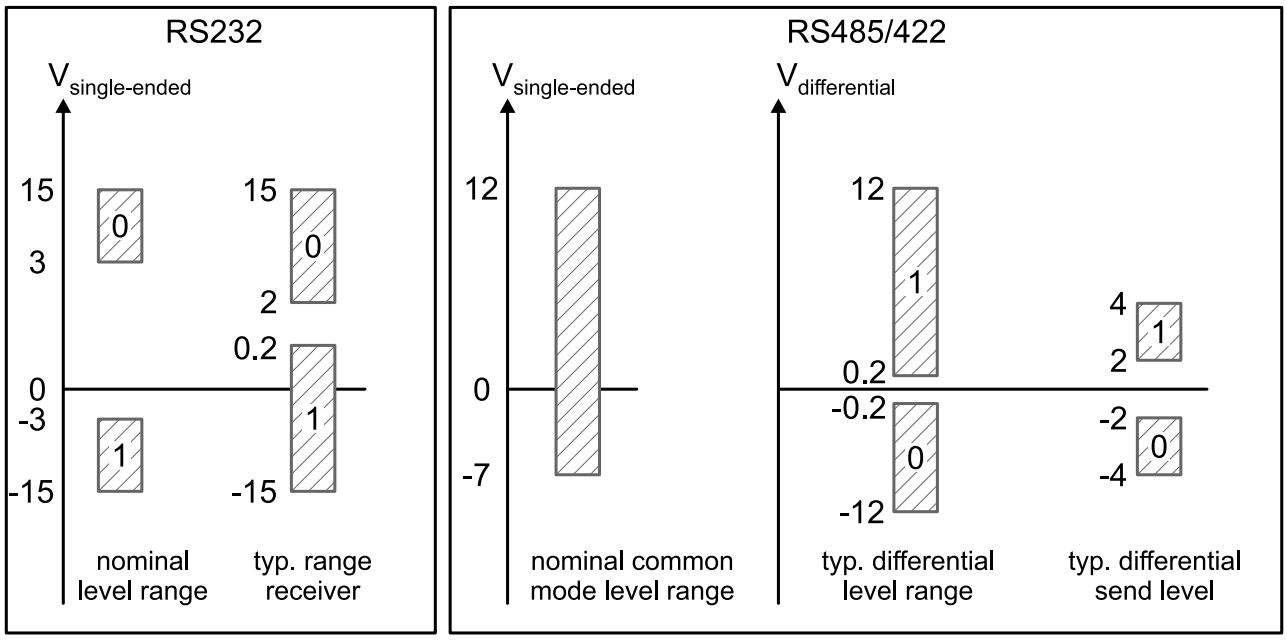
*) Real applicable approvals/markings see type plate on the side (product marking).

Additional tests

The devices have undergone the following additional tests:

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

3.1.1 Interface levels



Voltage levels depend on the load and the cabling

3.2 Scope of supply

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

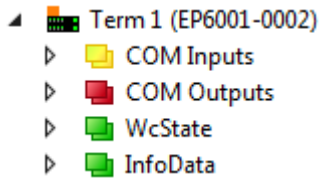
- 1x EtherCAT Box EP6001-0002
- 2x protective cap for EtherCAT socket, M8, green (pre-assembled)
- 1x protective cap for supply voltage input, M8, transparent (pre-assembled)
- 1x protective cap for supply voltage output, M8, black (pre-assembled)
- 10x labels, blank (1 strip of 10)

Pre-assembled protective caps do not ensure IP67 protection

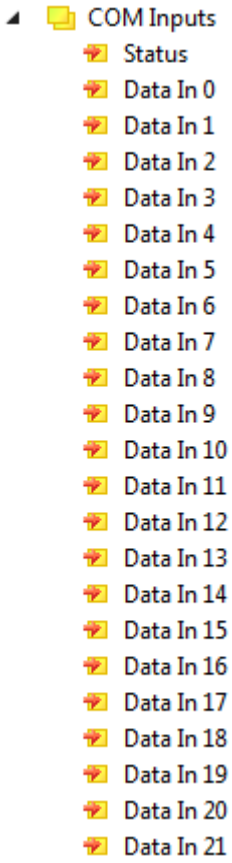
Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

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

3.3 Process image



COM Inputs



Status

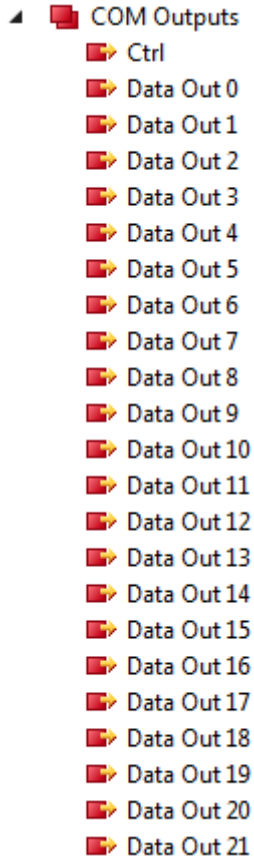
Status word for receive data, see chapter Status word [▶ 17].

Data In <n>

The PDO entries "Data In 0"–"Data In 22" each contain one byte of receive data (USINT).

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

COM outputs



Ctrl

Control word for transmit data, see chapter Control word [▶ 17].

Data Out <n>

The PDO entries "Data Out 0"–"Data Out 22" can each be filled with one byte of send data.

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

3.3.1 Status word

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

Key

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

3.3.2 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)	$1_{dec...}22_{dec}$	The number of output bytes available for the transmission from the controller to the box.
7 .. 4	reserved		
3	SC (SendContinuous)	rise	Continuous sending of data from the FIFO. The send buffer is filled (up to 128 bytes) by the controller. The buffer content is sent with rising edge of bit SC. Once the data has been transferred, this is acknowledged by the box to the controller by setting the SW.2 bit. SW.2 is cancelled with CW.3.
2	IR (InitRequest)	1_{bin}	The controller requests the box to initialize. The send and receive functions are blocked, the FIFO pointers are reset, and the interface is initialized with the values of the responsible objects (baud rate 4073, data frame 4074, feature bits 4075). The execution of the initialization is acknowledged by the box with the SW.2 (IA) bit.
		0_{bin}	The controller once again requests the box to prepare for serial data exchange.
1	RA (ReceiveAccepted)	toggle	The controller acknowledges receipt of data by changing the state of this bit. Only then can new data be transferred from the box to the controller.
0	TR (TransmitRequest)	toggle	Via a change of state of this bit the controller notifies the box that the DataOut bytes contain the number of bytes indicated via the OL bits. The box acknowledges receipt of the data in the status byte by changing the state of the SW.0 (TA) bit. Only then can new data be transferred from the controller to the box.

3.4 Technology

Transmission 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** transmission rate of:

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

can be achieved.

The next lower standard transmission 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 transmission rate can also be set higher, or the cycle time can be enlarged.

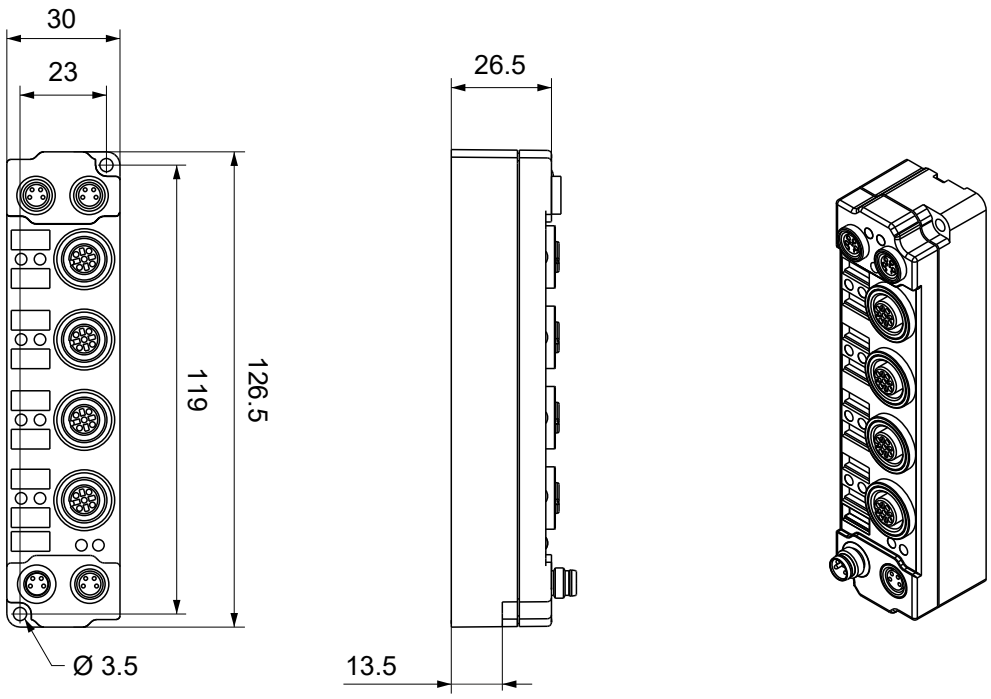
If the controller cannot fetch the data quickly enough from the box, they are buffered in the box's receive buffer. All further data are lost if the receive buffer is full.

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

4 Mounting and connection

4.1 Mounting

4.1.1 Dimensions



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

Housing features

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

4.1.2 Fixing

NOTICE

Dirt during assembly

Dirty connectors can lead to malfunctions. Protection rating 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 mounting holes in the corners of the module. The mounting holes have no thread.

4.1.3 Tightening torques for plug connectors

Tighten the connectors using a torque wrench. (e.g., Beckhoff ZB8801)

Connector diameter	Tightening torque
M8	0.4 Nm
M12	0.6 Nm

4.2 Supply voltages

⚠ WARNING

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV / PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

⚠ CAUTION

Observe the UL requirements

- When operating under UL conditions, observe the warnings in the chapter UL Requirements [▶ 30].

The EtherCAT Box has one input for two supply voltages:

- **Control voltage U_S**
  The following sub-functions are supplied from the control voltage U_S:
  - the fieldbus
  - the processor logic
  - typically the inputs and the sensors if the EtherCAT Box has inputs.
- **Peripheral voltage U_P**
  For EtherCAT Box modules with digital outputs the digital outputs are typically supplied from the peripheral voltage U_P. U_P can be supplied separately. If U_P is switched off, the fieldbus function, the function of the inputs and the supply of the sensors are maintained.

The exact assignment of U_S and U_P can be found in the pin assignment of the I/O connections.

Redirection of the supply voltages

The power IN and OUT connections are bridged in the module. Hence, the supply voltages U_s and U_p can be passed from EtherCAT Box to EtherCAT Box in a simple manner.

NOTICE

Note the maximum current!

Ensure that the permitted current for the connectors is not exceeded when routing the supply voltages U_S and U_P:

M8 connector: max. 4 A
7/8" connector: max 16 A

NOTICE

Unintentional cancellation of the electrical isolation possible

In some types of EtherCAT Box modules the ground potentials GND_S and GND_P are connected.

- If several EtherCAT Box modules are supplied with the same electrically isolated voltages, check whether there is an EtherCAT Box among them in which the ground potentials are connected.

4.2.1 Connectors

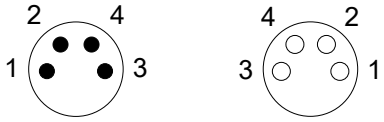
NOTICE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

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





Plug Socket

Input Forwarding

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

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

4.2.2 Status LEDs

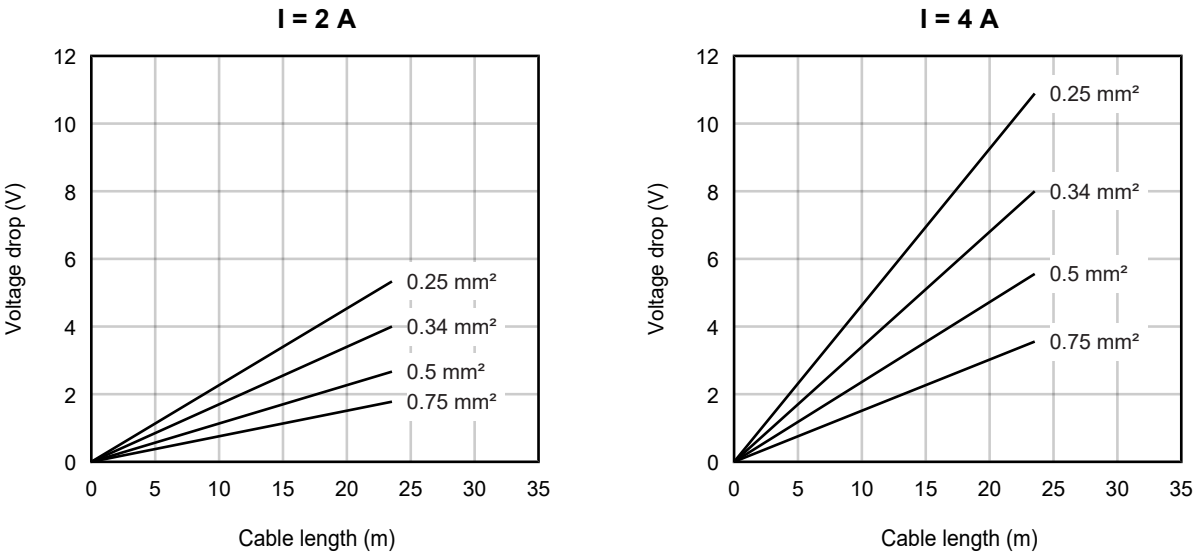


LED	Display	Meaning
U_S (control voltage)	off	The supply voltage U_S is not available.
	green illuminated	The supply voltage U_S is available.
U_P (peripheral voltage)	off	The supply voltage U_P is not available.
	green illuminated	The supply voltage U_P is available.

4.2.3 Conductor losses

Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage.
Variations in the voltage of the power supply unit must also be taken into account.

Voltage drop on the supply line



4.3 EtherCAT

4.3.1 Connectors

NOTICE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

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

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



Connection

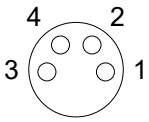


Fig. 4: M8 socket

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

[1)] Core colors according to EN 61918

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

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

4.3.2 Status LEDs



L/A (Link/Act)

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

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

Run

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

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

Description of the EtherCAT slave states

4.3.3 Cables

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

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

Detailed recommendations for the cabling of EtherCAT devices

4.4 RS232

4.4.1 Connector

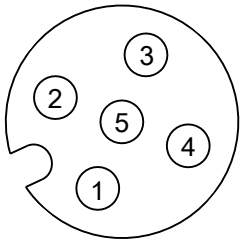


Fig. 5: M12 socket

M12 socket X01

Pin	Wire color	Signal	Description
1	brown	VCC	5 V_{DC} Supply voltage output
2	white	TxD	Send data
3	blue	GND_{ISO}	Ground
4	black	RxD	Receive data
5	gray	Shield	Shield

4.4.2 Status LEDs



Fig. 6: RS232 Status LEDs

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

4.5 RS422

4.5.1 Connectors

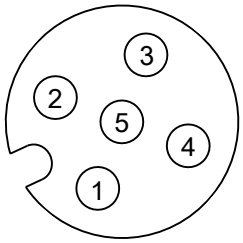


Fig. 7: M12 socket

M12 socket X02

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

M12 socket X03

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

4.5.2 Status LEDs

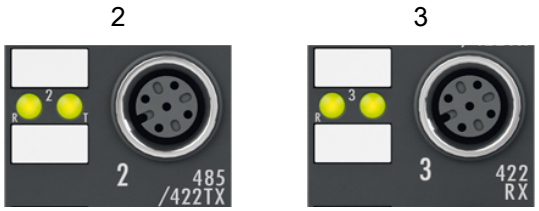


Fig. 8: RS422 Status LEDs

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

4.6 RS485

4.6.1 Connector

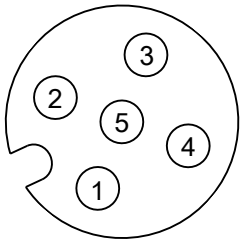


Fig. 9: M12 socket

M12 socket X02

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

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

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

RS485 bus structure - use termination resistors

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

4.6.2 Status LEDs



Fig. 10: RS485 Status LEDs

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

4.7 Digital combi channels

4.7.1 Connector

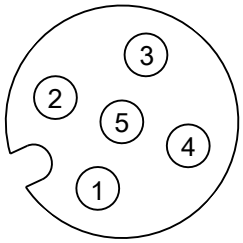


Fig. 11: M12 socket

M12 socket X04

Pin	Wire color	Signal	Description
1	brown	+24 V Us	Supply voltage
2	white	In-/Output B	Digital combi channel B
3	blue	GNDs	Ground
4	black	In-/Output A	Digital combi channel A
5	gray	reserved	-

4.7.2 Status LEDs



Fig. 12: Status LEDs for digital combi channels

LED	Display	Meaning
A	off	Digital combi channel A: low level
left	lights up green	Digital combi channel A: high level
B	off	Digital combi channel B: low level
right	lights up green	Digital combi channel B: high level

4.8 UL Requirements

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

Supply voltage

⚠ CAUTION

These 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

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

Networks

⚠ CAUTION

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

Ambient temperature range

⚠ CAUTION

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

Marking for UL

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



Fig. 13: UL label

4.9 Disposal



Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

5 Commissioning and configuration

5.1 Integrating into a TwinCAT project

The procedure for integration in a TwinCAT project is described in these Quick start guide.

5.2 Serial interface

5.2.1 Setting the interface type

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

Parameterization

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

5.2.1.1 RS232

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

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

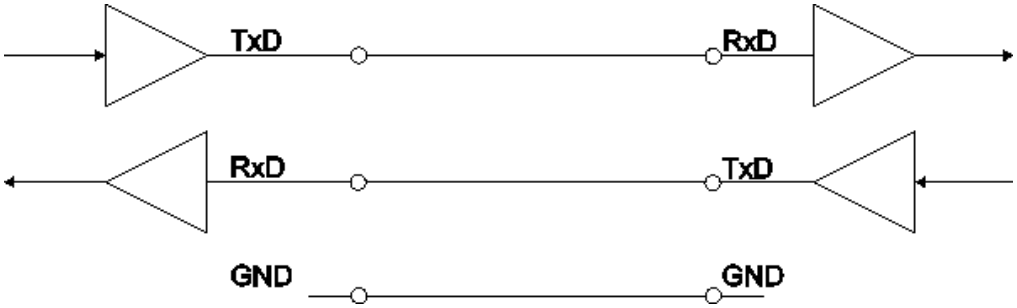


Fig. 14: Point-to-point connection to an RS232 device

The following CoE objects must be set

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

5.2.1.2 RS422

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

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

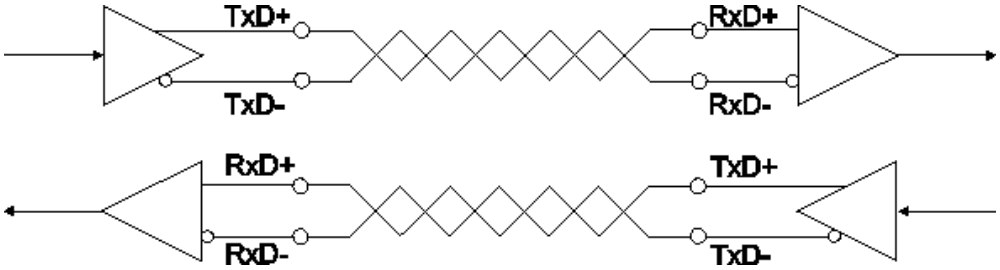


Fig. 15: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

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

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

5.2.1.3 RS485

You can operate the RS845 communication in two variants:

- Without diagnosis
- With diagnosis of the transmit data [▶ 36]

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

Bus structure, half duplex data transmission

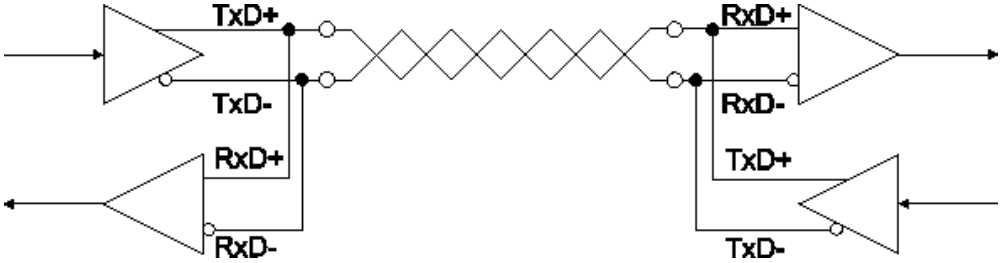


Fig. 16: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

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

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

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

Deactivated receive driver

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

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

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

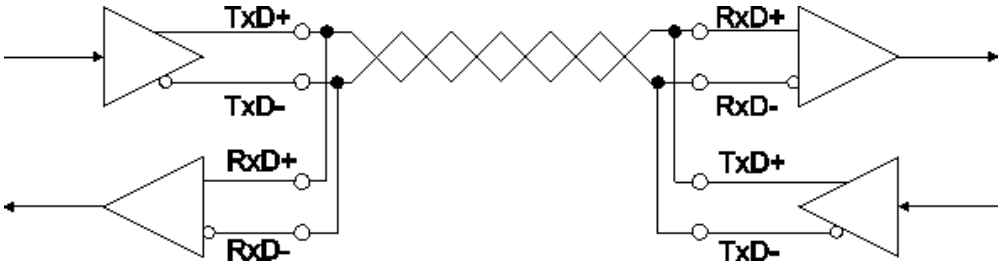


Fig. 17: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

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

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

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

Activated receive driver (from firmware version 03)

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

5.2.2 Setting the interface parameters

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

Parameterization via the CoE list (CAN over EtherCAT)

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

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

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

Fig. 18: CoE settings on object 0x8000 (default)

Continuous transmission of data

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

Optimization of transfer rates

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

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

5.2.3 Communication by PLC program

Initialization

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

Procedure:

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

⇨ The module is ready for data exchange.

Sending data

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

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

Receiving data

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

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

Prioritization

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

5.2.3.1 Samples

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

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

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

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

5.2.4 Communication via a virtual COM port

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

5.3 Digital combi channels

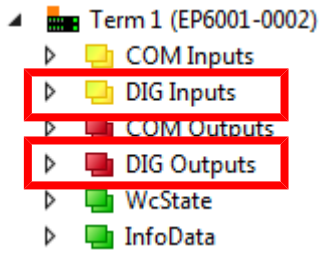
5.3.1 Activating process data

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

✓ Prerequisite: The EP6001-0002 has been added under **IO** in Solution Explorer.

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

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



5.3.2 Assignment of connector pins to process data

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

5.4 Restore the delivery state

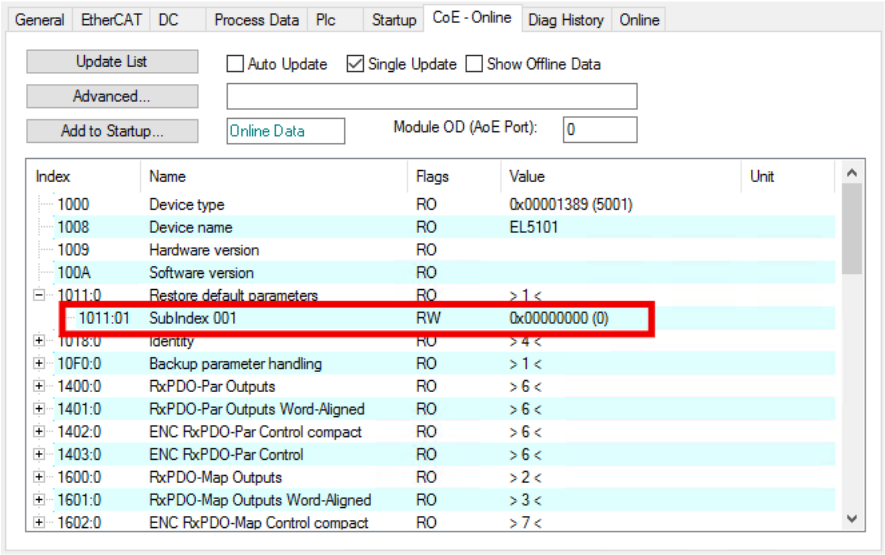
EtherCAT devices have two options for restoring the delivery status:

- **Function reset ("load")**: Resets changeable CoE parameters to their default values.
  This procedure is typically used during commissioning.
- **Data reset ("wipe")**:
  Deletes all user-specific data completely.
  This procedure is intended for a complete reset, e.g. when decommissioning.

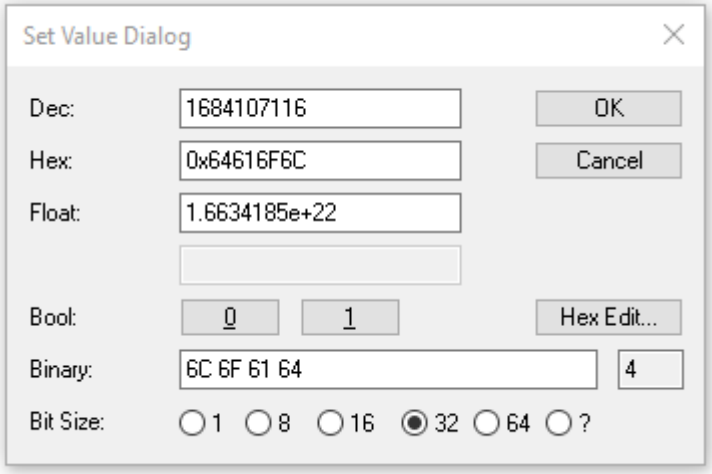
Function reset ("load")

You can restore the delivery status of all changeable CoE parameters with the function reset ("load") as follows:

1. Ensure that TwinCAT is in the CONFIG or RUN state and that EtherCAT communication is running error-free.
2. In CoE object 1011:0, "Restore default parameters" select parameter 1011:01."Subindex 001"



3. Double click on "Subindex 001".
   ⇨ The dialog box "Set Value Dialog" opens.
4. Enter the value 1684107116 in the field "Dec".
   Alternatively, enter the value 0x64616F6C in the field "Hex".



   For some EtherCAT devices with firmware version before approx. 2007, the value 1819238756 is required instead.
   Alternatively, enter the value 0x6C6F6164 in the field "Hex".

5. Confirm with "OK".

⇨ All changeable CoE parameters are reset to the default values.
Exception: CoE parameters that are protected by a write protection password.

Properties of the function reset ("load"):

- Depending on the EtherCAT devices, execution takes a few milliseconds to over 1 second.
- The execution is not confirmed.
- Entering an incorrect value has no effect.
- Depending on the device, individual CoE parameter values are not reset.
- Existing checksums for the parameter ranges will change.

Data reset ("wipe")

Proceed as for the function reset ("load"), but enter the value 2003398757 in the "Dec" field; alternatively, enter the value 0x77697065 in the "Hex" field.

Result:

- All user-specific data is deleted completely:
  - CoE
  - SoE
  - FoE
  - Register values
  - User memory
- Password protected data is also reset
- The following data will not be reset:
  - firmware
  - EtherCAT revision
  - persistent diagnostic data such as the object 0x1020 "Device Statistics", operating hours counter, peak values
  - manufacturer adjustment data

5.5 Decommissioning

⚠ WARNING

Risk of electric shock!

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

6 CoE parameters

Object directory

Index (hex)	Name
1000	Device type [▶ 47]
1008	Device name [▶ 47]
1009	Hardware version [▶ 47]
100A	Software version [▶ 47]
1011	Restore default parameters [▶ 45]
1018	Identity [▶ 47]
10F0	Backup parameter handling [▶ 47]
1400	COM RxPDO-Par Outputs [▶ 47]
1404	COM RxPDO-Par Outputs [▶ 47]
1600	COM RxPDO-Map Outputs [▶ 48]
1604	COM RxPDO-Map Outputs [▶ 49]
1608	DIG RxPDO-Map Outputs [▶ 49]
1800	COM TxPDO-Par Inputs [▶ 50]
1804	COM TxPDO-Par Inputs [▶ 50]
1A00	COM TxPDO-Map Inputs [▶ 51]
1A04	COM TxPDO-Map Inputs [▶ 52]
1A08	DIG TxPDO-Map Inputs [▶ 52]
1C00	Sync manager type [▶ 53]
1C12	RxPDO assign [▶ 53]
1C13	TxPDO assign [▶ 53]
1C32	SM output parameter [▶ 54]
1C33	SM input parameter [▶ 55]
6000	COM Inputs Ch. 1 [▶ 56]
6001	Status Ch. 1 [▶ 56]
6010	DIG Inputs [▶ 56]
7000	COM Outputs Ch. 1 [▶ 57]
7001	Ctrl Ch. 1 [▶ 57]
7010	DIG Outputs [▶ 57]
8000	COM Settings Ch. 1 [▶ 45]
A000	COM Diag data Ch. 1 [▶ 57]
F000	Modular device profile [▶ 58]
F008	Code word [▶ 58]
F010	Module list [▶ 58]
F800	COM Settings [▶ 46]

6.1 Configuration objects

Index 1011 Restore default parameters

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

Index 8000 COM Settings Ch.1

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

Index F800 COM Settings

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

6.2 Standard objects

Index 1000 Device type

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

Index 1008 Device name

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

Index 1009 Hardware version

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

Index 100A Software Version

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

Index 1018 Identity

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

Index 10F0 Backup parameter handling

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

Index 1400 COM RxPDO-Par Outputs

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

Index 1404 COM RxPDO-Par Outputs

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

Index 1600 COM RxPDO-Map Outputs

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

Index 1604 COM RxPDO-Map Outputs

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

Index 1608 DIG RxPDO-Map Outputs

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

Index 1800 COM TxPDO-Par Inputs

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

Index 1804 COM TxPDO-Par Inputs

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

Index 1A00 COM TxPDO-Map Inputs

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

Index 1A04 COM TxPDO-Map Inputs

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

Index 1A08 DIG TxPDO-Map Inputs

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

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

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

Index 1C13 TxPDO assign

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

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0003D090 (250000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 54])	UINT16	RO	0x0001 (1_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03 [▶ 54], 1C32:05 [▶ 54], 1C32:06 [▶ 54], 1C32:09 [▶ 54], 1C33:03 [▶ 55], 1C33:06 [▶ 54], 1C33:09 [▶ 55] 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 [▶ 54]	UINT32	RW	0x0003D090 (250000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 54] or 1C33:08 [▶ 55])	UINT16	RO	0x0001 (1_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 54]	UINT32	RO	0x0003D090 (250000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C33:08	Command	as 1C32:08 [▶ 54]	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 [▶ 54]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 54]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▶ 54]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▶ 54]	BOOLEAN	RO	0x00 (0_{dec})

6.3 Profile-specific objects

Index 6000 COM Inputs Ch.1

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

Index 6001 Status Ch.1

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

Index 6010 DIG Inputs

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

Index 7000 COM Outputs Ch.1

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

Index 7001 Ctrl Ch.1

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

Index 7010 DIG Outputs

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

Index A000 COM Diag data Ch.1

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

Index F000 Modular device profile

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

Index F008 Code word

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

Index F010 Module list

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

7 Appendix

7.1 General operating conditions

Protection rating according to IP code

The degrees of protection are defined and divided into different classes in the IEC 60529 standard (EN 60529). Degrees of protection are designated by the letters "IP" and two numerals: **IPxy**

- Numeral x: Dust protection and contact protection
- Numeral y: Protection against water

x	Meaning
0	Not protected
1	Protected against access to dangerous parts with the back of the hand. Protected against solid foreign objects of 50 mm Ø
2	Protected against access to dangerous parts with a finger. Protected against solid foreign objects of 12.5 mm Ø
3	Protected against access to dangerous parts with a tool. Protected against solid foreign objects of 2.5 mm Ø
4	Protected against access to dangerous parts with a wire. Protected against solid foreign objects of 1 mm Ø
5	Protection against access to dangerous parts with a wire. Dust-protected. Ingress of dust is not prevented completely, although the quantity of dust able to penetrate is limited to such an extent that the proper function of the device and safety are not impaired
6	Protection against access to dangerous parts with a wire. Dust-tight. No ingress of dust

y	Meaning
0	Not protected
1	Protection against vertically falling water drops
2	Protection against vertically falling water drops when enclosure tilted up to 15°
3	Protection against spraying water. Water sprayed at an angle of up to 60° on either side of the vertical shall have no harmful effects
4	Protection against splashing water. Water splashed against the enclosure from any direction shall have no harmful effects
5	Protection against water jets.
6	Protection against powerful water jets.
7	Protected against the effects of temporary immersion in water. Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is immersed in water at a depth of 1 m for 30 minutes

Chemical resistance

The resistance refers to the housing of the IP67 modules and the metal parts used. In the table below you will find some typical resistances.

Type	Resistance
Water vapor	unstable at temperatures > 100 °C
Sodium hydroxide solution (ph value > 12)	stable at room temperature unstable > 40 °C
Acetic acid	unstable
Argon (technically pure)	stable

Key

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

7.2 Accessories

Mounting

Ordering information	Description	Link
ZS5300-0011	Mounting rail	Website

Cables

A complete overview of pre-assembled cables can be found on the Beckhoff website: Link.

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

Labeling material, protective caps

Ordering information	Description	Link
ZS5000-0010	Blanking plug for M8 sockets, IP67 (50 pieces)	Website
ZS5000-0020	Blanking plug for M12 sockets, IP67 (50 pieces)	Website
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10	Data sheet
ZS5000-xxxx	Printed inscription labels on enquiry	n/a

Tools

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

Ordering information	Description
ZB8801-0003	Torque cable key for M12 field assembly / wrench size 18 for ZB8801-0000

Further accessories

You can find further accessories at https://www.beckhoff.com/accessories.

7.3 Version identification of EtherCAT devices

7.3.1 General notes on marking

Name

A Beckhoff EtherCAT device has a 14-digit technical designation consisting of

- Family key
- Type
- Version
- Revision

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

Notes

- The elements stated above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, for “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is composed of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (0000)
- The **revision** -0016 reflects the technical progress such as feature enhancement with regard to EtherCAT communication and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless otherwise specified, e.g. in the documentation.
  There is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file which is associated and synonymous with each revision, and this is available for download from the Beckhoff website.
  Since January 2014, the revision has been marked on the outside of the IP20 terminals; see Fig. *“EL2872 with revision 0022 and serial number 01200815.”*
- The type, version, and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

7.3.2 Version identification of IP67 modules

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

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

KK – week of production (calendar week)
YY – year of production
FF – firmware version
HH – hardware version

Example with serial number 12 06 3A 02:

12 – week of production 12
06 – year of production 2006
3A – firmware version 3A
02 – hardware version 02

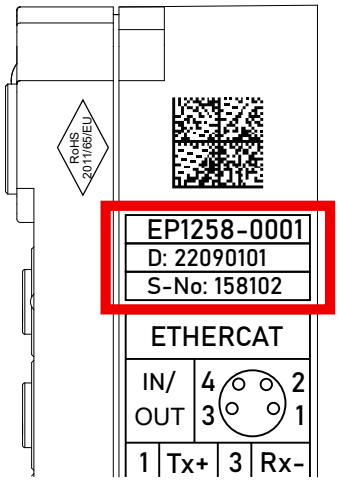


Fig. 19: EP1258-00001 IP67 EtherCAT Box with batch number/DateCode 22090101 and unique serial number 158102

7.3.3 Beckhoff Identification Code (BIC)

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

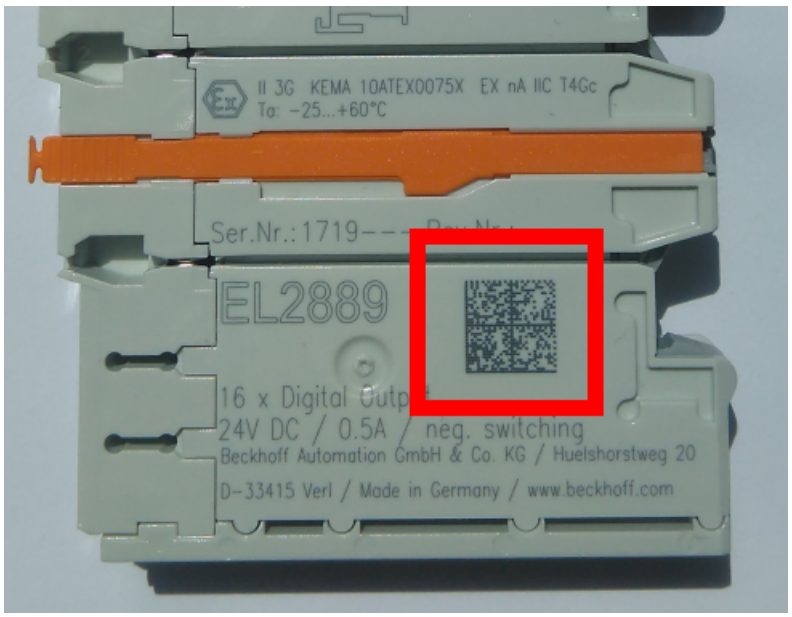


Fig. 20: BIC as a Data Matrix code (DMC, ECC200 code scheme)

The BIC is being introduced gradually across all product groups.
Depending on the product, it can be found in the following locations:

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

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

Each item of information is clearly identifiable with the data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length stated by the following table. If the items of information are shorter, spaces are added.

The following information may be present; items 1 through 4 are always included, while the others depend on the product family's requirements:

Item no.	Type of information	Explanation	Data identifier	Number of characters including the data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Previous serial number system, e.g., for safety products or calibrated terminals	51S	12	**51S**678294
7	Variant	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 2*K183
...					

Beckhoff uses additional types of information and data identifiers for internal processes.

Structure of the BIC

Example of composite information consisting of items 1 through 4 and the example value mentioned above in item 6. The data identifiers are highlighted in bold:

1P072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

The same number as a DMC:



Fig. 21: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

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 will replace all other serial number systems at Beckhoff in the future (e.g. batch designations on IO components, previous serial number circle for safety products, etc.). The BTN is also being introduced gradually, so it may not yet be coded in the BIC.

NOTICE

This information has been carefully prepared. However, the method described is constantly being refined. We reserve the right to revise and change procedures and documentation at any time and without notice. No claims for modification may be made on the basis of the data, diagrams, and descriptions in this documentation.

7.3.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

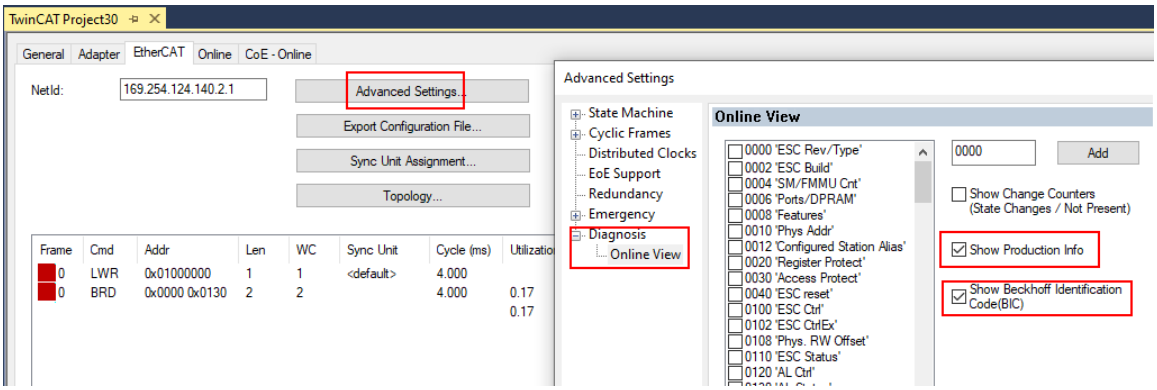
EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have an ESI-EEPROM which contains the EtherCAT identity with the revision number. The EtherCAT slave information, also colloquially known as the ESI/XML configuration file for the EtherCAT master, is stored in it. See the corresponding chapter in the EtherCAT system manual (Link) for the relationships.

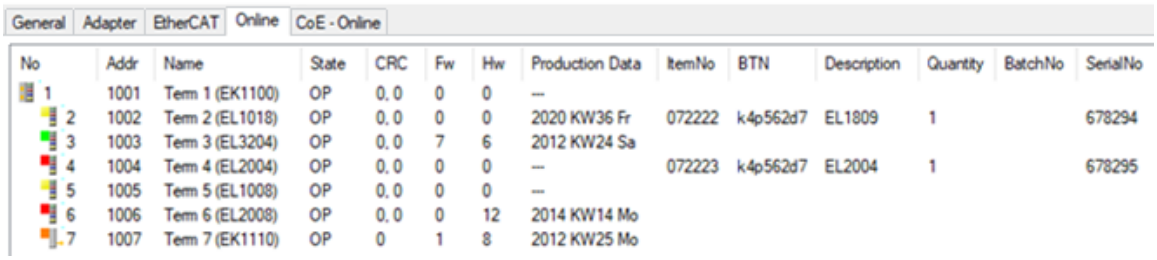
Beckhoff also writes the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM:
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
  - To do this, check the “Show Beckhoff Identification Code (BIC)” checkbox under EtherCAT → Advanced Settings → Diagnostics:

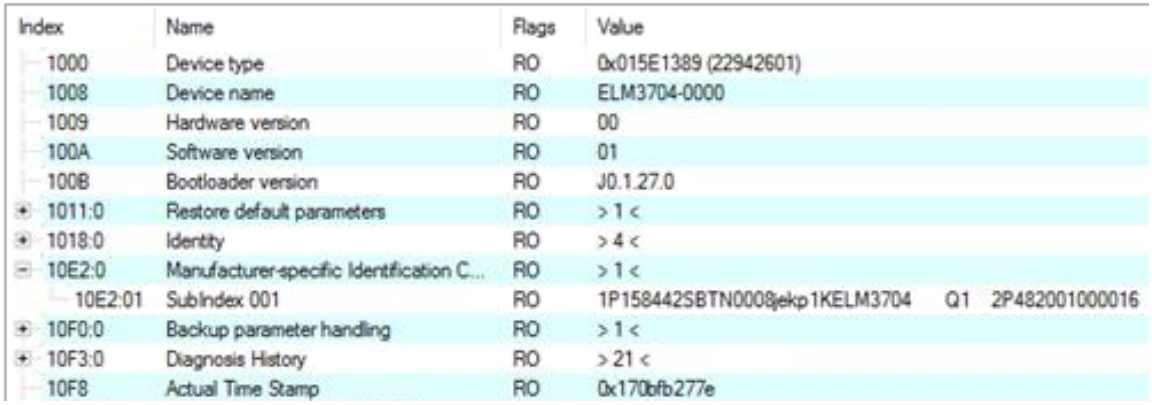


  - The BTN and its contents are then displayed:



  - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with “Show production info”.
  - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- The device must be in PREOP/SAFEOP/OP for access



Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the BIC sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic processing, the BTN is to be handled as a string(8); the identifier “SBTN” is not part of the BTN.
- On the technical background:
  The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50… 200 bytes of memory.
- Special cases
  - In a hierarchical configuration of multiple ESCs (EtherCAT Slave Controllers) within a single device, only the top-level ESC carries the eBIC information.
  - If multiple non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC information.
  - If the device consists of several sub-devices which each have their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of that ESC is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

7.4 Support and Service

Beckhoff Support and Service provides assistance with questions regarding Beckhoff products and system solutions, as well as with their planning, commissioning, and operation.

Beckhoff subsidiaries and representative offices

Please contact your Beckhoff subsidiary or your representative office for local support and service for Beckhoff products. You can find the addresses of Beckhoff subsidiaries and representative offices at: www.beckhoff.com/worldwide.

For Beckhoff products and technologies, we offer Training programs, tutorials, and webinars.

Beckhoff Support

Beckhoff Support provides assistance with technical questions regarding the use, planning, programming, and commissioning of Beckhoff products.

☎ Hotline:	+49 5246 963-157
✉ Email:	support@beckhoff.com, Support form
Web page:	www.beckhoff.com/support

Beckhoff Service

Our service specialists are here to assist you with any questions you may have regarding after-sales service processes, such as replacement parts, repairs, etc.

☎ Hotline:	+49 5246 963-460
✉ Email:	service@beckhoff.com, Service form
Web page:	www.beckhoff.com/service

Download finder

The Download finder serves as a central source for downloadable documents and files related to Beckhoff products, such as application reports, technical documentation, drawings, and configuration files.

Beckhoff Information System

The Beckhoff Information System is the central online documentation resource for Beckhoff products and contains documentation on products and automation topics, as well as sample code.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl, Germany

☎ Phone:	+49 5246 963-0
✉ Email:	info@beckhoff.com
Web page:	www.beckhoff.com

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com/ep6001-0002

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