BECKHOFF New Automation Technology

Documentation | EN

EPP6002-0002

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

2023-02-01 | Version: 1.3

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER

Serious risk of injury!

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

⚠ WARNING

Risk of injury!

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

⚠ CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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

ℹ **Tip or pointer**

This symbol indicates information that contributes to better understanding.

1.3 Documentation Issue Status

Version	Comment
1.3	• EtherCAT P status LEDs updated
1.2	• Dimensions updated • UL requirements updated
1.1	• New functions of firmware 04 und 05 added • Technical data updated • Structure update
1.0	• First release

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.

Documentation	Firmware	Hardware
1.1	05	04
1.0	03	04

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

Syntax of the batch number (D-number)

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

Further information on this topic: Version identification of EtherCAT devices [▶ 69].

2 Product group: EtherCAT P Box modules

EtherCAT P

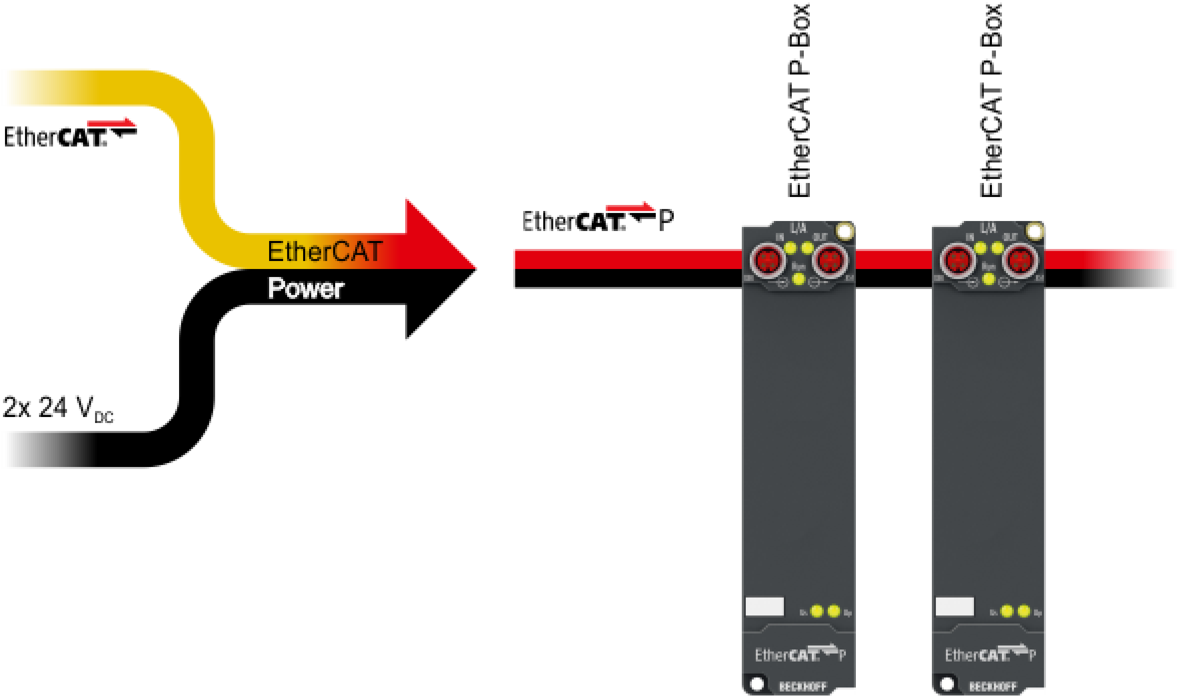
EtherCAT P supplements the EtherCAT technology with a process in which communication and supply voltages are transmitted on a common line. All EtherCAT properties are retained with this process.

Two supply voltages are transmitted per EtherCAT P line. The supply voltages are electrically isolated from each other and can therefore be switched individually. The nominal supply voltage for both is 24 V_{DC}.

EtherCAT P uses the same cable structure as EtherCAT: a 4-core Ethernet cable with M8 connectors. The connectors are mechanically coded so that EtherCAT connectors and EtherCAT P connectors cannot be interchanged.

EtherCAT P Box modules

EtherCAT P Box modules are EtherCAT P slaves with degree of protection IP67. They are designed for operation in wet, dirty or dusty industrial environments.



EtherCAT basics

A detailed description of the EtherCAT system can be found in the EtherCAT system documentation.

3 Product overview

3.1 Introduction

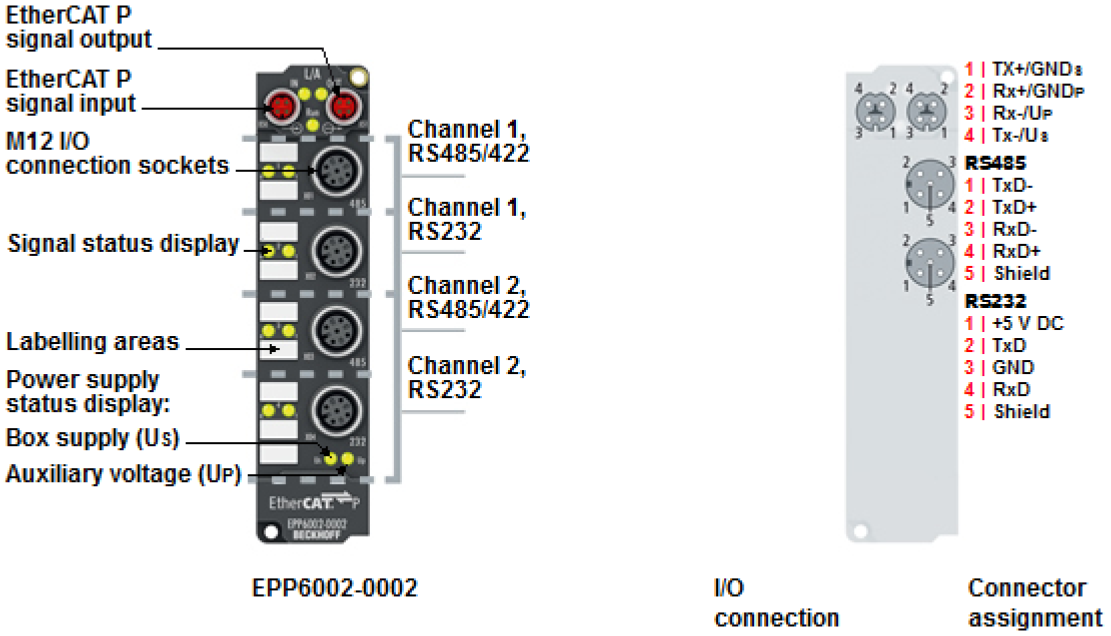


Fig. 1: EPP6002-0002

2-channel serial interface, RS232, RS422/RS485

The EPP6002-0002 serial interface module allows the connection of devices with an RS232 or RS422/RS485 interface. The module transmits the data in a fully transparent manner to the higher-level automation device. The active serial communication channel functions independently of the higher-level bus system in full duplex mode at up to 115,200 baud. The connector assignment depends on the interface. For each channel, RS232 or RS422/RS485 can be selected. In conjunction with the TwinCAT Virtual Serial COM Driver the EPP6002-0002 can be used as a normal Windows COM interface.

Quick links

Technical data
Process image [▶ 12]
Dimensions
RS232 Connection [▶ 24]
RS485/RS422 connection [▶ 25]
Commissioning [▶ 28]

3.2 Technical data

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

Technical data	EPP6002-0002
Fieldbus	
Fieldbus	EtherCAT
Connection	EtherCAT P: Combined connection for EtherCAT and supply voltages Input: 1 x M8 socket, 4-pin, P-coded Downstream connection: 1 x M8 socket, 4-pin, P-coded
Supply voltages	
Connection	See Fieldbus connection
Control voltage U_S	
Nominal voltage	24 V_{DC} (-15 % / +20 %)
Sum current	max. 3 A [1]
Current consumption from U_S	100 mA at 24 V_{DC}
Further consumers	Communication end devices (e.g. barcode scanners)
Peripheral voltage U_P	
Nominal voltage	24 V_{DC} (-15 % / +20 %)
Sum current	max. 3 A [1]
Current consumption from U_P	None. U_P is only forwarded.
Serial interface	
Number of channels	2
Interface type	Individually parameterizable for each channel: • RS232 • RS422 • RS485
Connection	1x M12 socket per channel
Remote station supply voltage [2]	5 V_{DC}, isolated potential max. 20 mA
Cable length	RS232: max. 15 m RS422/RS485: max. 1,000 m
Data transfer rate	Parameterizable: 300 .. 115,200 baud (bit/s)
Data format	Parameterizable: 8N1, 7E1, 7O1, 8N1, 8E1, 8O1, 7E2, 7O2, 8N2, 8E2, 8O2
Bit distortion	< 3 %
Receive buffer	From firmware 04: 1024 bytes Up to firmware 03: 864 bytes
Transmit buffer	From firmware 04: 1024 bytes Up to firmware 03: 128 bytes

[1] Sum current of consumers and power transmission.

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

Technical data	EPP6002-0002
Environmental conditions	
Ambient temperature during operation	-25…+60 °C -25…+55 °C according to cULus
Ambient temperature during storage	-40…+85 °C
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 Additional checks [▶ 11]
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 (conforms to EN 60529)
Mechanics	
Weight	approx. 165 g
Installation position	variable
Approvals and conformity	
Approvals	CE, cULus [▶ 26]

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

3.3 Scope of supply

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

- 1x EtherCAT P Box EPP6002-0002
- 2x protective cap for EtherCAT P socket, M8, red (pre-assembled)
- 10x labels, blank (1 strip of 10)

Pre-assembled protective caps do not ensure IP67 protection

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

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

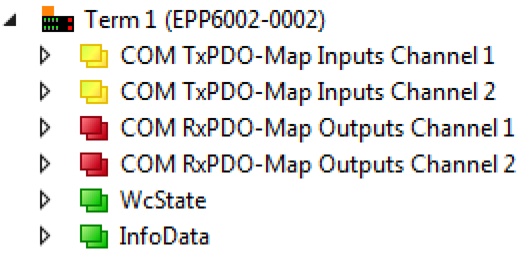
3.4 Process image

The size of the process image can be set from firmware 04 onwards.

This chapter describes the process image in the factory setting. It contains 22 bytes of receive data and 22 bytes of send data.

Among other things, the size of the process image determines the maximum continuous data transfer rate [▶ 17]. If your application requires a higher data transfer rate, you can set the size of the process image [▶ 16]from firmware 04 onwards.

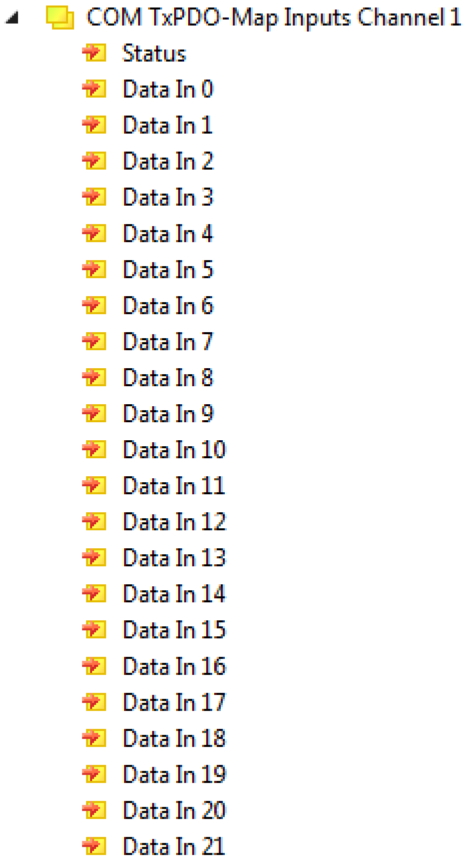
Complete process image



Term 1 (EPP6002-0002)
- COM TxPDO-Map Inputs Channel 1
- COM TxPDO-Map Inputs Channel 2
- COM RxPDO-Map Outputs Channel 1
- COM RxPDO-Map Outputs Channel 2
- WcState
- InfoData

COM TxPDO-Map Inputs

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



COM TxPDO-Map Inputs Channel 1
- Status
- Data In 0
- Data In 1
- Data In 2
- Data In 3
- Data In 4
- Data In 5
- Data In 6
- Data In 7
- Data In 8
- Data In 9
- Data In 10
- Data In 11
- Data In 12
- Data In 13
- Data In 14
- Data In 15
- Data In 16
- Data In 17
- Data In 18
- Data In 19
- Data In 20
- Data In 21

Status

Status word [▶ 15] for receive data.

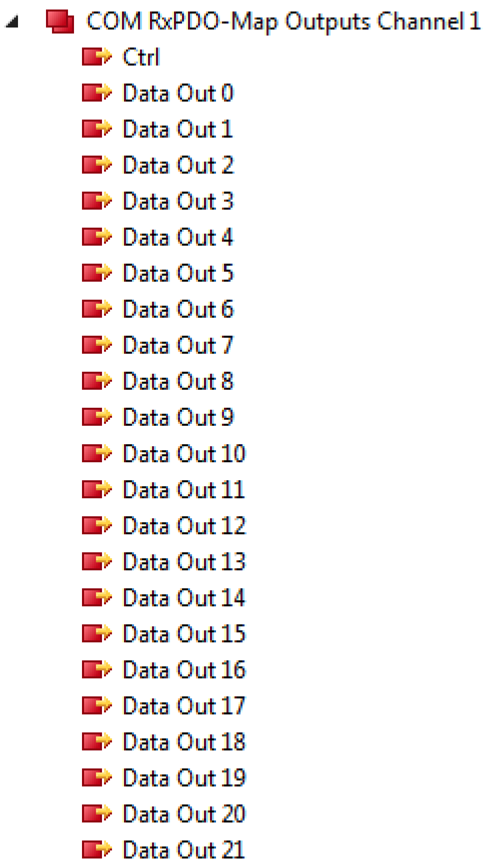
Data In [n]

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

„Data In 0“ contains the first-received byte.

COM RxPDO-Map Outputs

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



COM RxPDO-Map Outputs Channel 1
- Ctrl
- Data Out 0
- Data Out 1
- Data Out 2
- Data Out 3
- Data Out 4
- Data Out 5
- Data Out 6
- Data Out 7
- Data Out 8
- Data Out 9
- Data Out 10
- Data Out 11
- Data Out 12
- Data Out 13
- Data Out 14
- Data Out 15
- Data Out 16
- Data Out 17
- Data Out 18
- Data Out 19
- Data Out 20
- Data Out 21

Ctrl

Control word [▸ 14] for transmit data.

Data Out [n]

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

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

3.4.1 Assignment of connectors to process data

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

3.4.2 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. Note: If you have set the size of the send data [▶ 16] to 50 x 1 word, the output variables “Data Out *n*” have the data type “Word”. A Word has a size of two bytes. Multiply the number of output variables to be sent by two in order to calculate the number of output bytes.
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.

3.4.3 Status word

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

Key

Bit no.	Name	Description	
15 ... 8	IL7 ... IL0 (InLength)	1_{dec} .. 22_{dec}	The number of input bytes available for the transmission from the box to the controller. Note: If you have set the size of the receive data [▶ 16] to 50 x 1 word, the input variables “Data In *n*” have the data type “Word”. A Word has a size of two bytes. Divide the number of input bytes by two in order to determine the number of filled input variables.
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.

3.4.4 Setting the size of the process image

From firmware 04 onwards you can set the size of the process image, for example in order to increase the maximum continuous data transfer rate [▶ 17].

✓ Requirement: EPP6002 is integrated in a TwinCAT project (Procedure [▶ 28]).

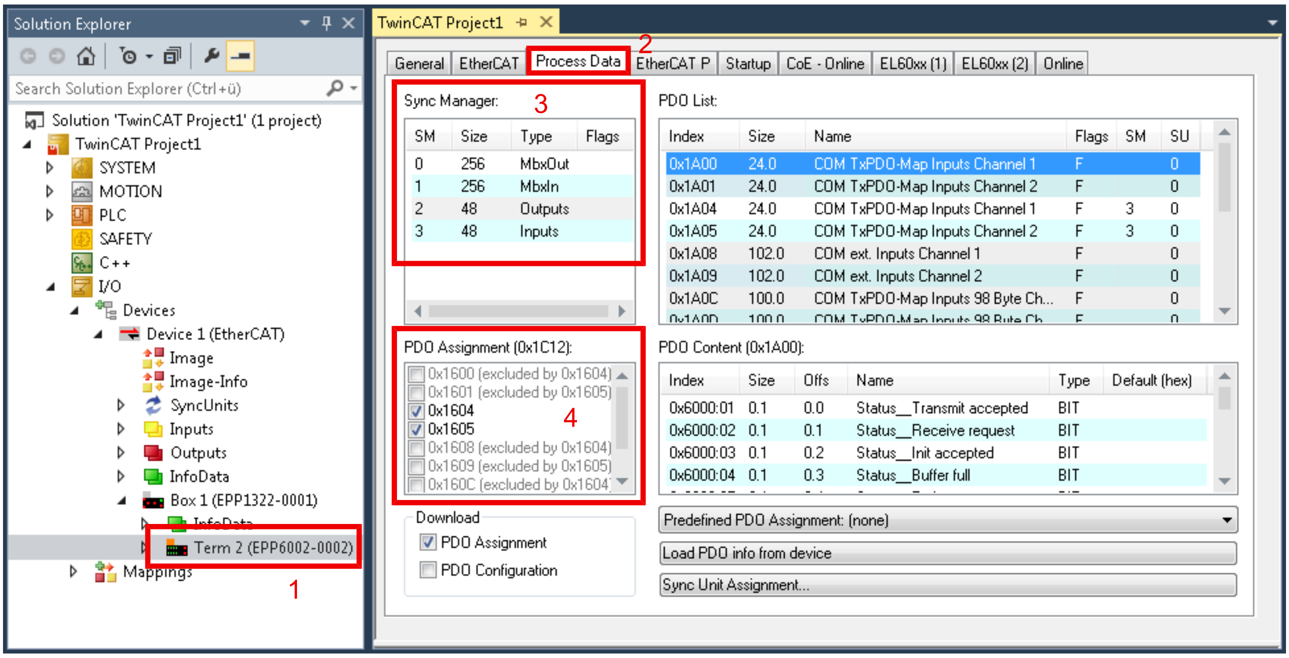
1. In the Solution Explorer: double-click the module "EPP6002-0002".
2. Click on the “Process Data” tab.
3. In the box “Sync Manager”: click on the “Outputs” entry
4. In the box “PDO Assignment (0x1C12)”: uncheck all checkboxes.

   ⇨ The other entries in this box are no longer grayed out.

5. Depending on the desired size of the process image, check the checkboxes next to the following entries:
   - for 22 x 1 byte send data: check the checkboxes next to 0x1604 and 0x1605 (factory setting).
   - for 98 x 1 byte send data: check the checkboxes next to 0x160C and 0x160D.
   - for 50 x 1 Word send data: check the checkboxes next to 0x1608 and 0x1609.
6. In the box “Sync Manager”: click the entry “Inputs”
7. In the box “PDO Assignment (0x1C13)”: uncheck all checkboxes.

   ⇨ The other entries in this box are no longer grayed out.

8. Depending on the desired size of the process image, check the checkboxes next to the following entries:
   - for 22 x 1 byte receive data: check the checkboxes next to 0x1A04 and 0x1A05 (factory setting).
   - for 98 x 1 byte receive data: check the checkboxes next to 0x1A0C, 0x1A0D
   - for 50 x 1 Word receive data: check the checkboxes next to 0x1A08, 0x1A09



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	48	Outputs	
3	48	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	24.0	COM TxPDO-Map Inputs Channel 1	F		0
0x1A01	24.0	COM TxPDO-Map Inputs Channel 2	F		0
0x1A04	24.0	COM TxPDO-Map Inputs Channel 1	F	3	0
0x1A05	24.0	COM TxPDO-Map Inputs Channel 2	F	3	0
0x1A08	102.0	COM ext. Inputs Channel 1	F		0
0x1A09	102.0	COM ext. Inputs Channel 2	F		0
0x1A0C	100.0	COM TxPDO-Map Inputs 98 Byte Ch...	F		0
0x1A0D	100.0	COM TxPDO-Map Inputs 98 Byte Ch	F		0

PDO Assignment (0x1C12):

☐ 0x1600 (excluded by 0x1604)
☐ 0x1601 (excluded by 0x1605)
☑ 0x1604
☑ 0x1605
☐ 0x1608 (excluded by 0x1604)
☐ 0x1609 (excluded by 0x1605)
☐ 0x160C (excluded by 0x1604)

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6000:01	0.1	0.0	Status__Transmit accepted	BIT	
0x6000:02	0.1	0.1	Status__Receive request	BIT	
0x6000:03	0.1	0.2	Status__Init accepted	BIT	
0x6000:04	0.1	0.3	Status__Buffer full	BIT	

Download
☑ PDO Assignment
☐ PDO Configuration

Predefined PDO Assignment: (none)

3.5 Technology

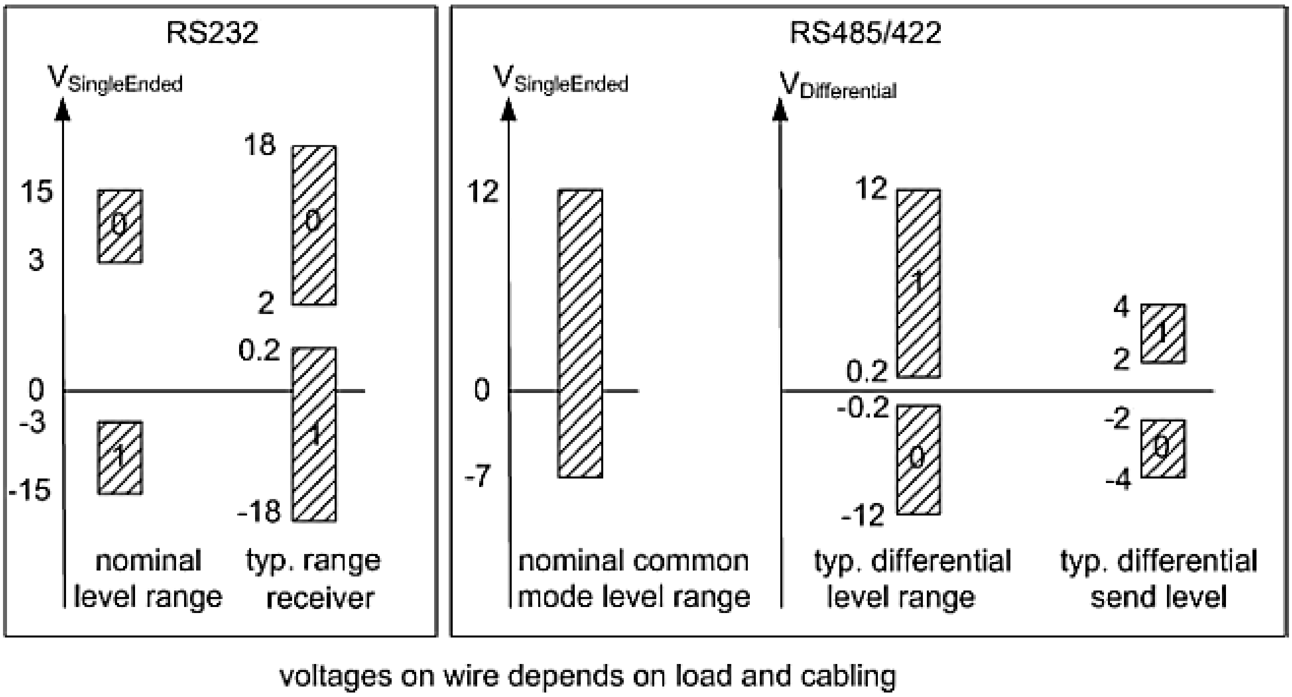


Fig. 2: Level of RS232, RS422, RS485 interfaces

Data transfer rate

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

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

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

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

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

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

can be achieved.

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

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

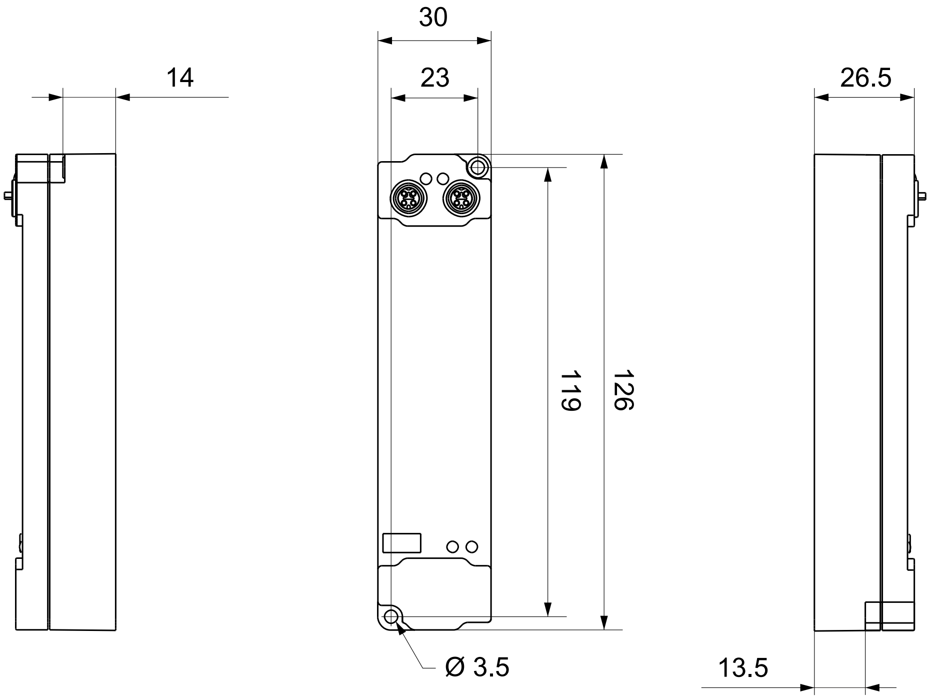
If the controller cannot fetch the data quickly enough from the box, they are buffered in the box's receive buffer. All further data are lost if the receive buffer is full.

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

4 Mounting and connection

4.1 Mounting

4.1.1 Dimensions



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

Housing features

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

4.1.2 Fixing

NOTE

Dirt during assembly

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

- Protect the plug connectors against dirt during the assembly.

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

4.1.3 Functional earth (FE)

The upper mounting holes also serves as a connection for functional earth (FE).

Make sure that the box is grounded to low impedance via the functional earth (FE) connection. You can achieve this, for example, by mounting the box on a grounded machine bed.

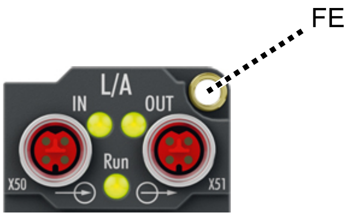


Fig. 3: Connection for functional earth (FE)

4.1.4 Tightening torques for plug connectors

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

Connector diameter	Tightening torque
M8	0.4 Nm
M12	0.6 Nm

4.2 Connection

4.2.1 EtherCAT P

⚠ **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 the EtherCAT P Power Sourcing Device (PSD).

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 (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor,
  a PELV (Protective Extra Low Voltage) 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 [▶ 26].

EtherCAT P transmits 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 P Box has inputs.
- **Peripheral voltage U_P**
  The digital outputs are typically supplied from the peripheral voltage U_P for EtherCAT P Box modules with digital outputs. 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 supply voltages are passed on internally from the "IN" connection to the "OUT" connection. Hence, the supply voltages U_S and U_P can be passed from one EtherCAT P Box to the next EtherCAT P Box in a simple manner.

NOTE

Note the maximum current.

Ensure that the maximum permitted current of 3 A for the M8 connectors is not exceeded when redirecting EtherCAT P.

4.2.1.1 Connectors

NOTE

Risk of damage to the device!

Bring the EtherCAT/EtherCAT P system into a safe, powered down state before starting installation, disassembly or wiring of the modules!

Two M8 sockets at the upper end of the modules are provided for supply and downstream connection of EtherCAT P:

- IN: left M8 socket for EtherCAT P supply
- OUT: right M8 socket for downstream connection of EtherCAT P

The metal threads of the M8 EtherCAT P sockets are internally linked to the FE connection via high impedance RC combination. See chapter Functional earth (FE) [▶ 19].

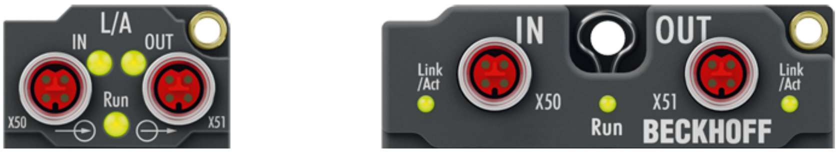


Fig. 4: Connectors for EtherCAT P

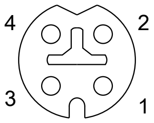


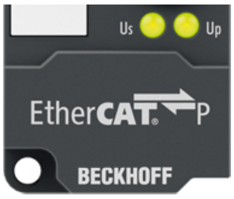
Fig. 5: M8 socket, p-coded

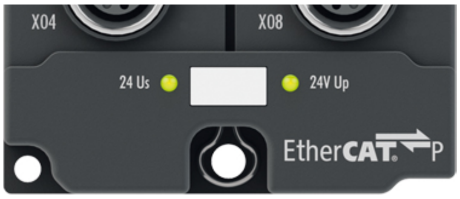
Contact	Signal	Voltage	Core color 1)
1	Tx +	GND$_S$	yellow
2	Rx +	GND$_P$	white
3	Rx -	U$_P$: peripheral voltage, +24 V$_{DC}$	blue
4	Tx -	U$_S$: control voltage, +24 V$_{DC}$	orange
Housing	Shield	Shield	Shield

1) The core colors apply to EtherCAT P cables and ECP cables from Beckhoff.

4.2.1.2 Status LEDs

4.2.1.2.1 Supply voltages

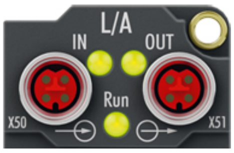


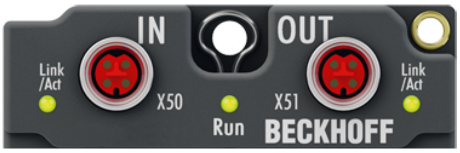


EtherCAT P Box modules indicate the status of the supply voltages via two status LEDs. The status LEDs are labeled with the designations of the supply voltages: Us and Up.

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

4.2.1.2.2 EtherCAT





L/A (Link/Act)

A green LED labeled "L/A" or "Link/Act" is located next to each EtherCAT/EtherCAT P 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

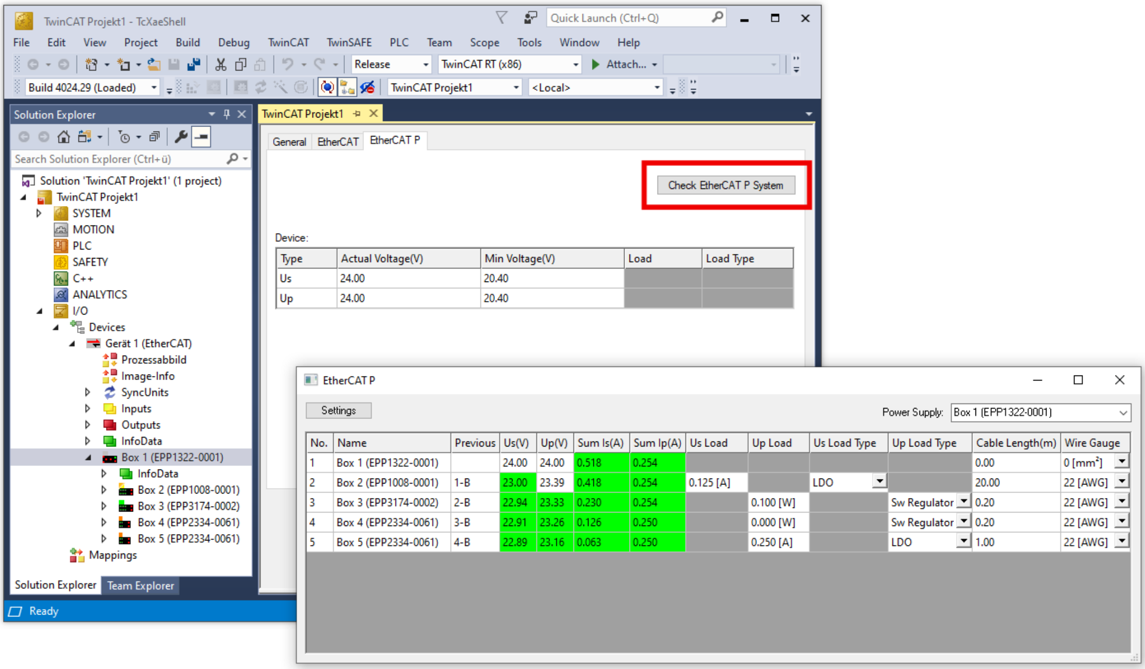
Description of the EtherCAT slave states

4.2.1.3 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.

Planning tool for EtherCAT P

You can plan cable lengths, voltages and currents of your EtherCAT P system using TwinCAT 3. The requirement for this is TwinCAT 3 Build 4020 or higher.



Device:

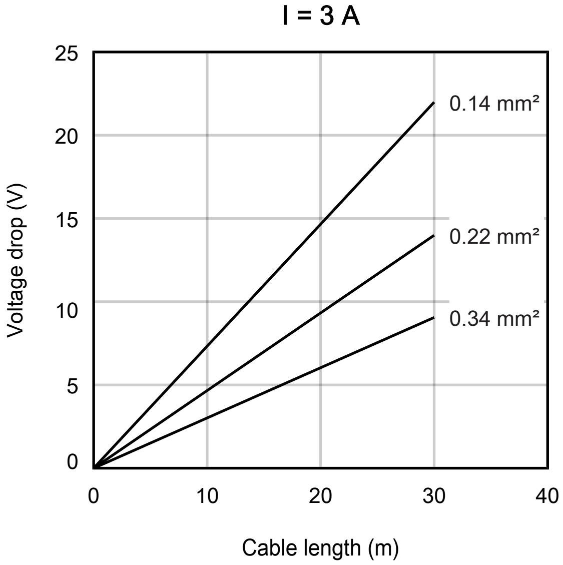
Type	Actual Voltage(V)	Min Voltage(V)	Load	Load Type
Us	24.00	20.40		
Up	24.00	20.40		

Power Supply: Box 1 (EPP1322-0001)

No.	Name	Previous	Us(V)	Up(V)	Sum Is(A)	Sum Ip(A)	Us Load	Up Load	Us Load Type	Up Load Type	Cable Length(m)	Wire Gauge
1	Box 1 (EPP1322-0001)		24.00	24.00	0.518	0.254					0.00	0 [mm²]
2	Box 2 (EPP1008-0001)	1-B	23.00	23.39	0.418	0.254	0.125 [A]		LDO		20.00	22 [AWG]
3	Box 3 (EPP3174-0002)	2-B	22.94	23.33	0.230	0.254		0.100 [W]		Sw Regulator	0.20	22 [AWG]
4	Box 4 (EPP2334-0061)	3-B	22.91	23.26	0.126	0.250		0.000 [W]		Sw Regulator	0.20	22 [AWG]
5	Box 5 (EPP2334-0061)	4-B	22.89	23.16	0.063	0.250		0.250 [A]		LDO	1.00	22 [AWG]

Further information can be found in the quick start guide IO configuration in TwinCAT in chapter "Configuration of EtherCAT P via TwinCAT".

Voltage drop on the supply line



4.2.2 RS232

4.2.2.1 Connectors

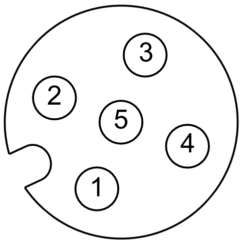


Fig. 6: M12 socket

M12 sockets X02 and X04

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

4.2.2.2 Status LEDs

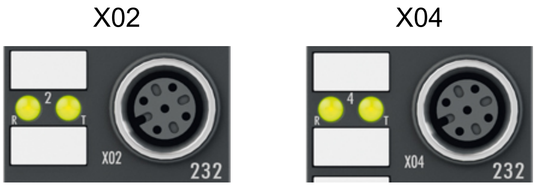


Fig. 7: RS232 Status LEDs

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

4.2.3 RS422 / RS485

4.2.3.1 Connectors

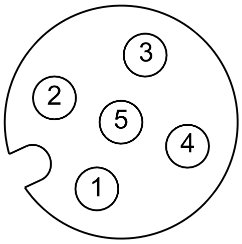


Fig. 8: M12 socket

M12 sockets X01 and X03

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

RS485 bus structure - use termination resistors

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

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

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

4.2.3.2 Status LEDs

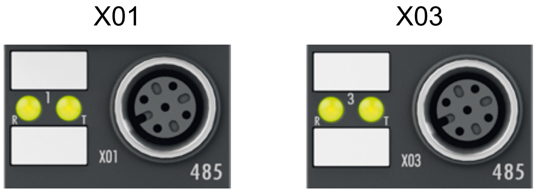


Fig. 9: RS422 / RS485 Status LEDs

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

4.3 UL Requirements

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

Supply voltage

⚠ CAUTION

CAUTION!

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

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

⚠ CAUTION

CAUTION!

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

Networks

⚠ CAUTION

CAUTION!

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

Ambient temperature range

⚠ CAUTION

CAUTION!

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

Marking for UL

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

Fig. 10: UL label

4.4 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/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 Configuring serial interfaces

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).

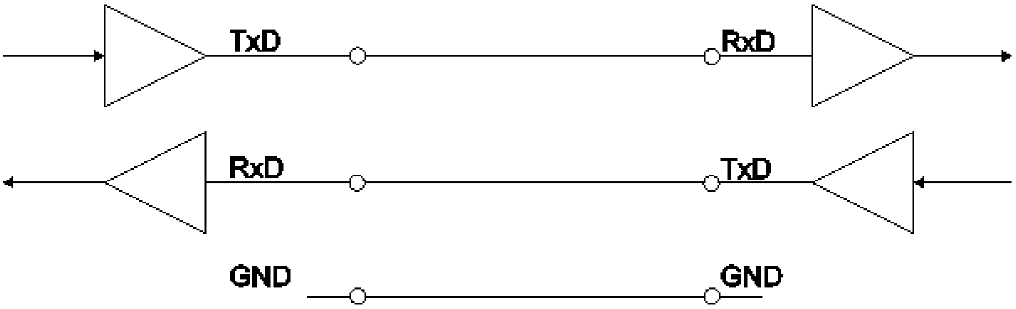


Fig. 11: Point-to-point connection to an RS232 device

The following CoE objects must be set

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

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.

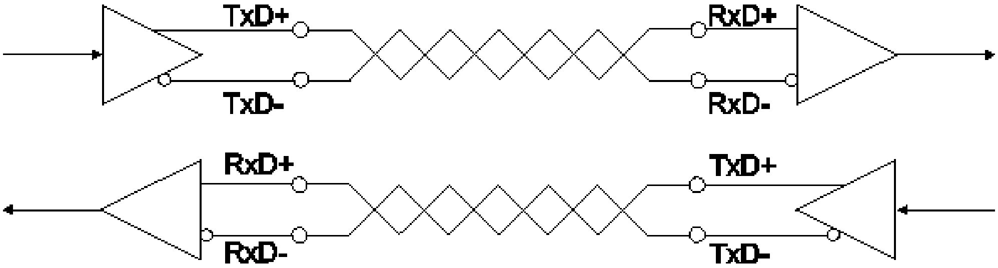


Fig. 12: 4-wire point-to-point connection to an RS422 device

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	1_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

5.2.1.3 RS485

You can operate the RS845 communication in two variants:

- Without diagnosis
- With diagnosis of the transmit data [▶ 32]

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

Bus structure, half duplex data transmission

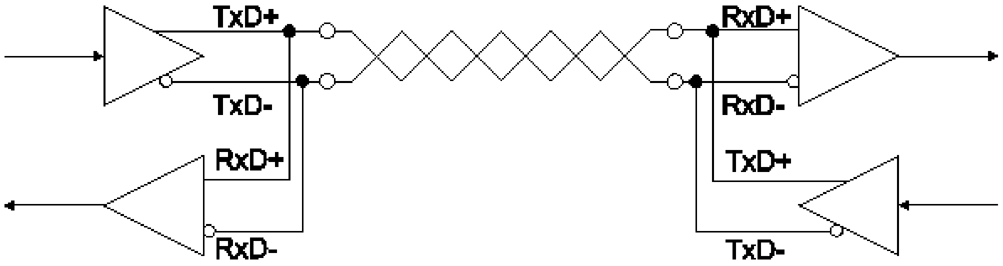


Fig. 13: 2-wire connection in bus structure to RS485 device(s)

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex channel n	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	1_{bin} (default)
		1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

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

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

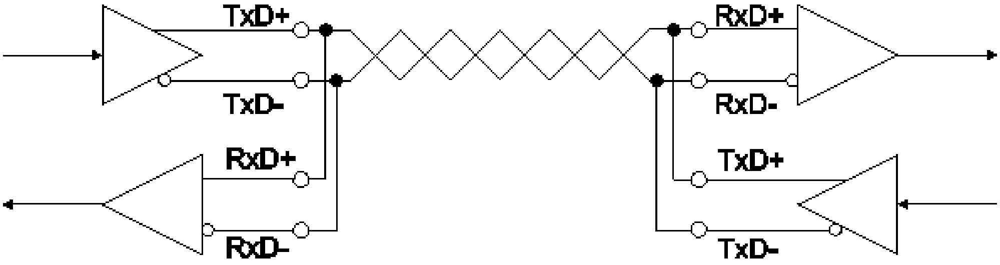


Fig. 14: 2-wire connection with external bridge in bus structure to RS485 device(s)

The following CoE objects must be set

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

Index	Name	Meaning		Data type	Flags	Setting
80n0:06	Enable half duplex	0_{bin}	Full duplex: transmitted data are monitored. The bit has no effect in RS232 and RS422 mode	BOOLEAN	RW	0_{bin}
	channel n	1_{bin}	Half duplex: The reception of the data transmitted by the box itself is suppressed			

Index	Name	Meaning		Data type	Flags	Setting
80n0:07	Enable point-to-point connection (RS422) Channel n	0_{bin}	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0_{bin}
		1_{bin}	The module is used for a point-to-point connection (RS422).			

5.2.2 Setting the interface parameters

5.2.2.1 Data transfer rate

The data transfer rate is specified in the unit "Baud". The following applies to the serial interface of EPP6002: 1 Baud = 1 bit per second.

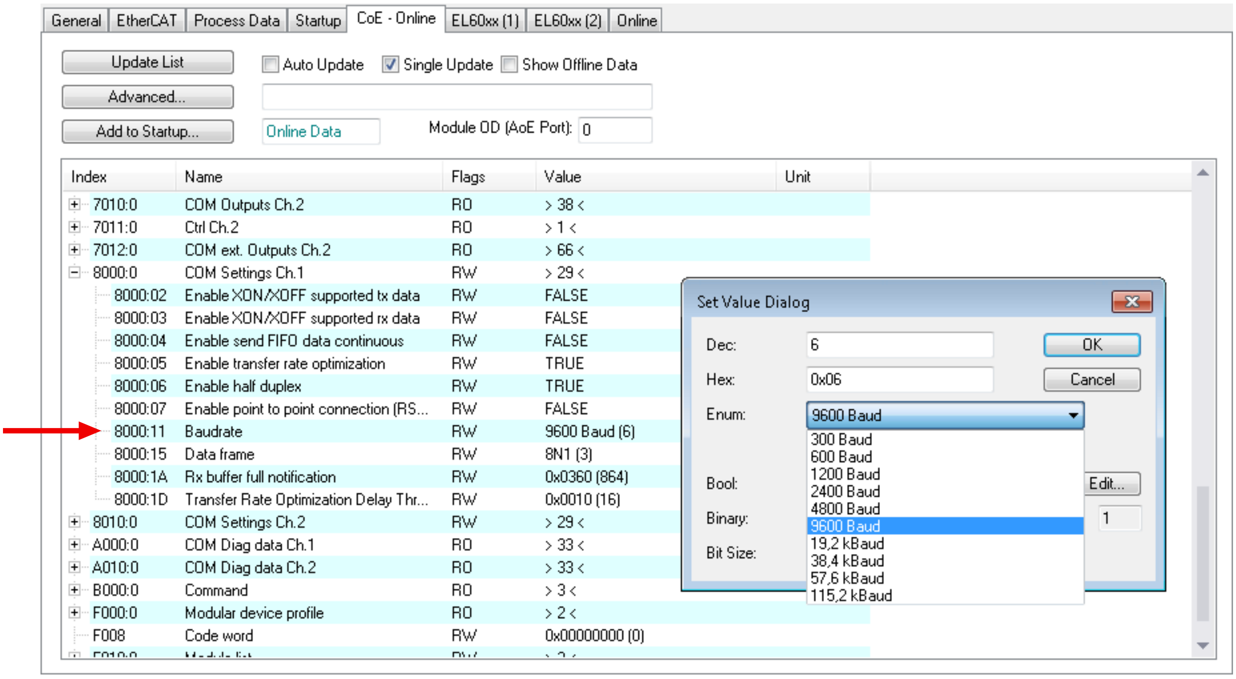
Set the data transfer rate in the following CoE parameters:

Channel	CoE object		Parameter	
1	8000_{hex}	COM Settings Ch.1	11_{hex}	Baudrate
2	8010_{hex}	COM Settings Ch.2	11_{hex}	Baudrate

Possible values

- 300 baud
- 600 baud
- 1200 baud
- 2400 baud
- 4800 baud
- 9600 baud (factory setting)
- 19.2 kbaud
- 38.4 kbaud
- 57.6 kbaud
- 115.2 kbaud

Example for channel 1



□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
7010:0	COM Outputs Ch.2	RO	> 38 <	
7011:0	Ctrl Ch.2	RO	> 1 <	
7012:0	COM ext. Outputs Ch.2	RO	> 66 <	
8000:0	COM Settings Ch.1	RW	> 29 <	
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 (RS...	RW	FALSE	
8000:11	Baudrate	RW	9600 Baud (6)	
8000:15	Data frame	RW	8N1 (3)	
8000:1A	Rx buffer full notification	RW	0x0360 (864)	
8000:1D	Transfer Rate Optimization Delay Thr...	RW	0x0010 (16)	
8010:0	COM Settings Ch.2	RW	> 29 <	
A000:0	COM Diag data Ch.1	RO	> 33 <	
A010:0	COM Diag data Ch.2	RO	> 33 <	
B000:0	Command	RO	> 3 <	
F000:0	Modular device profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	

Set Value Dialog

Dec: 6

Hex: 0x06

Enum: 9600 Baud

300 Baud
600 Baud
1200 Baud
2400 Baud
4800 Baud
9600 Baud
19,2 kBaud
38,4 kBaud
57,6 kBaud
115,2 kBaud

Bool:

Binary: 1

Bit Size:

Fig. 15: CoE parameter “Baudrate” for channel 1 (X01, X02)

5.2.2.2 Data format

Set the data format in the following CoE parameters:

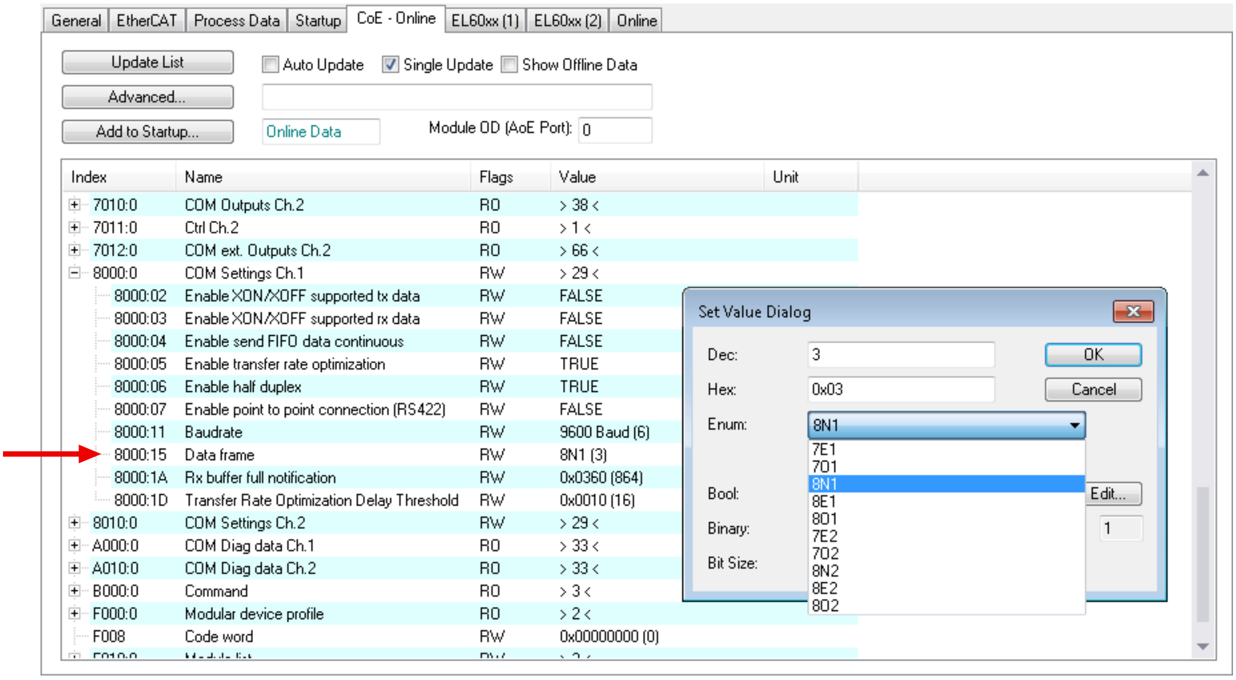
Channel	CoE object		Parameter	
1	8000_{hex}	COM Settings Ch.1	15_{hex}	Data frame
2	8010_{hex}	COM Settings Ch.2	15_{hex}	Data frame

Possible values

The data format is specified as a character string. Three parameters are coded in the character string:

Data format	Number of data bits	Parity bit	Number of stop bits
7E1	7	Even	1
7O1	7	Odd	1
8N1 (factory setting)	8	none	1
8E1	8	Even	1
8O1	8	Odd	1
7E2	7	Even	2
7O2	7	Odd	2
8N2	8	none	2
8E2	8	Even	2
8O2	8	Odd	2

Example for channel 1



□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 7010:0	COM Outputs Ch.2	RO	> 38 <	
+ 7011:0	Ctrl Ch.2	RO	> 1 <	
+ 7012:0	COM ext. Outputs Ch.2	RO	> 66 <	
- 8000:0	COM Settings Ch.1	RW	> 29 <	
8000:02	Enable XON/XOFF supported tx data	RW	FALSE	
8000:03	Enable XON/XOFF supported rx data	RW	FALSE	
8000:04	Enable send FIFO data continuous	RW	FALSE	
8000:05	Enable transfer rate optimization	RW	TRUE	
8000:06	Enable half duplex	RW	TRUE	
8000:07	Enable point to point connection (RS422)	RW	FALSE	
8000:11	Baudrate	RW	9600 Baud (6)	
8000:15	Data frame	RW	8N1 (3)	
8000:1A	Rx buffer full notification	RW	0x0360 (864)	
8000:1D	Transfer Rate Optimization Delay Threshold	RW	0x0010 (16)	
+ 8010:0	COM Settings Ch.2	RW	> 29 <	
+ A000:0	COM Diag data Ch.1	RO	> 33 <	
+ A010:0	COM Diag data Ch.2	RO	> 33 <	
+ B000:0	Command	RO	> 3 <	
+ F000:0	Modular device profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
+ F010:0	Module list	RW	> 2 <	

Set Value Dialog

Dec: 3

Hex: 0x03

Enum: 8N1

7E1
7O1
8N1
8E1
8O1
7E2
7O2
8N2
8E2
8O2

Bool:

Binary:

Bit Size: 1

OK

Cancel

Fig. 16: CoE parameter “Data frame” for channel 1 (X01, X02)

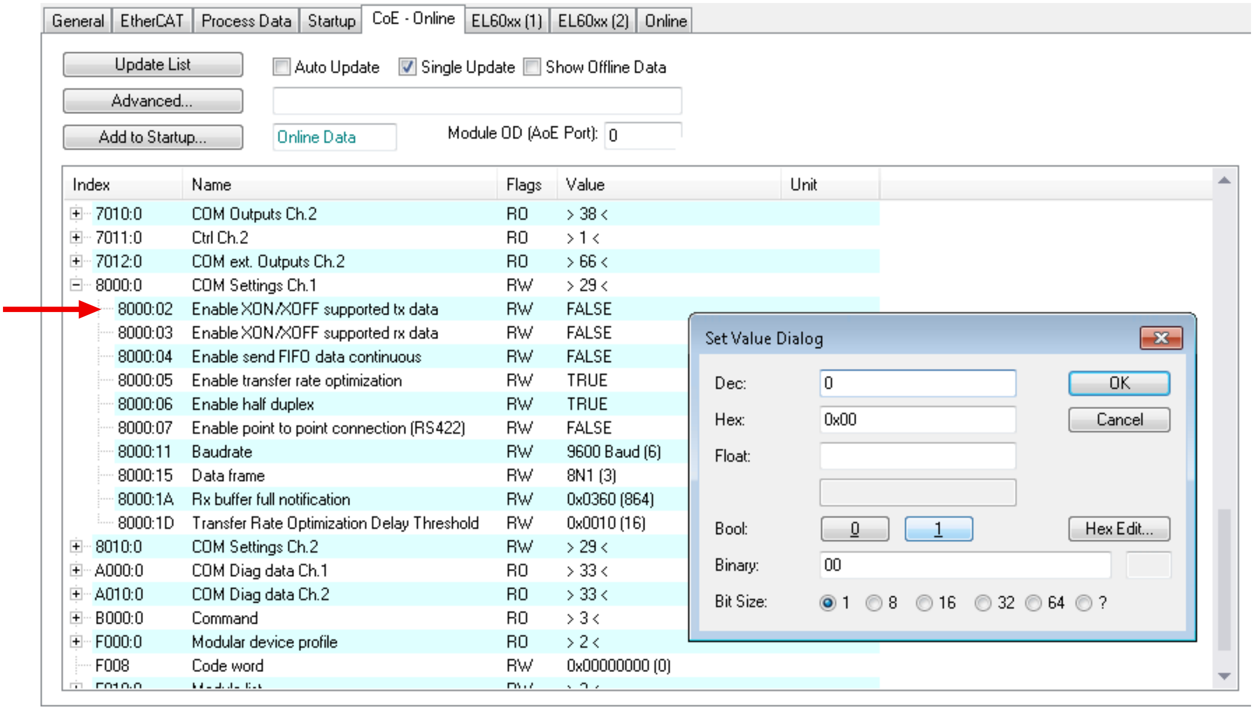
5.2.2.3 Flow control

EPP6002 supports a software flow control for send data and for receive data.

The flow control is disabled in the factory setting. Enable the flow control by setting the following CoE parameters to TRUE:

Chan-nel	Cable	CoE object		Parameter	
1	send data	8000_{hex}	COM Settings Ch.1	02_{hex}	Enable XON/XOFF supported tx data
	receive data	8000_{hex}	COM Settings Ch.1	03_{hex}	Enable XON/XOFF supported rx data
2	send data	8010_{hex}	COM Settings Ch.2	02_{hex}	Enable XON/XOFF supported tx data
	receive data	8010_{hex}	COM Settings Ch.2	03_{hex}	Enable XON/XOFF supported rx data

Example for channel 1



□ Auto Update ☑ Single Update □ Show Offline Data

Online Data   Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 7010:0	COM Outputs Ch.2	RO	> 38 <	
+ 7011:0	Ctrl Ch.2	RO	> 1 <	
+ 7012:0	COM ext. Outputs Ch.2	RO	> 66 <	
− 8000:0	COM Settings Ch.1	RW	> 29 <	
8000:02	Enable XON/XOFF supported tx data	RW	FALSE	
8000:03	Enable XON/XOFF supported rx data	RW	FALSE	
8000:04	Enable send FIFO data continuous	RW	FALSE	
8000:05	Enable transfer rate optimization	RW	TRUE	
8000:06	Enable half duplex	RW	TRUE	
8000:07	Enable point to point connection (RS422)	RW	FALSE	
8000:11	Baudrate	RW	9600 Baud (6)	
8000:15	Data frame	RW	8N1 (3)	
8000:1A	Rx buffer full notification	RW	0x0360 (864)	
8000:1D	Transfer Rate Optimization Delay Threshold	RW	0x0010 (16)	
+ 8010:0	COM Settings Ch.2	RW	> 29 <	
+ A000:0	COM Diag data Ch.1	RO	> 33 <	
+ A010:0	COM Diag data Ch.2	RO	> 33 <	
+ B000:0	Command	RO	> 3 <	
+ F000:0	Modular device profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
+ F010:0	Module list	RW	> 2 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Float:

Bool: 0 1

Binary: 00

Bit Size: ● 1 ○ 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 17: CoE parameter for the flow control of send data to channel 1 (X01, X02)

5.2.3 Continuous transmission of data

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".

5.2.4 Automatic summarizing of receive data

In serial communication, a data set is usually sent as a contiguous byte stream. There is a pause between two data sets.

On the basis of the pauses, EPP6002 can recognize when a data set begins and when it ends. This enables it to summarize the bytes of a data set and to forward them as contiguous byte stream to the controller.

Functioning

Several bytes received shortly after one another are initially collected in the receive buffer.

If a pause is detected after a byte, the collected receive data are transferred from the receive buffer into the process image. The “Receive Request” bit is inverted in order to show that new receive data exist.

Enabling / disabling

The automatic summarizing of receive data is enabled in the factory setting. It may be useful to disable it if you wish to receive the data sent by an end device as quickly as possible in the controller. Or if the end device sends continuously without pauses.

If the automatic summarizing of receive data is disabled, each byte received is transferred immediately to the process image.

You can disable the automatic summarizing of receive data by setting the following CoE parameters to FALSE:

Channel	CoE object		Parameter	
1	8000_{hex}	COM Settings Ch.1	05_{hex}	Enable transfer rate optimization
2	8010_{hex}	COM Settings Ch.2	05_{hex}	Enable transfer rate optimization

Length of the pause between two data sets

A pause between two data sets must exceed a minimum length in order to be detected as a pause.

From firmware 04: You can set the minimum length of the pause in parameter 80n0:1D “Transfer Rate Optimization Delay Threshold”. Factory setting: 16 bit times.

Up to firmware 03: The minimum length of the pause is 16 bit times.

Calculation: 1 bit time [s] = 1 / data transfer rate

5.2.5 Receive buffer overflow

Bit 3 “BUF_F” of the Status word [▶ 15] signals that the number of bytes in the receive buffer exceeds a certain value.

Set the desired number of bytes in the following CoE parameters:

Channel	CoE object		Parameter	
1	8000_{hex}	COM Settings Ch.1	$1A_{hex}$	Rx buffer full notification
2	8010_{hex}	COM Settings Ch.2	$1A_{hex}$	Rx buffer full notification

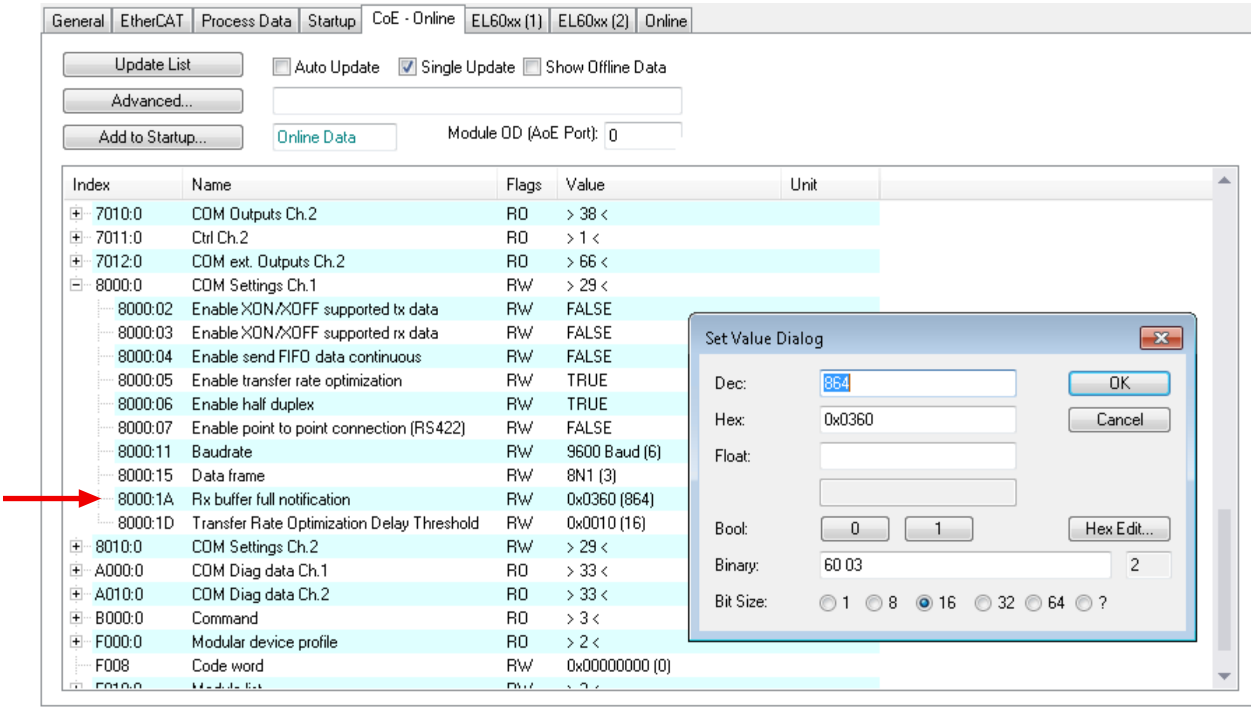
Possible values

In the factory setting the parameter is set to 864 bytes.

From firmware 04 onwards, the receive buffer is larger than the parameter in the factory setting. This means that in the factory setting, “BUF_F” warns of an impending receive buffer overflow.

Up to firmware 03, the parameter in the factory setting corresponds to the size of the receive buffer. This means that in the factory setting, “BUF_F” signals that the receive buffer has already overflowed.

Example for channel 1



☐ Auto Update ☑ Single Update ☐ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 7010:0	COM Outputs Ch.2	RO	> 38 <	
+ 7011:0	Ctrl Ch.2	RO	> 1 <	
+ 7012:0	COM ext. Outputs Ch.2	RO	> 66 <	
− 8000:0	COM Settings Ch.1	RW	> 29 <	
8000:02	Enable XON/XOFF supported tx data	RW	FALSE	
8000:03	Enable XON/XOFF supported rx data	RW	FALSE	
8000:04	Enable send FIFO data continuous	RW	FALSE	
8000:05	Enable transfer rate optimization	RW	TRUE	
8000:06	Enable half duplex	RW	TRUE	
8000:07	Enable point to point connection (RS422)	RW	FALSE	
8000:11	Baudrate	RW	9600 Baud (6)	
8000:15	Data frame	RW	8N1 (3)	
8000:1A	Rx buffer full notification	RW	0x0360 (864)	
8000:1D	Transfer Rate Optimization Delay Threshold	RW	0x0010 (16)	
+ 8010:0	COM Settings Ch.2	RW	> 29 <	
+ A000:0	COM Diag data Ch.1	RO	> 33 <	
+ A010:0	COM Diag data Ch.2	RO	> 33 <	
+ B000:0	Command	RO	> 3 <	
+ F000:0	Modular device profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	

Set Value Dialog

Dec: 864

Hex: 0x0360

Float:

Bool: 0 1

Binary: 60 03 2

Bit Size: ○ 1 ○ 8 ● 16 ○ 32 ○ 64 ○ ?

Fig. 18: CoE parameter for the receive buffer overflow to channel 1 (X01, X02)

5.3 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 [▶ 14] to the number of bytes to be transmitted.
3. Toggle the *Transmit Request* bit in the Control word [▶ 14].

⇨ The module acknowledges the data transmission in the Status word [▶ 15] via the *Transmit Accepted* parameter.

Receiving data

If the module in the Status word [▶ 15] toggles the *Receive Request* bit, there are new receive data in the process data.

1. Read the *Input Length* parameter from the Status word [▶ 15]. 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 [▶ 14]. 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.

5.3.1 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)

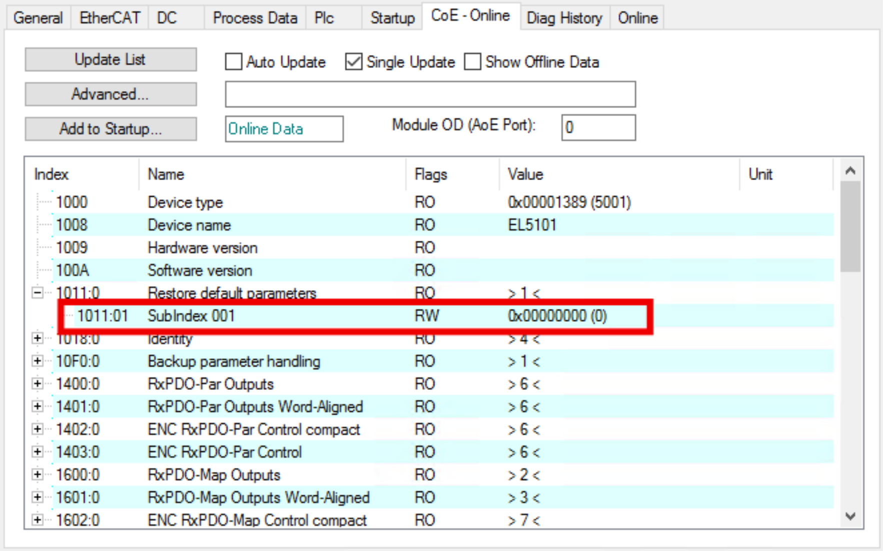
1. The module indicates that there is new data in the process image by changing the state of the "Receive request" bit.
2. The number of bytes received is written in "Input length"
3. The controller acknowledges acceptance of the bytes by changing the state of “Receive request”.

5.4 Communication via a virtual COM port

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

5.5 Restoring the delivery state

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



☐ Auto Update ☑ Single Update ☐ Show Offline Data

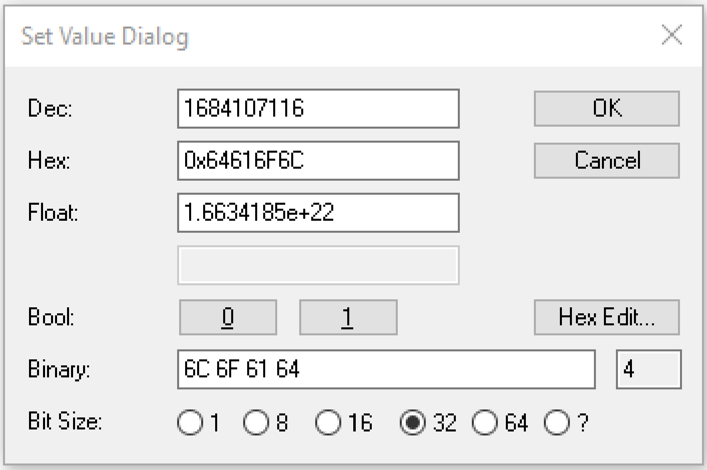
Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1000	Device type	RO	0x00001389 (5001)	
1008	Device name	RO	EL5101	
1009	Hardware version	RO		
100A	Software version	RO		
1011:0	Restore default parameters	RO	> 1 <	
1011:01	SubIndex 001	RW	0x00000000 (0)	
1018:0	Identity	RO	> 4 <	
10F0:0	Backup parameter handling	RO	> 1 <	
1400:0	RxPDO-Par Outputs	RO	> 6 <	
1401:0	RxPDO-Par Outputs Word-Aligned	RO	> 6 <	
1402:0	ENC RxPDO-Par Control compact	RO	> 6 <	
1403:0	ENC RxPDO-Par Control	RO	> 6 <	
1600:0	RxPDO-Map Outputs	RO	> 2 <	
1601:0	RxPDO-Map Outputs Word-Aligned	RO	> 3 <	
1602:0	ENC RxPDO-Map Control compact	RO	> 7 <	

Fig. 19: Selecting the Restore default parameters PDO

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

All backup objects are reset to the delivery state.



Dec: 1684107116

Hex: 0x64616F6C

Float: 1.6634185e+22

Binary: 6C 6F 61 64 4

Bit Size: ○ 1 ○ 8 ○ 16 ● 32 ○ 64 ○ ?

Fig. 20: Entering a restore value in the Set Value dialog

Alternative restore value

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

An incorrect entry for the restore value has no effect.

5.6 Decommissioning

⚠ **WARNING**

Risk of electric shock!

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

6 CoE parameters

6.1 Directory

Index (hex)	Name
1000	Device type [▶ 48]
1008	Device name [▶ 48]
1009	Hardware version [▶ 48]
100A	Software version [▶ 48]
1011	Restore default parameters [▶ 45]
1018	Identity [▶ 48]
10F0	Backup parameter handling [▶ 48]
1400	COM RxPDO-Par Outputs Ch. 1 [▶ 48]
1401	COM RxPDO-Par Outputs Ch. 2 [▶ 49]
1404	COM RxPDO-Par Outputs Ch. 1 [▶ 49]
1405	COM RxPDO-Par Outputs Ch. 2 [▶ 49]
1600	COM RxPDO-Map Outputs Ch. 1 [▶ 50]
1601	COM RxPDO-Map Outputs Ch. 2 [▶ 51]
1604	COM RxPDO-Map Outputs Ch. 1 [▶ 52]
1605	COM RxPDO-Map Outputs Ch. 2 [▶ 53]
1800	COM TxPDO-Par Inputs Ch. 1 [▶ 53]
1801	COM TxPDO-Par Inputs Ch. 2 [▶ 54]
1804	COM TxPDO-Par Inputs Ch. 1 [▶ 54]
1805	COM TxPDO-Par Inputs Ch. 2 [▶ 54]
1A00	COM TxPDO-Map Inputs Ch. 1 [▶ 55]
1A01	COM TxPDO-Map Inputs Ch. 2 [▶ 56]
1A04	COM TxPDO-Map Inputs Ch. 1 [▶ 57]
1A05	DIG TxPDO-Map Inputs Ch. 2 [▶ 58]
1C00	Sync manager type [▶ 58]
1C12	RxPDO assign [▶ 59]
1C13	TxPDO assign [▶ 59]
1C32	SM output parameter [▶ 60]
1C33	SM input parameter [▶ 61]
6000	COM Inputs Ch. 1 [▶ 62]
6001	Status Ch. 1 [▶ 62]
6010	COM Inputs Ch. 2 [▶ 63]
6011	Status Ch. 2 [▶ 63]
7000	COM Outputs Ch. 1 [▶ 64]
7001	Ctrl Ch. 1 [▶ 64]
7010	COM Outputs Ch. 2 [▶ 65]
7011	Ctrl Ch. 2 [▶ 65]
8000	COM Settings Ch. 1 [▶ 46]
8010	COM Settings Ch. 2 [▶ 47]
A000	COM Diag data Ch. 1 [▶ 65]
A010	COM Diag data Ch. 2 [▶ 66]
F000	Modular device profile [▶ 66]
F008	Code word [▶ 66]
F010	Module list [▶ 66]
F800	COM Settings [▶ 47]

6.2 Object description and parameterization

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

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

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

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▸ 45] during commissioning
- Objects required for the selection of the interface type [▸ 47]
- Objects intended for regular operation, e.g. through ADS access
- Objects for indicating internal settings [▸ 48] (may be fixed)
- Further profile-specific objects [▸ 61] indicating inputs, outputs and status information

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

Objects to be parameterized during commissioning

Index 1011 Restore default parameters

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

Index 8000 COM Settings Ch.1

Index (hex)	Name	Meaning		Data type	Flags	Default
8000:0	COM Settings Ch.1			UINT8	RO	0x1A (26_{dec})
8000:02	Enable XON/XOFF supported tx data	Enable flow control for send data.		BOOLEAN	RW	0x00 (0_{dec})
8000:03	Enable XON/XOFF supported rx data	Enable flow control for receive data.		BOOLEAN	RW	0x00 (0_{dec})
8000:04	Enable send FIFO data continuous	Enable Continuous transmission of data [▶ 36].		BOOLEAN	RW	0x00 (0_{dec})
8000:05	Enable transfer rate optimization	Enable Automatic summarizing of receive data [▶ 37].		BOOLEAN	RW	0x01 (1_{dec})
8000:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0_{dec})
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8000:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0_{dec})
		1	The module is used as a point-to-point connection (RS422)			
8000:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6_{dec})
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8000:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3_{dec})
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8000:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864_{dec})
8000:1D	Transfer Rate Optimization Delay Threshold	Minimum length of the pause between two data sets for the Automatic summarizing of receive data [▶ 37].		UINT16	RW	0x10 (16_{dec})

Index 8010 COM Settings Ch.2

Index (hex)	Name	Meaning		Data type	Flags	Default
8010:0	COM Settings Ch.2			UINT8	RO	0x1A (26$_{dec}$)
8010:02	Enable XON/XOFF supported tx data	Enable flow control for send data.		BOOLEAN	RW	0x00 (0$_{dec}$)
8010:03	Enable XON/XOFF supported rx data	Enable flow control for receive data.		BOOLEAN	RW	0x00 (0$_{dec}$)
8010:04	Enable send FIFO data continuous	Enable Continuous transmission of data [▶ 36].		BOOLEAN	RW	0x00 (0$_{dec}$)
8010:05	Enable transfer rate optimization	Enable Automatic summarizing of receive data [▶ 37].		BOOLEAN	RW	0x01 (1$_{dec}$)
8010:06	Enable half duplex	Half duplex for RS485 mode (this bit is not evaluated in RS232 and RS422 mode)		BOOLEAN	RW	0x00 (0$_{dec}$)
		0	Full duplex: The module monitors its transmitted data.			
		1	Half duplex: The module does not monitor the data that it has transmitted itself.			
8010:07	Enable point to point connection (RS422)	0	The module is used in a bus structure in accordance with the RS485 standard.	BOOLEAN	RW	0x00 (0$_{dec}$)
		1	The module is used as a point-to-point connection (RS422)			
8010:11	Baud rate	Baud Rate		BIT4	RW	0x06 (6$_{dec}$)
		0x01	300 baud			
		0x02	600 baud			
		0x03	1200 baud			
		0x04	2400 baud			
		0x05	4800 baud			
		0x06	9600 baud			
		0x07	19200 baud			
		0x08	38400 baud			
		0x09	57600 baud			
		0x0A	115200 baud			
8010:15	Data frame	Data frame / Stop bits		BIT4	RW	0x03 (3$_{dec}$)
		0x01	7E1			
		0x02	7O1			
		0x03	8N1			
		0x04	8E1			
		0x05	8O1			
		0x09	7E2			
		0x0A	7O2			
		0x0B	8N2			
		0x0C	8E2			
		0x0D	8O2			
8010:1A	Rx buffer full notification	The value specifies the number of data in the receive FIFO, from which the bit "buffer full" is set.		UINT16	RW	0x0360 (864$_{dec}$)
8010:1D	Transfer Rate Optimization Delay Threshold	Minimum length of the pause between two data sets for the Automatic summarizing of receive data [▶ 37].		UINT16	RW	0x10 (16$_{dec}$)

Index F800 COM Settings

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

Additional objects

Standard objects (0x1000-0x1FFF)

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x02581389 (39326601$_{dec}$)

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EPP6002-0002

Index 1009 Hardware version

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

Index 100A Software Version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4$_{dec}$)
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2$_{dec}$)
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x17724052 (393363538$_{dec}$)
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00100002 (1048578$_{dec}$)
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0$_{dec}$)

Index 10F0 Backup parameter handling

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

Index 1400 COM RxPDO-Par Outputs Ch.1

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

Index 1401 COM RxPDO-Par Outputs Ch.2

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

Index 1404 COM RxPDO-Par Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1404:0	COM RxPDO-Par Outputs Ch.1	PDO Parameter RxPDO 5	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[2]	RO	00 16

Index 1405 COM RxPDO-Par Outputs Ch.2

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

Index 1600 COM RxPDO-Map Outputs Ch.1

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

Index 1601 COM RxPDO-Map Outputs Ch.2

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

Index 1604 COM RxPDO-Map Outputs Ch.1

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

Index 1605 COM RxPDO-Map Outputs Ch.2

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

Index 1800 COM TxPDO-Par Inputs Ch.1

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

Index 1801 COM TxPDO-Par Inputs Ch.2

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

Index 1804 COM TxPDO-Par Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs Ch.1	PDO parameter TxPDO 5	UINT8	RO	0x06 (6$_{dec}$)
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 5	OCTET-STRING[2]	RO	00 1A

Index 1805 COM TxPDO-Par Inputs Ch.2

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

Index 1A00 COM TxPDO-Map Inputs Ch.1

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

Index 1A01 COM TxPDO-Map Inputs Ch.2

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

Index 1A04 COM TxPDO-Map Inputs Ch.1

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

Index 1A05 COM TxPDO-Map Inputs Ch.2

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

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1605 (5637_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A04 (6660_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A05 (6661_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000384 (900_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▸ 60])	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00002710 (10000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03 [▸ 60], 1C32:05 [▸ 60], 1C32:06 [▸ 60], 1C32:09 [▸ 60], 1C33:03 [▸ 61], 1C33:06 [▸ 60], 1C33:09 [▸ 61] 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 [▶ 60]	UINT32	RW	0x000F4240 (1000000$_{dec}$)
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900$_{dec}$)
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▶ 60] or 1C33:08 [▶ 61])	UINT16	RO	0xC007 (49159$_{dec}$)
1C33:05	Minimum cycle time	as 1C32:05 [▶ 60]	UINT32	RO	0x00002710 (10000$_{dec}$)
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900$_{dec}$)
1C33:08	Command	as 1C32:08 [▶ 60]	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 [▶ 60]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0C	Cycle exceeded counter	as 1C32:12 [▶ 60]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0D	Shift too short counter	as 1C32:13 [▶ 60]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:20	Sync error	as 1C32:32 [▶ 60]	BOOLEAN	RO	0x00 (0$_{dec}$)

Profile-specific objects (0x6000-0xFFFF)

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

Index 6000 COM Inputs Ch.1

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

Index 6001 Status Ch.1

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

Index 6010 COM Inputs Ch.2

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

Index 6011 Status Ch.2

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

Index 7000 COM Outputs Ch.1

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

Index 7001 Ctrl Ch.1

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

Index 7010 COM Outputs Ch.2

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

Index 7011 Ctrl Ch.2

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

Index A000 COM Diag data Ch.1

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

Index A010 COM Diag data Ch.2

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

Index F000 Modular device profile

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

Index F008 Code word

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

Index F010 Module list

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

7 Appendix

7.1 General operating conditions

Protection degrees (IP-Code)

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

1. Number: dust protection and touch guard	Definition
0	Non-protected
1	Protected against access to hazardous parts with the back of a hand. Protected against solid foreign objects of Ø 50 mm
2	Protected against access to hazardous parts with a finger. Protected against solid foreign objects of Ø 12.5 mm.
3	Protected against access to hazardous parts with a tool. Protected against solid foreign objects Ø 2.5 mm.
4	Protected against access to hazardous parts with a wire. Protected against solid foreign objects Ø 1 mm.
5	Protected against access to hazardous parts with a wire. Dust-protected. Intrusion of dust is not totally prevented, but dust shall not penetrate in a quantity to interfere with satisfactory operation of the device or to impair safety.
6	Protected against access to hazardous parts with a wire. Dust-tight. No intrusion of dust.
2. Number: water* protection	Definition
0	Non-protected
1	Protected against water drops
2	Protected against water drops when enclosure tilted up to 15°.
3	Protected against spraying water. Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects.
4	Protected against splashing water. Water splashed against the disclosure from any direction shall have no harmful effects
5	Protected against water jets
6	Protected against powerful water jets
7	Protected against the effects of temporary immersion in water. Intrusion of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water for 30 min. in 1 m depth.

*) These protection classes define only protection against water.

Chemical Resistance

The Resistance relates to the Housing of the IP67 modules and the used metal parts. In the table below you will find some typical resistance.

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

Key

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

7.2 Accessories

Mounting

Ordering information	Description	Link
ZS5300-0011	Mounting rail	Website

Cables

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

Ordering information	Description	Link
ZK2000-6xxx-xxxx	Sensor cable M12, 4-pin	Website
ZK2000-7xxx-0xxx	Sensor cable M12, 4-pin + shield	Website
ZK700x-xxxx-xxxx	EtherCAT P cable M8	Website

Labeling material, protective caps

Ordering information	Description
ZS5000-0012	Protective cap for M8 sockets, P-coded, IP67 (50 pieces)
ZS5000-0020	Protective cap for M12 sockets, IP67 (50 pcs.)
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10
ZS5000-xxxx	Printed inscription labels on enquiry

Tools

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

Further accessories

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

7.3 Version identification of EtherCAT devices

7.3.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

7.3.2 Version identification of EP/EPI/EPP/ER/ERI boxes

The serial number/ data 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 of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02

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

Syntax: D ww yy x y z u

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

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

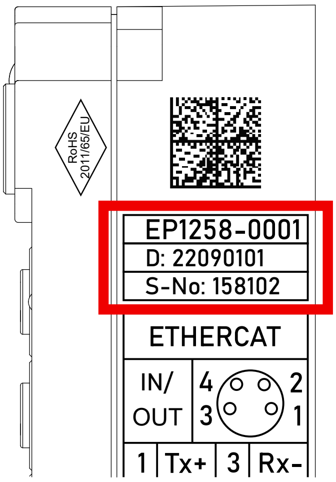


Fig. 21: EP1258-00001 IP67 EtherCAT Box with batch number/DateCode 22090101 and unique serial number 158102

7.3.3 Beckhoff Identification Code (BIC)

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

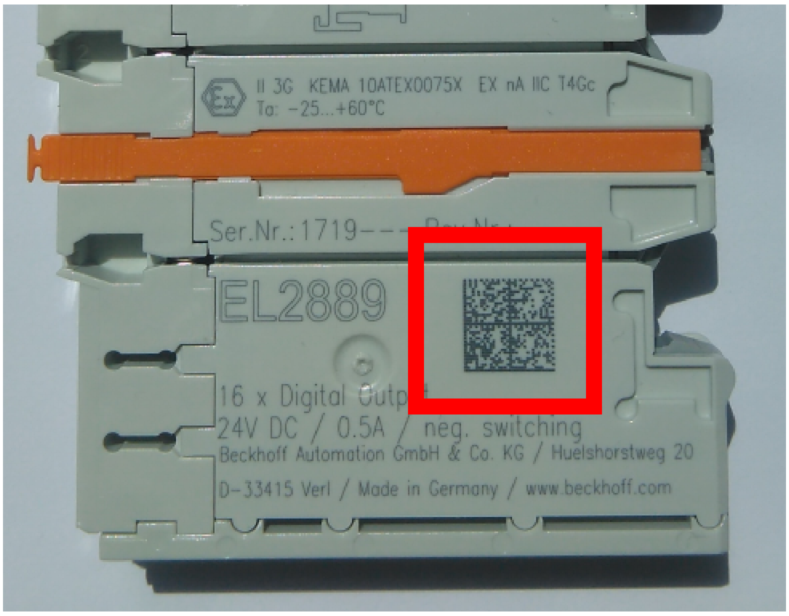

II 3G KEMA 10ATEX0075X EX nA IIC T4Gc
Ta: -25...+60°C
Ser.Nr.:1719---
EL2889
16 x Digital Output
24V DC / 0.5A / neg. switching
Beckhoff Automation GmbH & Co. KG / Huelshorstweg 20
D-33415 Verl / Made in Germany / www.beckhoff.com

Fig. 22: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

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

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

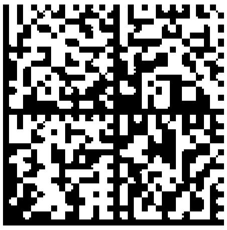


Fig. 23: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

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

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.

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

Currently, no electronic storage and readout is planned for these devices.

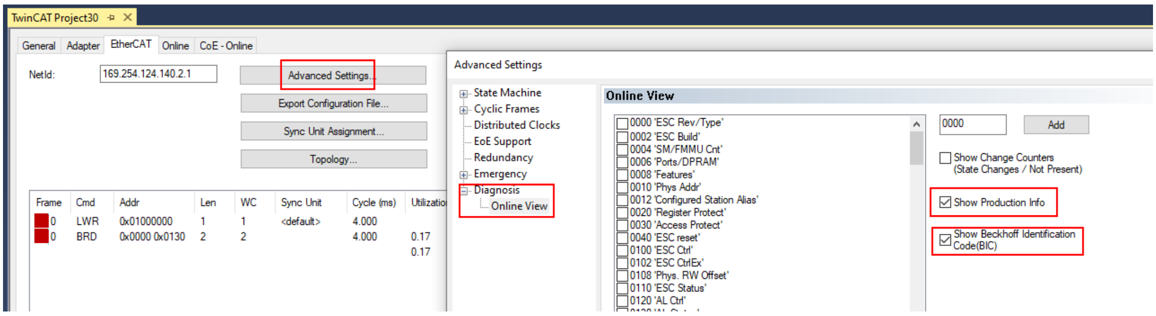
EtherCAT devices (IP20, IP67)

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

The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

The user can electronically access the eBIC (if existent) 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 checkbox "Show Beckhoff Identification Code (BIC)" under
      EtherCAT → Advanced Settings → Diagnostics:

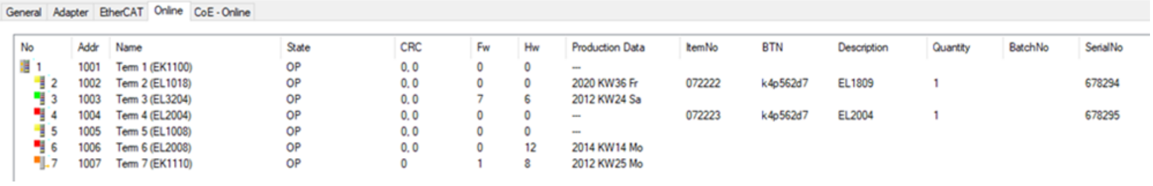


□ Show Change Counters (State Changes / Not Present)

☑ Show Production Info

☑ Show Beckhoff Identification Code(BIC)

- The BTN and its contents are then displayed:

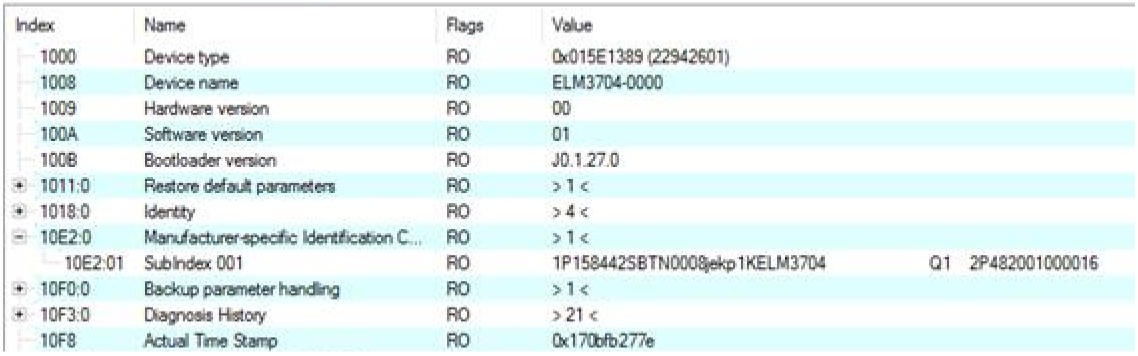


No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0, 0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0, 0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0, 0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0, 0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0, 0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0, 0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

- Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.

- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:

- 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 introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.

- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
  - If multiple, hierarchically arranged ESCs are installed in a 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 with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

Profibus/Profinet/DeviceNet… Devices

Currently, no electronic storage and readout is planned for these devices.

7.4 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

More Information:

www.beckhoff.com/epp6002-0002

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