

Documentation | EN

EL6688

IEEE 1588 PTP External Synchronization Interface



Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 Safety instructions	6
1.3 Documentation issue status	7
1.4 Version identification of EtherCAT devices	8
1.4.1 General notes on marking.....	8
1.4.2 Version identification of EL terminals.....	9
1.4.3 Beckhoff Identification Code (BIC).....	10
1.4.4 Electronic access to the BIC (eBIC).....	12
2 Product description	14
2.1 EL6688 - Introduction.....	14
2.2 EL6688 - Technical data	15
2.3 Basics IEEE1588	16
3 Basics communication	20
3.1 EtherCAT basics	20
3.2 General notes for setting the watchdog	20
3.3 EtherCAT State Machine	21
3.4 CoE Interface	23
3.5 Distributed Clock	28
4 Mounting and wiring	29
4.1 Explosion protection.....	29
4.1.1 ATEX - Special conditions (standard temperature range).....	29
4.1.2 Continuative documentation for ATEX and IECEx.....	31
4.2 UL notice	32
4.3 Note - Power supply.....	33
4.4 Mounting and demounting - terminals with front unlocking	34
4.5 Mounting and demounting - terminals with traction lever unlocking.....	36
4.6 Recommended mounting rails	38
4.7 Positioning of passive Terminals.....	39
4.8 Installation positions.....	40
4.9 EL6688 - LEDs and connection	42
4.10 Disposal	43
5 Commissioning	44
5.1 TwinCAT Development Environment	44
5.1.1 Installation of the TwinCAT real-time driver	44
5.1.2 Notes regarding ESI device description	50
5.1.3 TwinCAT ESI Updater.....	54
5.1.4 Distinction between Online and Offline	54
5.1.5 OFFLINE configuration creation.....	55
5.1.6 ONLINE configuration creation	60
5.1.7 EtherCAT subscriber configuration	68
5.1.8 Import/Export of EtherCAT devices with SCI and XTI.....	77
5.2 General Notes - EtherCAT Slave Application	84

5.3	Notes to Commissioning	92
5.4	Answers to frequently asked questions.....	98
5.5	Object description and parameterization.....	102
5.5.1	Objects for commissioning	102
5.5.2	Objects for regular operation.....	104
5.5.3	Other objects 0x1000-0xFFFF	104
6	Appendix.....	112
6.1	EtherCAT AL Status Codes	112
6.2	Firmware compatibility	112
6.3	Firmware Update EL/ES/EM/ELM/EPxxxx.....	113
6.3.1	Device description ESI file/XML	114
6.3.2	Firmware explanation.....	117
6.3.3	Updating controller firmware *.efw	117
6.3.4	FPGA firmware *.rbf.....	119
6.3.5	Simultaneous updating of several EtherCAT devices.....	123
6.4	Restoring the delivery state.....	124
6.5	Support and Service.....	125

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.

The logo for EtherCAT, featuring the word "EtherCAT" in a bold, black, sans-serif font. A red arrow points from the top of the "A" towards the right, ending above the "T". A registered trademark symbol (®) is located to the right of the "T".

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

DANGER

Serious risk of injury!

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

WARNING

Risk of injury!

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

CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
3.0	<ul style="list-style-type: none"> • Update chapter "Commissioning" • Update structure • Update revision status
2.9	<ul style="list-style-type: none"> • Update chapter "Basics communication" • Update chapter "Notes to commissioning" • Update structure • Update revision status
2.8	<ul style="list-style-type: none"> • Update chapter "Technical data" • Chapter "Note - Power supply" added • Update structure
2.7	<ul style="list-style-type: none"> • New title • Update chapter "Technical data"
2.6	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update chapter "Version identification of EtherCAT devices" • Update structure • Update Notes • Update revision status • Chapter Disposal added
2.5	<ul style="list-style-type: none"> • Chapter "EL6688 – Technical data" updated • Update revision status • Update structure
2.4	<ul style="list-style-type: none"> • Update chapter "UL notice" • Update chapter "Firmware compatibility!" • Update structure
2.3	<ul style="list-style-type: none"> • Chapter "Commissioning" updated • Update revision status • Update structure
2.2	<ul style="list-style-type: none"> • Chapter "EL6688 – Technical data" updated • Update revision status • Update structure
2.1	<ul style="list-style-type: none"> • Chapter "LEDs and connection" updated • Update revision status
2.0	<ul style="list-style-type: none"> • Migration • Chapter "Notes to Commissioning" updated
1.5	<ul style="list-style-type: none"> • Chapter "Basics IEEE1588" updated
1.4	<ul style="list-style-type: none"> • Chapter "Commissioning" updated
1.3	<ul style="list-style-type: none"> • Technical notes added
1.2	<ul style="list-style-type: none"> • Technical notes added
1.1	<ul style="list-style-type: none"> • Technical notes added
1.0	<ul style="list-style-type: none"> • Minor corrections, 1st public issue
0.1 – 0.3	<ul style="list-style-type: none"> • Provisional documentations for EL6688

1.4 Version identification of EtherCAT devices

1.4.1 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.

1.4.2 Version identification of EL terminals

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**

KK - week of production (CW, calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with serial number 12 06 3A 02:

12 - production week 12

06 - production year 2006

3A - firmware version 3A

02 - hardware version 02



Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.4.3 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.

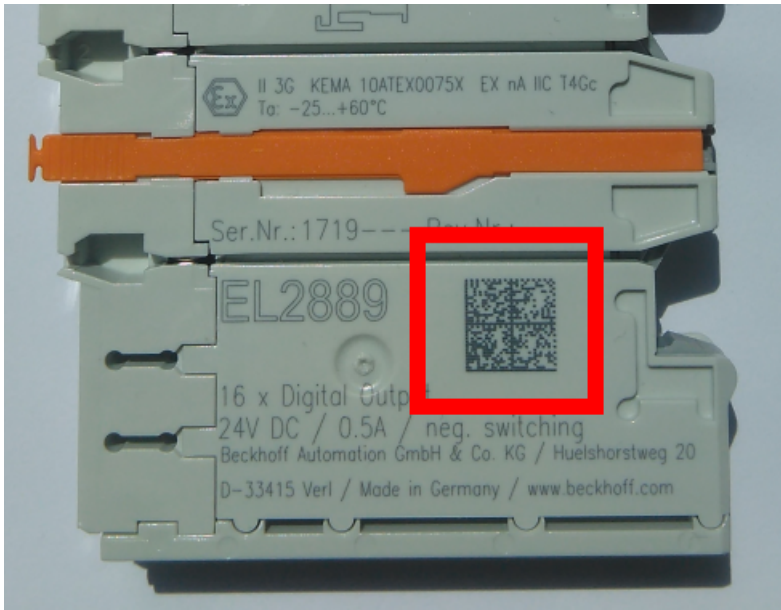


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	Beckhoff order number	1P	8	1P 072222
2	Beckhoff Traceability Number (BTN)	Unique serial number, see note below	SBTN	12	SBTN k4p562d7
3	Article description	Beckhoff article description, e.g. EL1008	1K	32	1K EL1809
4	Quantity	Quantity in packaging unit, e.g. 1, 10, etc.	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P 401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S 678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30P F971, 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**SBTN**k4p562d7**1K**EL1809 **Q1** **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q1** **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.

1.4.4 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.

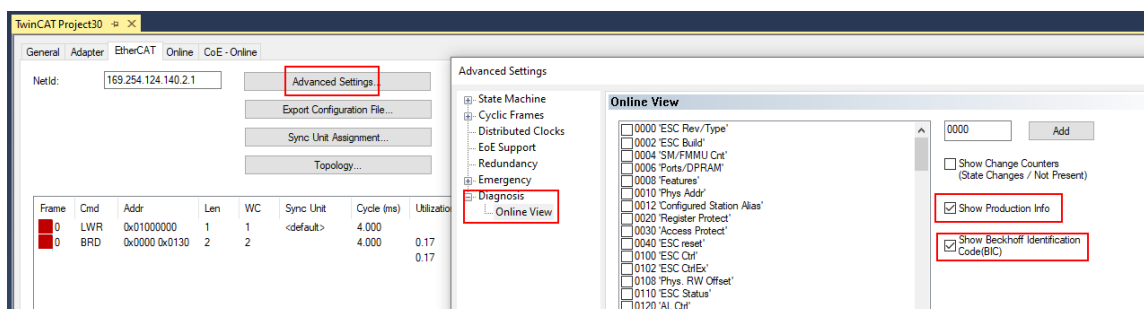
EtherCAT devices (IP20, IP67)

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:



- 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 be 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

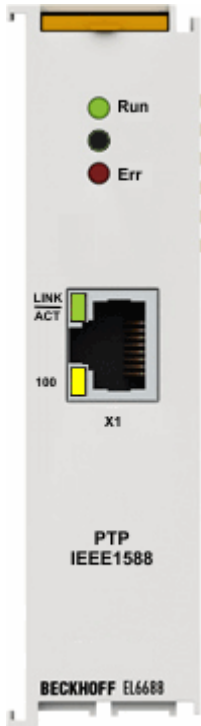
- the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the *Tc2_EtherCAT Library* 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.

2 Product description

2.1 EL6688 - Introduction



IEEE 1588 external synchronization interface

The EL6688 EtherCAT Terminal is a device in the IEEE-1588 synchronization system that supports the Ethernet-based precision time protocols PTPv1 (IEEE 1588-2002) and PTPv2 (IEEE 1588-2008).

On the one hand the EL6688 is an IEEE-1588 clock (master or slave) that is synchronized based on the protocol precision. On the other hand it is synchronized as an EtherCAT Terminal by the distributed clock system.

The following operating modes can be selected via the TwinCAT System Manager: "SlaveOnly", "MasterOnly" and "Best Master Clock" for the PTP side. In this way a consistent timebase can be created across applications for any number of spatially separated EtherCAT systems and machine sections, e.g. for application with axes or measurement technology. The compact EtherCAT Terminal enables flexible deployment depending on the application requirements.

2.2 EL6688 - Technical data

Technical data	EL6688
Supported TwinCAT version	from 2.11
IEEE1588 physics	Ethernet (IEEE 802.3), 10/100 Mbit/s
Supported network protocols	IPv4, DHCP, IGMP ^{*)} , PTP (multicast, IEEE 1588-2002 + IEEE 1588-2008)
IEEE1588 operating modes	PTPv1 Master/Slave, PTPv2 Master/Slave, default profile
Number of Ethernet ports	1
Ethernet interface	10BASE-T/100BASE-TX Ethernet with 1 x RJ45 10/100 Mbit/s, IEEE 802.3u Auto negotiation, half or full duplex at 10 and 100 Mbit/s possible, automatic settings
Cable length	up to 100 m acc. EN50173
Diagnosis	Status LEDs
Power supply	via the E-bus
Current consumption via E-bus	typ. 310 mA
Electrical isolation	500 V (E-Bus/Ethernet)
Configuration	TwinCAT System Manager/CAN-over-EtherCAT (CoE)
Weight	approx. 75 g
permissible ambient temperature range during operation	0°C ... + 55°C
permissible ambient temperature range during storage	-25°C ... + 85°C
permissible relative humidity	95 %, no condensation
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm)
Mounting [▶ 29]	on 35 mm mounting rail conforms to EN 50022
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Marking / Approval ^{*)}	CE, UKCA, EAC ATEX [▶ 29] cULus [▶ 32]

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

**) up to and including FW10 only supported in PTPv1, from FW11 also in PTPv2

Ex marking

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc

2.3 Basics IEEE1588

Definition

The PTP (**P**recision **T**ime **P**rotocol) defined in IEEE1588 is a protocol standard for synchronization of distributed clocks in networks. Each PTP clock cyclically runs the “Best Master Clock Algorithm” (BMCA) to determine the clock with the best precision. The clock with the best precision is referred to as grandmaster and is used as synchronization source for all other PTP clocks in the network. Current two standards are defined, PTPv1 and PTPv2. The PTP can be based on Ethernet as transport medium, for example.

If PTP devices are not directly connected and Ethernet traffic takes place via switches/routers instead, it is highly advisable to use PTP-capable switches/routers. If a switch supports PTP, it behaves like a slave clock vis-à-vis a connected master clock, i.e. it is synchronized by the master clock. Vis-à-vis other connected slave clocks it behaves like a master clock. The switch becomes a PTPv1/v2 boundary clock. In addition, PTPv2 offers a transparent clock option, in which case each switch enters a time correction value in a correction field in the PTP telegram. This enables downstream slaves to calculate the influence of the switch on the transmission link.

Different implementation types are available in the switch/router:

- No PTP support in the switch/router: This is likely to result in extremely reduced synchronization precision [ms range], and the application of such switches should therefore be avoided. Load-dependent instability in the control characteristics may occur.
- Software-based PTP support in the switch: This is likely to result in significantly reduced synchronization precision [ms range], and we would therefore advise against using this type of switch.
- Hardware-based PTP support in the switch: The application of this type of switch is recommended.

The architectural differences between the two stages are as follows:

- **IEEE-1588-2002 V1**
V1 was developed in principle for testing & measurement and industrial automation. The Multicast protocol is conceived for LAN use and achieves better accuracy than NTP.
From a technical point of view higher-priced boundary clocks have to be used with V1 in order to achieve optimum results. The use of V1 is less widespread in practice.
- **IEEE-1588-2008 V2**
V2 is an improvement on V1. It is only possible to use either V1 or V2 in a network. V2 was extended by many features, for which reason it has been well accepted by the market. The selection of devices on the market is greater than for V1. From a technical point of view simpler switches with the clock type “Transparent clock” can be used with V2.

V1 and V2 barely differ in terms of the theoretically attainable accuracy; an accuracy of $< 1 \mu\text{s}$ is attainable with both versions in the ideal case, i.e. with an optimum infrastructure. The maximally attainable accuracy depends alone on the type of time stamping that can be implemented in hardware or software.

Functioning

Time determination, 2-step procedure

A basic function in all PTP protocols is the option of allocating exact timestamps for sending and receiving of Ethernet telegrams in the Ethernet hardware. This enables the delay times between the grandmaster and the slaves to be calculated.

1. At time t_1 the grandmaster sends a *SyncMessage* containing the estimated send time to the slaves. It arrives at the slave at time t_2 .
2. Shortly afterwards the grandmaster sends a *FollowUp* containing the actual send time t_1 of the previous telegram.
3. This process is cyclically repeated at the *SyncInterval*, e.g. every 2 seconds. Defined/supported Sync Intervals are 1 s, 2 s, 4 s, 8 s, 16 s, 32 s (PTPv1) and 0.5 s, 1 s, 2 s, 4 s (PTPv2).

At longer intervals the slave examines the return path:

1. The slave sends a *DelayRequest* to the master, which receives it at time t_4 .
2. It responds with a *DelayResponse* message containing the time of arrival t_4 .

This can be used to calculate the latency periods on both sides and correct any drift.

In Fig. "PTPv1 traffic in 2-step mode" the grandmaster "192.168.200.2" triggers a *Sync* every 2 seconds. The slave "192.168.200.1" responds with a *DelayRequest* every 8 seconds.

3799	1.501307	192.168.200.2	224.0.1.130	PTPv1	Sync Message
3800	0.000295	192.168.200.2	224.0.1.130	PTPv1	Follow_Up Message
3801	2.001432	192.168.200.2	224.0.1.130	PTPv1	Sync Message
3802	0.000359	192.168.200.2	224.0.1.130	PTPv1	Follow_Up Message
3803	2.001513	192.168.200.2	224.0.1.130	PTPv1	Sync Message
3804	0.000304	192.168.200.2	224.0.1.130	PTPv1	Follow_Up Message
3805	2.001435	192.168.200.2	224.0.1.130	PTPv1	Sync Message
3806	0.000322	192.168.200.2	224.0.1.130	PTPv1	Follow_Up Message
3807	0.420229	192.168.200.1	224.0.1.130	PTPv1	Delay_Request Message
3808	0.079926	192.168.200.2	224.0.1.130	PTPv1	Delay_Response Message
3809	1.501252	192.168.200.2	224.0.1.130	PTPv1	Sync Message

```

⊕ Frame 3799 (182 bytes on wire (182 bytes captured)
⊖ Ethernet II, Src: Beckhoff_03:04:05 (00:01:05:03:04:05), Dst: IPv4mcast_00:01:82 (01:00:5e:00:01:82)
  ⊕ Destination: IPv4mcast_00:01:82 (01:00:5e:00:01:82)
  ⊕ Source: Beckhoff_03:04:05 (00:01:05:03:04:05)
    Type: IP (0x0800)
    Trailer: 0101051000008000F879DDEA
    Frame check sequence: 0x1d340000 [incorrect, should be 0x61fb99d8]
⊕ Internet Protocol, Src: 192.168.200.2 (192.168.200.2), Dst: 224.0.1.130 (224.0.1.130)
⊕ User Datagram Protocol, Src Port: ptp-event (319), Dst Port: ptp-event (319)
  Source port: ptp-event (319)
  Destination port: ptp-event (319)
  Length: 132
  ⊕ Checksum: 0x98a7 [correct]
⊕ Precision Time Protocol (IEEE1588)
  versionPTP: 1
  versionNetwork: 1
  subdomain: _ALT1
  messageType: Event Message (1)
  sourceCommunicationTechnology: IEEE 802.3 (Ethernet) (1)
  sourceUuid: Beckhoff_03:04:05 (00:01:05:03:04:05)
  sourcePortId: 1
  sequenceId: 1218
    
```

Fig. 4: PTPv1 traffic in 2-step mode

Routing

The telegrams are usually routed via IP/UDP through multicast via ports 319 (event messages: SyncMsg, DelayReq) and 320 (general messages: FollowUp, Announce, DelayResp, Management). These ports should be prioritized, if possible.

Four multicast domains are defined, so that up to four independent time networks can be established within a network:

- `_DFLT`: 224.0.1.129
- `_ALT1`: 224.0.1.130
- `_ALT2`: 224.0.1.131
- `_ALT3`: 224.0.1.132

The domains in the grandmaster and slave clocks must match. A dedicated IP address for the slave is only required if the PTP master requires this.

Terminology

Stratum: measure for the quality of a clock. Stratum-1 is the highest level.

Grandmaster Clock selection

A balanced PTP system according to IEEE1588 only has a single time source, the Grandmaster Clock. Each PTP clock can start off as a Master or Slave Clock. A PTP clock uses different criteria (Stratum, Preferred flag) to indicate the quality of the time it supplies, depending on whether it is based on GPS, quartz, atomic clock or similar, for example. If more than one PTP clock exists in a network, the PTP clocks select the best clock within the network as timer (Grandmaster Clock), based on the standardized BMCA, which should be integrated in the same way in all PTP clocks. The remaining clocks switch to the passive mode, as long as the selected Grandmaster Clock sends out Sync messages.

Through its parameters a PTP clock can enforce Grandmaster Clock status within a network (by setting its parameter optimally) or Slave Clock status (by setting its parameter in the worst possible way).

All devices automatically run the BMCA in each cycle (generally every few seconds). This ensures that the system can respond to changes in topology and always selects the best Grandmaster Clock.

Infrastructure

If more than two directly connected PTP clocks are used, FastEthernet switches have to be used for 100 Mbit. Since the synchronization mechanism takes into account the signal run times for compensation of network delays, irregular latency periods along the signal route have a negative effect on the synchronization precision.

Normal switches based on the store and forward concept are therefore not very suitable. The synchronization precision of special IEEE1588 switches with "boundary clocks" depends on the protocol.

PTPv1

The boundary clock switches appear as Slave Clocks relative to the Grandmaster Clock, although at their outgoing ports they appear as Grandmaster Clock relative to the connected slaves. The switches insert their own send time into the telegram and therefore act as "transparent clocks".

PTPv2

PTPv2 has a further clock type in addition to boundary clocks, the transparent clock. Transparent clock switches insert their own cycle time into the telegram. (Cycle times are added up in a CorrectionField). Transparent clocks are invisible for other PTP clocks.

PTPv1 vs. PTPv2

- PTPv1:
 - PTPv1 according to IEEE 1588-2002
 - PTPv1 via UDP (ISO/OSI layer 3, EtherType: UDP x0800)
 - shortest Sync (synchronization) interval: 1 second
 - compensation of network delays (assumption: the path and return path are time-symmetric)
- PTPv2:
 - PTPv2 according to IEEE 1588-2008
 - PTPv2 via UDP (ISO/OSI layer 3, EtherType: UDP x0800) or via Ethernet (layer 2, EtherType x88F7)
 - shorter Sync intervals: e.g. 500 ms
 - CorrectionField: each switch component can use this field for adding up its cycle time for analysis by the slaves
 - compensation of network delays (assumption: path and return path time-asymmetric --> more realistic)
 - "transparent" switches, "transparent clocks": insert their own send time into the telegram when forwarding Sync telegrams

PTP modes

1. "SlaveOnly":

The terminal is permanently configured as synchronization slave through lower settings and cannot become grandmaster.

The BMCA is run cyclically to determine the best available synchronization source in the network.

2. "Grandmaster":

The terminal is configured as grandmaster through high settings and is available as synchronization source for other clocks in the network. The BMCA is also run in this mode, although it is only used to check whether another grandmaster with higher quality is available. If this is the case, terminal switches to passive mode.

3. "Best Master Clock":

The terminal can act as master clock or slave clock.

The BMCA is run cyclically to determine the best available clock.

If no better clock is found, the terminal becomes grandmaster.

3 Basics communication

3.1 EtherCAT basics

Please refer to the [EtherCAT System Documentation](#) for the EtherCAT fieldbus basics.

3.2 General notes for setting the watchdog

The ELxxxx terminals are equipped with a safety device (watchdog) which, e.g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e.g. to FALSE (off) or an output value.

The EtherCAT slave controller (ESC) features two watchdogs:

- SM watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

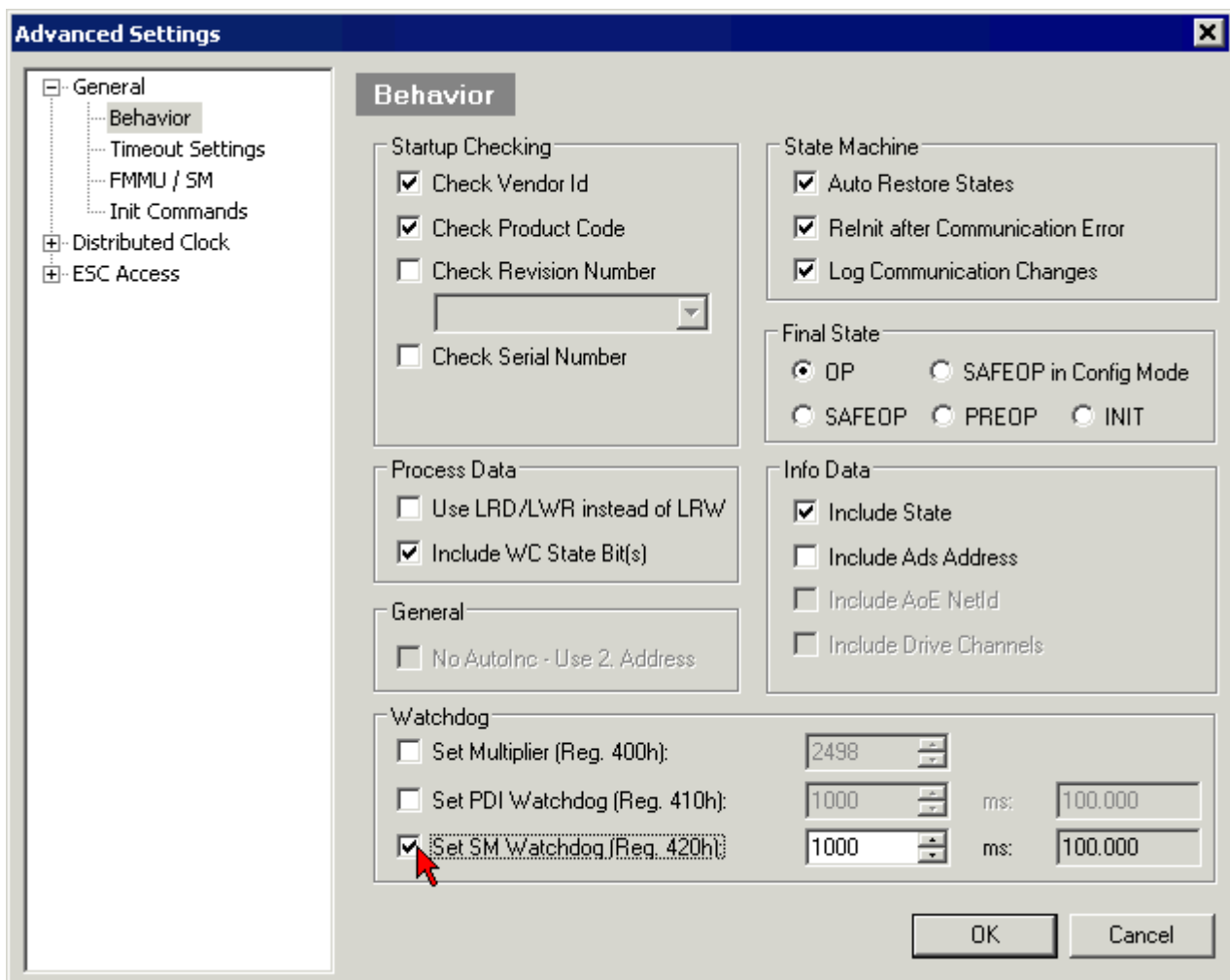


Fig. 5: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i.e. x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.

- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers x0400/0410/0420: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The SyncManager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to ~170 seconds. For "complex" EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via Reg. 400/420 but executed by the μ C and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the EtherCAT slave controller (ESC) for longer than the set and activated PDI watchdog time, this watchdog is triggered.

PDI (Process Data Interface) is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

Calculation

Watchdog time = $[1/25 \text{ MHz} * (\text{Watchdog multiplier} + 2)] * \text{PDI/SM watchdog}$

Example: default setting Multiplier=2498, SM watchdog=1000 -> 100 ms

The value in Multiplier + 2 corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

The function for switching off of the SM watchdog via SM watchdog = 0 is only implemented in terminals from version -0016. In previous versions this operating mode should not be used.

⚠ CAUTION

Damage of devices and undefined state possible!

If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is the deactivation of the watchdog! Set outputs are NOT set in a safe state if the communication is interrupted.

3.3 EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.

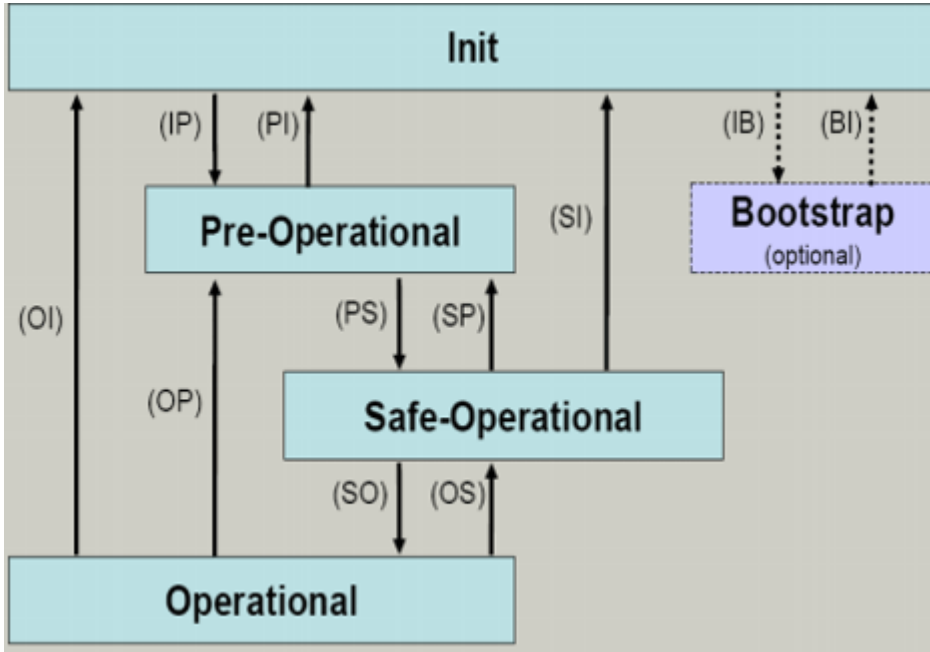


Fig. 6: States of the EtherCAT State Machine

Init

After switch-on the EtherCAT slave in the *Init* state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)

During the transition between *Init* and *Pre-Op* the EtherCAT slave checks whether the mailbox was initialized correctly.

In *Pre-Op* state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)

During transition between *Pre-Op* and *Safe-Op* the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

● Outputs in SAFEOP state

i The default set [watchdog_1 \[p. 20\]](#) monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from *Safe-Op* to *Op* it must transfer valid output data.

In the *Op* state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

Boot

In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state.

In the *Boot* state mailbox communication via the *file access over EtherCAT* (FoE) protocol is possible, but no other mailbox communication and no process data communication.

3.4 CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read only) or variable parameters which they require for operation, diagnostics or commissioning.

CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus. The EtherCAT master (TwinCAT System Manager) can access the local CoE lists of the slaves via EtherCAT in read or write mode, depending on the attributes.

Different CoE parameter types are possible, including string (text), integer numbers, Boolean values or larger byte fields. They can be used to describe a wide range of features. Examples of such parameters include manufacturer ID, serial number, process data settings, device name, calibration values for analog measurement or passwords.

The order is specified in two levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

- Index: 0x0000 ...0xFFFF (0...65535_{dec})
- SubIndex: 0x00...0xFF (0...255_{dec})

A parameter localized in this way is normally written as 0x8010:07, with preceding "0x" to identify the hexadecimal numerical range and a colon between index and subindex.

The relevant ranges for EtherCAT fieldbus users are:

- 0x1000: This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.
- 0x8000: This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

Other important ranges are:

- 0x4000: here are the channel parameters for some EtherCAT devices. Historically, this was the first parameter area before the 0x8000 area was introduced. EtherCAT devices that were previously equipped with parameters in 0x4000 and changed to 0x8000 support both ranges for compatibility reasons and mirror internally.
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

● Availability

Not every EtherCAT device must have a CoE list. Simple I/O modules without dedicated processor usually have no variable parameters and therefore no CoE list.

If a device has a CoE list, it is shown in the TwinCAT System Manager as a separate tab with a listing of the elements:

Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PwM RxDPO-Par Ch.1	RO	> 6 <
1401:0	PwM RxDPO-Par Ch.2	RO	> 6 <
1402:0	PwM RxDPO-Par h.1 Ch.1	RO	> 6 <
1403:0	PwM RxDPO-Par h.1 Ch.2	RO	> 6 <
1600:0	PwM RxDPO-Map Ch.1	RO	> 1 <

Fig. 7: "CoE Online" tab

The figure above shows the CoE objects available in device "EL2502", ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

NOTE

Changes in the CoE directory (CAN over EtherCAT), program access

When using/manipulating the CoE parameters observe the general CoE notes in chapter "[CoE interface](#)" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the [Beckhoff website](#)),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see [TwinCAT3 | PLC Bibliothek: Tc2 EtherCAT](#) and [Example program R/W CoE](#))

Data management and function "NoCoeStorage"

Some parameters, particularly the setting parameters of the slave, are configurable and writeable. This can be done in write or read mode

- via the System Manager (Fig. "CoE Online" tab) by clicking
This is useful for commissioning of the system/slaves. Click on the row of the index to be parameterized and enter a value in the "SetValue" dialog.
- from the control system/PLC via ADS, e.g. through blocks from the TcEtherCAT.lib library
This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

i Data management

If slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart. The situation may be different with other manufacturers.

An EEPROM is subject to a limited lifetime with respect to write operations. From typically 100,000 write operations onwards it can no longer be guaranteed that new (changed) data are reliably saved or are still readable. This is irrelevant for normal commissioning. However, if CoE parameters are continuously changed via ADS at machine runtime, it is quite possible for the lifetime limit to be reached. Support for the NoCoeStorage function, which suppresses the saving of changed CoE values, depends on the firmware version.

Please refer to the technical data in this documentation as to whether this applies to the respective device.

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE 0xF008 and remains active as long as the code word is not changed. After switching the device on it is then inactive. Changed CoE values are not saved in the EEPROM and can thus be changed any number of times.
- Function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

i Startup list

Changes in the local CoE list of the terminal are lost if the terminal is replaced. If a terminal is replaced with a new Beckhoff terminal, it will have the default settings. It is therefore advisable to link all changes in the CoE list of an EtherCAT slave with the Startup list of the slave, which is processed whenever the EtherCAT fieldbus is started. In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

If EtherCAT slaves are used which are unable to store local CoE values permanently, the Startup list must be used.

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager
The values are stored locally in the EtherCAT slave
- If the value is to be stored permanently, enter it in the Startup list.
The order of the Startup entries is usually irrelevant.

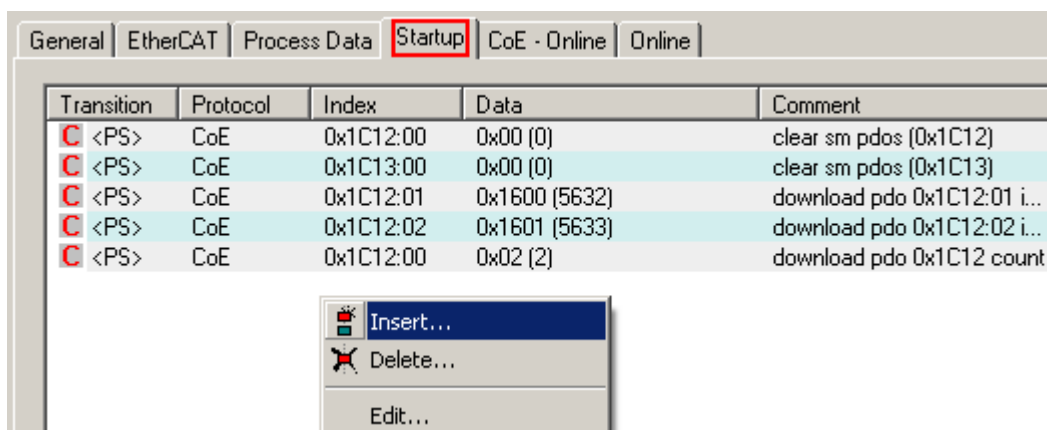


Fig. 8: Startup list in the TwinCAT System Manager

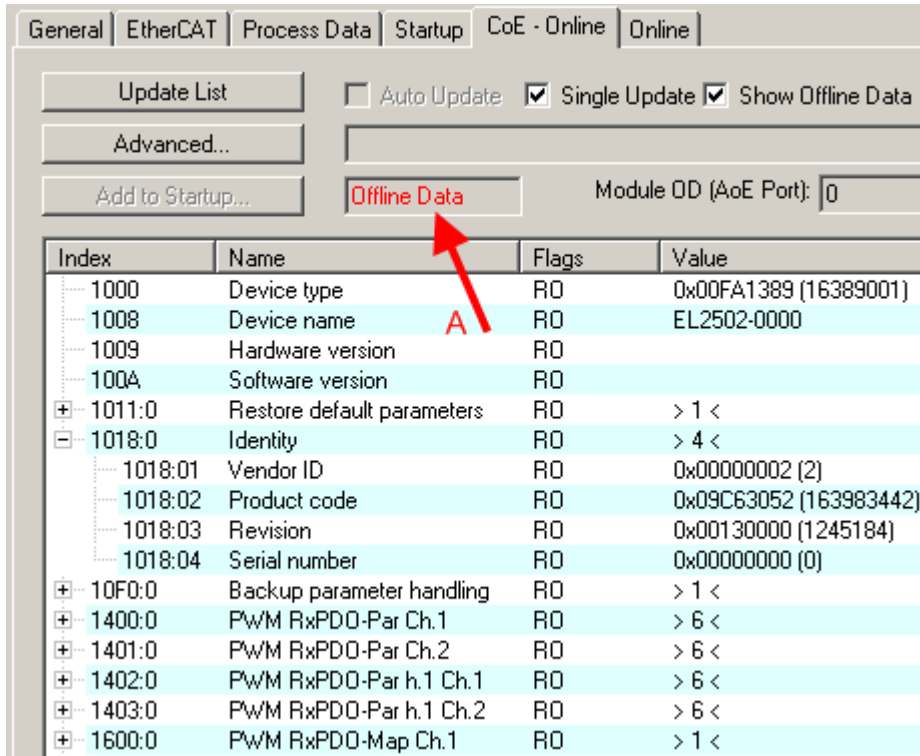
The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

Online/offline list

While working with the TwinCAT System Manager, a distinction has to be made whether the EtherCAT device is “available”, i.e. switched on and linked via EtherCAT and therefore **online**, or whether a configuration is created **offline** without connected slaves.

In both cases a CoE list as shown in Fig. “CoE online tab” is displayed. The connectivity is shown as offline/online.

- If the slave is offline
 - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
 - The configured status is shown under Identity.
 - No firmware or hardware version is displayed, since these are features of the physical device.
 - **Offline** is shown in red.



The screenshot shows the 'CoE - Online' tab in the Beckhoff software. The 'Offline Data' section is highlighted in red, and a red arrow points to the 'Offline Data' label. The table below shows the details of the offline data.

Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PwM RxDPO-Par Ch.1	RO	> 6 <
1401:0	PwM RxDPO-Par Ch.2	RO	> 6 <
1402:0	PwM RxDPO-Par h.1 Ch.1	RO	> 6 <
1403:0	PwM RxDPO-Par h.1 Ch.2	RO	> 6 <
1600:0	PwM RxDPO-Map Ch.1	RO	> 1 <

Fig. 9: Offline list

- If the slave is online
 - The actual current slave list is read. This may take several seconds, depending on the size and cycle time.
 - The actual identity is displayed
 - The firmware and hardware version of the equipment according to the electronic information is displayed
 - **Online** is shown in green.

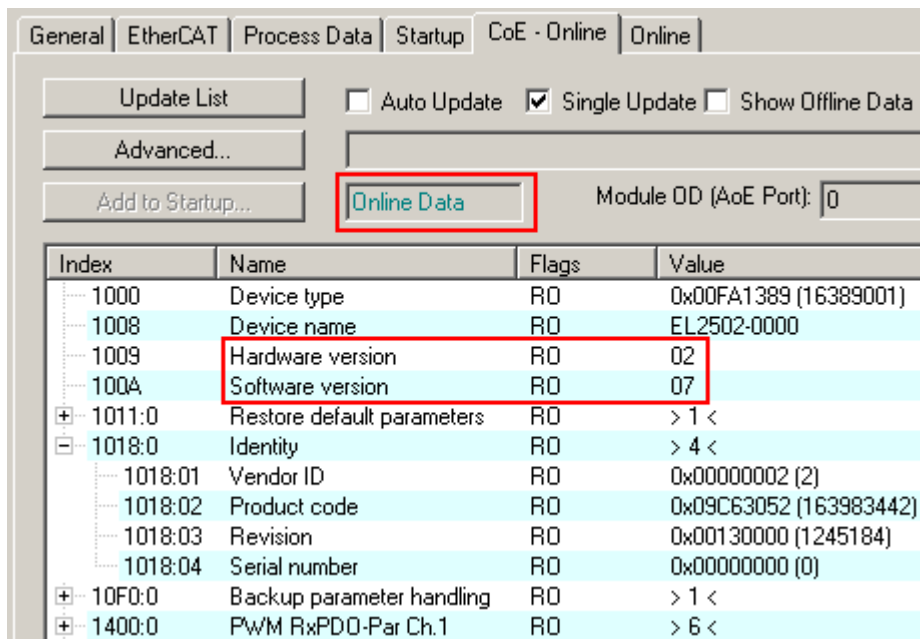


Fig. 10: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 0...10 V input terminal also has four logical channels and therefore four identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder “n” tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in $16_{dec}/10_{hex}$ steps. The parameter range 0x8000 exemplifies this:

- Channel 0: parameter range 0x8000:00 ... 0x800F:255
- Channel 1: parameter range 0x8010:00 ... 0x801F:255
- Channel 2: parameter range 0x8020:00 ... 0x802F:255
- ...

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the [EtherCAT system documentation](#) on the Beckhoff website.

3.5 Distributed Clock

The distributed clock represents a local clock in the EtherCAT slave controller (ESC) with the following characteristics:

- Unit *1 ns*
- Zero point *1.1.2000 00:00*
- Size *64 bit* (sufficient for the next 584 years; however, some EtherCAT slaves only offer 32-bit support, i.e. the variable overflows after approx. 4.2 seconds)
- The EtherCAT master automatically synchronizes the local clock with the master clock in the EtherCAT bus with a precision of < 100 ns.

For detailed information please refer to the [EtherCAT system description](#).

4 Mounting and wiring

4.1 Explosion protection

4.1.1 ATEX - Special conditions (standard temperature range)

WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. KEMA 10ATEX0075 X Issue 9)

Marking

The Beckhoff fieldbus components with standard temperature range certified according to the ATEX directive for potentially explosive areas bear one of the following markings:



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 ... +55°C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: 0 ... +55°C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

4.1.2 Continulative documentation for ATEX and IECEx

NOTE



Continulative documentation about explosion protection according to ATEX and IECEx




Pay also attention to the continuative documentation

Ex. Protection for Terminal Systems

Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx,

that is available for [download](#) within the download area of your product on the Beckhoff homepage www.beckhoff.com!

4.2 UL notice

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

Basic principles

UL certification according to UL508. Devices with this kind of certification are marked by this sign:



4.3 Note - Power supply

⚠ WARNING**Power supply from SELV/PELV power supply unit!**

SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV/PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV (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.

4.4 Mounting and demounting - terminals with front unlocking

The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e.g. mounting rail TH 35-15).

● Fixing of mounting rails

i The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the recommended mounting rails under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

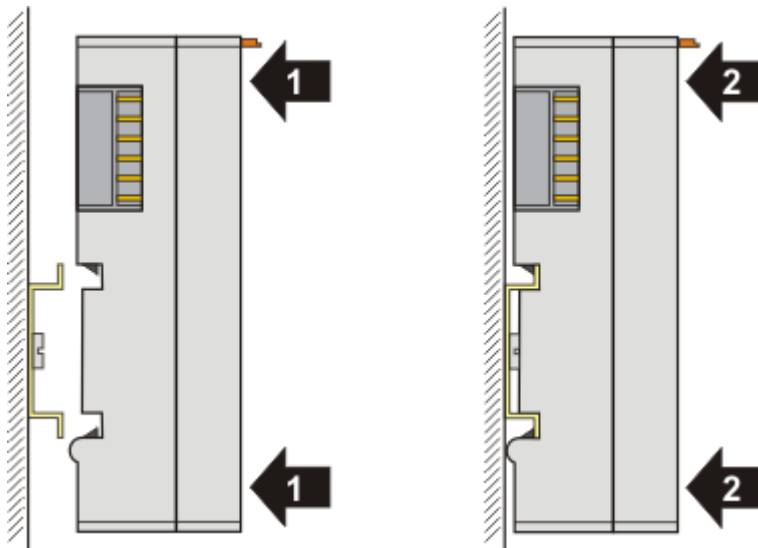
⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Mounting

- Fit the mounting rail to the planned assembly location.

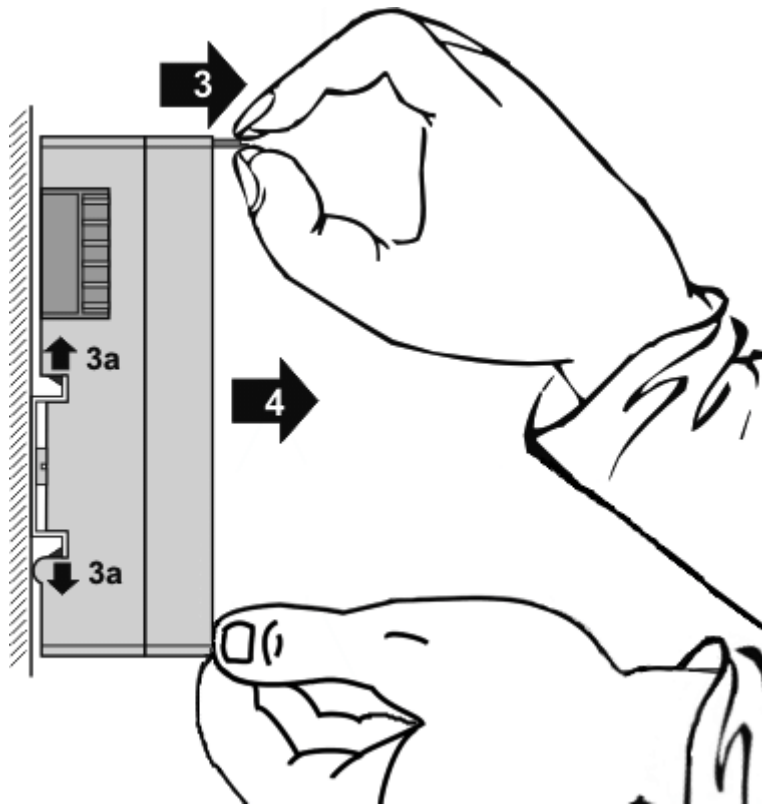


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- Attach the cables.

Demounting

- Remove all the cables.
- Lever the unlatching hook back with thumb and forefinger (3). An internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module.



- Pull (4) the terminal module away from the mounting surface.
Avoid canting of the module; you should stabilize the module with the other hand, if required.

4.5 Mounting and demounting - terminals with traction lever unlocking

The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e.g. mounting rail TH 35-15).

● Fixing of mounting rails

i The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the recommended mounting rails under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

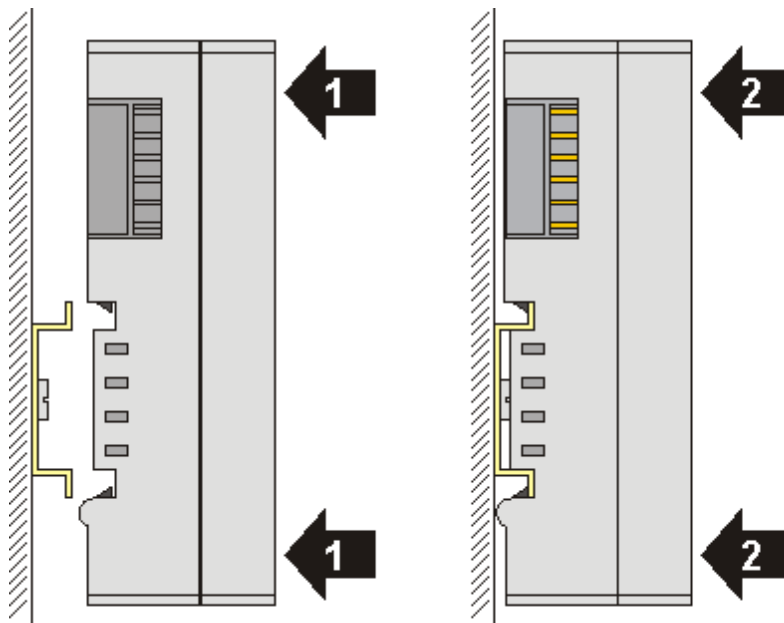
⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Mounting

- Fit the mounting rail to the planned assembly location.

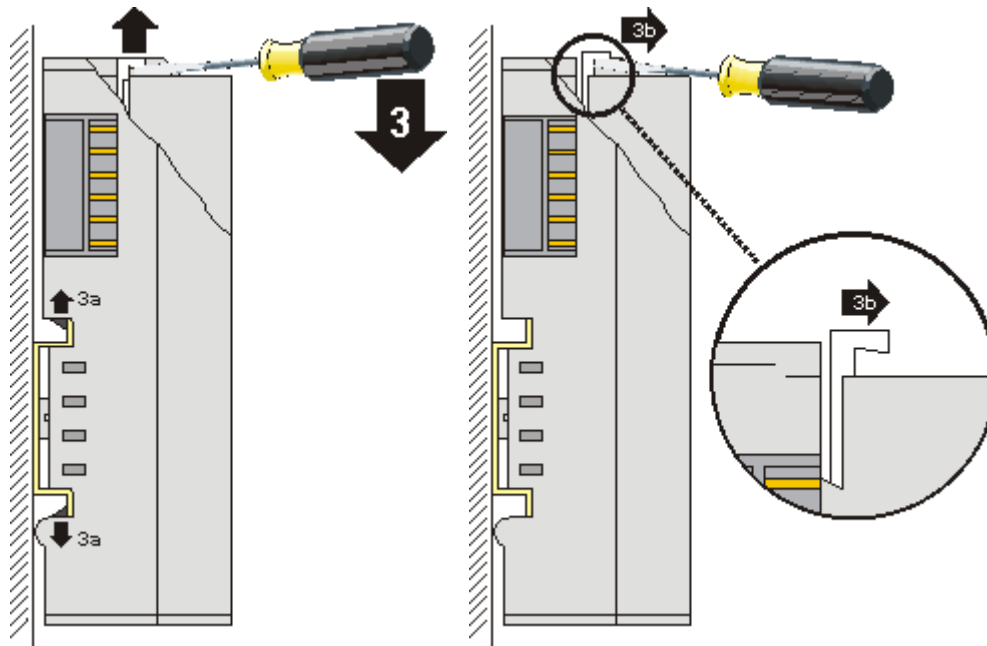


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

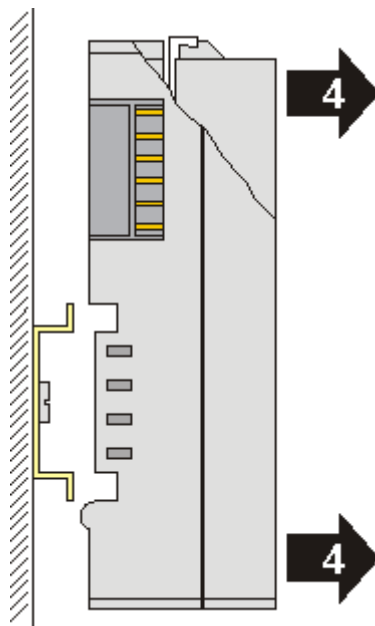
- Attach the cables.

Demounting

- Remove all the cables. Thanks to the KM/EM connector, it is not necessary to remove all the cables separately for this, but for each KM/EM connector simply undo 2 screws so that you can pull them off (fixed wiring)!
- Lever the unlatching hook on the left-hand side of the terminal module upwards with a screwdriver (3). As you do this
 - an internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module,
 - the unlatching hook moves forwards (3b) and engages



- In the case 32 and 64 channel terminal modules (KMxxx4 and KMxxx8 or EMxxx4 and EMxxx8) you now lever the second unlatching hook on the right-hand side of the terminal module upwards in the same way.
- Pull (4) the terminal module away from the mounting surface.



4.6 Recommended mounting rails

Terminal Modules and EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series can be snapped onto the following recommended mounting rails:

- mounting rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)
- mounting rail TH 35-15 with 1.5 mm material thickness

● Pay attention to the material thickness of the mounting rail

i Terminal Modules and EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series does not fit to the mounting rail TH 35-15 with 2.2 to 2.5 mm material thickness (according to EN 60715)!

4.7 Positioning of passive Terminals

i Hint for positioning of passive terminals in the bus terminal block

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.

To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)

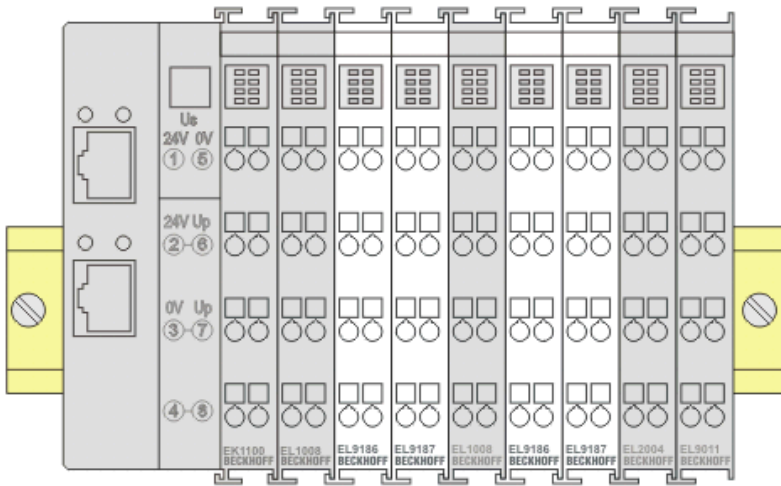


Fig. 11: Correct positioning

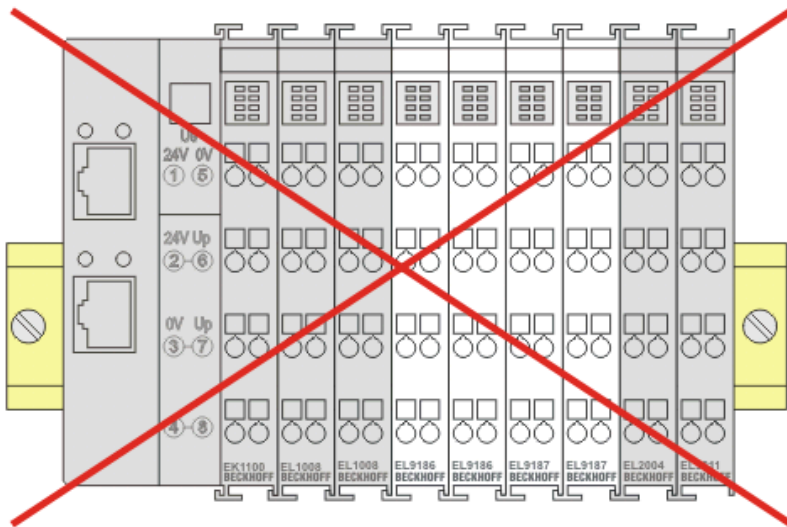


Fig. 12: Incorrect positioning

4.8 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

Please refer to the technical data for a terminal to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *Recommended distances for standard installation position*). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.

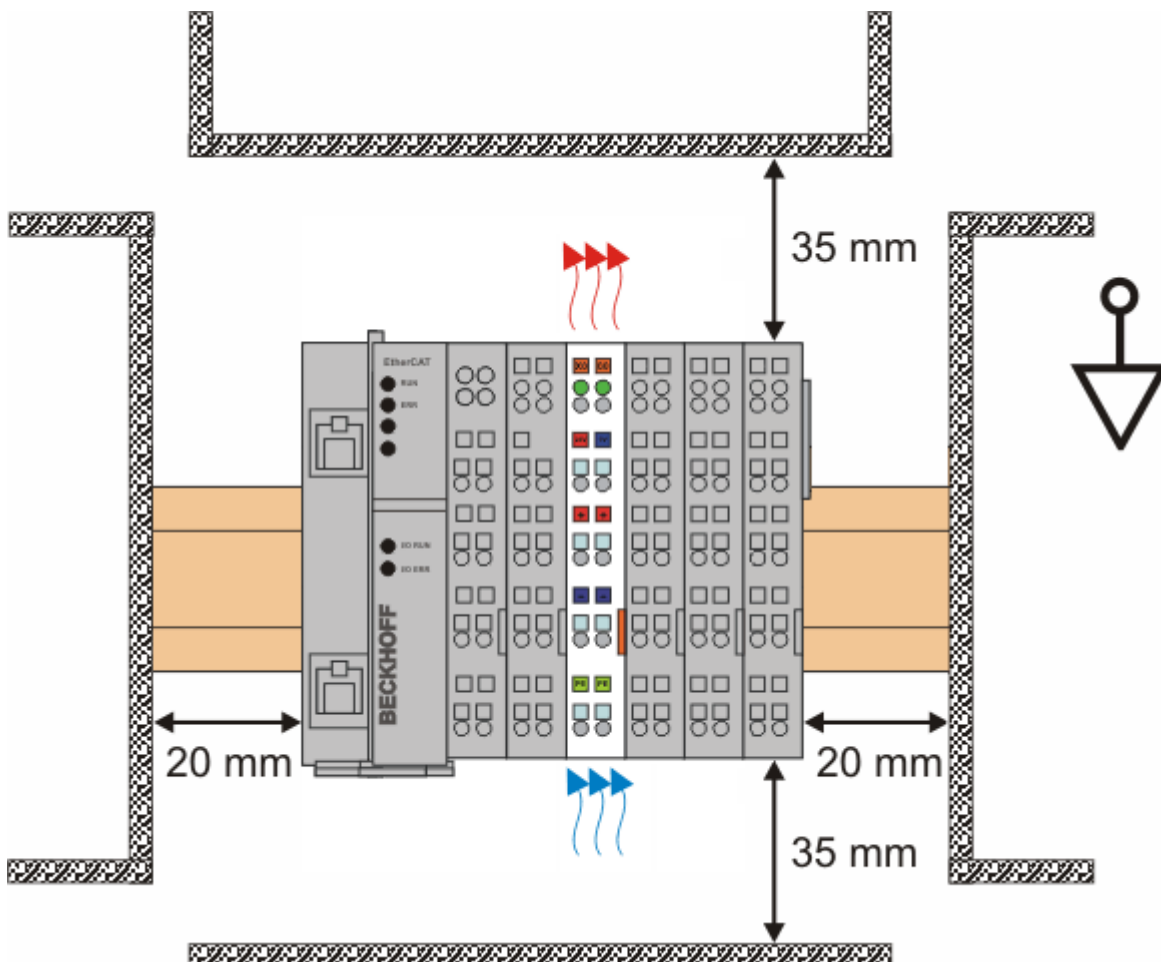


Fig. 13: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommended.

Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Fig *Other installation positions*.

The minimum distances to ambient specified above also apply to these installation positions.

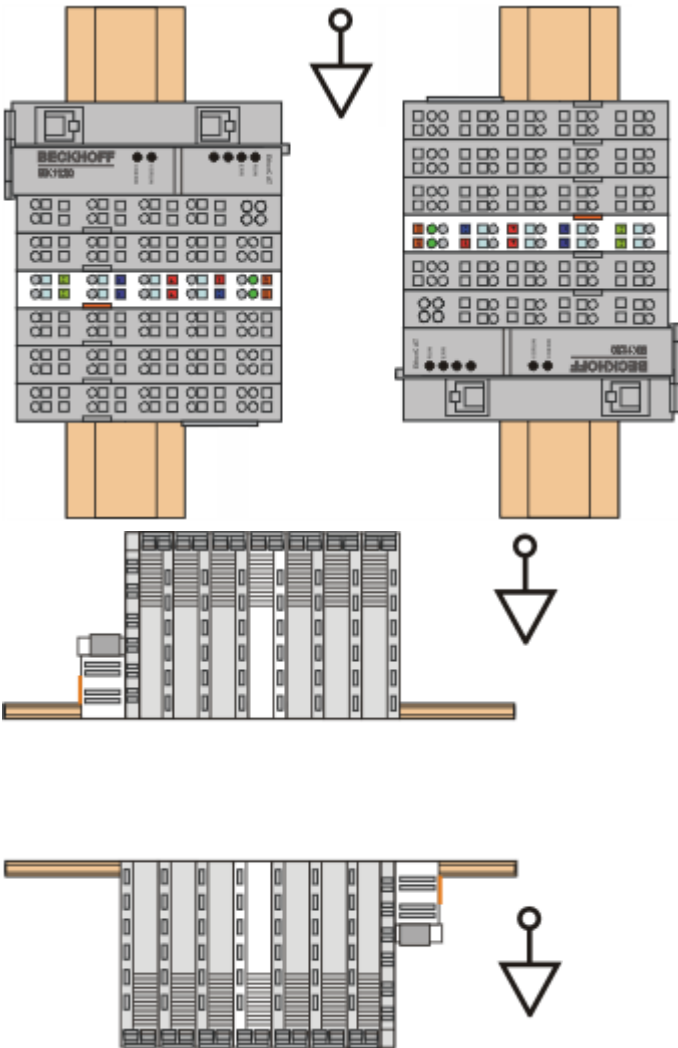
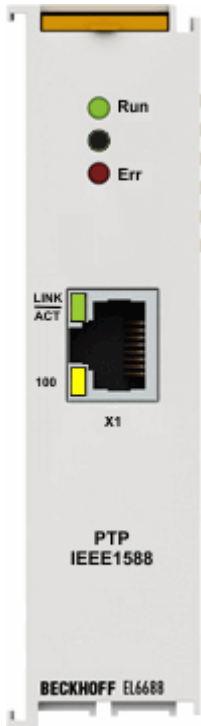


Fig. 14: Other installation positions

4.9 EL6688 - LEDs and connection



LEDs

LED	Color	Meaning	
Run	green	These LEDs indicate the terminal's operating state:	
		off	State of the EtherCAT State Machine [▶ 68]: INIT = initialization of the terminal or BOOTSTRAP = function for <u>firmware updates</u> [▶ 113] of the terminal
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: SAFEOP = verification of the <u>Sync Manager</u> [▶ 68] channels and the distributed clocks. Outputs remain in safe state
on	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible		
Err	red	Fault indication EtherCAT	
Lnk/Act	green	This LED shows the connection state of the Ethernet port:	
		off	no Link signal
		on	Link signal present, Ethernet connection established
flashing	Data traffic		
100	yellow	off	Data transmission with 10 Mbit/s
		on	Data transmission with 100 Mbit/s

Connections

1 x RJ45 with 10BASE-T/100BASE-TX Ethernet

4.10 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

5.1 TwinCAT Development Environment

The Software for automation TwinCAT (The Windows Control and Automation Technology) will be distinguished into:

- TwinCAT 2: System Manager (Configuration) & PLC Control (Programming)
- TwinCAT 3: Enhancement of TwinCAT 2 (Programming and Configuration takes place via a common Development Environment)

Details:

- **TwinCAT 2:**
 - Connects I/O devices to tasks in a variable-oriented manner
 - Connects tasks to tasks in a variable-oriented manner
 - Supports units at the bit level
 - Supports synchronous or asynchronous relationships
 - Exchange of consistent data areas and process images
 - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
 - Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
 - Interconnection to all common fieldbusses
 - More...

Additional features:

- **TwinCAT 3 (eXtended Automation):**
 - Visual-Studio®-Integration
 - Choice of the programming language
 - Supports object orientated extension of IEC 61131-3
 - Usage of C/C++ as programming language for real time applications
 - Connection to MATLAB®/Simulink®
 - Open interface for expandability
 - Flexible run-time environment
 - Active support of Multi-Core- and 64-Bit-Operatingsystem
 - Automatic code generation and project creation with the TwinCAT Automation Interface
 - More...

Within the following sections commissioning of the TwinCAT Development Environment on a PC System for the control and also the basically functions of unique control elements will be explained.

Please see further information to TwinCAT 2 and TwinCAT 3 at <http://infosys.beckhoff.com>.

5.1.1 Installation of the TwinCAT real-time driver

In order to assign real-time capability to a standard Ethernet port of an IPC controller, the Beckhoff real-time driver has to be installed on this port under Windows.

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.

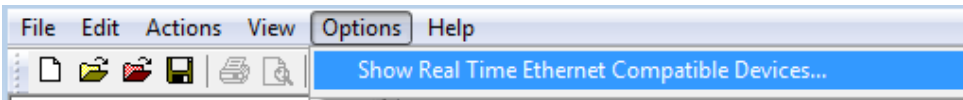


Fig. 15: System Manager “Options” (TwinCAT 2)

This have to be called up by the menu “TwinCAT” within the TwinCAT 3 environment:

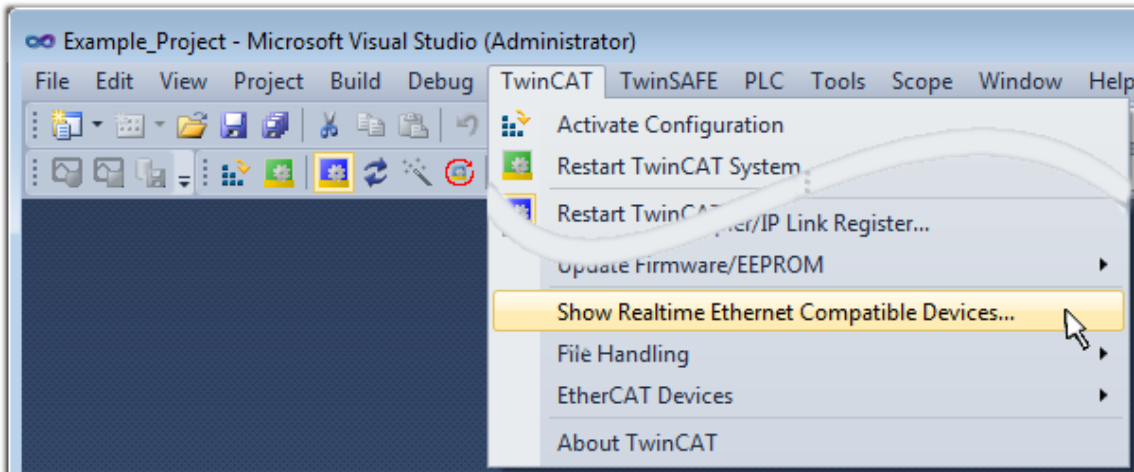


Fig. 16: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

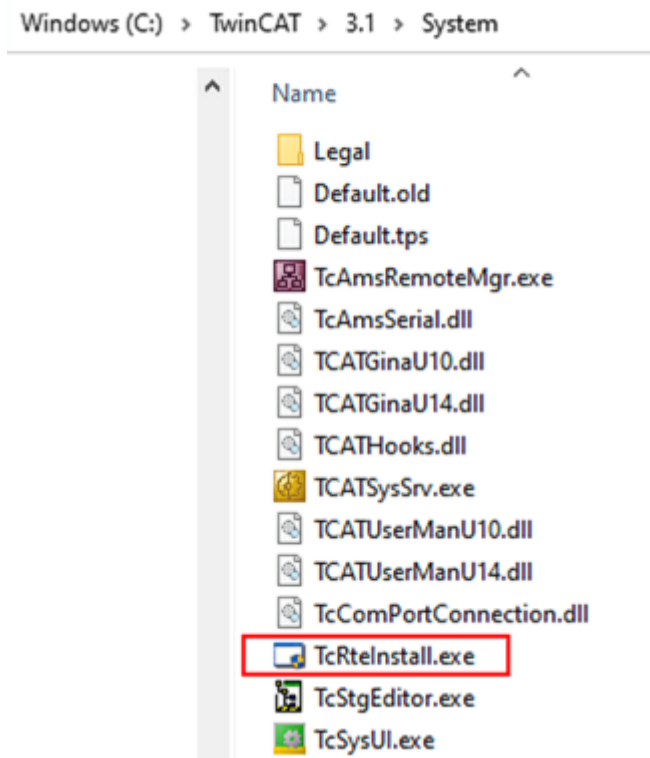


Fig. 17: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

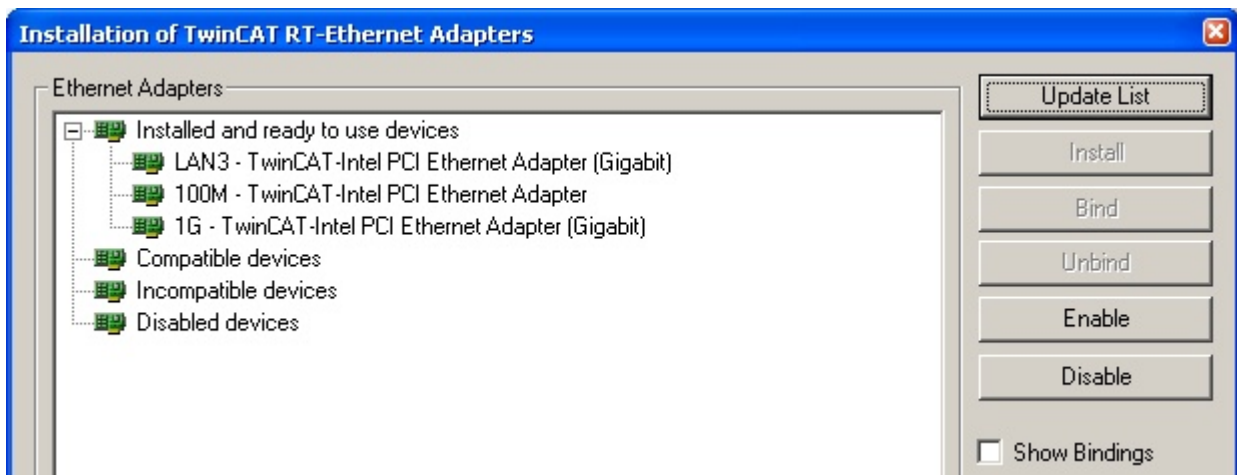


Fig. 18: Overview of network interfaces

Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

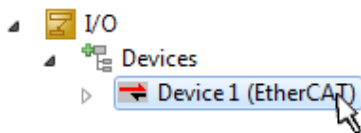
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation](#), section “Creating the EtherCAT device” [► 55] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 19: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices...” of tab “Adapter”

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)

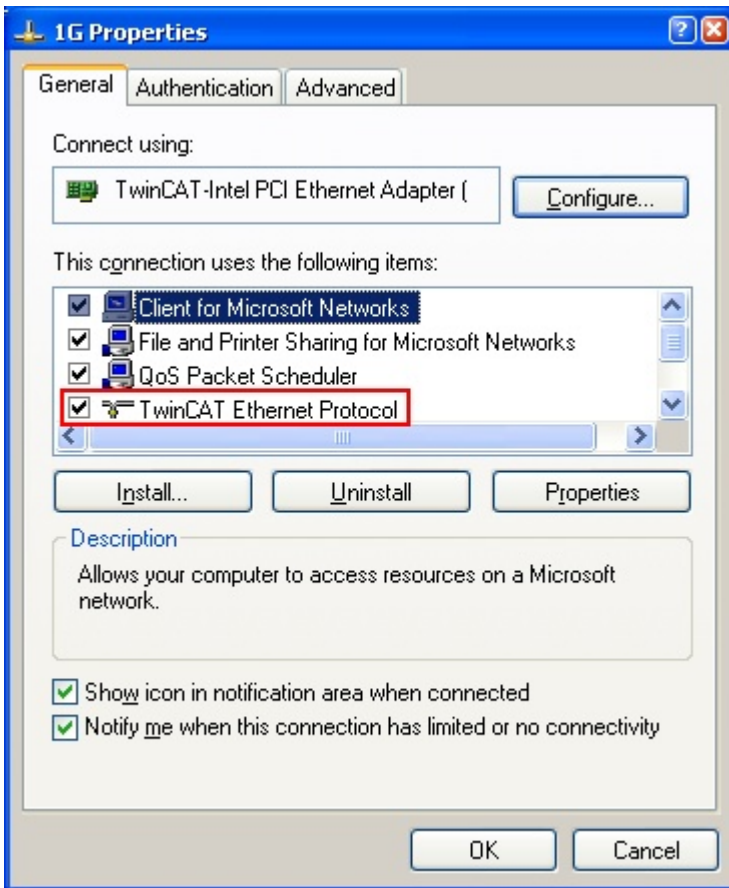


Fig. 20: Windows properties of the network interface

A correct setting of the driver could be:

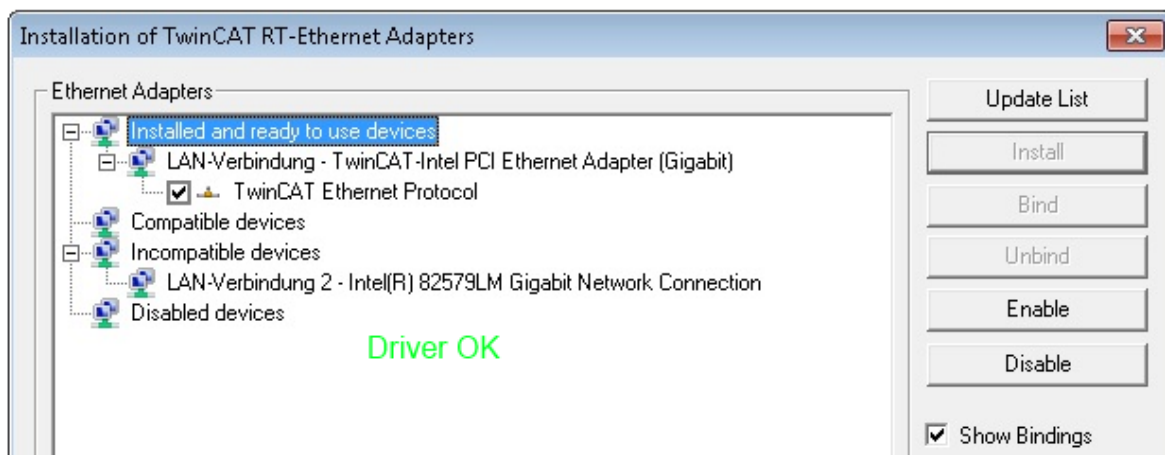


Fig. 21: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

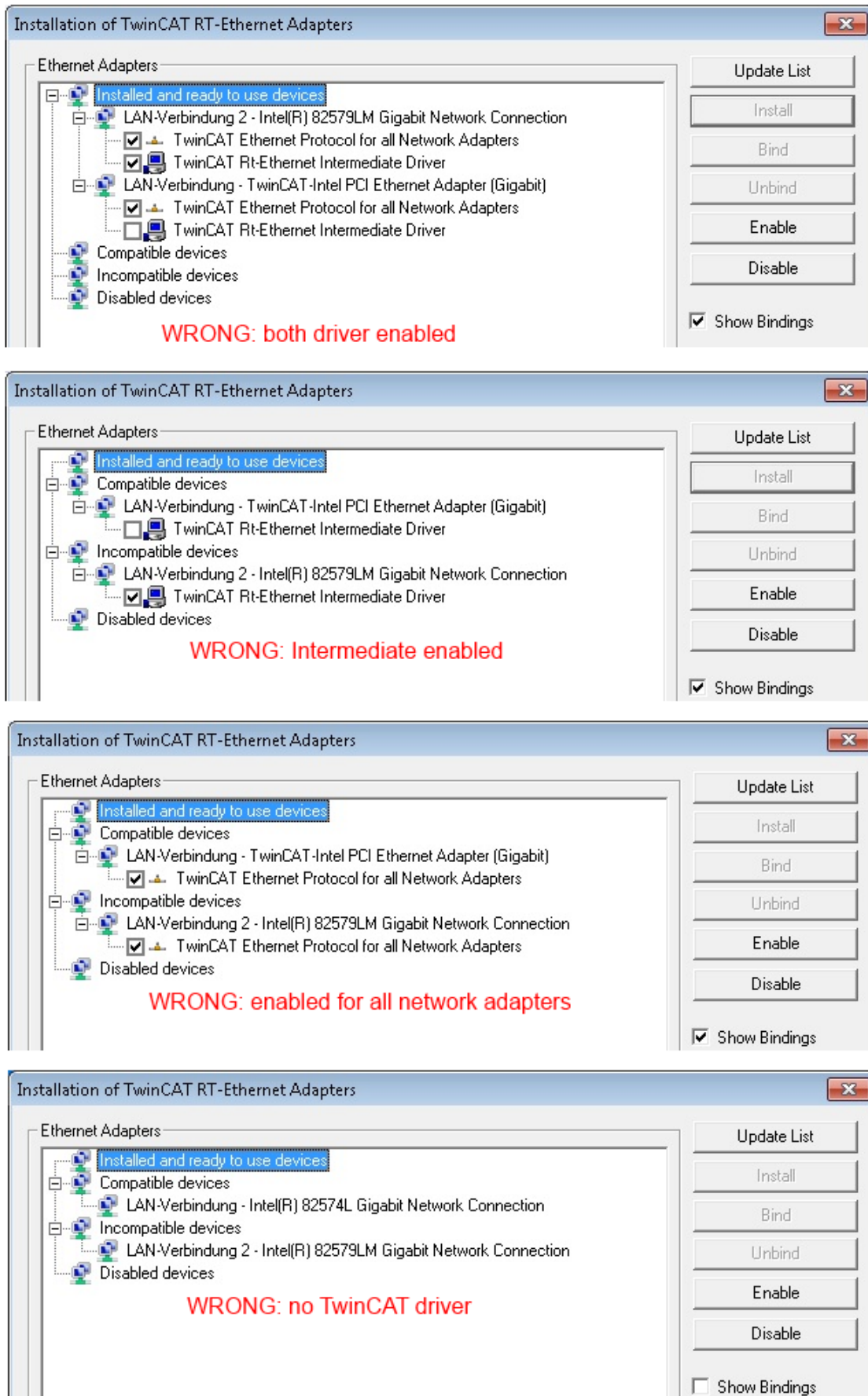


Fig. 22: Incorrect driver settings for the Ethernet port

IP address of the port used

i IP address/DHCP

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the “Internet Protocol TCP/IP” driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.

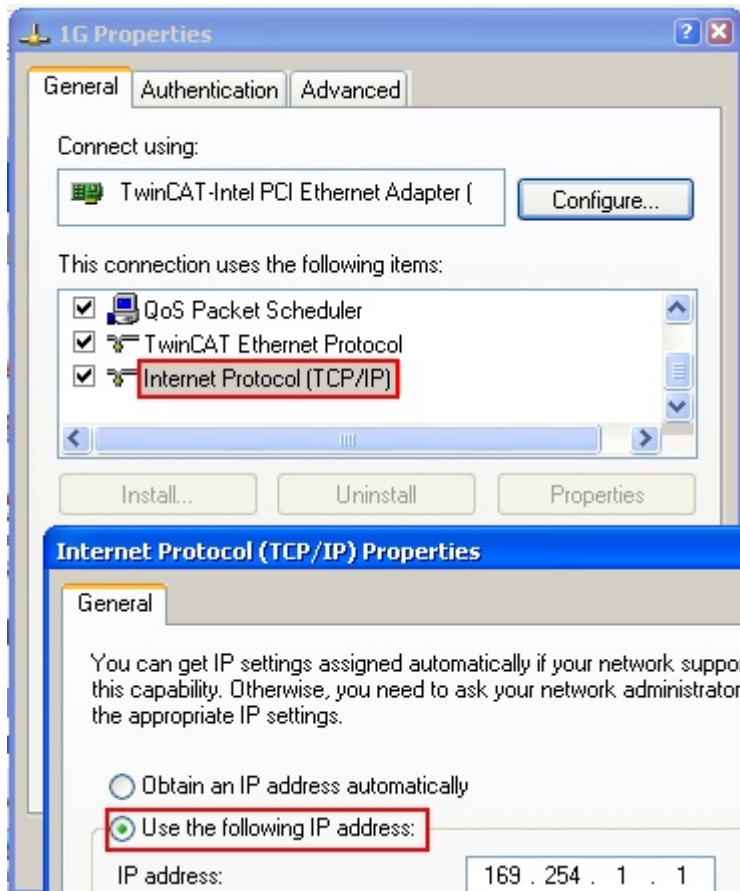


Fig. 23: TCP/IP setting for the Ethernet port

5.1.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to generate the configuration in online or offline mode. The device descriptions are contained in the so-called ESI files (EtherCAT Slave Information) in XML format. These files can be requested from the respective manufacturer and are made available for download. An *.xml file may contain several device descriptions.

The ESI files for Beckhoff EtherCAT devices are available on the [Beckhoff website](#).

The ESI files should be stored in the TwinCAT installation directory.

Default settings:

- **TwinCAT 2:** C:\TwinCAT\IO\EtherCAT
- **TwinCAT 3:** C:\TwinCAT\3.1\Config\Io\EtherCAT

The files are read (once) when a new System Manager window is opened, if they have changed since the last time the System Manager window was opened.

A TwinCAT installation includes the set of Beckhoff ESI files that was current at the time when the TwinCAT build was created.

For TwinCAT 2.11/TwinCAT 3 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet; by

- **TwinCAT 2:** Option → “Update EtherCAT Device Descriptions”
- **TwinCAT 3:** TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)...”

The [TwinCAT ESI Updater \[► 54\]](#) is available for this purpose.



ESI

The *.xml files are associated with *.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

Device differentiation

EtherCAT devices/slaves are distinguished by four properties, which determine the full device identifier. For example, the device identifier EL2521-0025-1018 consists of:

- family key “EL”
- name “2521”
- type “0025”
- and revision “1018”

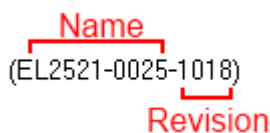


Fig. 24: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress 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. Each revision has its own ESI description. See further notes.

Online description

If the EtherCAT configuration is created online through scanning of real devices (see section Online setup) and no ESI descriptions are available for a slave (specified by name and revision) that was found, the System Manager asks whether the description stored in the device should be used. In any case, the System Manager needs this information for setting up the cyclic and acyclic communication with the slave correctly.

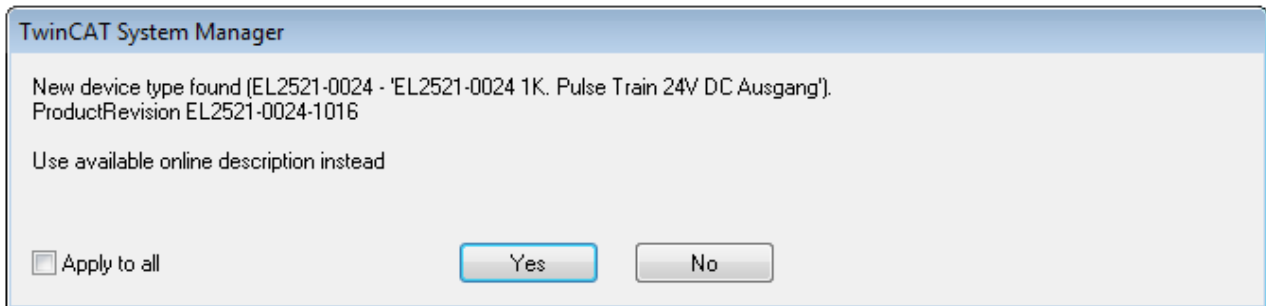


Fig. 25: OnlineDescription information window (TwinCAT 2)

In TwinCAT 3 a similar window appears, which also offers the Web update:

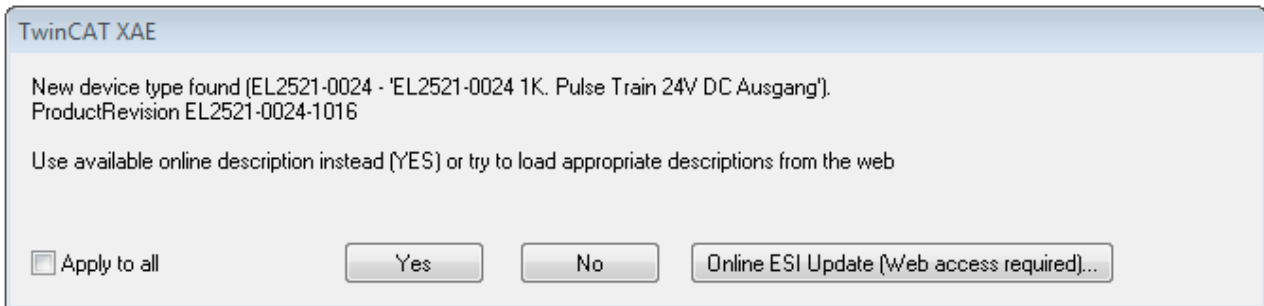


Fig. 26: Information window OnlineDescription (TwinCAT 3)

If possible, the Yes is to be rejected and the required ESI is to be requested from the device manufacturer. After installation of the XML/XSD file the configuration process should be repeated.

NOTE

Changing the “usual” configuration through a scan

- ✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019
 - a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).
 - b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter “General notes on the use of Beckhoff EtherCAT IO components” and for manual configuration to the chapter “Offline configuration creation [▶ 55]”.

If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore it’s recommended using an offline ESI file with priority in such a case.

The System Manager creates for online recorded device descriptions a new file “OnlineDescription0000...xml” in its ESI directory, which contains all ESI descriptions that were read online.

OnlineDescriptionCache00000002.xml

Fig. 27: File OnlineDescription.xml created by the System Manager

If a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol ">" in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example*).

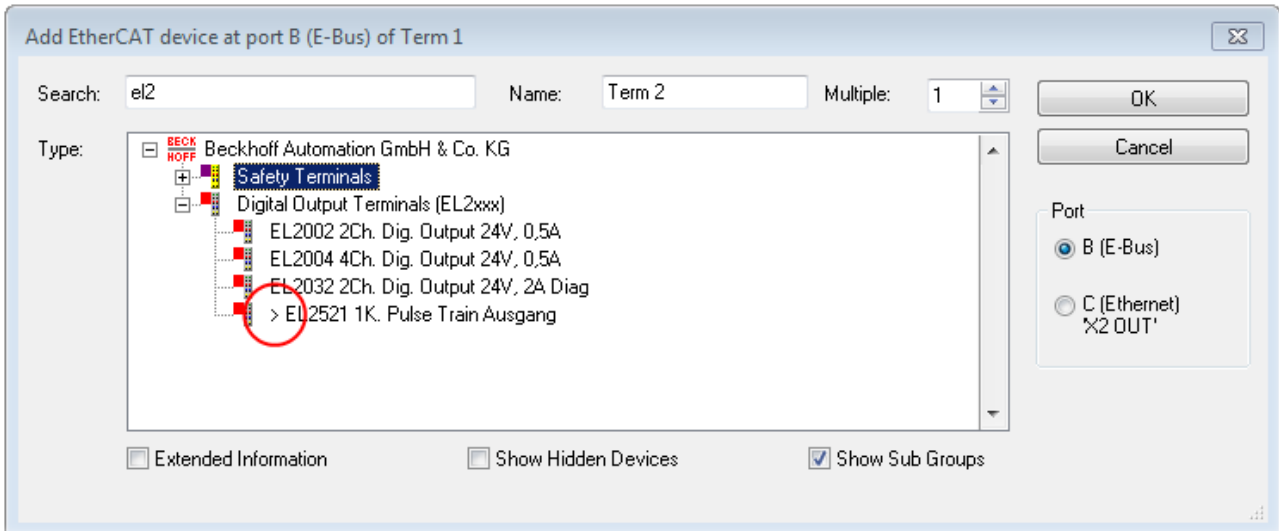


Fig. 28: Indication of an online recorded ESI of EL2521 as an example

If such ESI files are used and the manufacturer's files become available later, the file OnlineDescription.xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete "OnlineDescription0000...xml"
- restart TwinCAT System Manager

This file should not be visible after this procedure, if necessary press <F5> to update

i OnlineDescription for TwinCAT 3.x

In addition to the file described above "OnlineDescription0000...xml", a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

`C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml`

(Please note the language settings of the OS!)

You have to delete this file, too.

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.

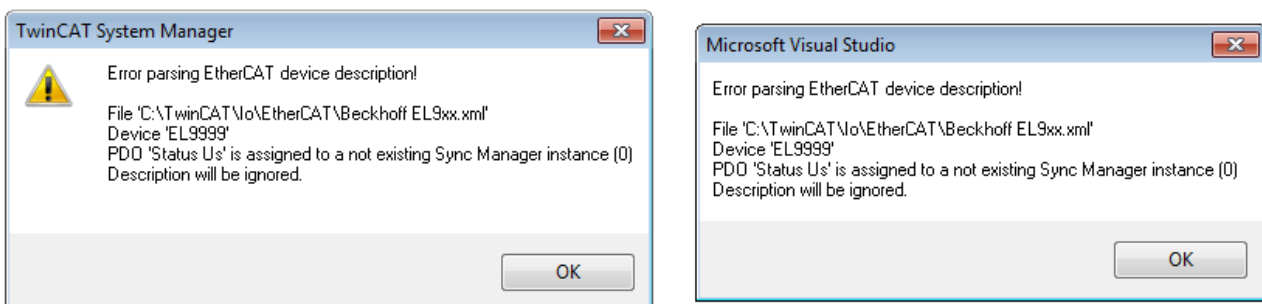


Fig. 29: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

- Structure of the *.xml does not correspond to the associated *.xsd file → check your schematics
- Contents cannot be translated into a device description → contact the file manufacturer

5.1.3 TwinCAT ESI Updater

For TwinCAT 2.11 and higher, the System Manager can search for current Beckhoff ESI files automatically, if an online connection is available:

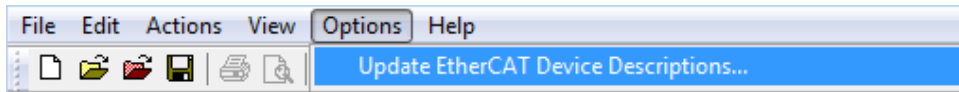


Fig. 30: Using the ESI Updater (>= TwinCAT 2.11)

The call up takes place under:
 “Options” → “Update EtherCAT Device Descriptions”

Selection under TwinCAT 3:

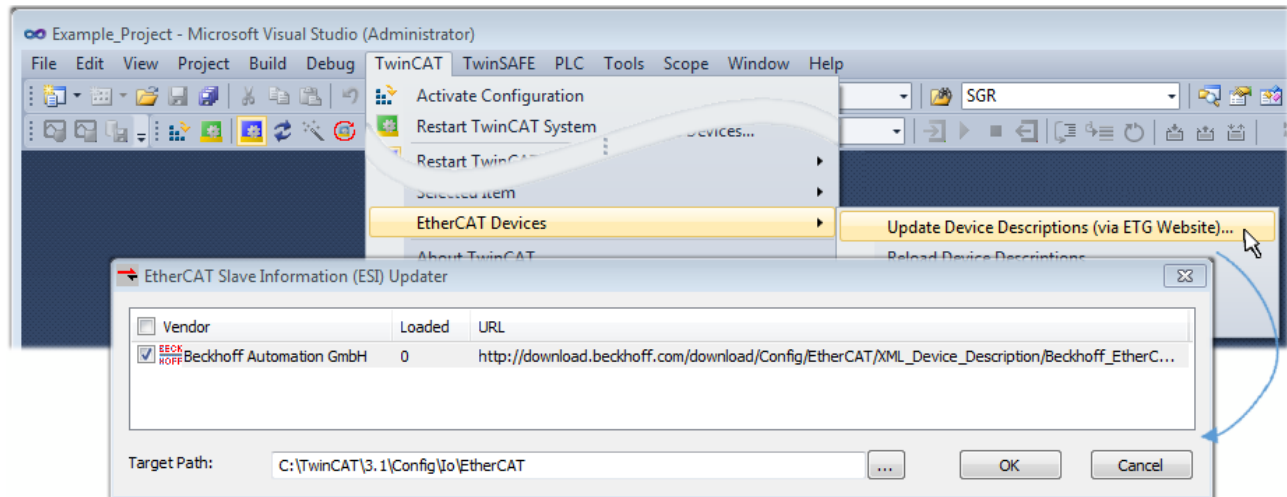


Fig. 31: Using the ESI Updater (TwinCAT 3)

The ESI Updater (TwinCAT 3) is a convenient option for automatic downloading of ESI data provided by EtherCAT manufacturers via the Internet into the TwinCAT directory (ESI = EtherCAT slave information). TwinCAT accesses the central ESI ULR directory list stored at ETG; the entries can then be viewed in the Updater dialog, although they cannot be changed there.

The call up takes place under:
 “TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)...”.

5.1.4 Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals, EJ-modules). If the configuration is to be prepared in advance of the system configuration as a programming system, e.g. on a laptop, this is only possible in “Offline configuration” mode. In this case all components have to be entered manually in the configuration, e.g. based on the electrical design.

If the designed control system is already connected to the EtherCAT system and all components are energised and the infrastructure is ready for operation, the TwinCAT configuration can simply be generated through “scanning” from the runtime system. This is referred to as online configuration.

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to [note “Installation of the latest ESI-XML device description” \[▶ 50\]](#).

For preparation of a configuration:

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed
- the devices/modules must be connected via EtherCAT cables or in the terminal/ module strand in the same way as they are intended to be used later

- the devices/modules be connected to the power supply and ready for communication
- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 60] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 61]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 64]

The scan with existing configuration [▶ 65] can also be carried out for comparison.

5.1.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

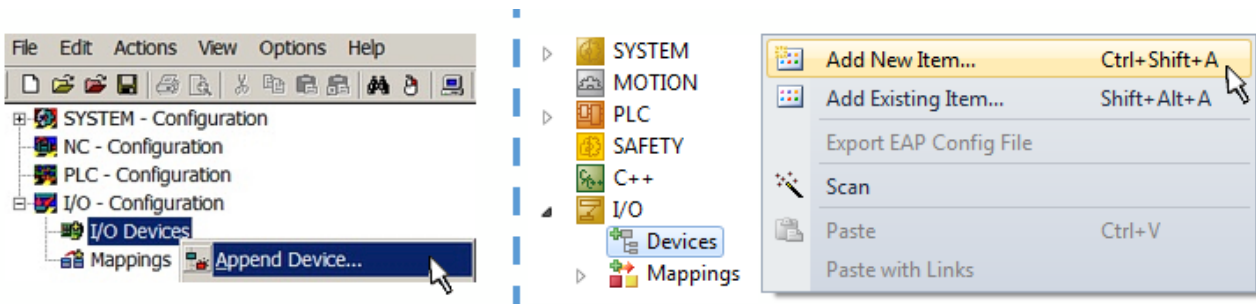


Fig. 32: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

Select type “EtherCAT” for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/ subscriber service in combination with an EL6601/EL6614 terminal select “EtherCAT Automation Protocol via EL6601”.

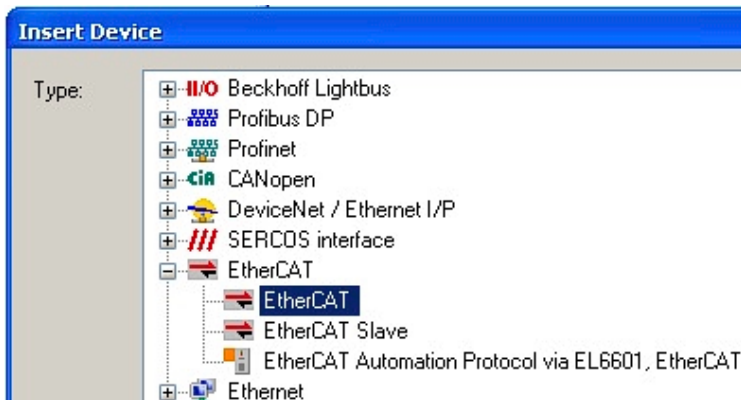


Fig. 33: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.

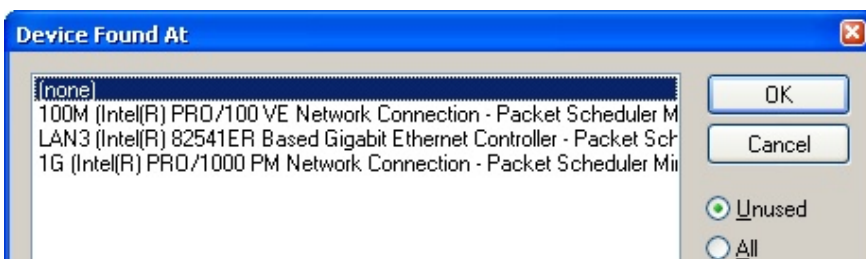


Fig. 34: Selecting the Ethernet port

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/modified later in the properties dialog; see Fig. “EtherCAT device properties (TwinCAT 2)”.

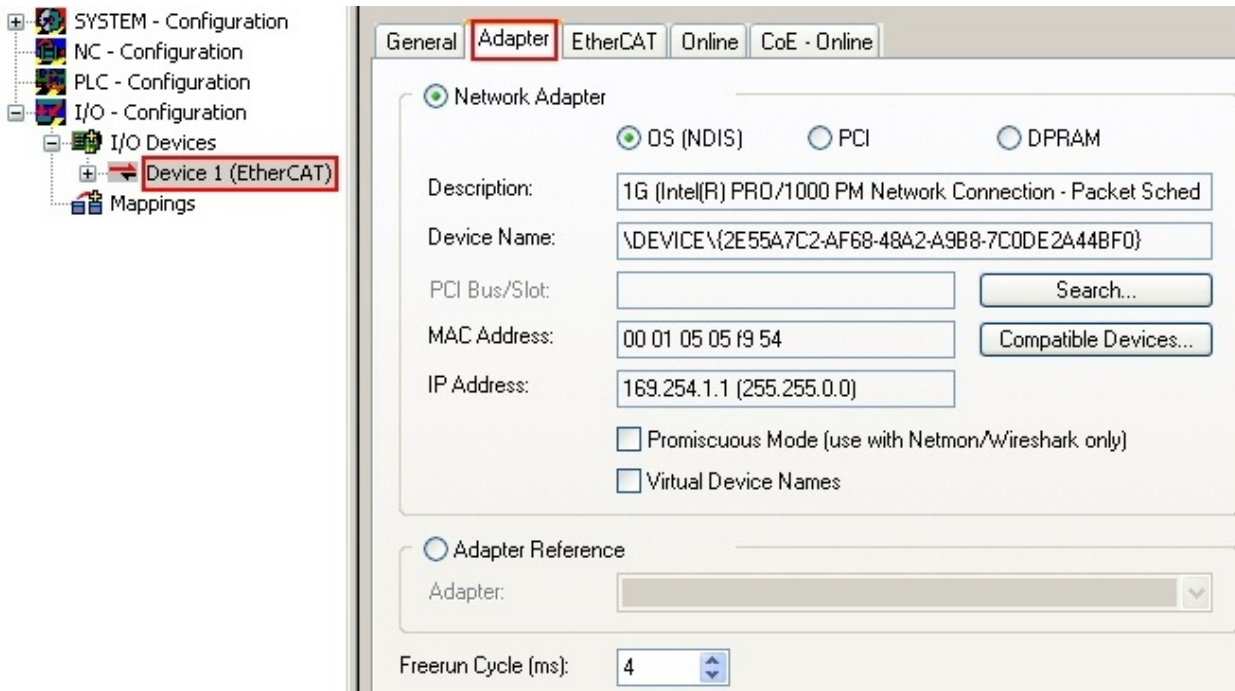
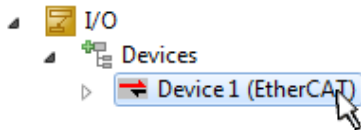


Fig. 35: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



i **Selecting the Ethernet port**

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page \[▶ 44\]](#).

Defining EtherCAT slaves

Further devices can be appended by right-clicking on a device in the configuration tree.

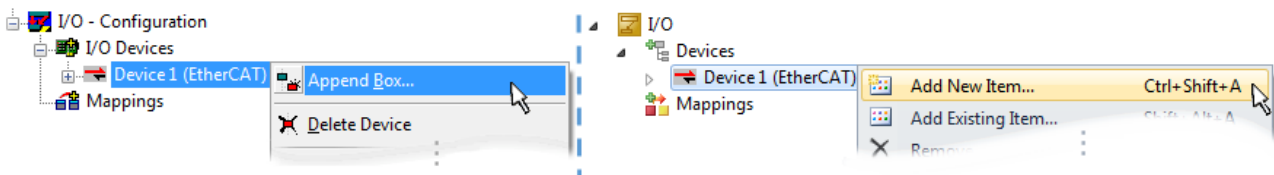


Fig. 36: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

The dialog for selecting a new device opens. Only devices for which ESI files are available are displayed.

Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then also only cable-based devices are available, as shown in Fig. “Selection dialog for new EtherCAT device”. If the preceding device has several free ports (e.g. EK1122 or EK1100), the required port can be selected on the right-hand side (A).

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).

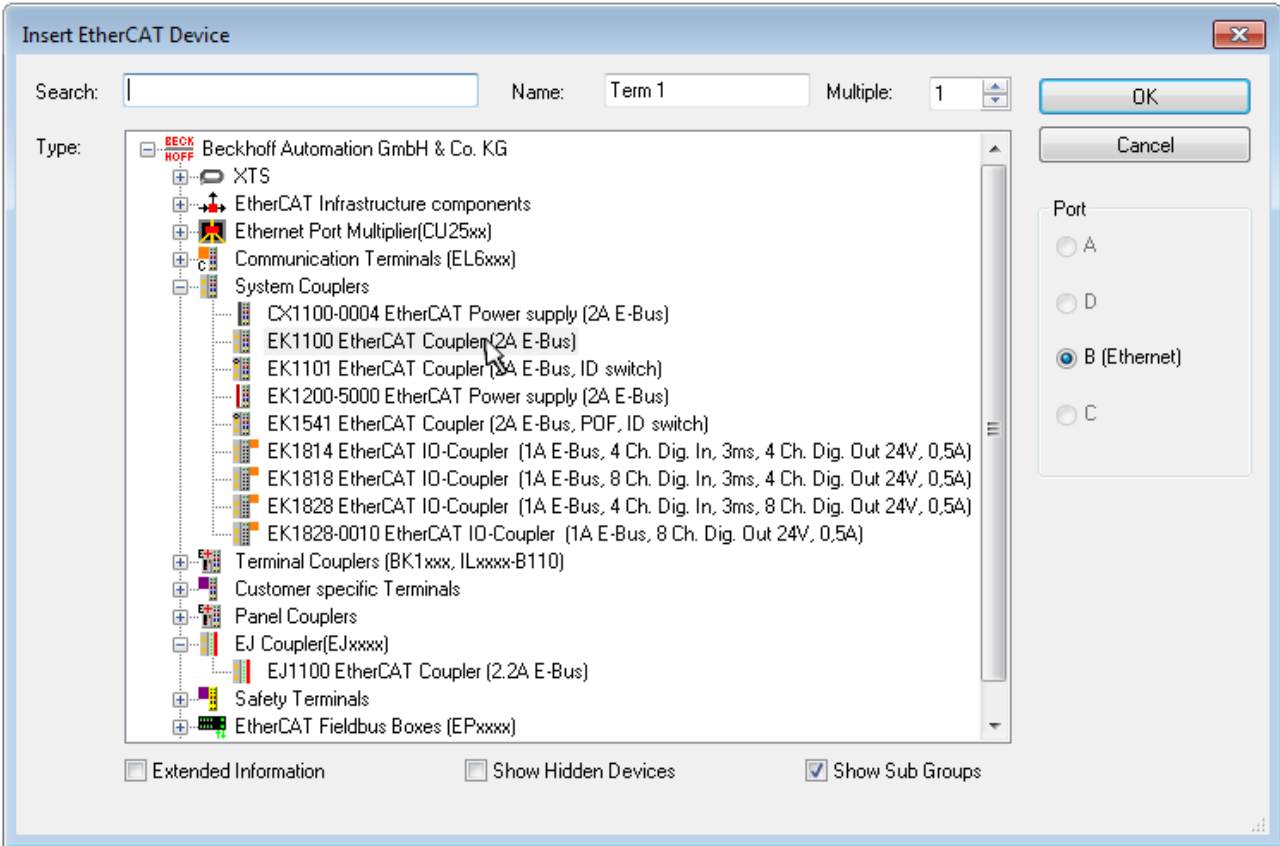


Fig. 37: Selection dialog for new EtherCAT device

By default, only the name/device type is used as selection criterion. For selecting a specific revision of the device, the revision can be displayed as “Extended Information”.

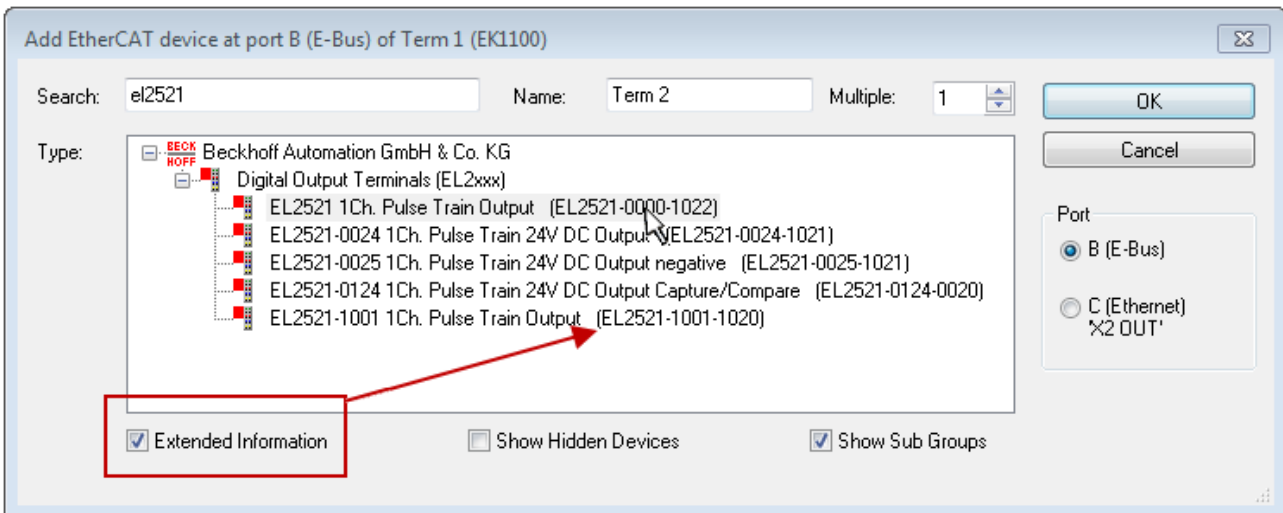


Fig. 38: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. “Selection dialog for new EtherCAT device”) only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the “Show Hidden Devices” check box, see Fig. “Display of previous revisions”.

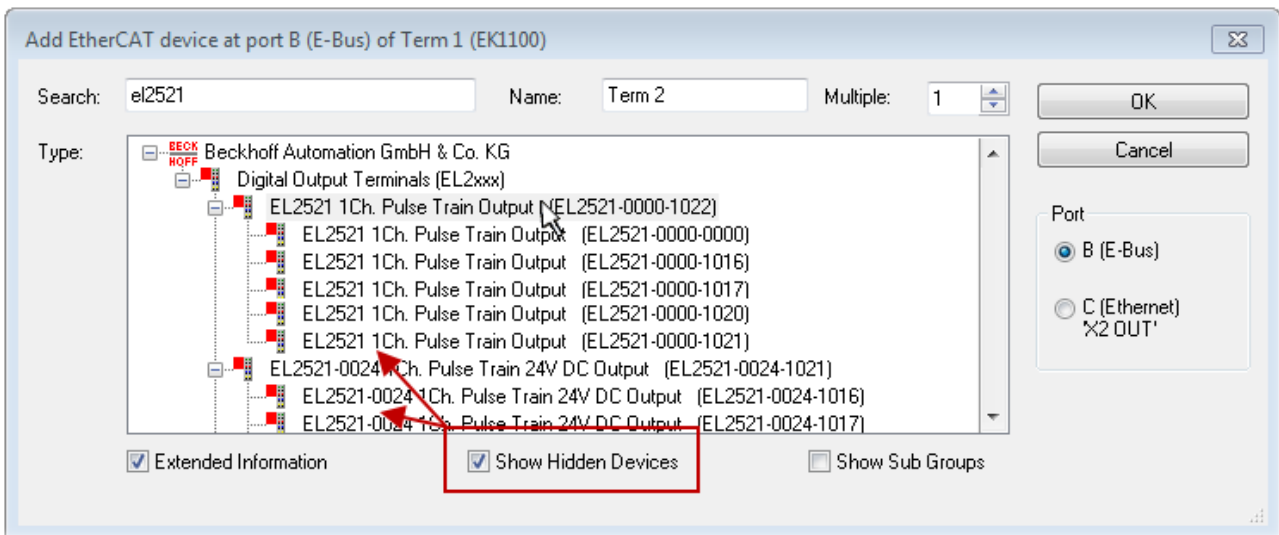


Fig. 39: Display of previous revisions

i Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-1018 is specified in the configuration, an EL2521-0025-1018 or higher (-1019, -1020) can be used in practice.

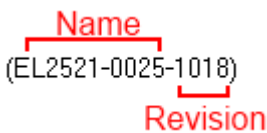


Fig. 40: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

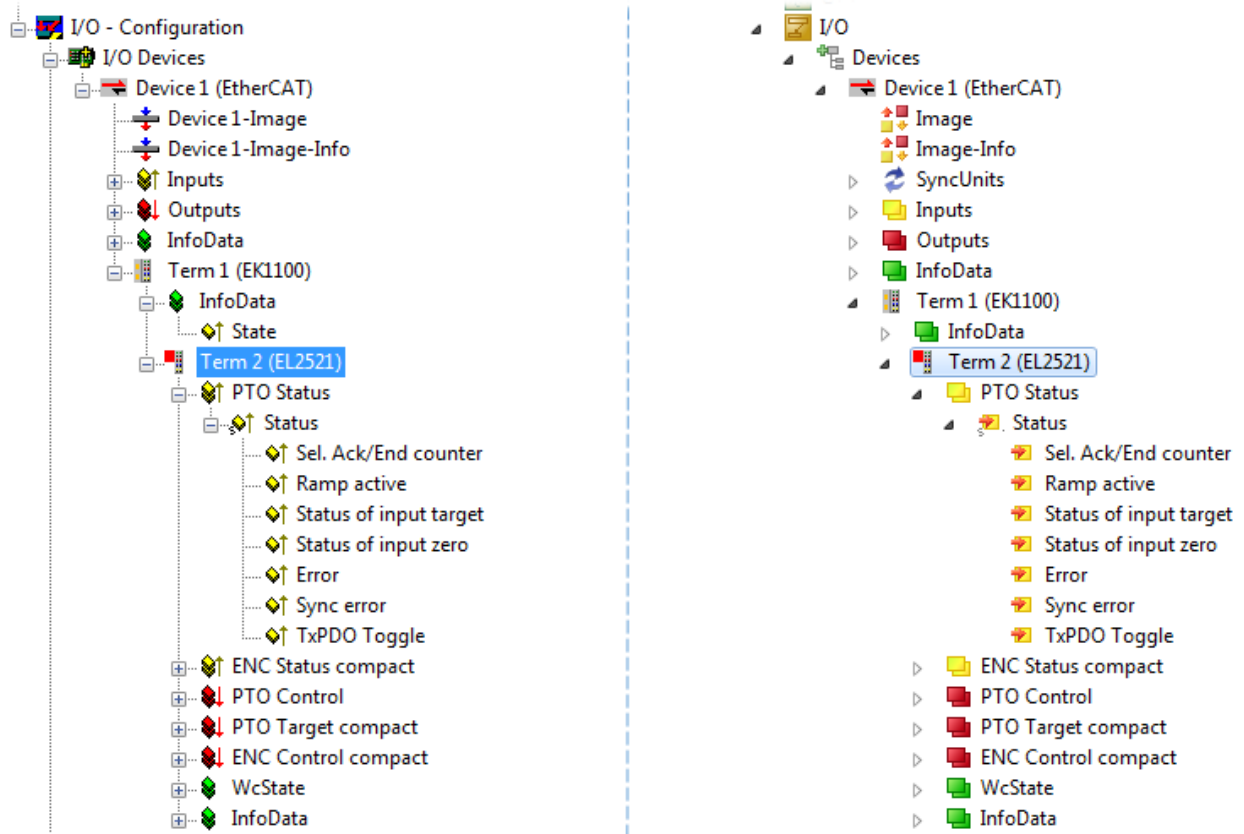




Fig. 41: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.1.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

The online device search can be used if the TwinCAT system is in CONFIG mode. This can be indicated by a symbol right below in the information bar:



- on TwinCAT 2 by a blue display “Config Mode” within the System Manager window:  .
- on TwinCAT 3 within the user interface of the development environment by a symbol  .

TwinCAT can be set into this mode:

- TwinCAT 2: by selection of  in the Menubar or by “Actions” → “Set/Reset TwinCAT to Config Mode...”
- TwinCAT 3: by selection of  in the Menubar or by “TwinCAT” → “Restart TwinCAT (Config Mode)”

● Online scanning in Config mode

i The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.

The TwinCAT 2 icon () or TwinCAT 3 icon () within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.

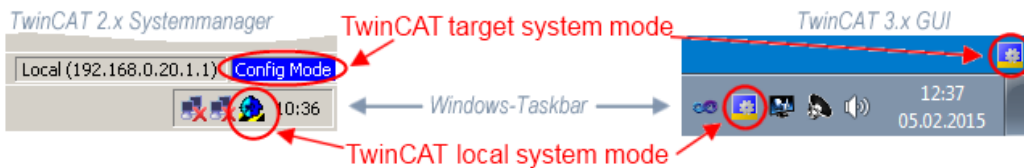


Fig. 42: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on “I/O Devices” in the configuration tree opens the search dialog.

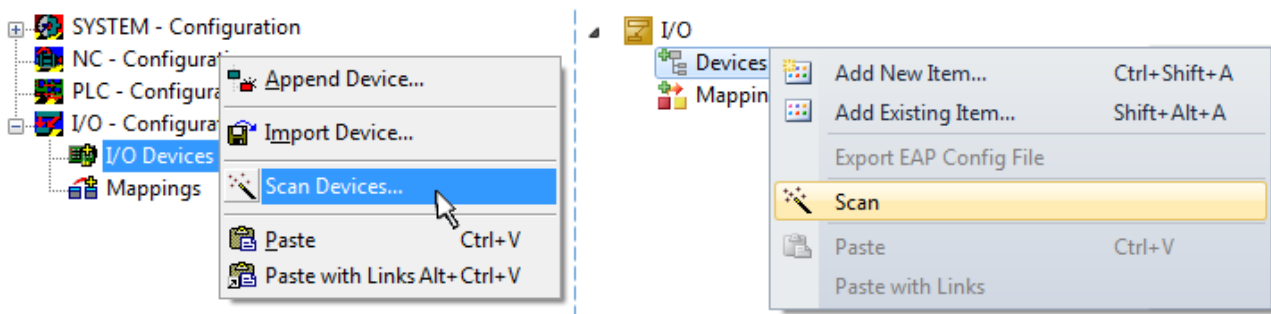


Fig. 43: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVDRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.

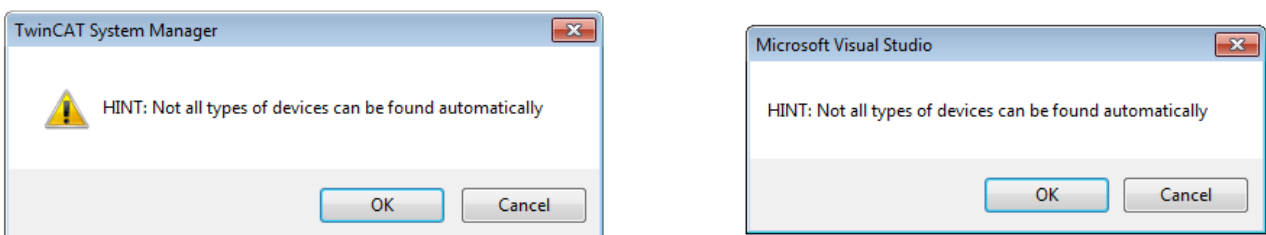


Fig. 44: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as “RT Ethernet” devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an “EtherCAT Device” .

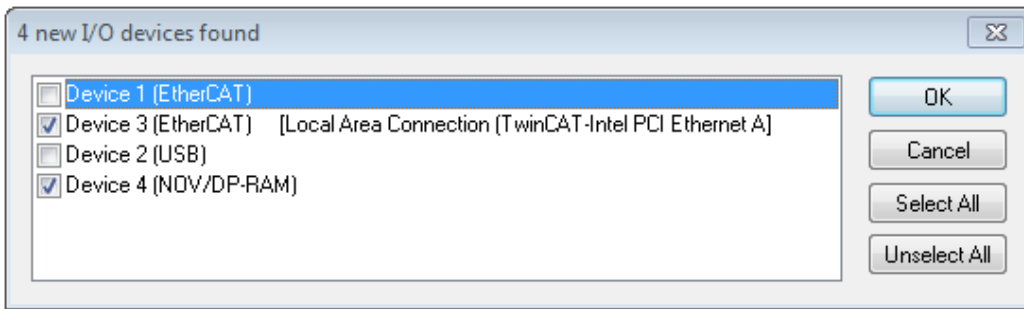


Fig. 45: Detected Ethernet devices

Via respective checkboxes devices can be selected (as illustrated in Fig. “Detected Ethernet devices” e.g. Device 3 and Device 4 were chosen). After confirmation with “OK” a device scan is suggested for all selected devices, see Fig.: “Scan query after automatic creation of an EtherCAT device”.

● Selecting the Ethernet port



Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page](#) [▶ 44].

Detecting/Scanning the EtherCAT devices

● Online scan functionality



During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.

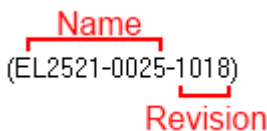


Fig. 46: Example default state

NOTE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for [comparison](#) [▶ 65] with the defined initial configuration. Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration “B.tsm” is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:

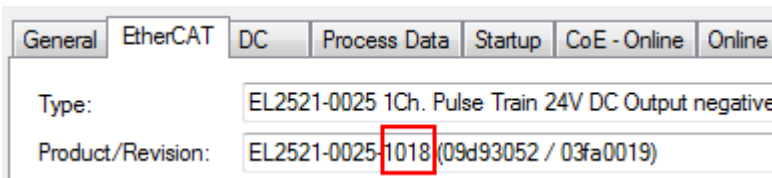


Fig. 47: Installing EthetCAT terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC “B.pro” or the NC. (the same applies correspondingly to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the series machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming series-produced machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [► 65] against the initial configuration “B.tsm” in order to check the built machine.

However, if the series machine production department now doesn't use “B.tsm”, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

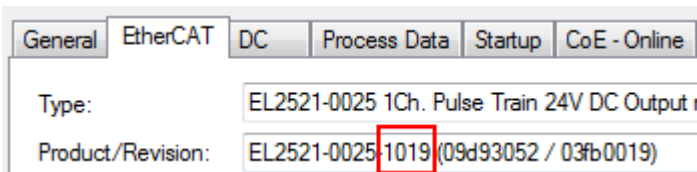


Fig. 48: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since virtually a new configuration is created. According to the compatibility rule, however, this means that no EL2521-0025-**1018** should be built into this machine as a spare part (even if this nevertheless works in the vast majority of cases).

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. If series machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.



Fig. 49: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

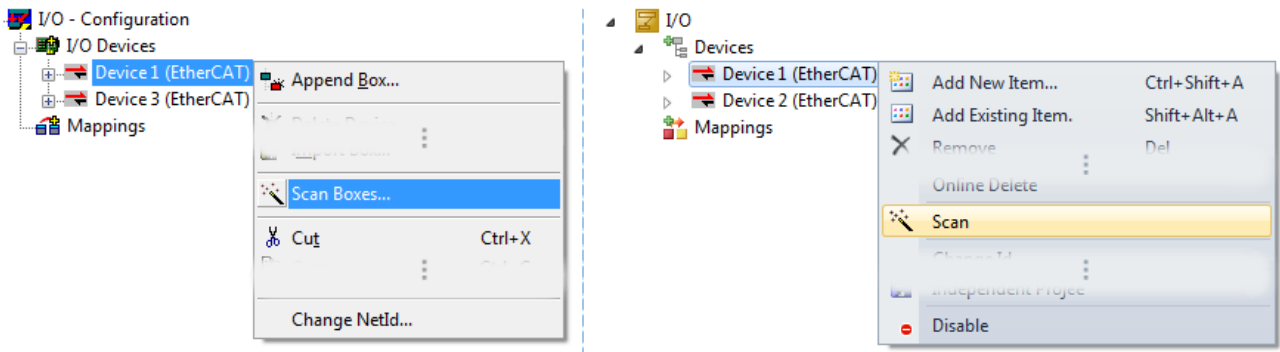


Fig. 50: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

In the System Manager (TwinCAT 2) or the User Interface (TwinCAT 3) the scan process can be monitored via the progress bar at the bottom in the status bar.



Fig. 51: Scan progress exemplified by TwinCAT 2

The configuration is established and can then be switched to online state (OPERATIONAL).



Fig. 52: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).



Fig. 53: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 54: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

The EtherCAT system should then be in a functional cyclic state, as shown in Fig. *Online display example*.

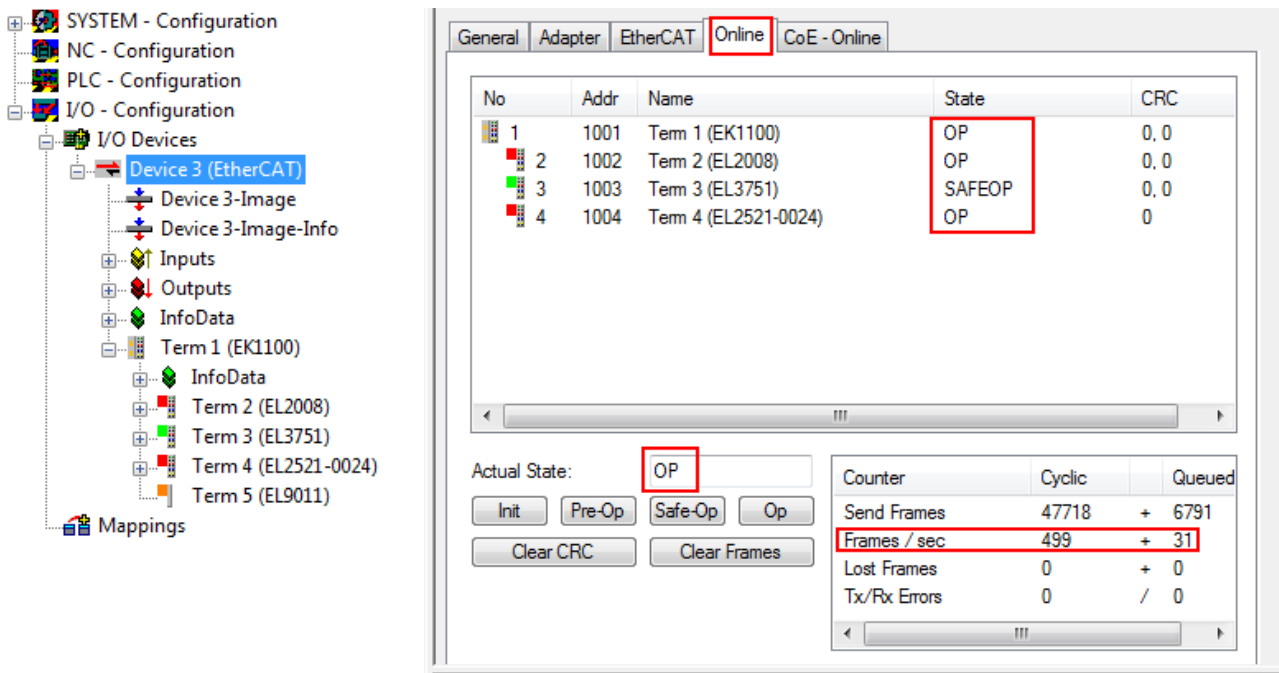


Fig. 55: Online display example

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in “Actual State” OP
- “frames/sec” should match the cycle time taking into account the sent number of frames
- no excessive “LostFrames” or CRC errors should occur

The configuration is now complete. It can be modified as described under [manual procedure \[► 55\]](#).

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter “Notes regarding ESI device description”.

- **Device are not detected properly**

Possible reasons include:

- faulty data links, resulting in data loss during the scan
- slave has invalid device description

The connections and devices should be checked in a targeted manner, e.g. via the emergency scan.

Then re-run the scan.

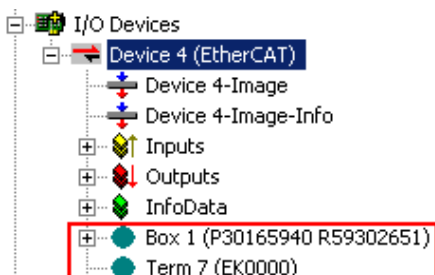


Fig. 56: Faulty identification

In the System Manager such devices may be set up as EK0000 or unknown devices. Operation is not possible or meaningful.

Scan over existing Configuration

NOTE

Change of the configuration after comparison

With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A “ChangeTo” or “Copy” should only be carried out with care, taking into consideration the Beckhoff IO compatibility rule (see above). The device configuration is then replaced by the revision found; this can affect the supported process data and functions.

If a scan is initiated for an existing configuration, the actual I/O environment may match the configuration exactly or it may differ. This enables the configuration to be compared.



Fig. 57: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

If differences are detected, they are shown in the correction dialog, so that the user can modify the configuration as required.

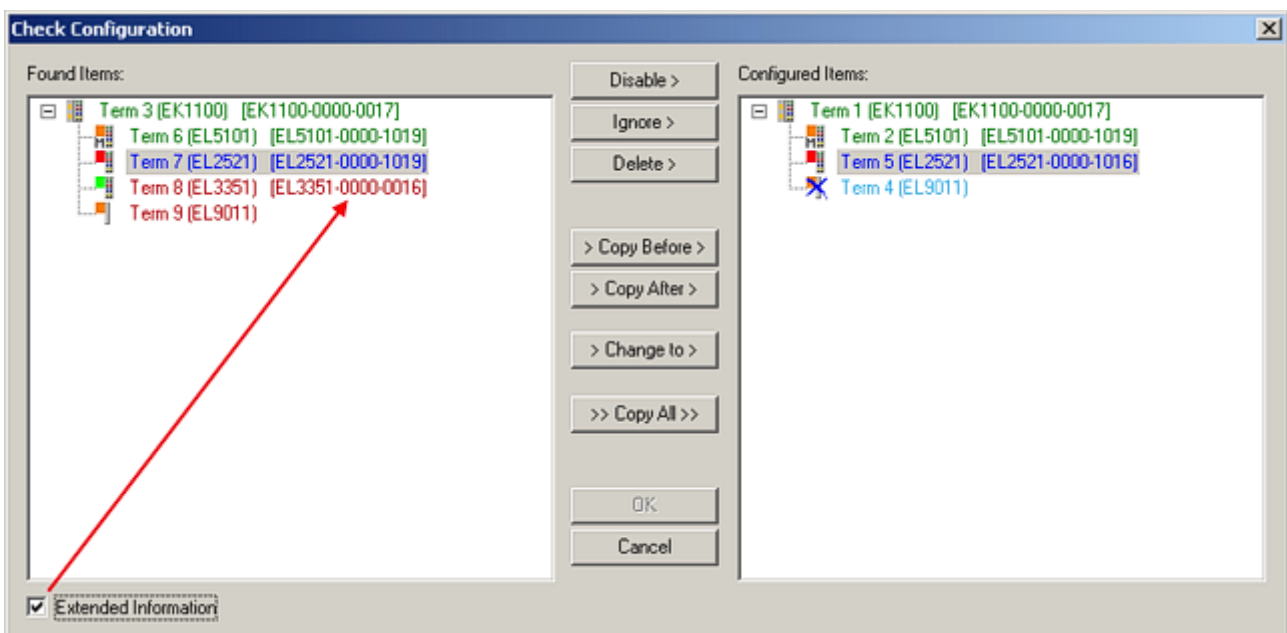


Fig. 58: Correction dialog

It is advisable to tick the “Extended Information” check box to reveal differences in the revision.

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	<ul style="list-style-type: none"> This EtherCAT slave is not present on the other side. It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

i Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system \geq device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (**-1019**, **-1020**) can be used in practice.

Name
(EL2521-0025-1018)
Revision

Fig. 59: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

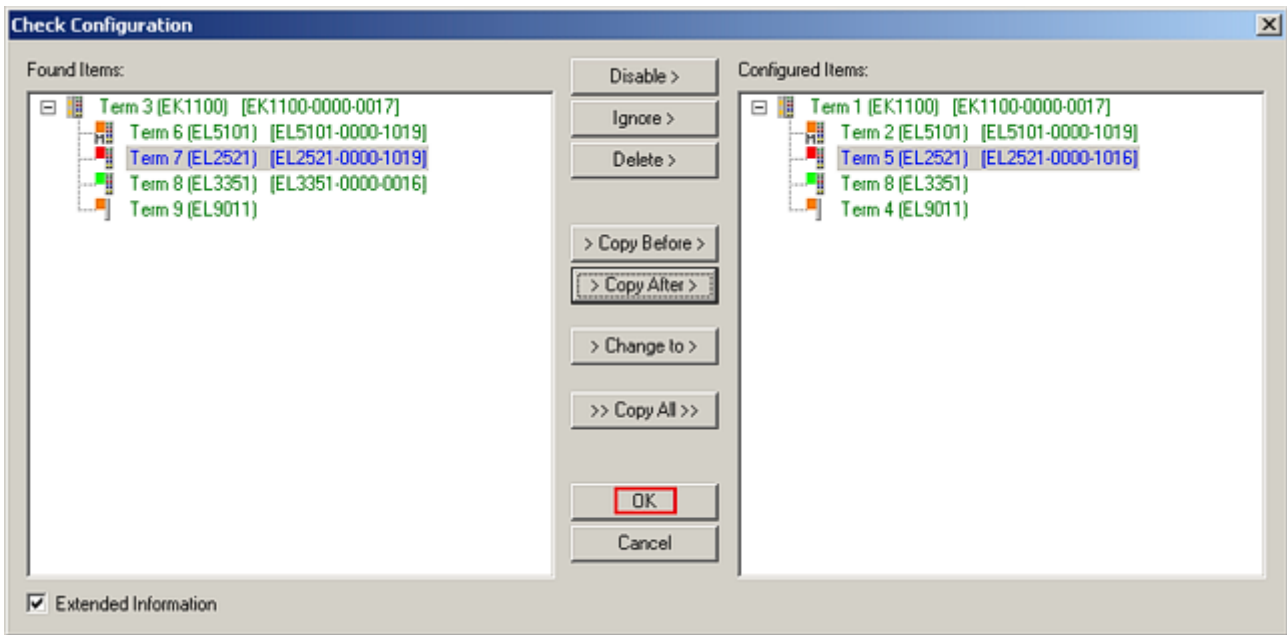


Fig. 60: Correction dialog with modifications

Once all modifications have been saved or accepted, click “OK” to transfer them to the real *.tsm configuration.

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type...* for the exchange of a device whilst retaining the links in the task.

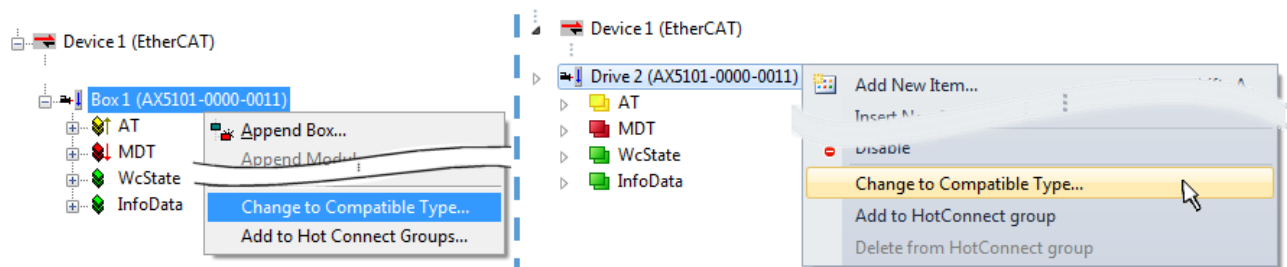


Fig. 61: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is indicated as "compatible":

- Physics (e.g. RJ45, Ebus...)
- FMMU (additional ones are allowed)
- SyncManager (SM, additional ones are allowed)
- EoE (attributes MAC, IP)
- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Ent-ry.Datatype)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type

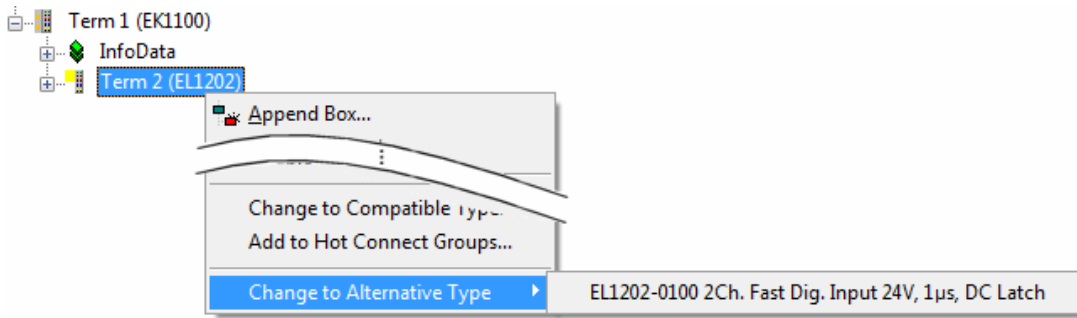


Fig. 62: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL1202-0000) for details of compatible devices contained there. The configuration is changed and the ESI-EEPROM is overwritten at the same time – therefore this process is possible only in the online state (ConfigMode).

5.1.7 EtherCAT subscriber configuration

In the left-hand window of the TwinCAT 2 System Manager or the Solution Explorer of the TwinCAT 3 Development Environment respectively, click on the element of the terminal within the tree you wish to configure (in the example: EL3751 Terminal 3).

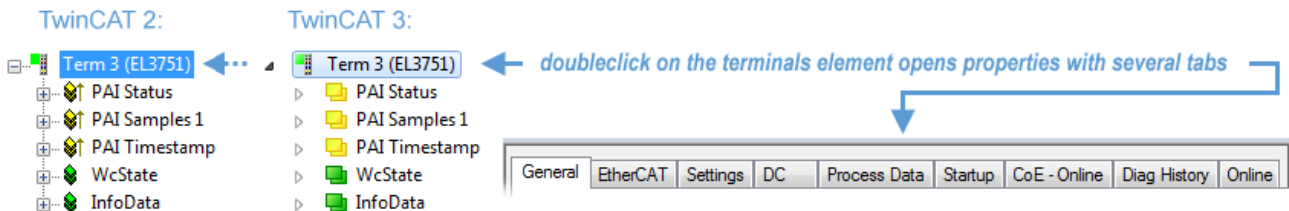


Fig. 63: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs “General”, “EtherCAT”, “Process Data” and “Online” are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so “EL6695” in this case. A specific tab “Settings” by terminals with a wide range of setup options will be provided also (e.g. EL3751).

“General” tab

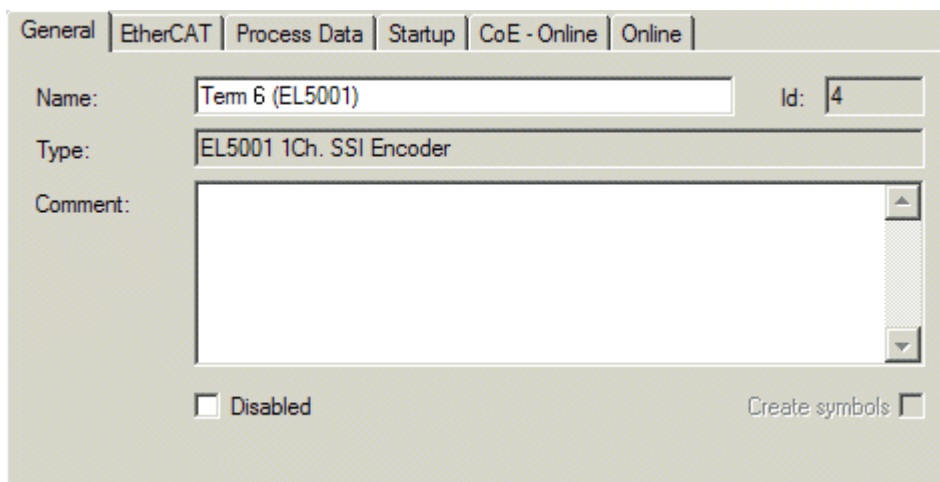


Fig. 64: “General” tab

Name	Name of the EtherCAT device
Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this control box is activated.

“EtherCAT” tab

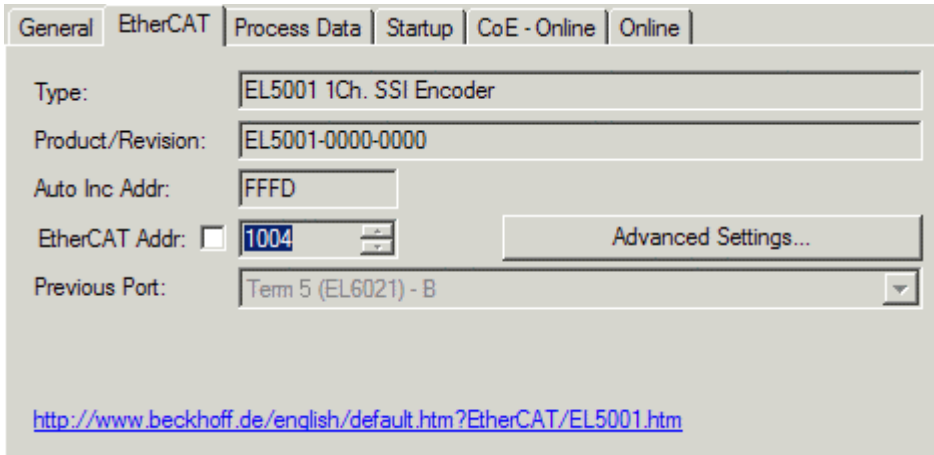


Fig. 65: “EtherCAT” tab

Type	EtherCAT device type
Product/Revision	Product and revision number of the EtherCAT device
Auto Inc Addr.	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000 _{hex} . For each further slave the address is decremented by 1 (FFFF _{hex} , FFFE _{hex} etc.).
EtherCAT Addr.	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.
Previous Port	Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.
Advanced Settings	This button opens the dialogs for advanced settings.

The link at the bottom of the tab points to the product page for this EtherCAT device on the web.

“Process Data” tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (**Process Data Objects, PDOs**). The user can select a PDO via PDO assignment and modify the content of the individual PDO via this dialog, if the EtherCAT slave supports this function.

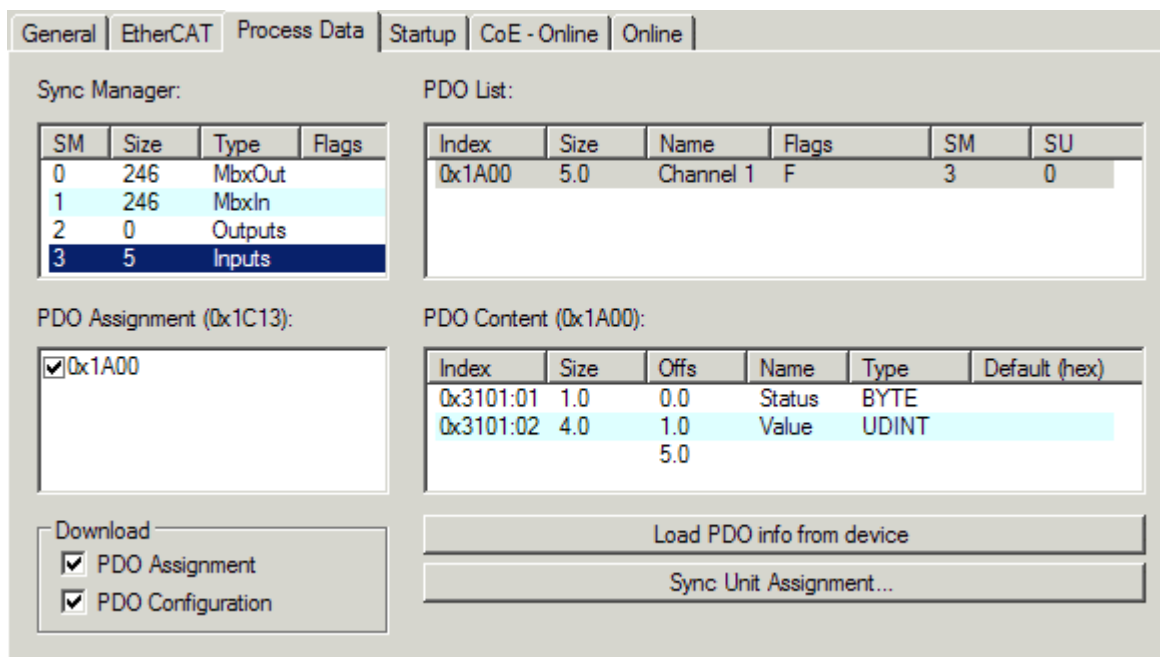


Fig. 66: "Process Data" tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation. Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called "intelligent" EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the "Process Data" tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record ("predefined PDO settings").

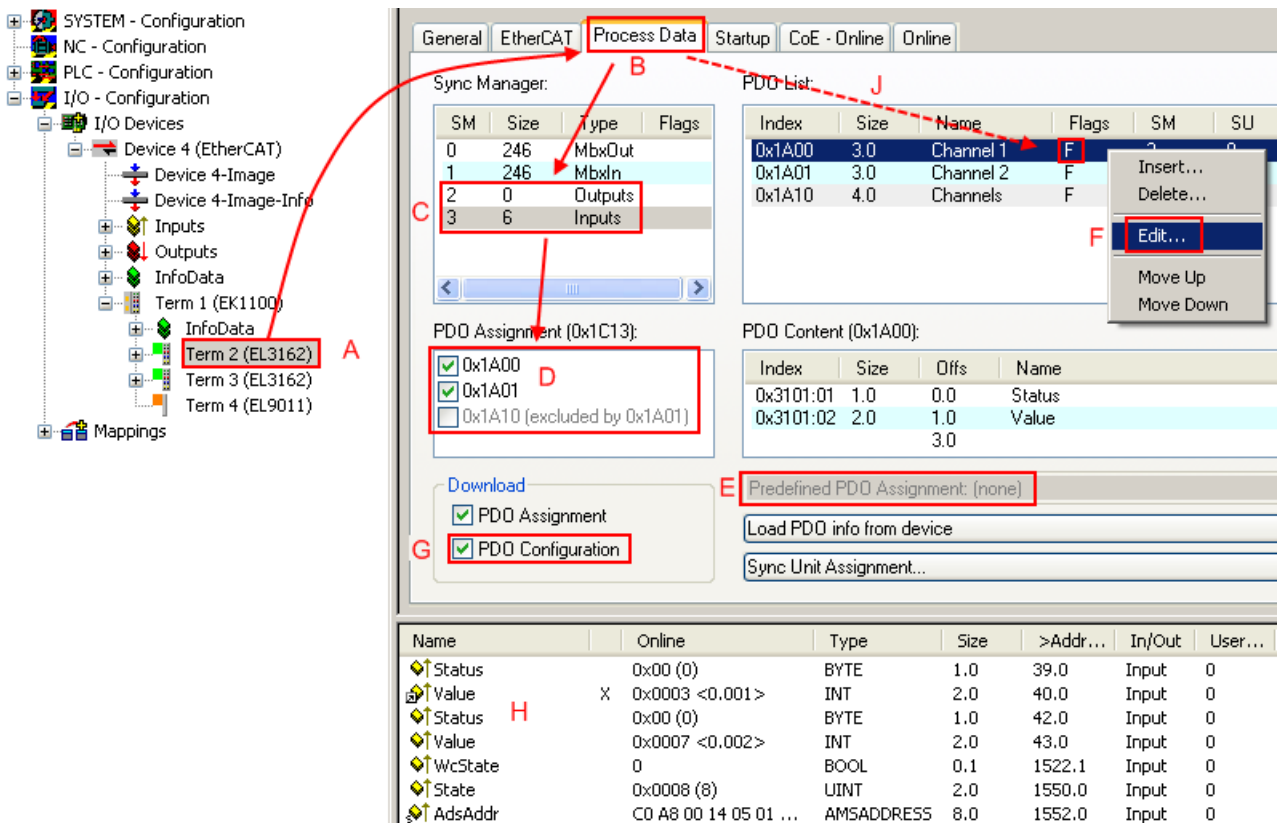


Fig. 67: Configuring the process data

i Manual modification of the process data

According to the ESI description, a PDO can be identified as “fixed” with the flag “F” in the PDO overview (Fig. *Configuring the process data*, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog (“Edit”). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, “G”. In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an “invalid SM cfg” logger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A detailed description [▶ 76] can be found at the end of this section.

“Startup” tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

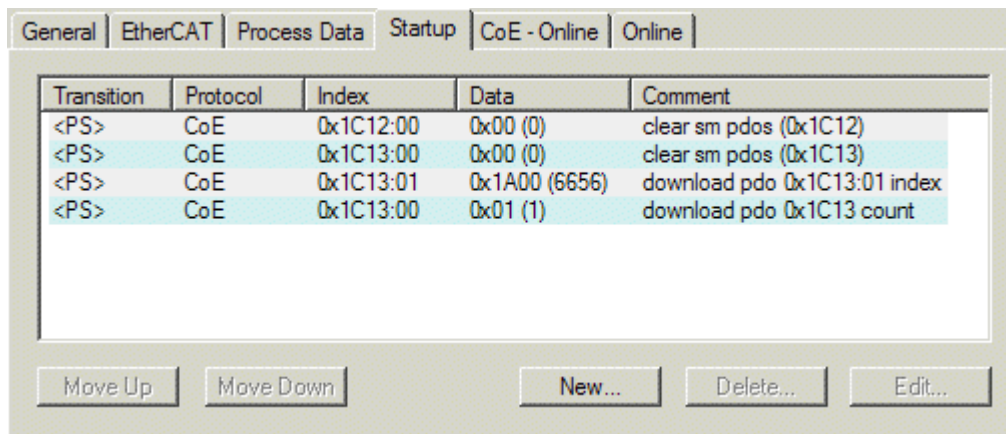


Fig. 68: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> the transition from pre-operational to safe-operational (PS), or the transition from safe-operational to operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

- Move Up** This button moves the selected request up by one position in the list.
- Move Down** This button moves the selected request down by one position in the list.
- New** This button adds a new mailbox download request to be sent during startup.
- Delete** This button deletes the selected entry.
- Edit** This button edits an existing request.

"CoE - Online" tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

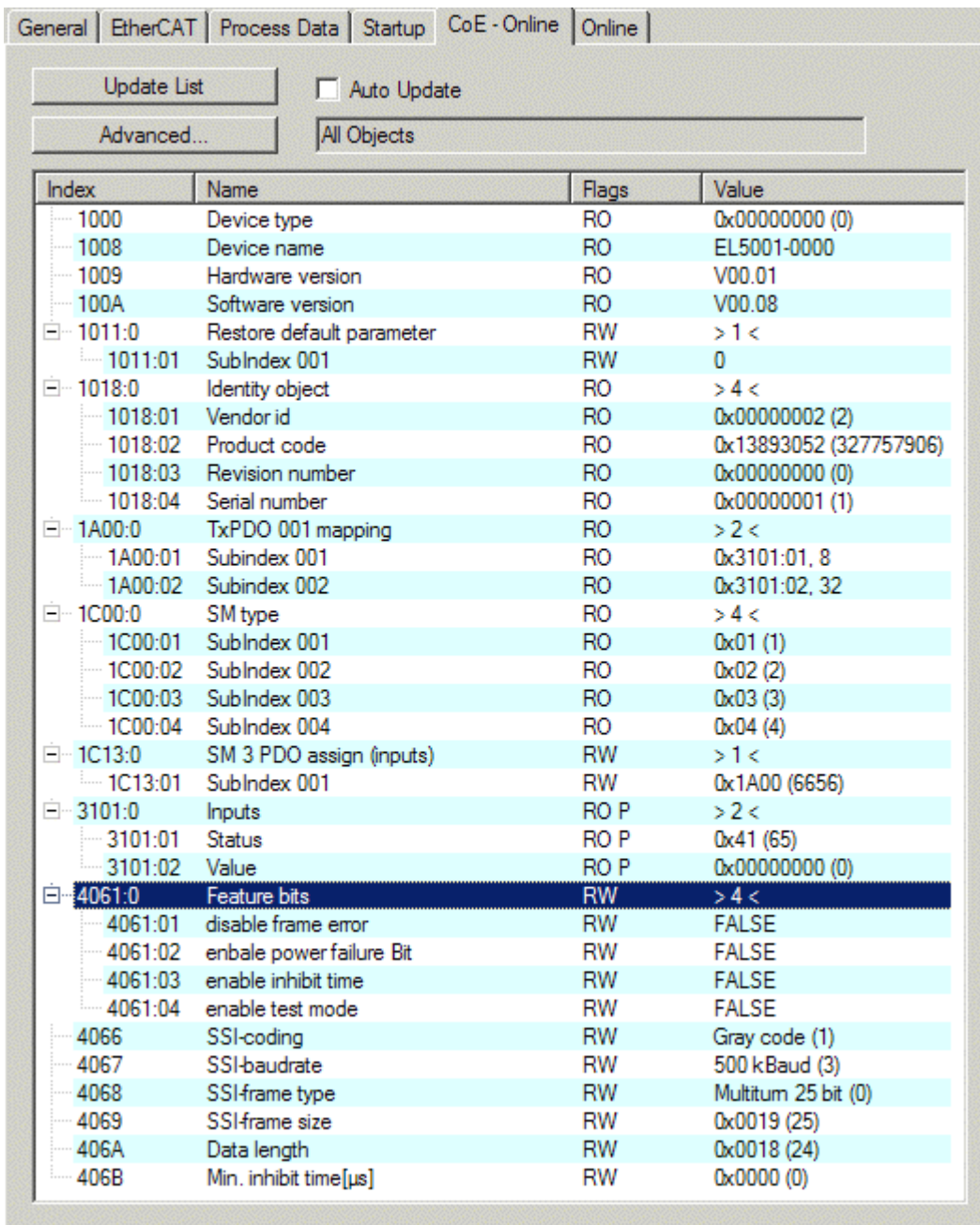


Fig. 69: "CoE - Online" tab

Object list display

Column	Description
Index	Index and sub-index of the object
Name	Name of the object
Flags	RW The object can be read, and data can be written to the object (read/write)
	RO The object can be read, but no data can be written to the object (read only)
	P An additional P identifies the object as a process data object.
Value	Value of the object

Update List The *Update list* button updates all objects in the displayed list

Auto Update If this check box is selected, the content of the objects is updated automatically.

Advanced The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

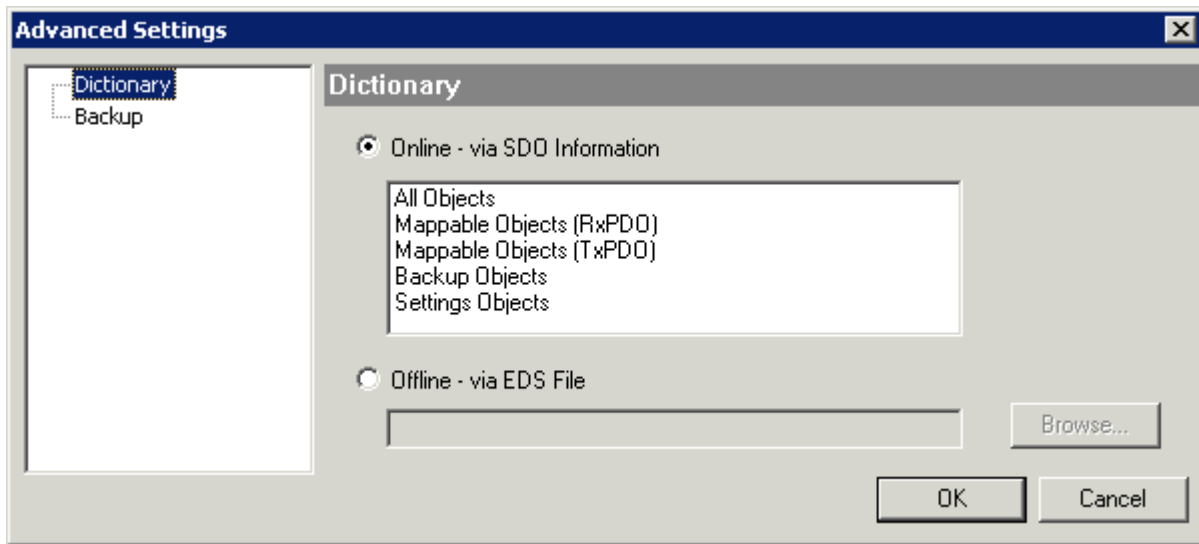


Fig. 70: Dialog “Advanced settings”

Online - via SDO Information If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.

Offline - via EDS File If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

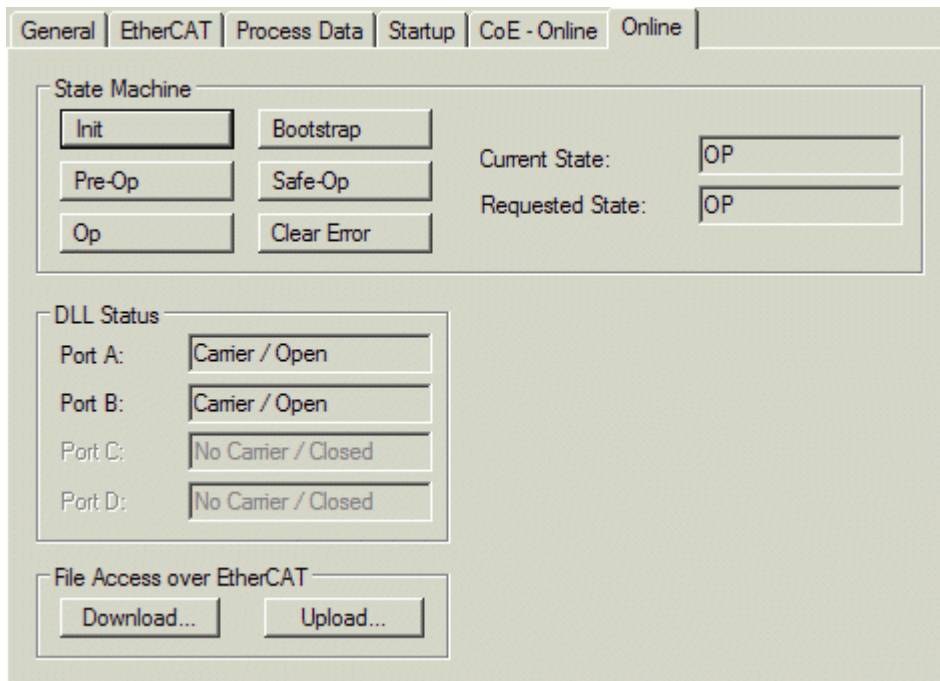


Fig. 71: “Online” tab

State Machine

- Init** This button attempts to set the EtherCAT device to the *Init* state.
- Pre-Op** This button attempts to set the EtherCAT device to the *pre-operational* state.
- Op** This button attempts to set the EtherCAT device to the *operational* state.
- Bootstrap** This button attempts to set the EtherCAT device to the *Bootstrap* state.
- Safe-Op** This button attempts to set the EtherCAT device to the *safe-operational* state.
- Clear Error** This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag.

Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
- Current State** Indicates the current state of the EtherCAT device.
- Requested State** Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

- Download** With this button a file can be written to the EtherCAT device.
- Upload** With this button a file can be read from the EtherCAT device.

“DC” tab (Distributed Clocks)

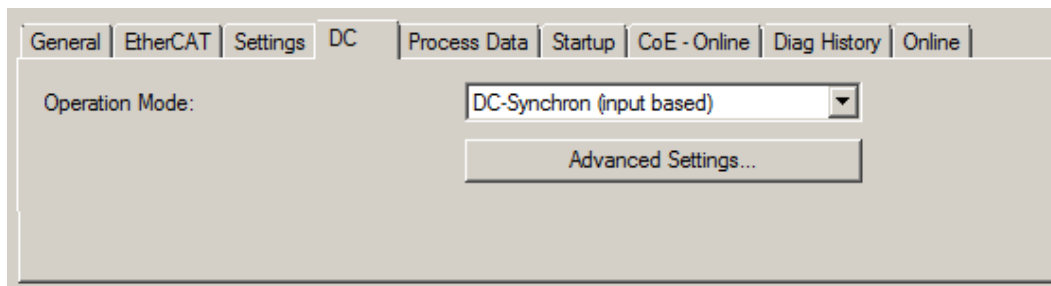


Fig. 72: “DC” tab (Distributed Clocks)

- Operation Mode** Options (optional):
 - FreeRun
 - SM-Synchron
 - DC-Synchron (Input based)
 - DC-Synchron
- Advanced Settings...** Advanced settings for readjustment of the real time determinant TwinCAT-clock

Detailed information to Distributed Clocks is specified on <http://infosys.beckhoff.com>:

Fieldbus Components → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks

5.1.7.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).

SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

If an input is selected, the corresponding PDO assignment is displayed in the *PDO Assignment* list below.

PDO Assignment



PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here:

- If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed.
- If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the *Name* parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is deactivated (not selected and greyed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a greyed out PDO, the currently selected PDO has to be deselected first.

i Activation of PDO assignment

- ✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,
 - a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see [Online tab \[▶ 74\]](#)),
 - b) and the System Manager has to reload the EtherCAT slaves

( button for TwinCAT 2 or  button for TwinCAT 3)

PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the <i>PDO Assignment</i> list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync unit to which this PDO is assigned.	

PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the [Startup \[► 71\]](#) tab.

PDO Configuration

If this check box is selected, the configuration of the respective PDOs (as shown in the PDO list and the PDO Content display) is downloaded to the EtherCAT slave.

5.1.8 Import/Export of EtherCAT devices with SCI and XTI

SCI and XTI Export/Import – Handling of user-defined modified EtherCAT slaves

5.1.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

- Cyclic process data (PDO)
- Synchronization (Distributed Clocks, FreeRun, SM-Synchron)
- CoE parameters (acyclic object dictionary)

Note: Not all three elements may be present, depending on the slave.

For a better understanding of the export/import function, let's consider the usual procedure for IO configuration:

- The user/programmer processes the IO configuration in the TwinCAT system environment. This involves all input/output devices such as drives that are connected to the fieldbuses used.
Note: In the following sections, only EtherCAT configurations in the TwinCAT system environment are considered.
- For example, the user manually adds devices to a configuration or performs a scan on the online system.
- This results in the IO system configuration.
- On insertion, the slave appears in the system configuration in the default configuration provided by the vendor, consisting of default PDO, default synchronization method and CoE StartUp parameter as defined in the ESI (XML device description).
- If necessary, elements of the slave configuration can be changed, e.g. the PDO configuration or the synchronization method, based on the respective device documentation.

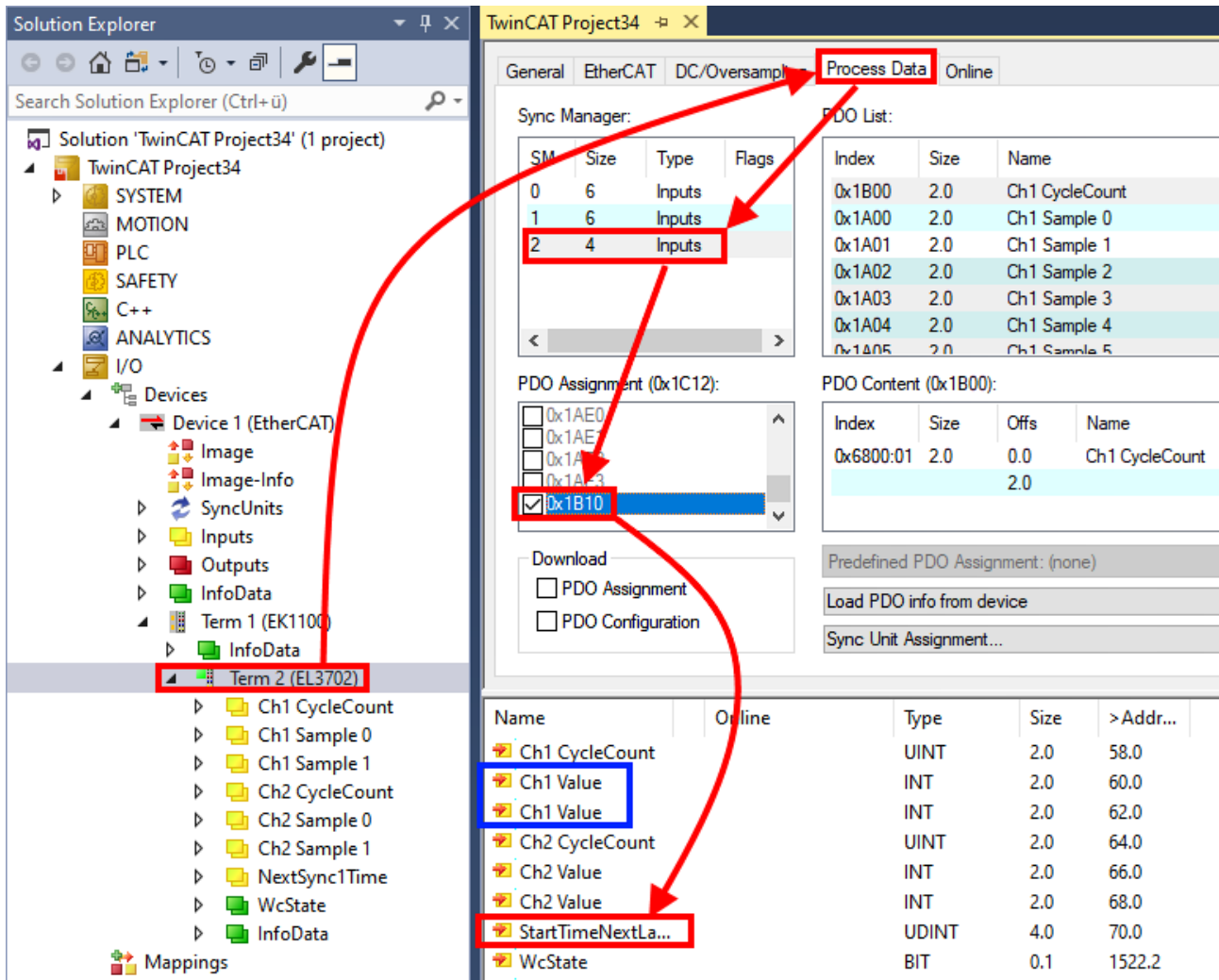
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- Export the slave configuration from the project,
- Store and transport as a file,
- Import into another EtherCAT project.

TwinCAT offers two methods for this purpose:

- within the TwinCAT environment: Export/Import as **x**ti file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **s**ci file.

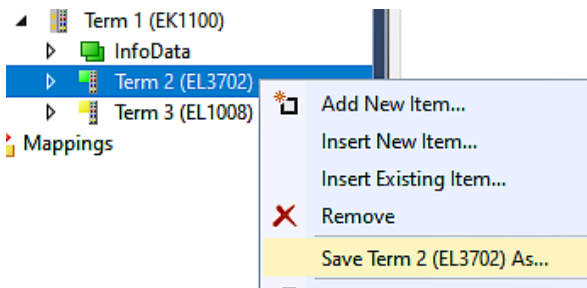
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



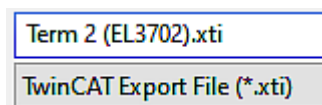
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.1.8.2 Procedure within TwinCAT with xti files

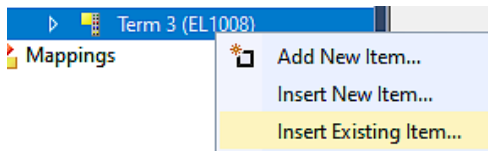
Each IO device can be exported/saved individually:



The xti file can be stored:



and imported again in another TwinCAT system via "Insert Existing item":



5.1.8.3 Procedure within and outside TwinCAT with sci file

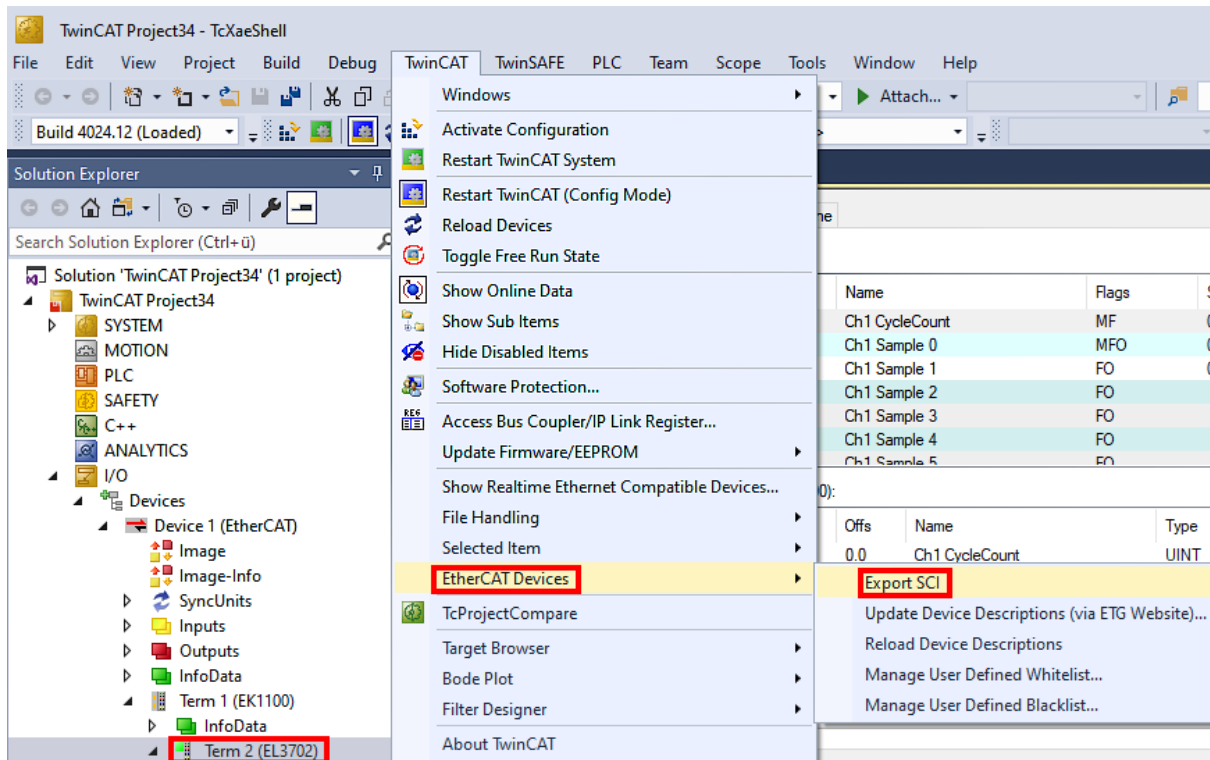
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

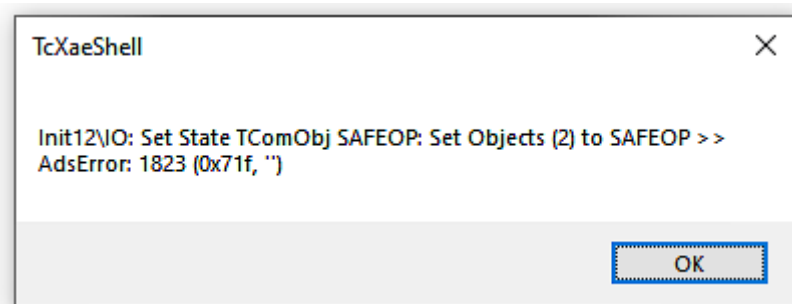
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

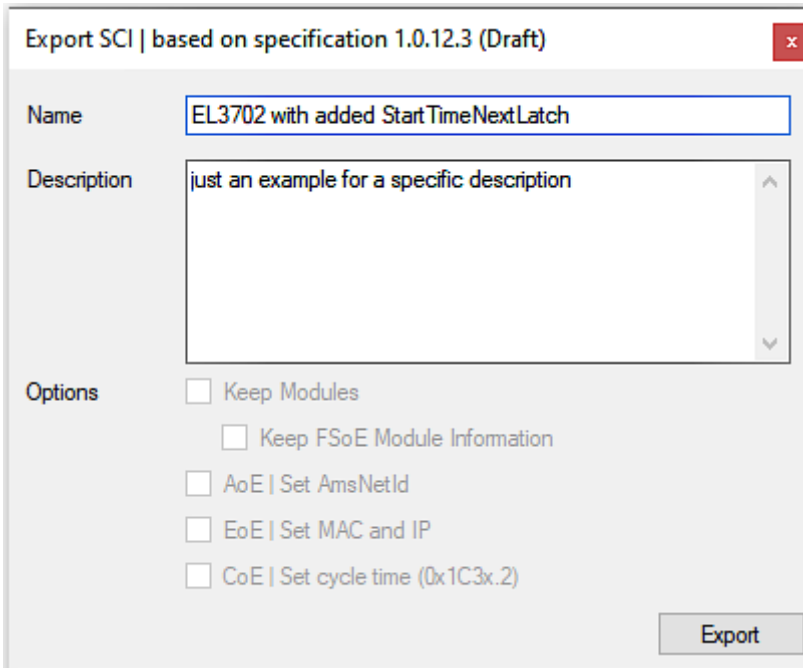
- select a single device via the menu (multiple selection is also possible):
TwinCAT → EtherCAT Devices → Export SCI.



- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:



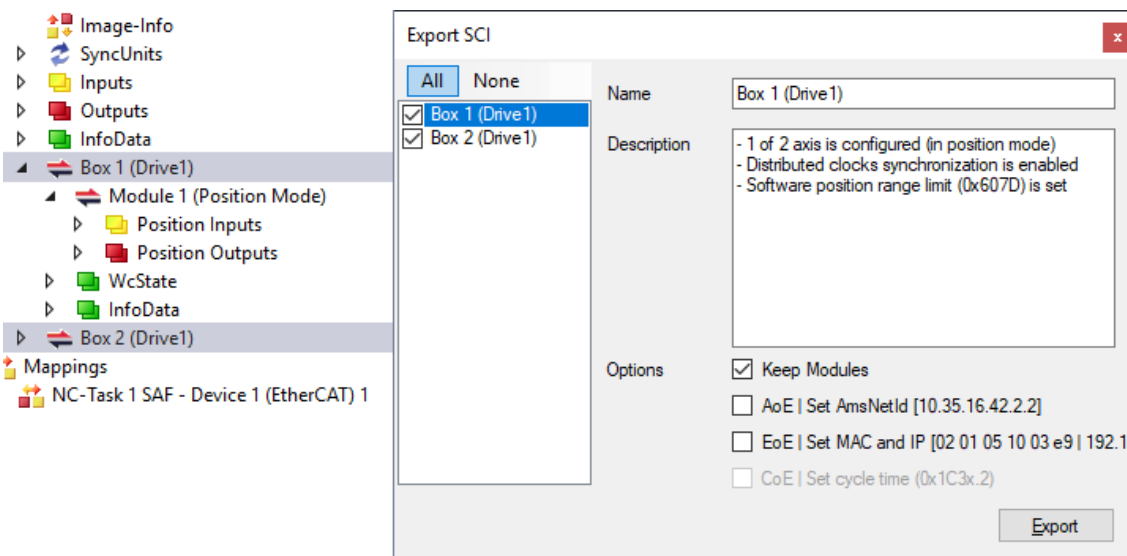
- A description may also be provided:



- Explanation of the dialog box:

Name	Name of the SCI, assigned by the user.	
Description	Description of the slave configuration for the use case, assigned by the user.	
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined during export.
	AoE Set AmsNetId	The configured AmsNetId is exported. Usually this is network-dependent and cannot always be determined in advance.
	EoE Set MAC and IP	The configured virtual MAC and IP addresses are stored in the SCI. Usually these are network-dependent and cannot always be determined in advance.
	CoE Set cycle time(0x1C3x.2)	The configured cycle time is exported. Usually this is network-dependent and cannot always be determined in advance.
ESI	Reference to the original ESI file.	
Export	Save SCI file.	

- A list view is available for multiple selections (*Export multiple SCI files*):

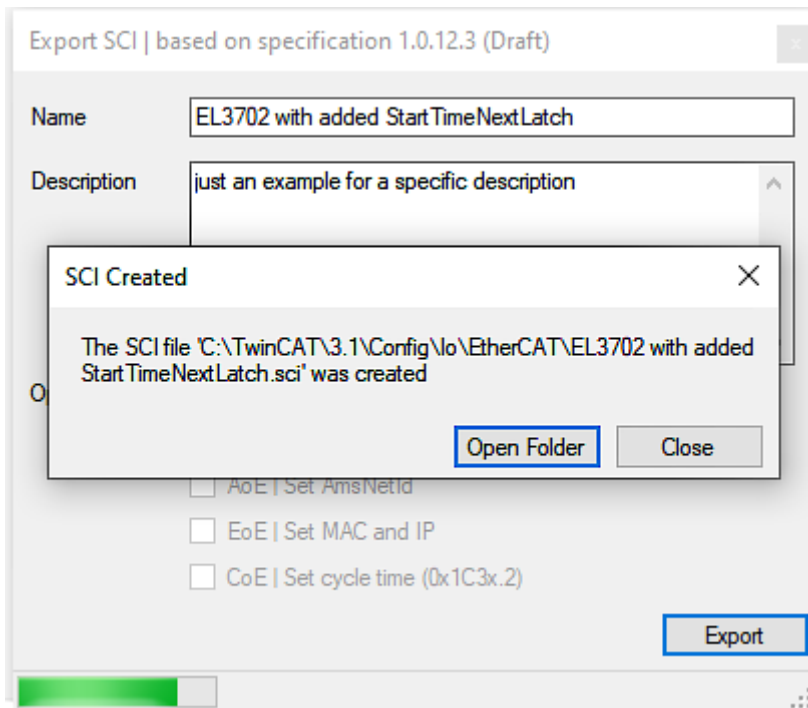


- Selection of the slaves to be exported:
 - All:
 - All slaves are selected for export.

- None:
All slaves are deselected.
- The sci file can be saved locally:

Dateiname:
 Dateityp:

- The export takes place:

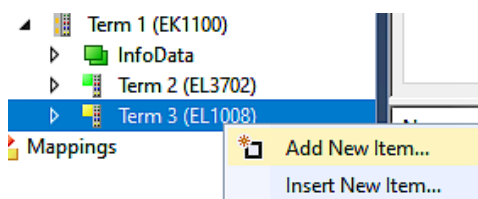


Import

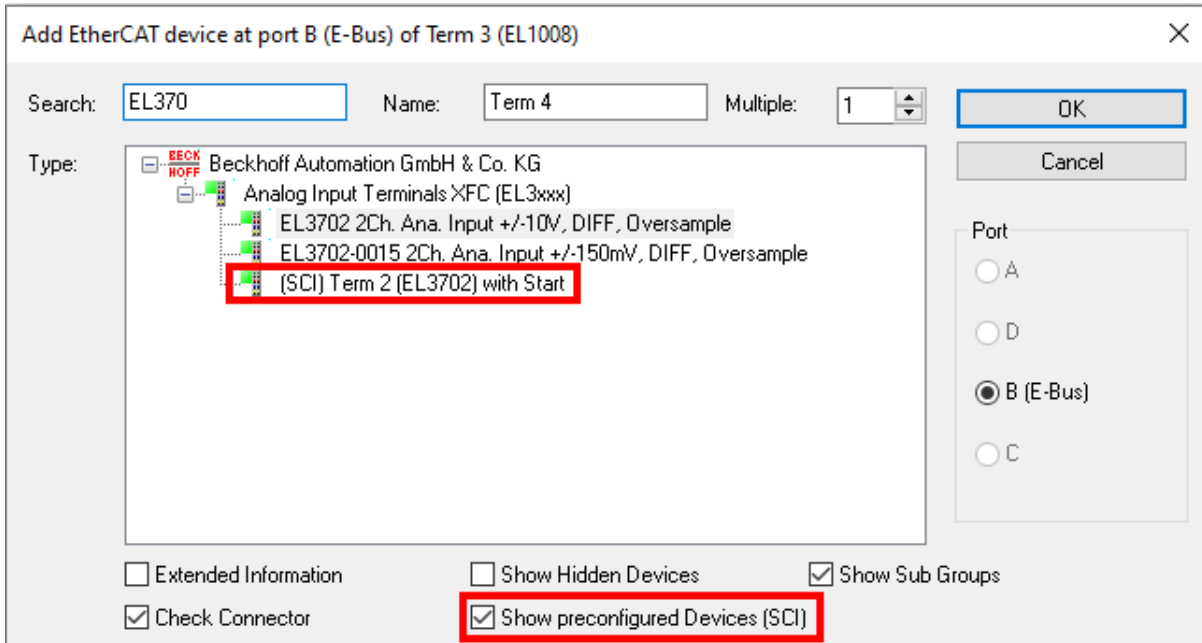
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
C:\TwinCAT\3.1\Config\Io\EtherCAT

	EL3702 with added StartTimeNextLatch.sci	11.01.2021 13:29	SCI-Datei	6 KB
--	--	------------------	-----------	------

- Open the selection dialog:

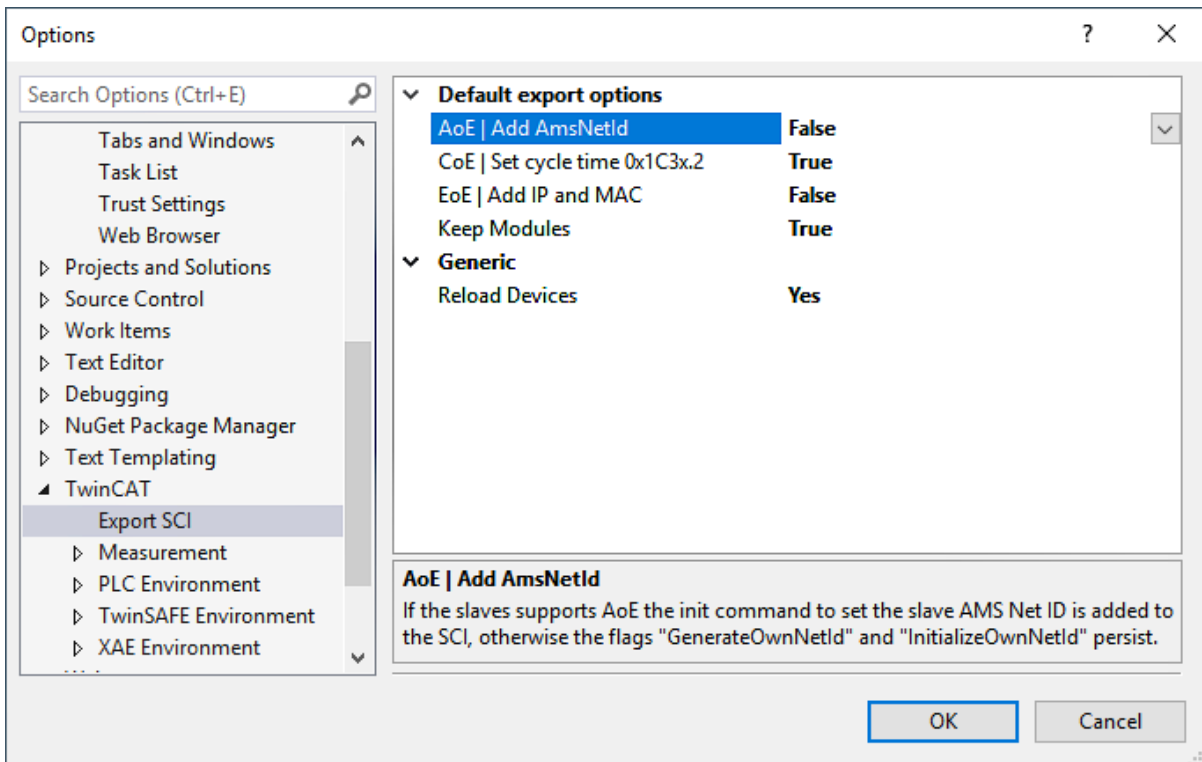


- Display SCI devices and select and insert the desired device:



Additional Notes

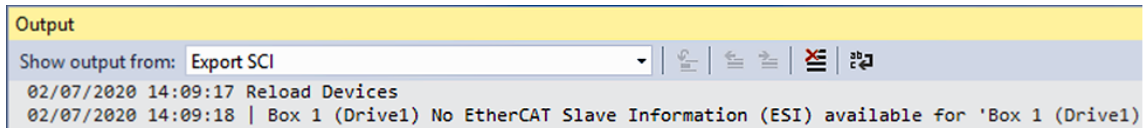
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

Default export options	AoE Set AmsNetId	Default setting whether the configured AmsNetId is exported.
	CoE Set cycle time(0x1C3x.2)	Default setting whether the configured cycle time is exported.
	EoE Set MAC and IP	Default setting whether the configured MAC and IP addresses are exported.
	Keep modules	Default setting whether the modules persist.
Generic	Reload Devices	Setting whether the Reload Devices command is executed before the SCI export. This is strongly recommended to ensure a consistent slave configuration.

SCI error messages are displayed in the TwinCAT logger output window if required:



5.2 General Notes - EtherCAT Slave Application

This summary briefly deals with a number of aspects of EtherCAT Slave operation under TwinCAT. More detailed information on this may be found in the corresponding sections of, for instance, the EtherCAT System Documentation.

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

Generally speaking an EtherCAT Slave provides a variety of diagnostic information that can be used by the controlling task.

This diagnostic information relates to differing levels of communication. It therefore has a variety of sources, and is also updated at various times.

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

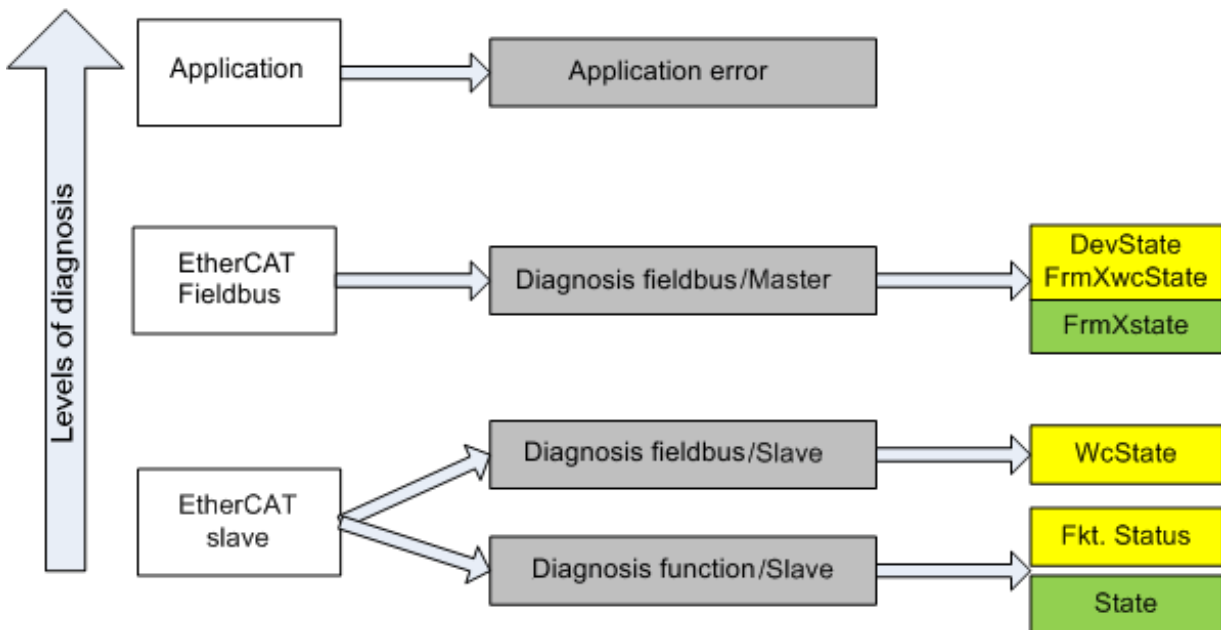


Fig. 73: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the exchange of process data, and correct operating mode)
This diagnosis is the same for all slaves.

as well as

- function diagnosis typical for a channel (device-dependent)
See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	Meaning
yellow	Input variables from the Slave to the EtherCAT Master, updated in every cycle
red	Output variables from the Slave to the EtherCAT Master, updated in every cycle
green	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. Basic EtherCAT Slave Diagnosis in the PLC shows an example of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

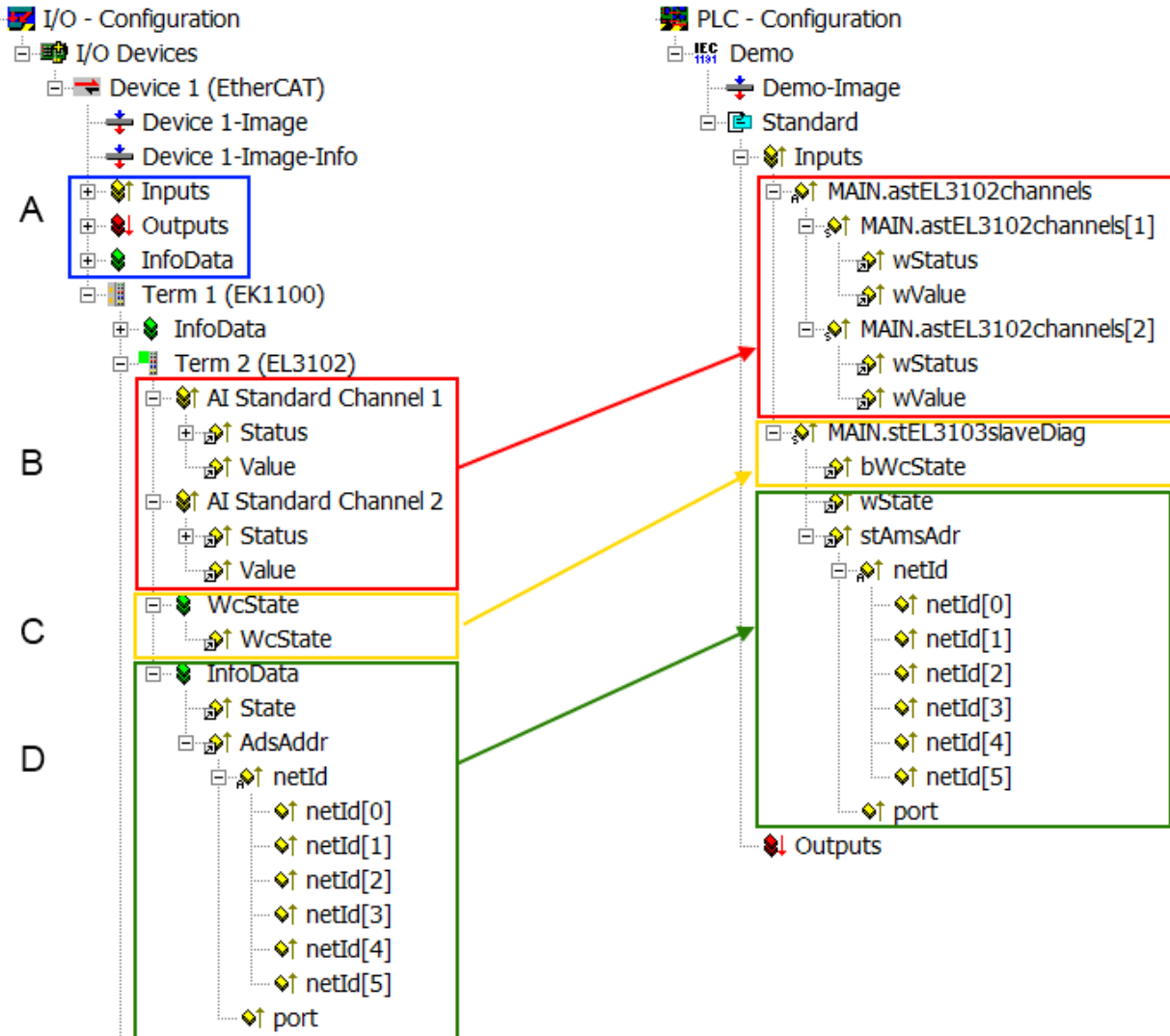


Fig. 74: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: <ul style="list-style-type: none"> • CoE in the Master for communication with/through the Slaves • Functions from <i>TcEtherCAT.lib</i> • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status <ul style="list-style-type: none"> • the bit significations may be found in the device documentation • other devices may supply more information, or none that is typical of a slave 	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager <ol style="list-style-type: none"> 1. at the EtherCAT Slave, and, with identical contents 2. as a collective variable at the EtherCAT Master (see Point A) for linking.	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it <ul style="list-style-type: none"> • is only rarely/never changed, except when the system starts up • is itself determined acyclically (e.g. EtherCAT Status) 	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. <i>AdsAddr</i> The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the <i>port</i> (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

NOTE

Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:

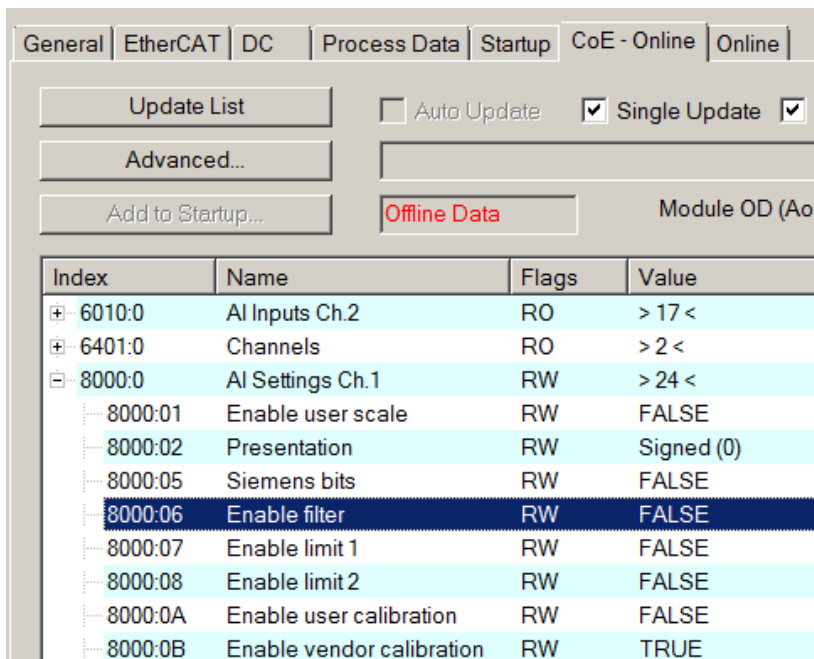


Fig. 75: EL3102, CoE directory

i EtherCAT System Documentation

The comprehensive description in the [EtherCAT System Documentation](#) (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.

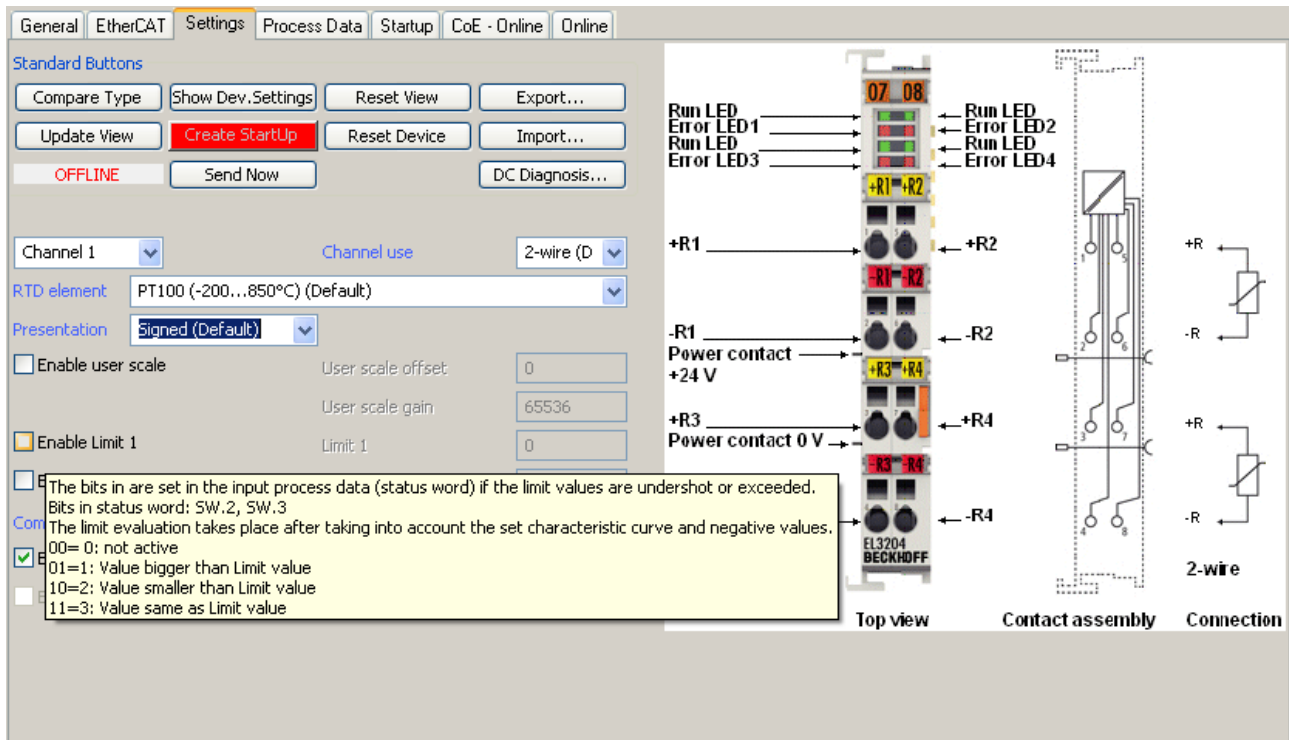


Fig. 76: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the “Process Data”, “DC”, “Startup” and “CoE-Online” that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of [Communication, EtherCAT State Machine \[▶ 21\]](#)" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

The target state wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach the target states.

Standard setting

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP
This setting applies equally to all Slaves.

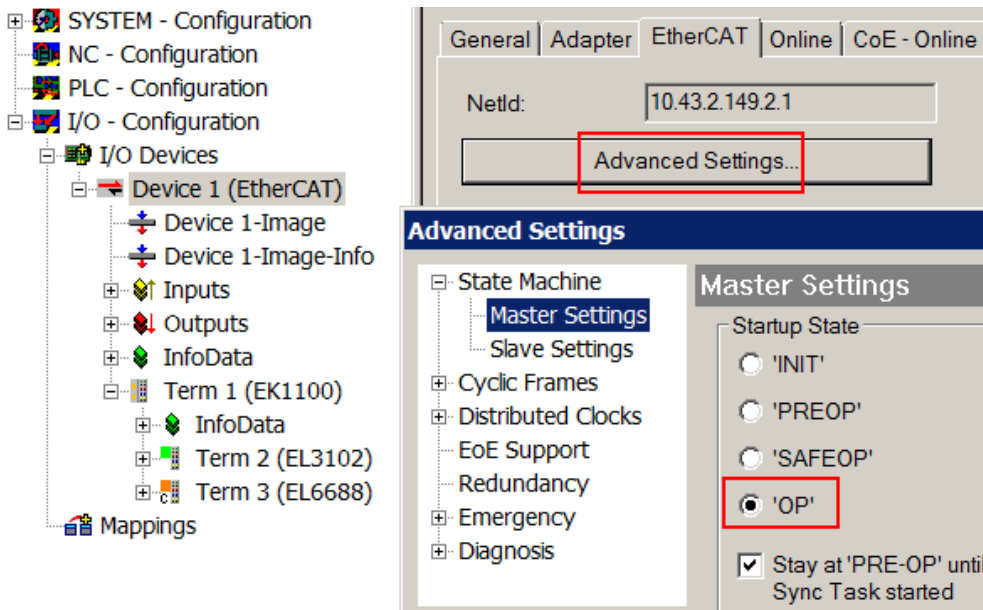


Fig. 77: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the “Advanced Settings” dialogue; the standard setting is again OP.

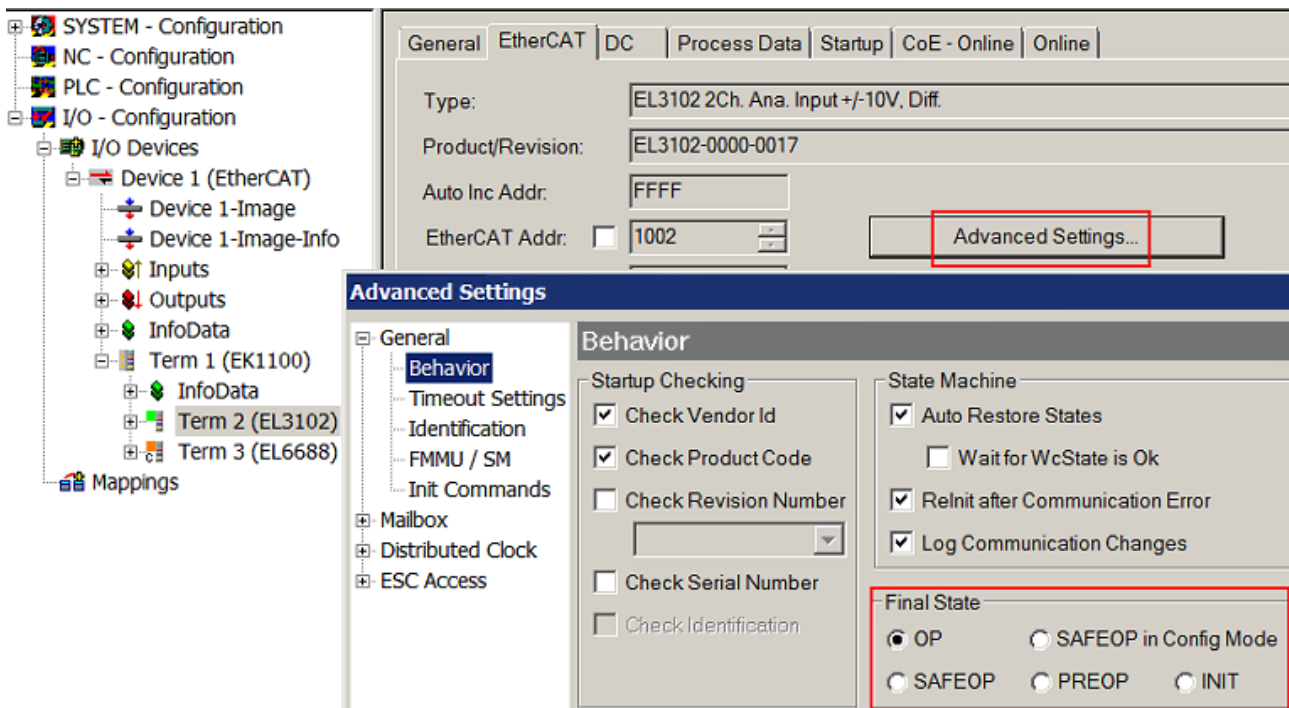


Fig. 78: Default target state in the Slave

Manual Control

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.

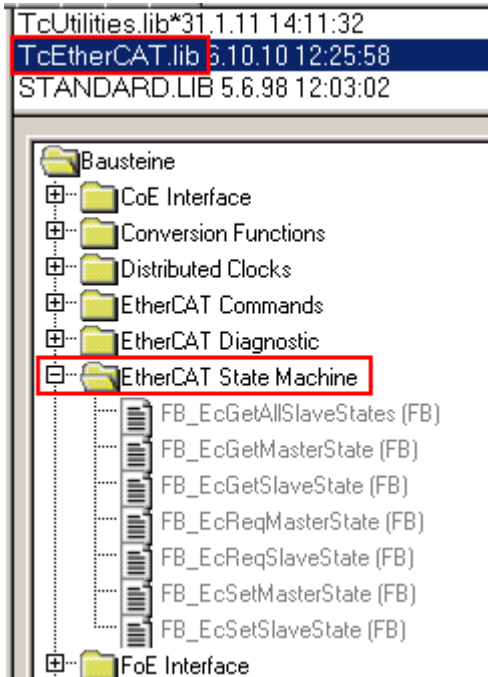


Fig. 79: PLC function blocks

Note regarding E-Bus current

EL/ES terminals are placed on the DIN rail at a coupler on the terminal strand. A Bus Coupler can supply the EL terminals added to it with the E-bus system voltage of 5 V; a coupler is thereby loadable up to 2 A as a rule. Information on how much current each EL terminal requires from the E-bus supply is available online and in the catalogue. If the added terminals require more current than the coupler can supply, then power feed terminals (e.g. EL9410) must be inserted at appropriate places in the terminal strand.

The pre-calculated theoretical maximum E-Bus current is displayed in the TwinCAT System Manager as a column value. A shortfall is marked by a negative total amount and an exclamation mark; a power feed terminal is to be placed before such a position.

General Adapter EtherCAT Online CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Fig. 80: Illegally exceeding the E-Bus current

From TwinCAT 2.11 and above, a warning message “E-Bus Power of Terminal...” is output in the logger window when such a configuration is activated:

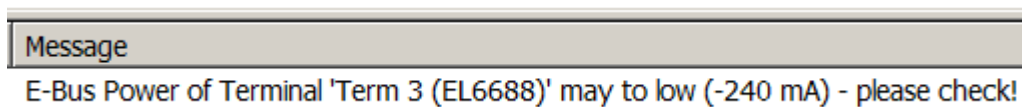


Fig. 81: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!
The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!

5.3 Notes to Commissioning

NOTE

Basic information about external TwinCAT synchronization

The EL6688 is an EtherCAT slave for time synchronization. Please regard the explanations in the EtherCAT system documentation from chapter "EtherCAT synchronization".

Quick start as slave clock

(Item 3 optional - StartUp list for the exchange case)

1. Generate configuration in TwinCAT: Right-click on EtherCAT devices and "Scan boxes" or manually create configuration

2. Make CoE settings for the EL6688 (in PreOP – see note)

- Set PTP common (0xF880:01) to "IEEE1588-2008 (PTPv2) – Slave Only Clock" (corresponds to 32_{dec})
- Set the PTPv2 settings (0xF882) as in the PTP Master (grandmaster) (see Mutual Settings [► 95])
- Ethernet Settings (0xF8E0): enter valid IP address and subnet mask
- Subsequently, changes must be permanently saved by writing "0x65766173" into index 0x1010 (see note)

3. Add CoE settings to the StartUp list (for the exchange case)

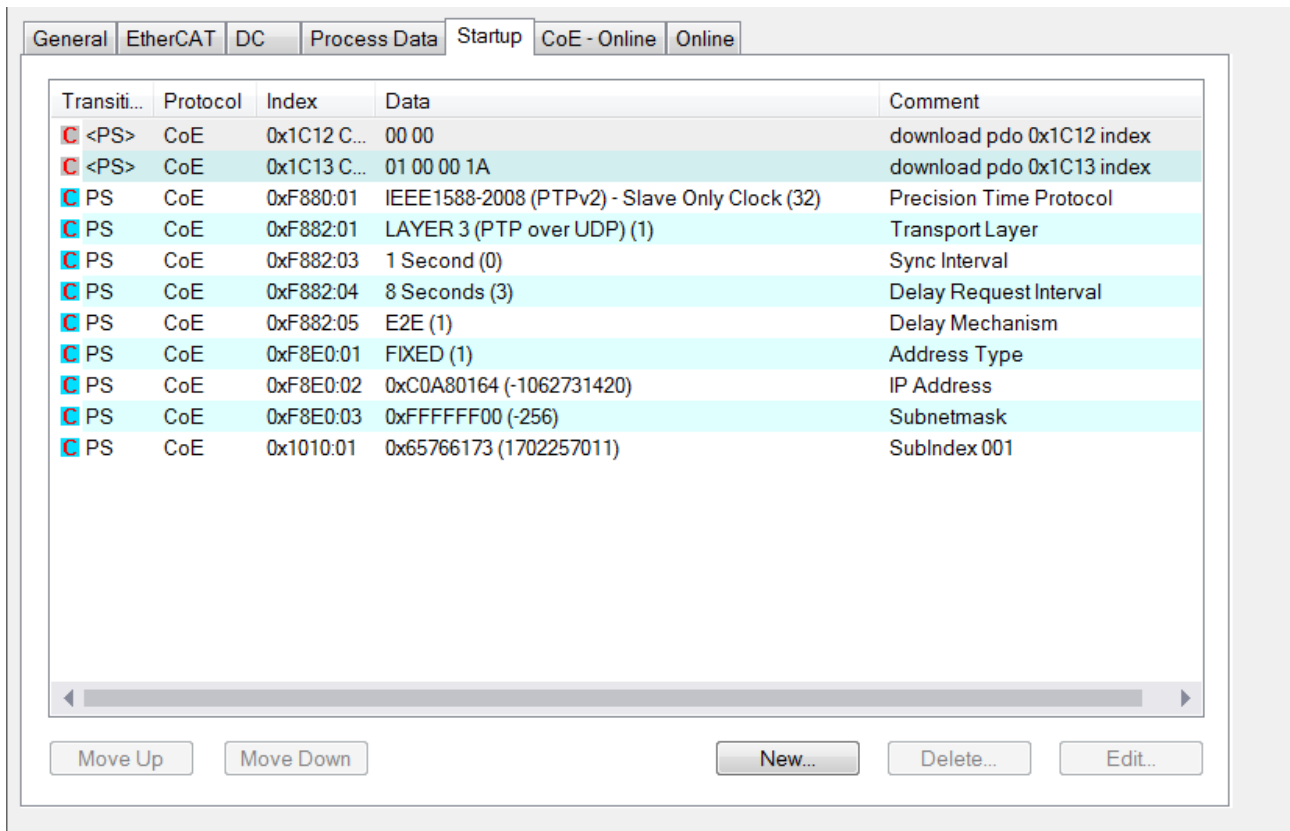


Fig. 82: Start-up list

Special features or start-up behavior

If a terminal is replaced with a new Beckhoff terminal, it will have the factory settings. It is therefore recommended to link all changes in the CoE directory of an EtherCAT slave in the startup list of the slave, which is sent to the slave at each startup of the EtherCAT fieldbus in the PreOP/SafeOP or SafeOP/OP transition.

In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

Please note for the EL6688: immediately when the terminal starts up, the PTP clock is started with the parameters stored in the CoE, i.e. already in the INIT state. A change of the PTP settings in the CoE, e. g. by the StartUp list in the later PreOp/SafeOP, only becomes effective at the next/subsequent start from INIT, since the PTP clock cannot be changed during operation. After changing the PTP settings the terminal must pass through the INIT state once, e.g. by restarting EtherCAT, PowerOn/Off of the terminal or state change from the PLC.

This also applies to the special case of initial commissioning with factory settings or after changes to the PTP settings: only during the subsequent startup after INIT does the PTP Clock operate as intended.

4. Check EtherCAT master settings

- a) Master settings → Info Data: enable "Include DC Time Offsets"
- b) Distributed Clocks → DC Mode: DC in use -> DC Time controlled by External Sync Device (External Mode) - as a result, this TwinCAT system now becomes a "time slave" and follows the external PTP time

5. If not already done, create a task and link it with EtherCAT

e.g. Cycle time: 1ms, if necessary with autostart:

6. Check that, at the latest from now on, the EL6688 is connected to the grandmaster via the X1-port (RJ45).

7. Activate TwinCAT configuration and start TwinCAT in RUN mode

8. Diagnosis via PTP Diag (0xFA80)

- "PTP State" should change from "LISTENING" to "SLAVE" after 10 - 15 seconds
- "Sync Event Sequence Counter" should count up according to the sync interval of the grandmaster

The screenshot shows the configuration interface for the PTP Diag (0xFA80). The 'PTP State' is set to 'SLAVE (9)' and the 'Sync Event Sequence Number' is 0xE69B (59035). Other parameters include PTP Version (PTPv2 (IEEE1588)), Clock Identity (00 01 05 FF FE 30), Parent Port Identity (00 00 BC FF FE 2E), Grandmaster Identity (FF FF 00 01 00 2C), Offset From Master (ns) (-148), Mean Path Delay (ns) (0), and Steps Removed (0x0002 (2)).

Index	Name	Flags	Value
F882:0	PTPv2 Settings	RW	> 9 <
F8E0:0	Ethernet Settings	RW	> 4 <
F8F0:0	Vendor data	RW	> 1 <
FA80:0	PTP Diag	RO	> 15 <
FA80:01	PTP Version	RO	PTPv2 (IEEE1588-
FA80:02	PTP State	RO	SLAVE (9)
FA80:03	Clock Identity	RO	00 01 05 FF FE 30
FA80:04	Parent Port Identity	RO	00 00 BC FF FE 2E
FA80:05	Grandmaster Identity	RO	FF FF 00 01 00 2C
FA80:06	Offset From Master (ns)	RO	-148
FA80:07	Mean Path Delay (ns)	RO	0
FA80:08	Steps Removed	RO	0x0002 (2)
FA80:09	Sync Event Sequence Number	RO	0xE69B (59035)

Fig. 83: Diagnosis via PTP Diag (0xFA80)

9. Diagnosis via FB_EcExtSyncCheck (TC2) or FB_EcExtSyncCheck64 (TC3)

- "Offset From Master" should now settle around "0" in accordance with the hardware in use and its accuracy (typ. less than ± 1000 ns = 1 μ s)
- For this purpose the following sample program can be used.



Sample program <https://infosys.beckhoff.com/content/1033/el6688/Resources/12797784843/.zip>

For further information, see chapter "[Answers to frequently asked questions \[▶ 98\]](#)"

General notes

● Changing the settings



Always use the following procedure for changing CoE settings in the EL6688:

1. Set EL6688 to PREOP
2. Change parameters
3. Enter the value 0x65766173 in index [0x1010 \[▶ 102\]:01](#).
4. Set EL6688 to INIT and then OP

- EtherCAT cycle time: 1 ms is recommended; the slower the cycle time, the less accurate the synchronization
- Due to the BMCA it takes several seconds before the EL6688 can supply analyzable data. After the clock startup the bus is monitored for approx. 10 seconds so that any available grandmasters can be found. The EL6688 is ready for operation when the toggle bits in the process data are operational.
- Synchronization accuracy: with the EL6688 precisions of up to ± 350 ns can be achieved, depending on the hardware type used.
The shorter the sync interval selected, the better the synchronization accuracy and settling will be after the TwinCAT start.
- Basics regarding the time synchronization direction EL6688 (Slave) → TwinCAT:
In this mode of operation the EL6688 receives the PTP frames and operates itself as a PTP Slave Clock. In the TwinCAT/EtherCAT system it operates as an external reference clock, but not necessarily as the EtherCAT reference clock at the same time. The forwarding of the time control is incumbent on the EtherCAT master employed (TwinCAT). Hence, the control sequence as follows:
 - The processor of the EL6688 receives the PTP frames several times per second. The frame contains the so-called external timestamp, if necessary with adjustment information.
In the same moment as it receives a PTP frame, the processor reads the DC time – the internal timestamp – from the local EtherCAT Distributed Clock in the ESC.
Hence pairs are continuously formed from internal and external timestamp, from which the readjustment requirement can be derived.
 - These pairs are not processed in the EL6688, but conveyed as cyclic process data to the EtherCAT master (TwinCAT).
 - The latter evaluates these pairs at its own discretion/ability and
 - if necessary regulates its own real-time, which controls the sending of the EtherCAT frames.
 - adjusts the DC reference clock, which is usually realized in the first DC-capable EtherCAT slave.
Hence, the Distributed Clock system follows the external PTP Master Clock. As a consequence of this the cyclic ARMW command of the EtherCAT master in turn distributes this "DC reference" time to subordinated DC slaves further back in the EtherCAT strand
- The functional chain is thus represented as follows: External PTP master clock → EL6688 as PTP slave clock → Internal /external timestamp as PDO to the EtherCAT master → Adjustment of the internal real-time and adjustment of the DC reference clock.
The position of the EL6688 in the EtherCAT strand is thus insignificant for the time control; the EL6688 itself has no possibility to affect its own Distributed Clock.
- Procedure 1-Step/2-Step with PTP v1/v2: no setting is possible in this context.
 - EL6688 as master: only 2-step possible, i.e. with follow-up message.
Against 2-step slave: possible
Against 1-step slave: depends exclusively on the slave itself and the intermediate network infrastructure (switch, boundary clock...). The slave and the infrastructure must be able to evaluate the 2-step flag and to cope with the 2-step sync and follow-up messages of the grandmaster.

- EL6688 as slave
 - Against 2-step master: possible
 - Against 1-step master: possible

Process data

Sync Mode	0: no synchronization 1: device operates as SYNC master 2: device operates as SYNC slave
Control Value Update Toggle	only for SYNC slave. toggles every time when the control value was updated
Time Stamp Update Toggle	only for SYNC slave. toggles every time when the control value was updated
External device not connected	TRUE: no external synchronization found (always TRUE if device operates as SYNC master)
Internal Time Stamp	only for SYNC slave. DC time stamp at the same time as the external time stamp This is a cyclically calculated time, not the DC absolute time.
External Time Stamp	only for SYNC slave. external time stamp recalculated in DC units (ns) This is a cyclically calculated time, not the absolute time from the last PTP telegram.
Time Control Value	TwinCAT does not use this value.

Default state

- PTPv1 Slave (Index 0xF880:01)
- SyncInterval (Index 0xF881:03): 2 seconds
- DomainName (Index 0xF881:01): _DFLT
- Own IP: 0.0.0.0
- Own MAC: 00.01.05.xx.xx.xx (Beckhoff)

Supported PTP modes

Operation mode	Note
PTPv1 SlaveOnly	from SW02
PTPv1 Grandmaster	from SW03
PTPv1 Best Master Clock	on request
PTPv2 SlaveOnly	from SW03
PTPv2 Grandmaster	from SW07
PTPv2 Best Master Clock	on request

Mutual settings

Depending on the protocol used, the following settings must match in the grandmaster clock and the slave clock(s). For setting the EL6688 see the CoE list from index 0xF880. Changes must be saved permanently by writing 0x65766173 into index 0x1010:01.

- PTPv1
 - DomainName
 - SyncInterval
 - Delay Request Interval
- PTPv2
 - TransportLayer
 - DomainNumber
 - DelayMechanism (disabled, End2End, Peer2Peer [from FW10])
 - SyncInterval
 - Delay Request Interval

Quality settings

For setting the EL6688 see the CoE list from index 0xF880. Changes must be saved permanently by writing 0x65766173 into index 0x1010:01.

- PTPv1
"Clock Stratum": the smaller this value, the better quality the master certifies itself
"Preferred": this flag indicates that this clock is to be used preferentially as master
Both values influence the BMCA in the network.
- PTPv2
Priority1+2 correspond to Stratum+Preferred

TwinCAT settings

Distributed clocks (DC) timing settings

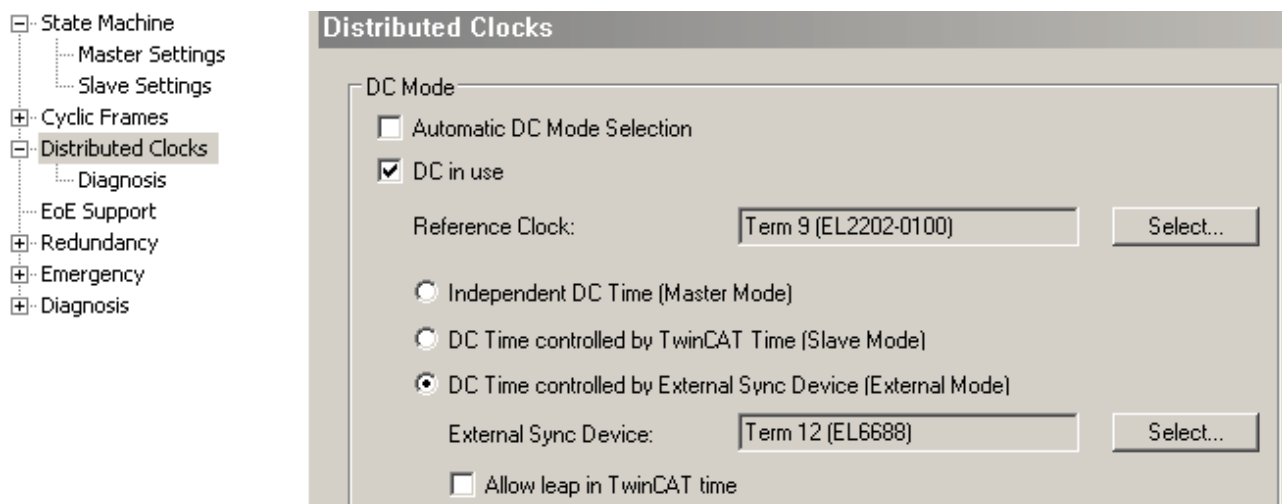


Fig. 84: TwinCAT 2.11 DC settings

We recommended manually selecting the EL6688 as reference clock through "DC in use".

- **Independent DC Time:** one of the EL terminals (generally the first terminal supporting distributed clocks (DC)) is the reference clock to which all other DC terminals are adjusted. Even an EL6688 that may be present in the system is adjusted to this reference clock as a DC slave. Selection of the reference clock in the dialog above.
This setting is useful if the EL6688 is operated as PTP grandmaster clock and another EtherCAT system is to be synchronized.
- **DC Time controlled by TwinCAT:** the DC reference clock is adjusted to the local TwinCAT time.
- **DC Time controlled by External Sync Device:** if the EtherCAT system is to be adjusted to a higher-level clock, the External Sync Device can be selected here.
This setting is useful if the EL6688 is operated as a PTP slave clock and another EtherCAT system is to synchronize its own.

TwinCAT behavior

After the EtherCAT startup the EL6688 takes a few seconds before it can supply data for the synchronization. This is indicated by toggling bits in the process data. Depending on the environment, a few more seconds may elapse before full synchronization is achieved.

Operation of the EL6688 as PTPv1 master

The SyncInterval of the sent SyncTelegrams can be changed. The default is 2 sec.

Stratum = 0 is automatically set, forcing this clock to act as grandmaster for the network.

In master mode the EL6688 reports "External device not connected" in the process data

Operation of the EL6688 as PTPv2 master

The SyncInterval as well as the AnnounceInterval of the sent Sync or Announce telegrams can be changed, default: Sync=1 s, Announce=2 s

The master operation can be forced by setting Priority1 = 1 and Priority2 = 1 as grandmaster in the network.

In master mode the EL6688 reports "External device not connected" in the process data

Operation of the EL6688 as PTPv1 slave

Stratum = 255 is set to prevent this clock being use as grandmaster in the network.

Own IP address: the EL6688 can be allocated a dedicated IP address, under which it sends its DelayRequest messages, for example. It can also be used for ping control. As far as the set Delay Request Interval is not equal to "disabled", then an IP address $\neq 0$ is mandatory.

Operation of the EL6688 as PTPv2 slave

Slave operation can be forced by setting Priority1 = 255 and Priority2 = 255, so this clock will never be used as a grandmaster in the network.

Own IP address: the EL6688 can be allocated a dedicated IP address, under which it sends its (P)DelayRequest messages, for example. It can also be used for ping control. PTPv2 defines two different transport layers:

Layer 2 "PTP over ETHERNET": IP address $\neq 0$ is not required for delay mechanism (only relevant for ping control)

Layer 3 "PTP over UDP": IP address $\neq 0$ is required for delay mechanism (E2E or P2P) and ping control

5.4 Answers to frequently asked questions

- [What time base is suitable for the synchronization of distributed systems? \[▶ 98\]](#)
- [What needs to be considered if sun times \(e. g. UTC\) are to be used? \[▶ 99\]](#)
- [How is an absolute time, e. g. UTC, calculated? \[▶ 100\]](#)
- [Why is DcToTcTimeOffset sometimes 0 and DcToExtTimeOffset other times 0? \[▶ 100\]](#)
- [How is a T_DCTIME64 variable turned into something user readable? \[▶ 100\]](#)
- [Error in PTP diagnosis: PTP state should change from "LISTENING" to "SLAVE" \[▶ 100\]](#)
- [Error in PTP diagnosis: "Offset From Master" does not settle around "0" \[▶ 101\]](#)

What time base is suitable for the synchronization of distributed systems?

PTP/IEEE1588 is designed in such a way that TAI is distributed as a time source and, if required, the UTC time can be calculated in the respective end devices using the "UTC Offset".

In order avoid time jumps (which can lead to undesired effects within a plant), it is advisable to use and distribute a constantly advancing time (e. g. without leap seconds), such as the international atomic time [French **T**emps **A**tomique **I**nternational (TAI)].

Coordinated **U**niversal **T**ime (UTC), on the other hand, involves the risk of time jumps (due to leap seconds), but is synchronous with solar time and thus more "suitable for everyday use" for many processes.

Nevertheless, the direct use of UTC is not recommended, since (for the control unpredictable) time jumps can considerably disturb the proper operation of a plant. If UTC is still required for the application, the indirect way via the UTC offset should be chosen. The TAI can be converted to UTC using the UTC Offset in the application.

The base time provided by the grandmaster is used as the base time of the EL6688! This base time will almost always be the TAI (because other time sources/bases can be jumpy). If necessary, commercially available grandmasters show UTC in the display, but TAI is distributed in the network (if the device has not been adjusted).

What needs to be considered if sun times (e. g. UTC) are to be used?

The indirect way via TAI and the UTC offset should always be chosen. The TAI can be converted to UTC using the UTC Offset in the application. If the grandmaster provides information about the current UTC offset, this information is stored in the CoE of the EL6688 so that it can be processed in the controller. The current number of leap seconds (CoE: CurrentUtcOffset (0xFA80.0B)) as well as announcements of possible further leap events (CoE: Leap61 (0xFA80.0D) for plus 1 s and Leap59 (0xFA80.0E) for minus 1 s) are transmitted and should be evaluated by the user.

Concretely: if the Grandmaster has determined a correct UTC offset e. g. via GPS, it will provide this value in the network:

Index	Name	Flags	Value
FA80:0	PTP Diag	RO	> 15 <
FA80:01	PTP Version	RO	PTPv2 (IEEE1588-2008) (32)
FA80:02	PTP State	RO	MASTER (6)
FA80:03	Clock Identity	RO	00 01 05 FF FE 30 15 88
FA80:04	Parent Port Identity	RO	00 00 00 00 00 00 00 00 00
FA80:05	Grandmaster Identity	RO	00 00 00 00 00 00 00 00
FA80:06	Offset From Master (ns)	RO	0
FA80:07	Mean Path Dealy (ns)	RO	0
FA80:08	Steps Removed	RO	0x0000 (0)
FA80:09	Sync Event Sequence Number	RO	0x0000 (0)
FA80:0A	Timescale	RO	PTP (TAI) (1)
FA80:0B	CurrentUtcOffset	RO	0
FA80:0C	CurrentUtcOffsetIsValid	RO	FALSE (0)
FA80:0D	Leap61	RO	FALSE (0)
FA80:0E	Leap59	RO	FALSE (0)
FA80:0F	Epoch_Number	RO	0x0000 (0)

- 0xFA80:0A Timescale, transmitted by the grandmaster: for TAI the value "1" is given
- 0xFA80:0B CurrentUtcOffset, transmitted by the grandmaster: offset to UTC time in seconds
- 0xFA80:0C CurrentUtcOffsetValid, transmitted by the grandmaster:
 - TRUE if offset is known/valid,
 - FALSE if invalid
- 0xFA80:0D Leap61, transmitted by the Grandmaster: leap second event pending: 1 second added at 23:59:60 (no effect on TAI)
- 0xFA80:0E Leap59, transmitted by the grandmaster: leap second event pending: 1 second subtracted at 23:59:59 (no effect on TAI)

The leap second events "leap59" or "leap61" do not affect TAI or a device synchronized with TAI time source, this information is only used by the application, which can use it to completely map or calculate the UTC time and, if necessary, take suitable protective measures in the event of a leap second.

How is an absolute time, e. g. UTC, calculated?

If an external time source (e. g. a GPS-assisted PTP master) is connected with TwinCAT via the EL6688, the UTC can be calculated using the following equation (see example project for TC2 resp. TC3) (reference [example PLC project \[► 94\]](#)):

$$\text{UTCabsolut} := \text{F_GetCurExtTime64} (\text{DcToExtTimeOffset}, \text{DcToTcTimeOffset}) - \text{UTC_Offset_NSec}$$

or

$$\text{UTCabsolut} := \text{F_ConvTcTimeToExtTime64} (\text{GetCurDcTaskTime64}(), \text{DcToTcTimeOffset}, \text{DcToExtTimeOffset}) - \text{UTC_Offset_NSec}$$

- UTCabsolute - task current UTC time expressed as a distance from 1/1/2000 00:00. in nanoseconds.
- F_GetCurExtTime64() - supplies the start time (ExtTime, as T_DCTIME64 alias ULINT) of the task in which it is called.
- F_GetCurDcTaskTime64() – supplies the start time (TcTime, as T_DCTIME64 alias ULINT) of the task in which it is called.
- F_ConvTcTimeToExtTime64()- converts TcTime to ExtTime.

(GetCurExtTime64, GetCurDcTaskTime64 as well as ConvTcTimeToExtTime64 are part of the PLC library Tc2_EtherCAT.lib)

- DcToTcTimeOffset - during the TwinCAT system start the TC time is initialized by the Windows clock. If the external clock is also available during the system startup, deviations between these clocks are thus compensated.
- UTC_Offset_NSec - since TwinCAT works internally with the TAI time, any leap seconds introduced up to now must be subtracted (status 2017: 37 seconds) (see also EL6688-CoE: CurrentUtcOffset (0xFA80.0B)).
- DcToExtTimeOffset - TwinCAT uses this offset variable to handle larger deviations between the external clock and the DC time that occur during operation (e. g. due to loss of the connection to the external clock) and can no longer be automatically compensated.

Why is DcToTcTimeOffset sometimes 0 and DcToExtTimeOffset other times 0?

During the development phase of a controller, the "Auto Boot" setting in "Boot Settings" in the TwinCAT system may still be set to "Config Mode". If this is the case, TwinCAT cannot access the external clock during the startup phase (even with a physical connection between PTP master and EL6688) - thus DcToTcTimeOffset = 0 and the difference is entered by TwinCAT in DcToExtTimeOffset.

Only by setting "Auto Boot" to "Run Mode (Enable)" TwinCAT can also access the external clock during the startup phase and thus process the offset and store it in the variable "DcToTcTimeOffset".

How is a T_DCTIME64 variable turned into something user readable?

$$\text{sXXX} := \text{DCTIME64_TO_STRING}(\text{tXXX})$$

- tXXX - 503764149978700026 - DCTIME64 alias ULINT: time as distance in nanoseconds from 1/1/2000 00:00
- sXXX - '2015-12-18-14:29:09.978700026' - STRING(29):
- DCTIME64_TO_STRING() - transforms a T_DCTIME64 to a 29 character-string. Part of PLC library: Tc2_EtherCAT

Error in PTP diagnosis: PTP state should change from "LISTENING" to "SLAVE"

- Check the correspondence of the PTP Settings between PTP Master and EL6688 (CoE: 0xF882).
- Check the network settings between the PTP master and the EL6688 (CoE: 0xF8E0). The IP address must not be zero.

Error in PTP diagnosis: "Offset From Master" does not settle around "0"

- Check that "Auto Boot" is set to "Run Mode" and restart the system.
- Depending on the hardware used and its accuracy (for example: less than ± 1000 ns = 1 μ s for a GPS-supported system), the "Offset From Master" value oscillates according to this accuracy. Increasing the Sync Interval (CoE: 0xF882.3) can improve the accuracy, but also increases the network-side traffic.

5.5 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 terminal 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 information \[► 23\]](#) 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
- “CoE-Reload” for resetting the changes

Introduction

The CoE overview contains objects for different intended applications:

- [Objects required for parameterization \[► 102\]](#) during commissioning
- [Objects intended for regular operation \[► 104\]](#), e.g. through ADS access.
- [Objects for indicating internal settings \[► 104\]](#) (may be fixed)

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

5.5.1 Objects for commissioning

Index 0x1010 Store parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1010:0	Store parameters	Max. Subindex	UINT8	RO	0x01 (1 _{dec})
1010:01	SubIndex 001	Changes in the configuration can be saved permanently by entering 0x65766173	UINT32	RW	0x00000000 (0 _{dec})

Index 0x1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	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 0xF880 PTP Common

Index (hex)	Name	Meaning	Data type	Flags	Default
F880:0	PTP Common	Max. Subindex	UINT8	RO	0x01 (1 _{dec})
F880:01	Precision Time Protocol	EL6688 mode setting: PTPV1/2	UINT16	RW	0x0010 (16 _{dec})

Index 0xF881 PTPv1 Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F881:0	PTPv1 Settings	Max. Subindex	UINT8	RO	0x07 (7 _{dec})
F881:01	Subdomain Name	Name of the SubDomain: _DFLT, _ALT1, _ALT2, _ALT3	UINT16	RW	0x0000 (0 _{dec})
F881:02	Subdomain Name (user)	Subdomain Name (user)	STRING	RW	
F881:03	Sync Interval	Interval of the sync telegrams from the PTP master	UINT16	RW	0x0001 (1 _{dec})
F881:04	Delay Request Interval	Interval of the DelayRequest telegrams from the PTP slave	UINT16	RW	0x0005 (5 _{dec})
F881:05	Clock is preferred	1: The clock tries to become grandmaster	UINT16	RW	0x0000 (0 _{dec})
F881:06	Clock Stratum	Information about the time quality of the own clock within the PTP network	UINT16	RW	0x0004 (4 _{dec})
F881:07	Management	Management	UINT16	RW	0x0000 (0 _{dec})

Index 0xF882 PTPv2 Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F882:0	PTPv2 Settings	Max. Subindex	UINT8	RO	0x09 (9 _{dec})
F882:01	Transport Layer	Transport Layer	UINT16	RW	0x0001 (1 _{dec})
F882:02	Domain Number	Domain Number	UINT16	RW	0x0000 (0 _{dec})
F882:03	Sync Interval	Sync Interval	UINT16	RW	0x0000 (0 _{dec})
F882:04	Delay Request Interval	Delay Request Interval	UINT16	RW	0x0000 (0 _{dec})
F882:05	Delay Mechanism	Delay Mechanism	UINT16	RW	0x0001 (1 _{dec})
F882:06	Announce Interval	Announce Interval	UINT16	RW	0x0001 (1 _{dec})
F882:07	Announce Interval Timeout	Announce Interval Timeout	UINT16	RW	0x0003 (3 _{dec})
F882:08	Priority1	Priority1	UINT16	RW	0x0080 (128 _{dec})
F882:09	Priority2	Priority2	UINT16	RW	0x0080 (128 _{dec})

Index 0xF8E0 Ethernet Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F8E0:0	Ethernet Settings	Max. Subindex	UINT8	RO	0x04 (4 _{dec})
F8E0:01	Address Type	default: FIXED	UINT16	RW	0x0001 (1 _{dec})
F8E0:02	IP Address	IP address under which a PTP device is accessible or sends its telegrams	UINT32	RW	0x00000000 (0 _{dec})
F8E0:03	Subnetmask	Subnetmask	UINT32	RW	0x00000000 (0 _{dec})
F8E0:04	Gateway	Gateway	UINT32	RW	0x00000000 (0 _{dec})

5.5.2 Objects for regular operation

The EL6688 has no such objects.

5.5.3 Other objects 0x1000-0xFFFF**Complete overview****Standard objects (0x1000-0x1FFF)**

Standard objects have the same meaning for all EtherCAT slaves.

Index 0x1000 Device type

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

Index 0x1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL6688

Index 0x1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	02

Index 0x100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	02

Index 0x1018 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	0x1A203052 (438317138 _{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	0x00110000 (1114112 _{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 0x10F0 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 0x10F4 External synchronization status

Index (hex)	Name	Meaning	Data type	Flags	Default
10F4:0	External synchronization status	Max. Subindex	UINT8	RO	0x13 (19 _{dec})
10F4:01	Sync Mode	Sync Mode	BIT2	RO	0x00 (0 _{dec})
10F4:0E	Control value update toggle	Control value update toggle	BOOLEAN	RO	0x00 (0 _{dec})
10F4:0F	Time stamp update toggle	Time stamp update toggle	BOOLEAN	RO	0x00 (0 _{dec})
10F4:10	External device not connected	External device not connected	BOOLEAN	RO	0x00 (0 _{dec})
10F4:11	Internal time stamp	Internal time stamp	UINT64	RO	
10F4:12	External time stamp	External time stamp	UINT64	RO	
10F4:13	Control Value for DC Master Clock	Control Value for DC Master Clock	INT32	RO	0x00000000 (0 _{dec})

Index 0x10F5 External synchronization settings

Index (hex)	Name	Meaning	Data type	Flags	Default
10F5:0	External synchronization settings	Max. Subindex	UINT8	RO	0x12 (18 _{dec})
10F5:01	Sync master	Sync master	BOOLEAN	RW	0x00 (0 _{dec})
10F5:02	32 Bit time stamps	32 Bit time stamps	BOOLEAN	RW	0x00 (0 _{dec})
10F5:11	Control Interval (ms)	Control Interval (ms)	UINT16	RW	0x0000 (0 _{dec})
10F5:12	Additional System Time	Additional System Time	UINT64	RW	

Index 0x1800 TxPDO-Par External Sync

Index (hex)	Name	Meaning	Data type	Flags	Default
1800:0	TxPDO-Par External Sync	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[4]	RO	01 1A 02 1A

Index 0x1801 TxPDO-Par External Sync (32 Bit)

Index (hex)	Name	Meaning	Data type	Flags	Default
1801:0	TxPDO-Par External Sync (32 Bit)	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[4]	RO	00 1A 02 1A

Index 0x1802 TxPDO-Par External Sync Compact

Index (hex)	Name	Meaning	Data type	Flags	Default
1802:0	TxPDO-Par External Sync Compact	PDO Parameter TxPDO 3	UINT8	RO	0x06 (6 _{dec})
1802:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 3	OCTET-STRING[4]	RO	00 1A 01 1A

Index 0x1A00 TxPDO-Map External Sync

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	TxPDO-Map External Sync	PDO Mapping TxPDO 1	UINT8	RO	0x09 (9 _{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x02)	UINT32	RO	0x10F4:02, 2
1A00:02	SubIndex 002	2. PDO Mapping entry (6 bits align)	UINT32	RO	0x0000:00, 6
1A00:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0E (Control value update toggle))	UINT32	RO	0x10F4:0E, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0F (Time stamp update toggle))	UINT32	RO	0x10F4:0F, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10 (External device not connected))	UINT32	RO	0x10F4:10, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x11 (Internal time stamp))	UINT32	RO	0x10F4:11, 64
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x12 (External time stamp))	UINT32	RO	0x10F4:12, 64
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x13 (Control Value for DC Master Clock))	UINT32	RO	0x10F4:13, 32

Index 0x1A01 TxPDO-Map External Sync (32 Bit)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	TxPDO-Map External Sync (32 Bit)	PDO Mapping TxPDO 2	UINT8	RO	0x09 (9 _{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x02)	UINT32	RO	0x10F4:02, 2
1A01:02	SubIndex 002	2. PDO Mapping entry (6 bits align)	UINT32	RO	0x0000:00, 6
1A01:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0E (Control value update toggle))	UINT32	RO	0x10F4:0E, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0F (Time stamp update toggle))	UINT32	RO	0x10F4:0F, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10 (External device not connected))	UINT32	RO	0x10F4:10, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x11 (Internal time stamp))	UINT32	RO	0x10F4:11, 32
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x12 (External time stamp))	UINT32	RO	0x10F4:12, 32
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x13 (Control Value for DC Master Clock))	UINT32	RO	0x10F4:13, 32

Index 0x1A02 TxPDO-Map External Sync Compact

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	TxPDO-Map External Sync Compact	PDO Mapping TxPDO 3	UINT8	RO	0x06 (6 _{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x02)	UINT32	RO	0x10F4:02, 2
1A02:02	SubIndex 002	2. PDO Mapping entry (6 bits align)	UINT32	RO	0x0000:00, 6
1A02:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A02:04	SubIndex 004	4. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0E (Control value update toggle))	UINT32	RO	0x10F4:0E, 1
1A02:05	SubIndex 005	5. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0F (Time stamp update toggle))	UINT32	RO	0x10F4:0F, 1
1A02:06	SubIndex 006	6. PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10 (External device not connected))	UINT32	RO	0x10F4:10, 1

Index 0x1C00 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 0x1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x00 (0 _{dec})

Index 0x1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1 _{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 _{dec})

Index 0x1C33 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: <ul style="list-style-type: none"> • 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	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	0x00000000 (0 _{dec})
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0: free run is supported • Bit 1: Synchronous with SM 2 Event is supported (outputs available) • Bit 1: Synchronous 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 or 1C33:08 [▶ 110]) 	UINT16	RO	0xC007 (49159 _{dec})
1C33:05	Minimum cycle time	as 1C32:05	UINT32	RO	0x000F4240 (1000000 _{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})
1C33:08	Command	as 1C32:08	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0 B	SM event missed counter	as 1C32:11	UINT16	RO	0x0000 (0 _{dec})
1C33:0 C	Cycle exceeded counter	as 1C32:12	UINT16	RO	0x0000 (0 _{dec})
1C33:0 D	Shift too short counter	as 1C32:13	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	as 1C32:32	BOOLEAN	RO	0x00 (0 _{dec})

Profile-specific objects (0x6000-0xFFFF)

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

Index 0xF000 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	0x0000 (0 _{dec})

Index 0xF008 Code word

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

Index 0xF8F0 Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
F8F0:0	Vendor data	Max. Subindex	UINT8	RO	0x01 (1 _{dec})
F8F0:01	MAC_Address	MAC_Address	OCTET-STRING[6]	RW	{0}

6 Appendix

6.1 EtherCAT AL Status Codes

For detailed information please refer to the [EtherCAT system description](#).

6.2 Firmware compatibility

Beckhoff EtherCAT devices are delivered with the latest available firmware version. Compatibility of firmware and hardware is mandatory; not every combination ensures compatibility. The overview below shows the hardware versions on which a firmware can be operated.

Note

- It is recommended to use the newest possible firmware for the respective hardware
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page. If a device is placed in BOOT-STRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL6688			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
02	01	EL6688-0000-0016	05/2008
02 - 04	02	EL6688-0000-0017	05/2009
	03		05/2009
03 - 09	04	EL6688-0000-0018	03/2010
	05	EL6688-0000-0019	02/2011
		EL6688-0000-0020	10/2012
	06	EL6688-0000-0021	07/2014
10 - 14*	07	EL6688-0000-0022	02/2016
	08		07/2016
	09		02/2017
	10		10/2017
	11*		11/2022

*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff web page whether more up-to-date [documentation](#) is available.

6.3 Firmware Update EL/ES/EM/ELM/EPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK and EP series. A firmware update should only be carried out after consultation with Beckhoff support.

NOTE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website <https://www.beckhoff.com/en-us/>.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

An EtherCAT slave stores operating data in up to three locations:

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (<https://www.beckhoff.com>). All ESI files are accessible there as zip files.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.

The TwinCAT System Manager offers mechanisms for programming all three parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a *.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxx-xxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

NOTE**Risk of damage to the device!**

- ✓ Note the following when downloading new device files
 - a) Firmware downloads to an EtherCAT device must not be interrupted
 - b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
 - c) The power supply must adequately dimensioned. The signal level must meet the specification.
- ⇒ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

6.3.1 Device description ESI file/XML**NOTE****Attention regarding update of the ESI description/EEPROM**

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

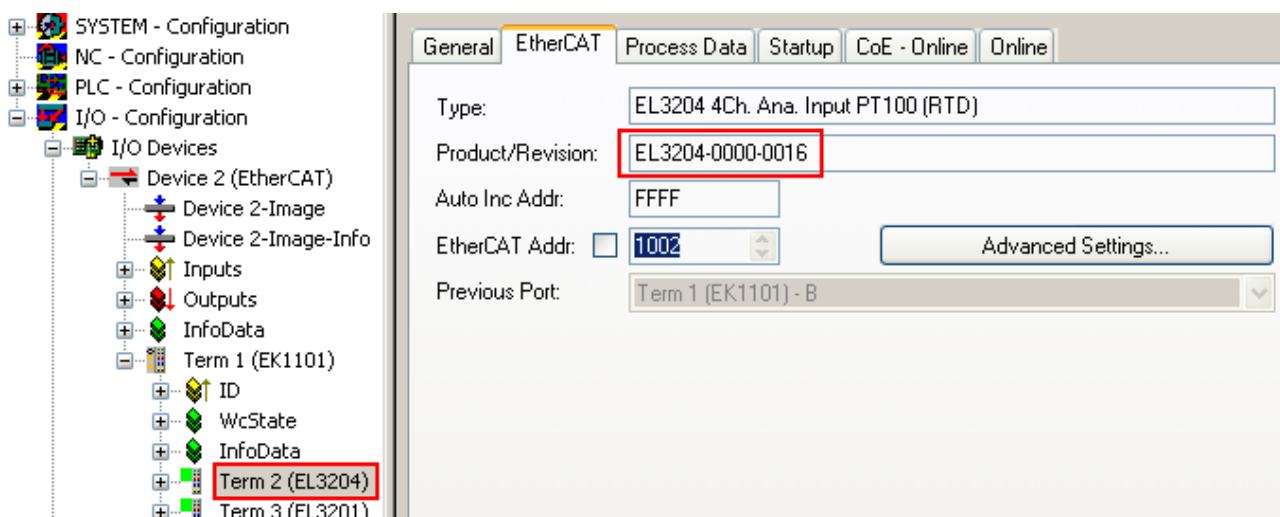


Fig. 85: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the [EtherCAT system documentation](#).

● Update of XML/ESI description

i The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

Display of ESI slave identifier

The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

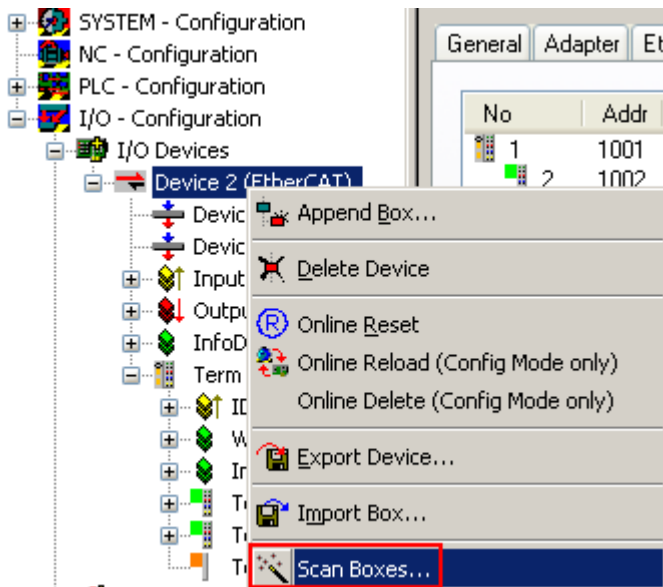


Fig. 86: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows



Fig. 87: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

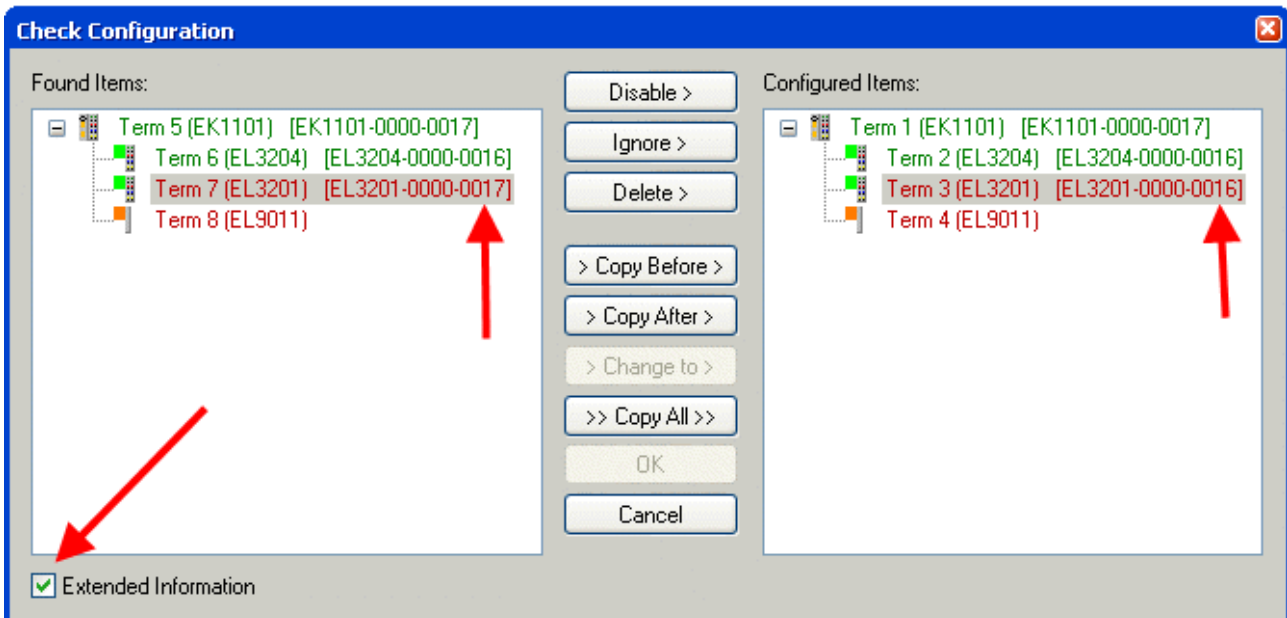


Fig. 88: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-0017 was found, while an EL3201-0000-0016 was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

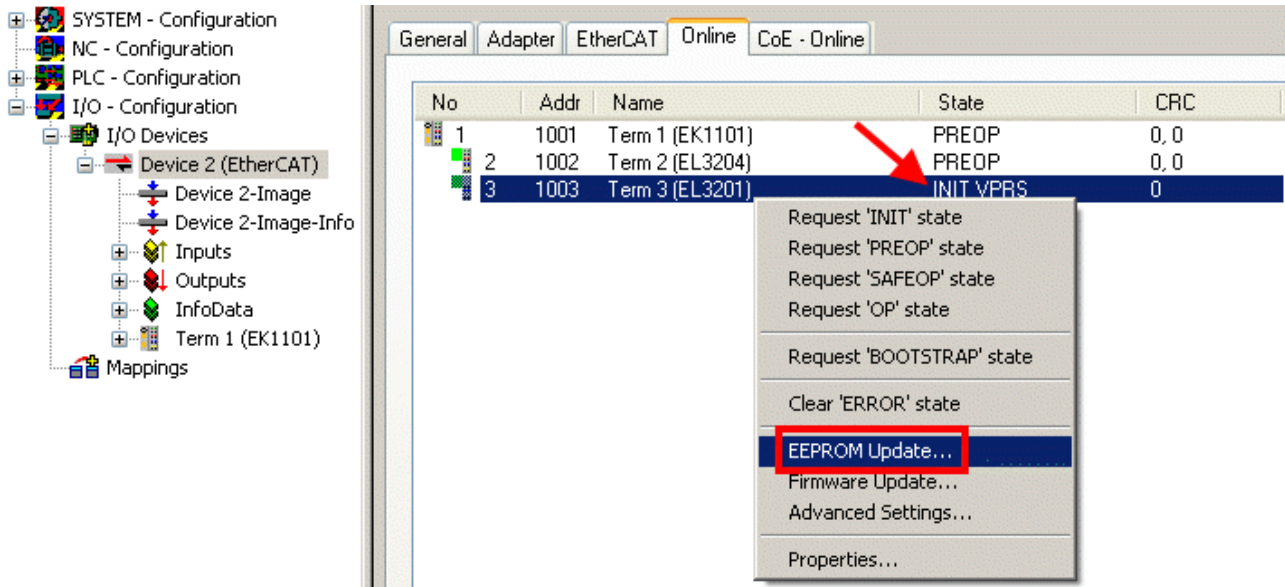


Fig. 89: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI*. The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

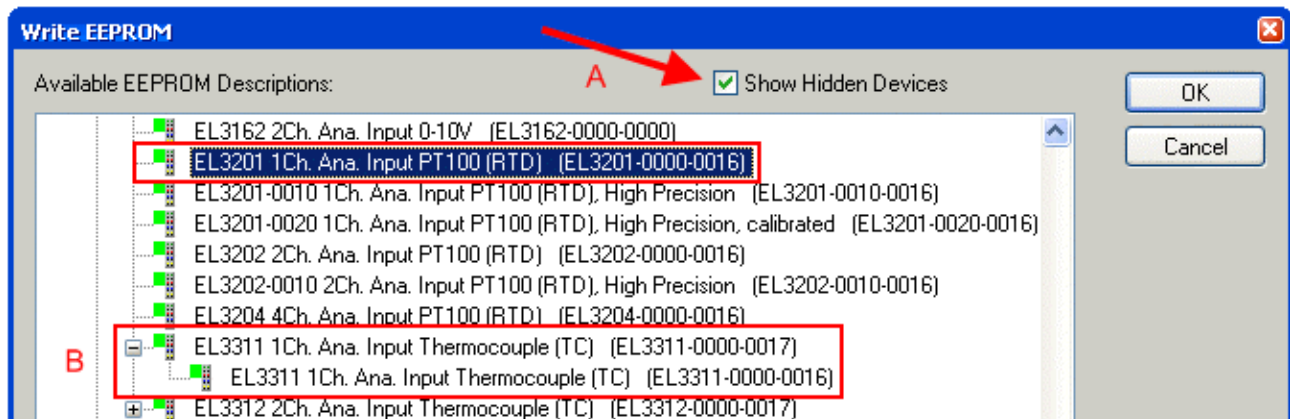


Fig. 90: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

i The change only takes effect after a restart.

Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

6.3.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

i **CoE Online and Offline CoE**

Two CoE directories are available:

- **online:** This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
- **offline:** The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. "Beckhoff EL5xxx.xml").

The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.

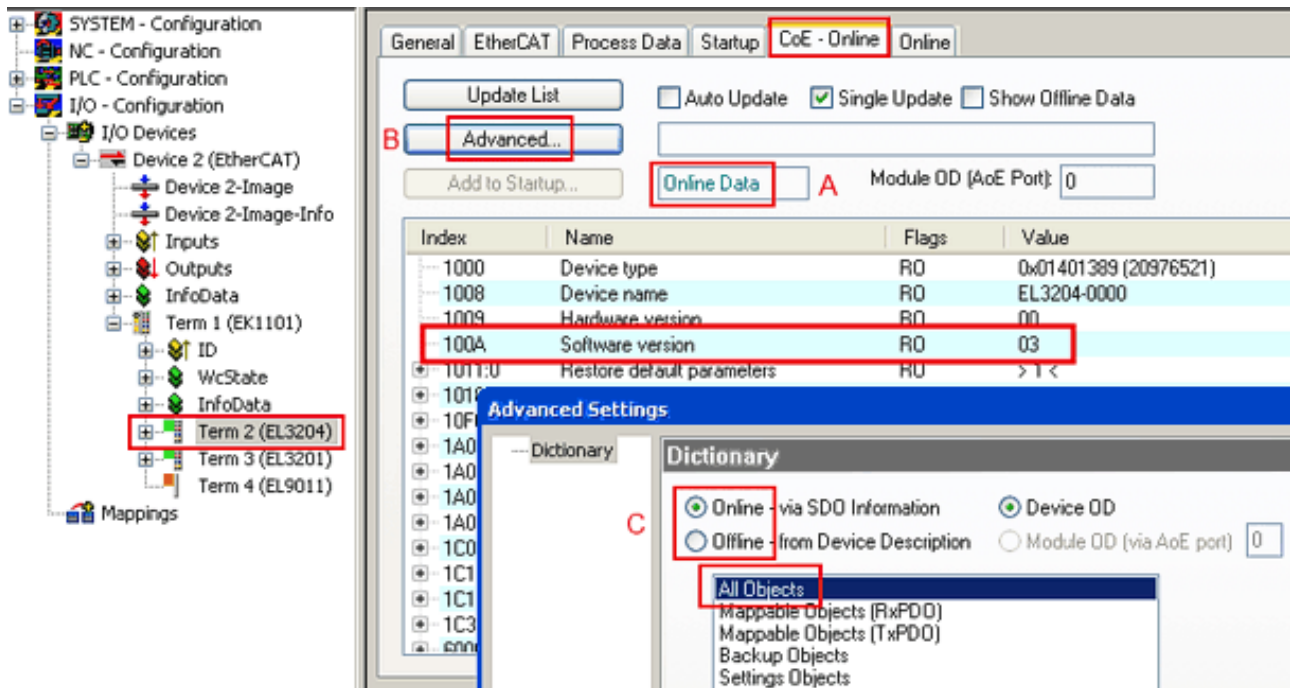


Fig. 91: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

6.3.3 Updating controller firmware *.efw

i **CoE directory**

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update*.

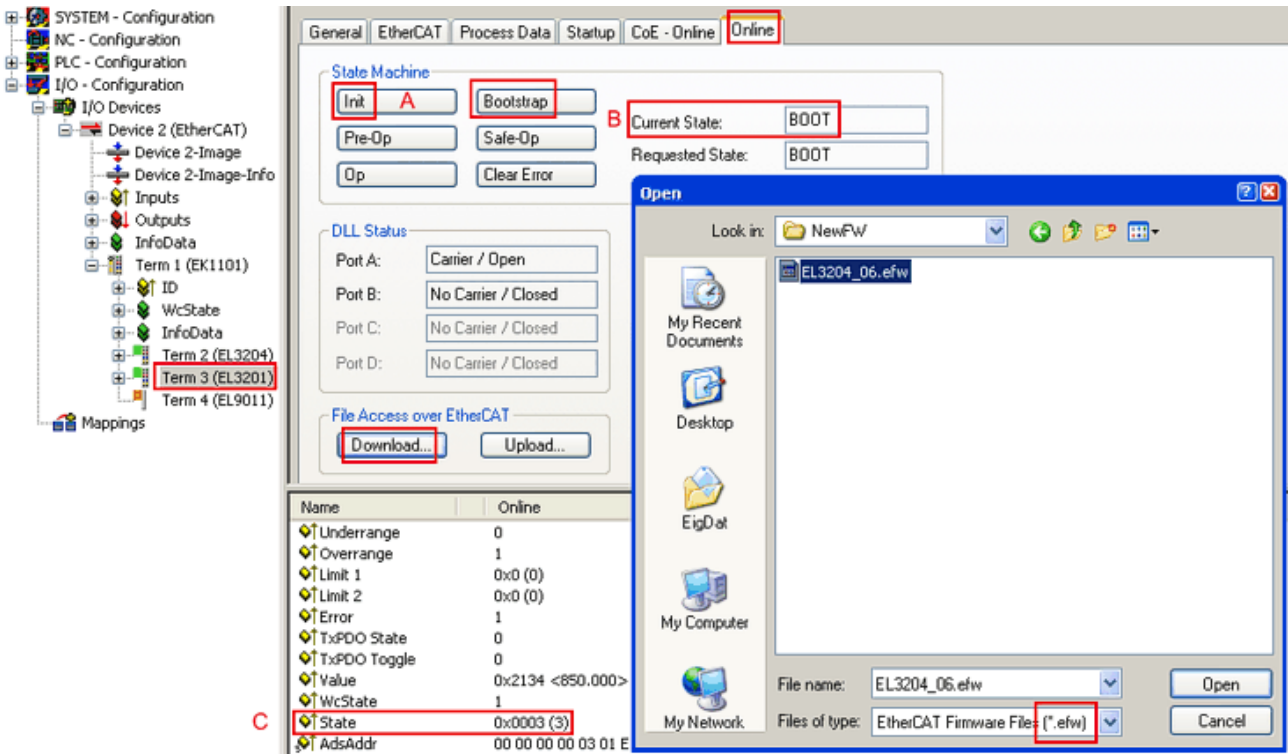
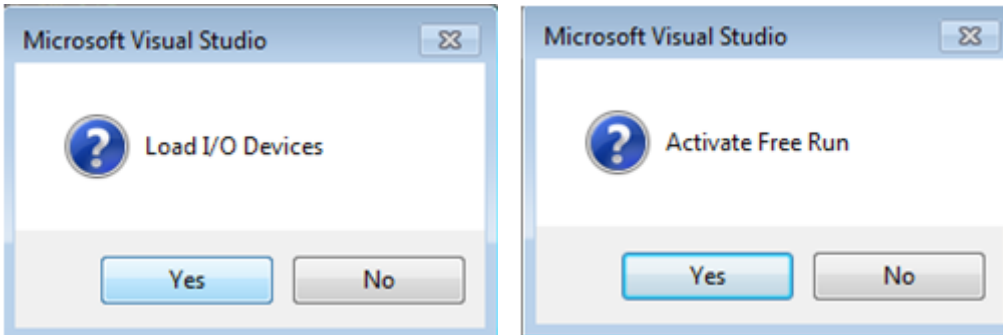


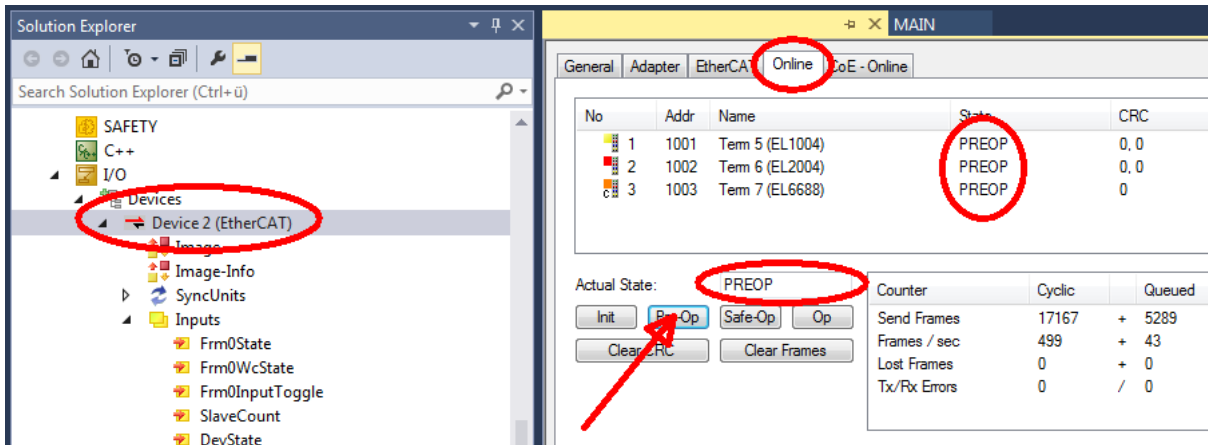
Fig. 92: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

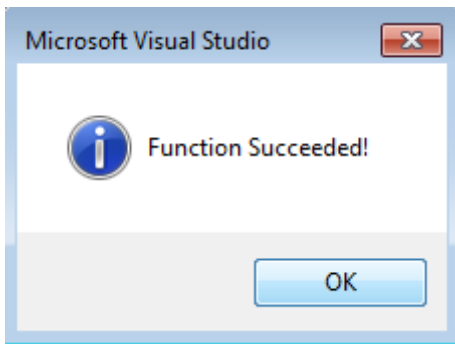


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

6.3.4 FPGA firmware *.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an *.rbf file.

- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

Determining the version via the System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.

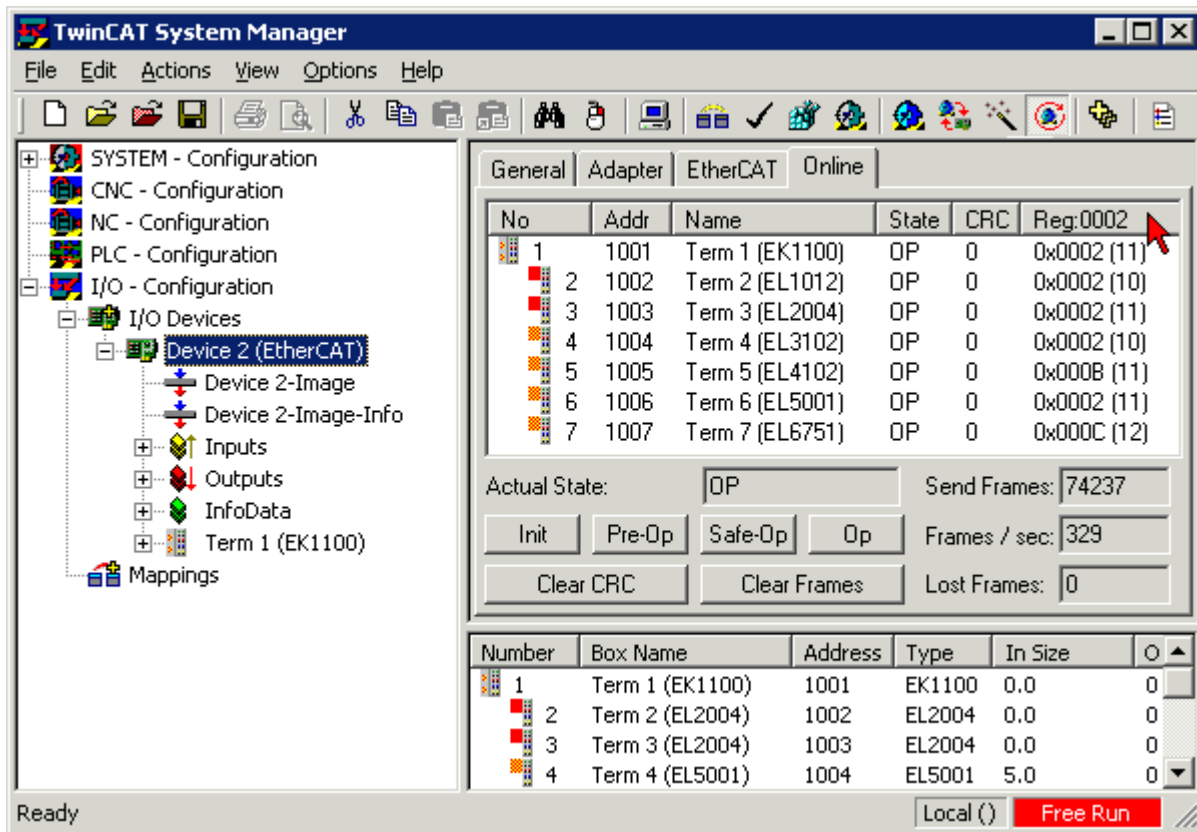


Fig. 93: FPGA firmware version definition

If the column *Reg:0002* is not displayed, right-click the table header and select *Properties* in the context menu.

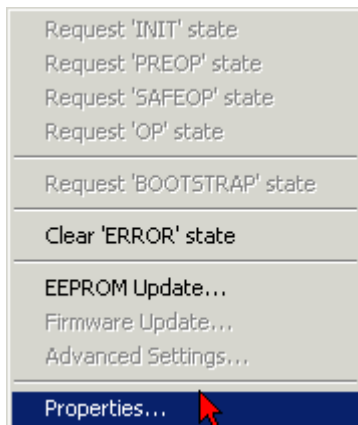


Fig. 94: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/Online View* select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.

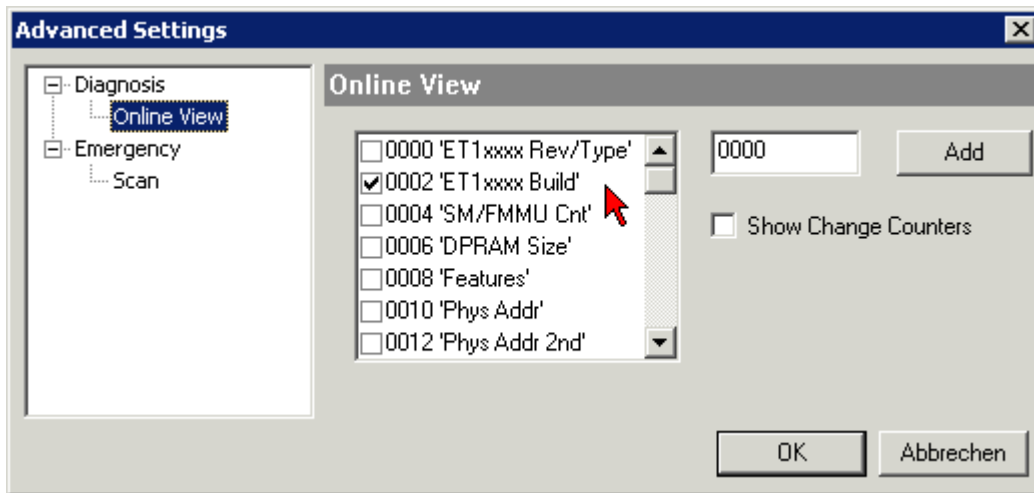


Fig. 95: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

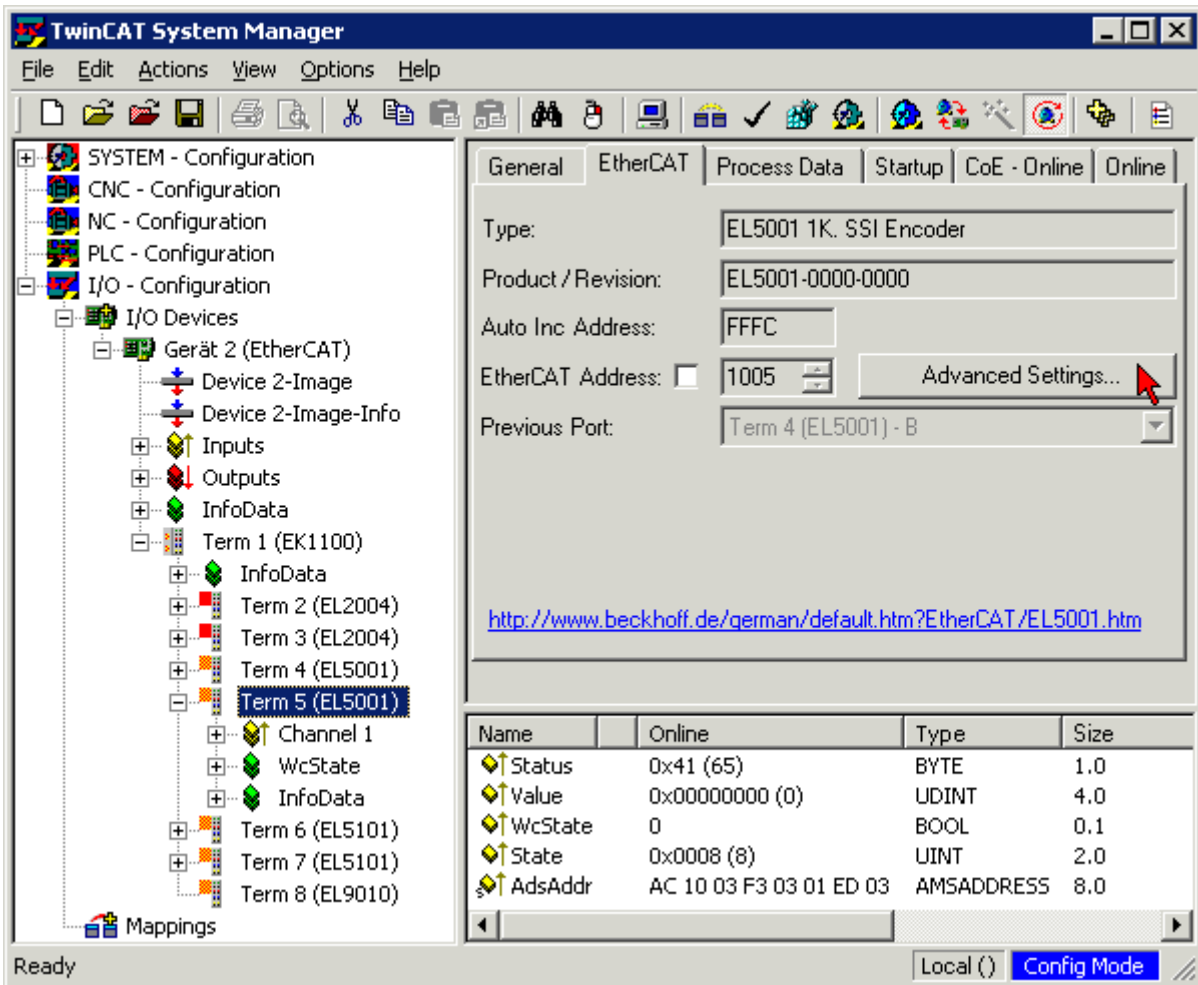
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

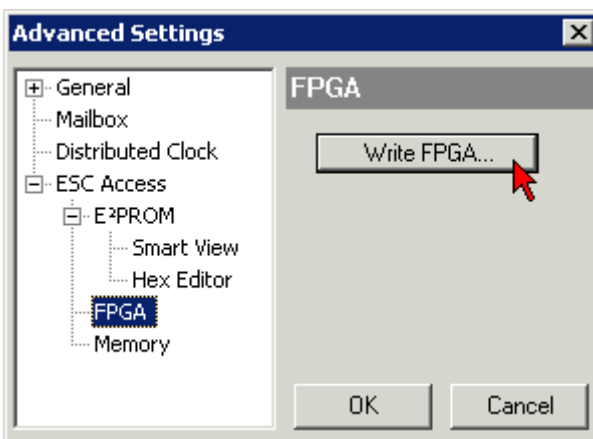
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

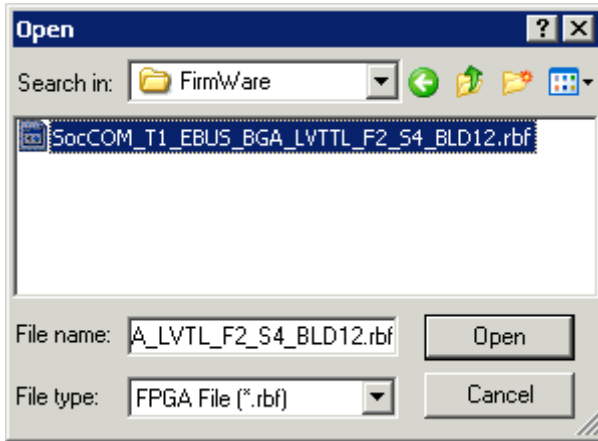
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM/FPGA* click on *Write FPGA* button:



- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTE

Risk of damage to the device!

A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

6.3.5 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

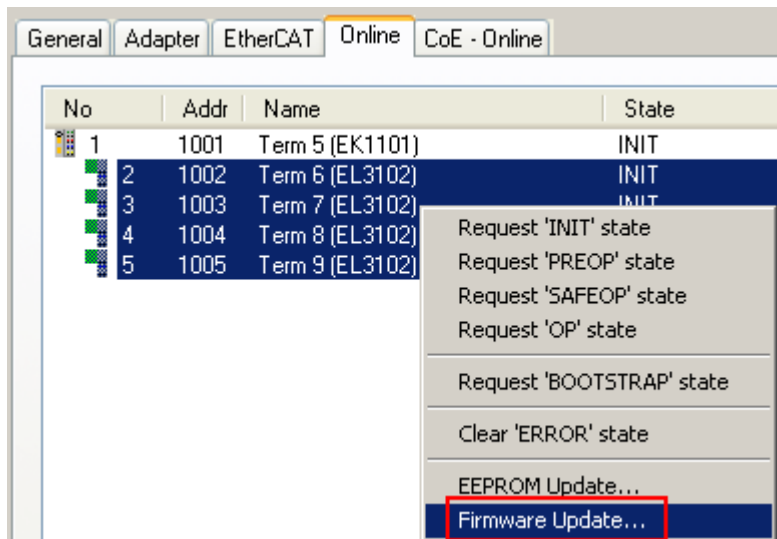


Fig. 96: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.

6.4 Restoring the delivery state

To restore the delivery state (factory settings) for backup objects in ELxxx terminals, the CoE object Restore default parameters, *SubIndex 001* can be selected in the TwinCAT System Manager (Config mode) (see Fig. *Selecting the Restore default parameters PDO*)

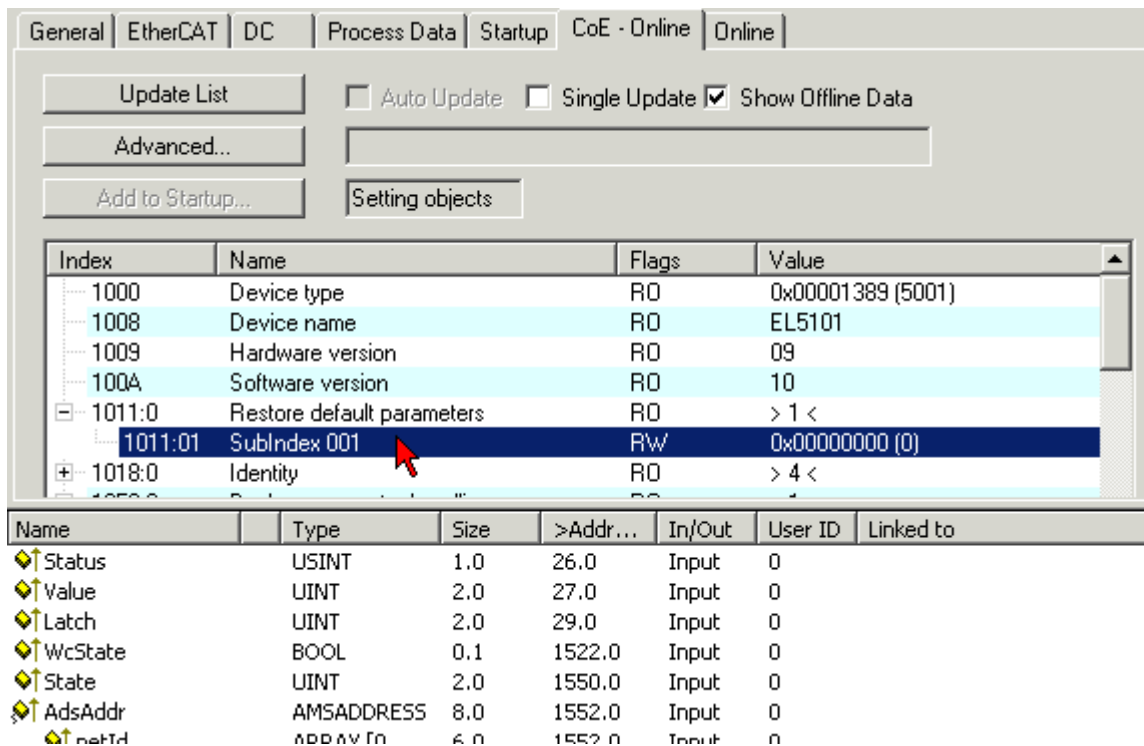


Fig. 97: Selecting the *Restore default parameters* PDO

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* (Fig. *Entering a restore value in the Set Value dialog*). All backup objects are reset to the delivery state.

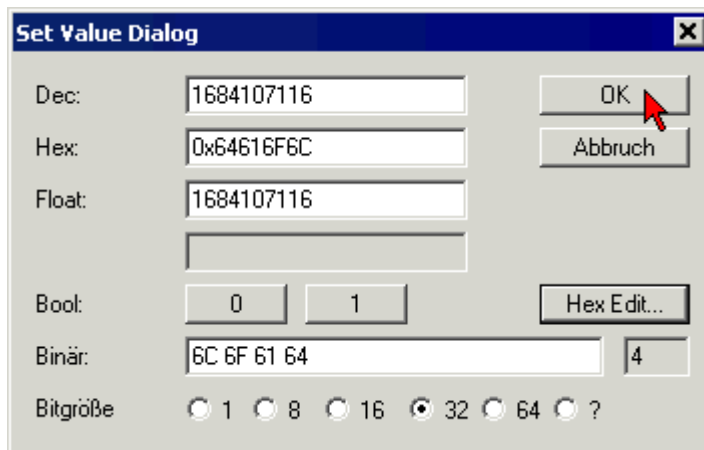


Fig. 98: Entering a restore value in the Set Value dialog

● Alternative restore value

i In some older terminals the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164An incorrect entry for the restore value has no effect.

6.5 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/EL6688

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

