



Documentation

FM33xx-B110-xxxx

12 and 32 channel analog thermocouple input plug

Version: 1.4
Date: 2018-12-04

BECKHOFF

Table of contents

1 Foreword	5
1.1 Notes on the documentation.....	5
1.2 Safety instructions	6
1.3 Documentation issue status	7
1.4 Version identification of EtherCAT devices	8
2 Product overview	12
2.1 FM33xx-B110-xxxx - Introduction.....	12
2.2 FM33xx-B110-xxxx - Technical data	13
2.3 Start	14
3 Basics communication	15
3.1 EtherCAT basics.....	15
3.2 EtherCAT State Machine.....	15
3.3 CoE Interface.....	16
3.4 Distributed Clock	21
4 Connection and diagnosis	22
4.1 Diagnosis-LEDs and connection	22
4.1.1 Diagnosis	22
4.1.2 Connection.....	24
4.2 Cabling	27
4.3 Power cables	29
4.4 UL notice	30
5 Commissioning and basic function principles	31
5.1 Basic function principles	31
5.2 Notices and measurement accuracy	36
5.3 QUICK START.....	38
5.4 TwinCAT Development Environment	46
5.4.1 Installation of the TwinCAT real-time driver.....	46
5.4.2 Notes regarding ESI device description.....	52
5.4.3 OFFLINE configuration creation	56
5.4.4 ONLINE configuration creation.....	61
5.4.5 EtherCAT slave process data settings.....	69
5.4.6 EtherCAT subscriber configuration.....	70
5.4.7 Object description and parameterization	80
5.5 General Notes - EtherCAT Slave Application	86
6 Appendix	94
6.1 EtherCAT AL Status Codes	94
6.2 Firmware compatibility	94
6.3 Firmware Update EL/ES/EM/EPxxxx	95
6.3.1 Device description ESI file/XML.....	96
6.3.2 Firmware explanation	99
6.3.3 Updating controller firmware *.efw.....	100
6.3.4 FPGA firmware *.rbf.....	101
6.3.5 Simultaneous updating of several EtherCAT devices.....	105

6.4	Restoring the delivery state	106
6.5	Support and Service	107

1 Foreword

1.1 Notes on the documentation

Intended audience

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

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.

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

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 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
1.4	<ul style="list-style-type: none">• Update title page• Update chapter <i>Connection and Diagnosis</i>• Correction chapter <i>Power cable</i>• Update structure
1.3	<ul style="list-style-type: none">• Update Technical data
1.2	<ul style="list-style-type: none">• Update Technical data
1.1	<ul style="list-style-type: none">• Update structure
1.0	<ul style="list-style-type: none">• Release version for documentation for FM33xx-B110
0.6	<ul style="list-style-type: none">• Addenda & corrections, preliminary documentation #6 for FM33xx-B110-xxxx
0.5	<ul style="list-style-type: none">• Addenda & corrections, preliminary documentation #5 for FM33xx-B110-xxxx
0.4	<ul style="list-style-type: none">• Addenda & corrections, preliminary documentation #4 for FM33xx-B110-xxxx
0.3	<ul style="list-style-type: none">• Addenda, preliminary documentation #3 for FM33xx-B110-xxxx
0.2	<ul style="list-style-type: none">• Addenda, preliminary documentation #2 for FM33xx-B110-xxxx
0.1	<ul style="list-style-type: none">• Preliminary documentation for FM33xx-B110-xxxx

1.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

Notes

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

Identification number

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

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

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

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

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

Example with

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

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

Syntax: D ww yy x y z u

D - prefix designation

ww - calendar week

yy - year

x - firmware version of the bus PCB

y - hardware version of the bus PCB

z - firmware version of the I/O PCB

u - hardware version of the I/O PCB

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

Unique serial number/ID, ID number

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

See also the further documentation in the area

- IP67: [EtherCAT Box](#)
- Safety: [TwinSafe](#)
- Terminals with factory calibration certificate and other measuring terminals

Examples of markings



Fig. 1: EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01)



Fig. 2: EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number



Fig. 3: CU2016 switch with serial/ batch number

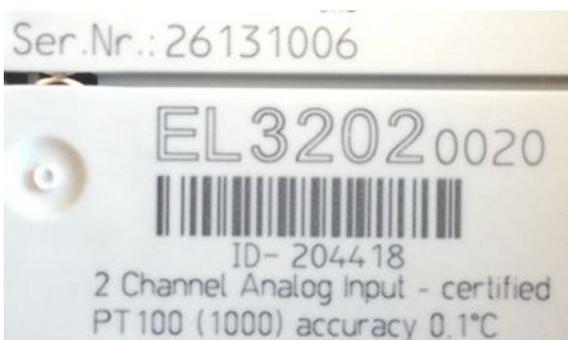


Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

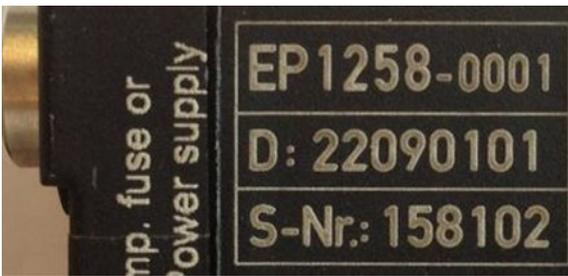


Fig. 5: EP1258-0001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102



Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070



Fig. 7: EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701



Fig. 8: ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201

2 Product overview

2.1 FM33xx-B110-xxxx - Introduction



Fig. 9: *Thermocouple input plug*

12 and 32 channel analog thermocouple input plug with open-circuit recognition (TC plug)

The FM33xx-B110-xxxx thermocouple input plugs allow thermocouples to be connected directly. The TC plug circuit can operate thermocouple sensors using the 2-wire technique. Linearization over the full temperature range is realized with the aid of a microprocessor. The temperature range can be selected freely. The error LEDs indicates a broken wire. Compensation for the cold junction is made through an internal temperature measurement at the terminals.

2.2 FM33xx-B110-xxxx - Technical data

Technical data	FM3312-B110-0010	FM3332-B110-1010
Power supply	24 V _{DC} , -15...+20% (20.4 .. 28.8 V)	
typical operating current	120 mA (24 V)	150 mA (24 V)
max. operating current	150 mA (24 V)	180 mA (24 V)
Power supply input protection Reverse voltage Over voltage	min. -35 V max. 35 V	
Potential separation Thermocouples / Fieldbus	500 V	
Potential separation Thermocouples / Supply Voltage	500 V	
Potential separation Supply Voltage / Fieldbus	100 V	
Operating temperature	0 °C ... + 55 °C	
Storage temperature	-25 °C ... + 85 °C	
Type of protection	IP65	
Vibration	EN 60068-2-6 10 Hz ≤ f ≤ 57 Hz: 0,075 mm amplitude 57 Hz ≤ f ≤ 150 Hz: 1,0 g	
Shock	EN 60068-2-27 35 g @ 6 ms	
Vibration	conforms to EN 60068-2-6 / EN 60068-2-27	
EMC immunity / EMC emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Approvals	CE, cULus [▶ 30]	
Fieldbus/Configuration	EtherCAT/ via TwinCAT System Manager	
Number of inputs	12	32
Type of Input	Differential Input, mV measurement, high impedance grounded no electrical separation	
Connection	2-wire, direct connection	
Wiring length (max.)	30 m	
Thermocouple sensor type	Type K:-100...1370°C, Type J:-100...900°C	
Common Mode input voltage range	-8 V...+8 V	
Common mode rejection	typical 90 dB (independent from conversion time) typical 150 dB (at 50 Hz or 60 Hz Rejection)	
Normal mode rejection	typical 98 dB (at 50 Hz or 60 Hz Rejection)	
Resolution	0.1 °C per Digit	
Total measuring error (temperature)	see chapter "TC plug measurement accuracy [▶ 36]"	
Cold junction compensation	Internal (temperature measurement in the plug)	
Conversation time (per channel)	Adjustable 50 ms (60 Hz Common Mode Rejection) 60 ms (50 Hz Common Mode Rejection)	
Data update time	200 ms (60 Hz), 230 ms (50 Hz)	
Wiring fail indication	Yes (Open Thermocouple)	
Back Voltage Range	>10 V, < -10 V	
Switch off time	3 ms max.	
Input resistance for current limitation	2400 R, typ.	
Input voltage	100 V _{AC} between TC+ and TC- of each channel 240 V _{AC} above Protective Earth 415 V _{AC} to other TC	
Input protection mechanism	Current limitation, Channel disconnect	
Range of Input protection	240 V _{AC/DC}	
Earth connection	The plug must be connected to protective earth with the earth-connector in the plug!	
Dimensions (L x W x H)	approx. 120 mm x 52 mm x 129 mm	
Connector hood	Han24B (Harting)	
Connector to Thermocouples	Han24E	Han64D
Contacts	Hard gold plated	
Coding system	With coding pins in the connector	
Connctor power supply	M8	
Connector fieldbus	M12	
Weight	approx. 500 g	

2.3 Start

For commissioning:

- mount the FM33xx-B110-xxxx as explained in the chapter "[Connection and diagnosis \[▶ 22\]](#)"
- configure the FM33xx-B110-xxxx in TwinCAT as explained in the chapter "[Commissioning \[▶ 31\]](#)"

3 Basics communication

3.1 EtherCAT basics

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

3.2 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.

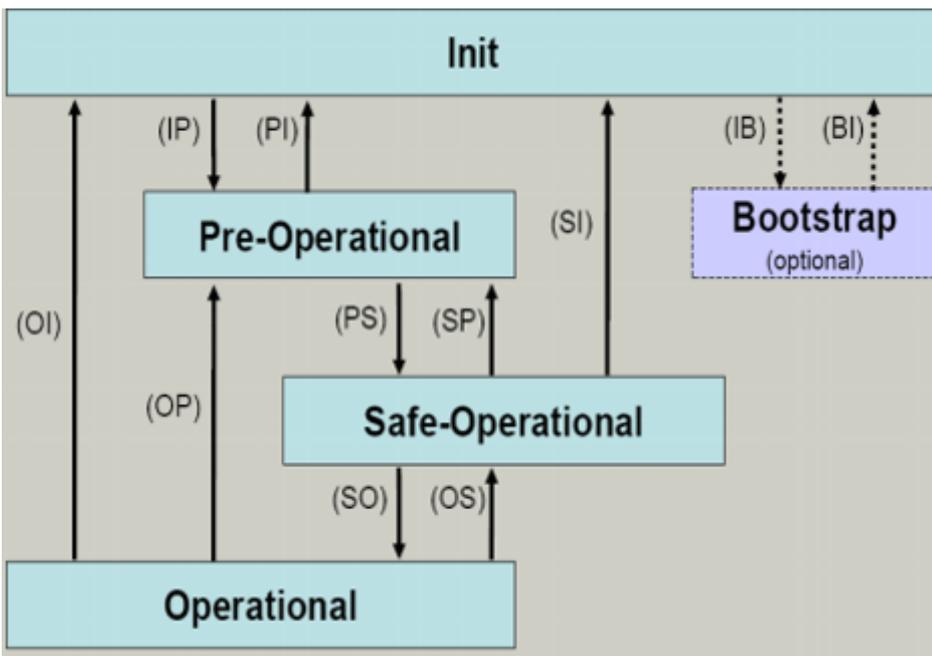


Fig. 10: States of the EtherCAT State Machine

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

The default set watchdog 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.

3.3 CoE Interface

General description

The CoE interface (CANopen 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 2 levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

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

A parameter localized in this way is normally written as 0x8010:07, with preceding "x" 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: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

Availability

i 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:

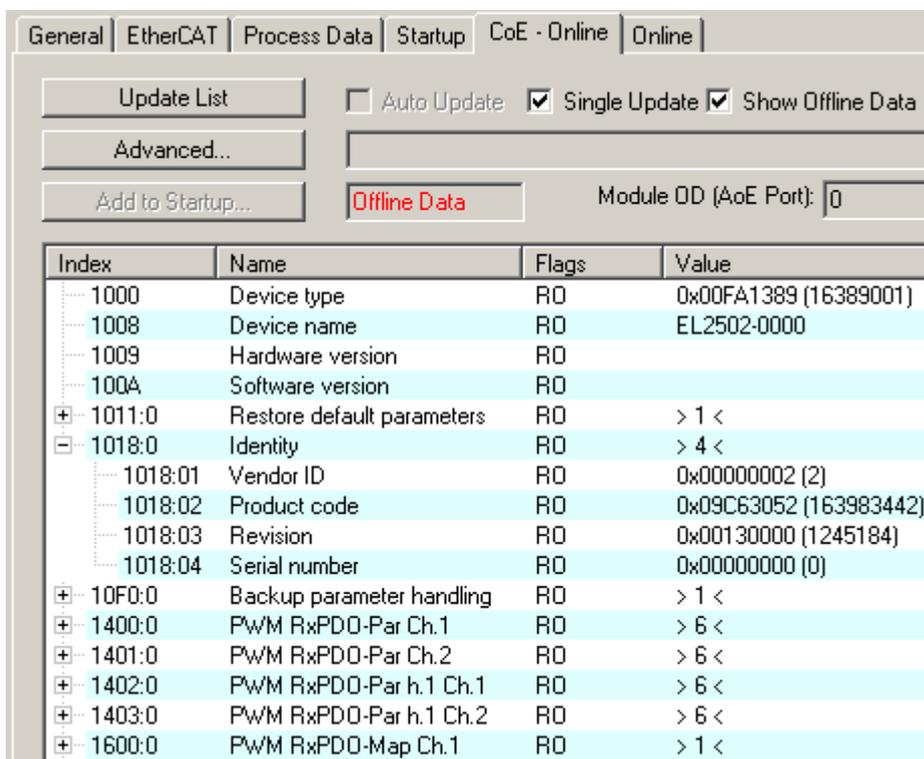


Fig. 11: "CoE Online " tab

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

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 parameterised 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.

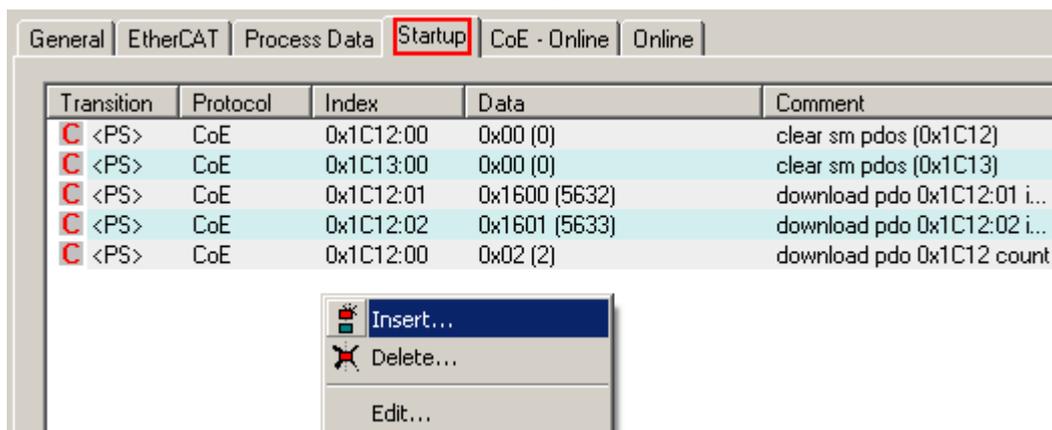


Fig. 12: Startup list in the TwinCAT System Manager

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.

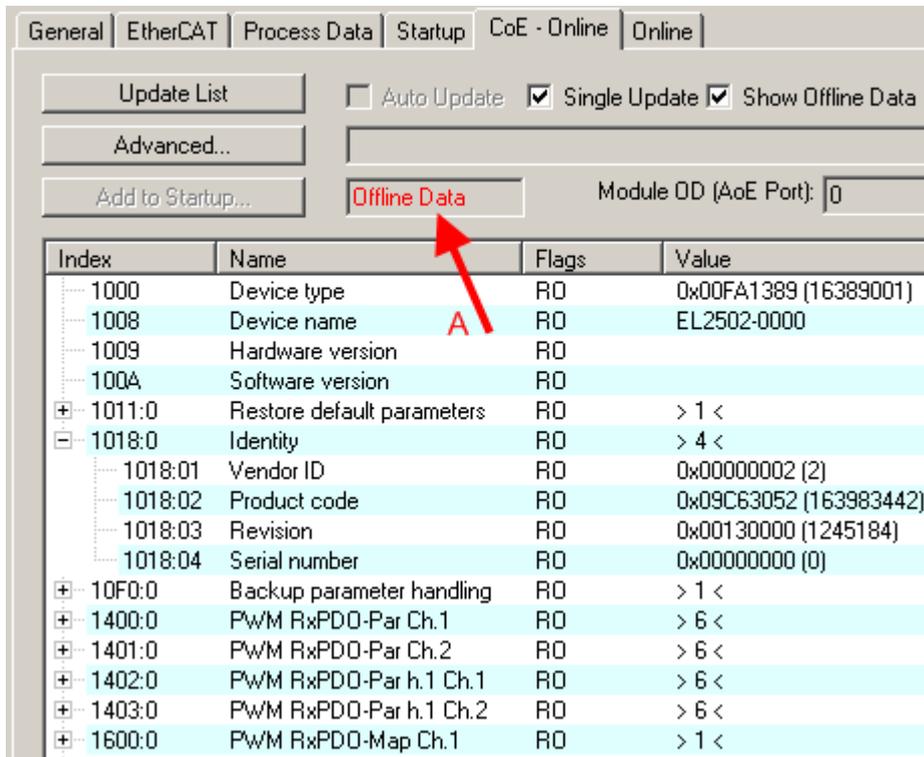


Fig. 13: Offline list

- 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.

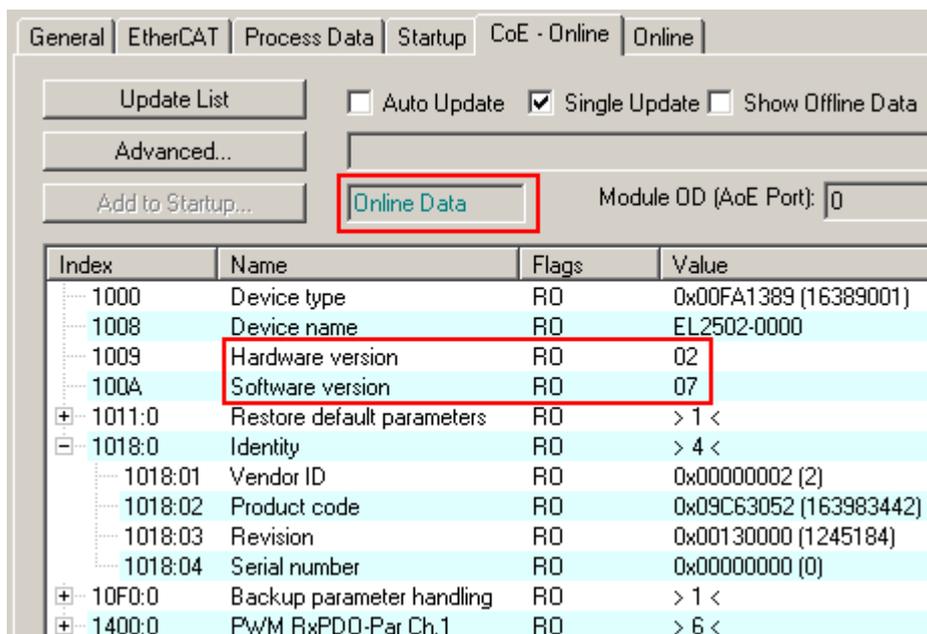


Fig. 14: Online list

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 4 logical channels and therefore 4 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.

3.4 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 Connection and diagnosis

4.1 Diagnosis-LEDs and connection

4.1.1 Diagnosis

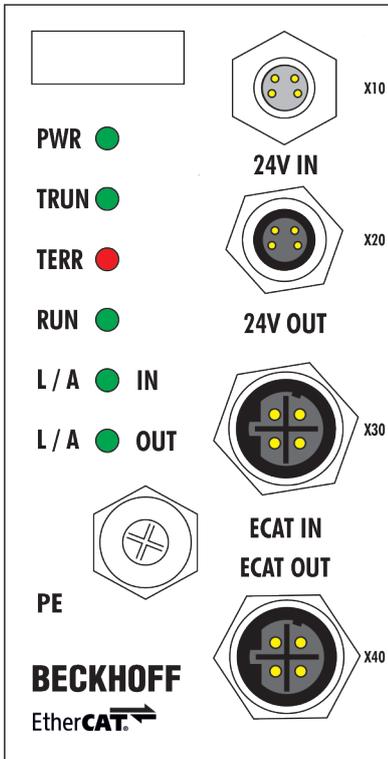


Fig. 15: Top view TC Modul

LED	Color	Meaning	
PWR	green	off	24 V _{DC} supply voltage is not present
		on	24 V _{DC} supply voltage is present
TRUN	green	off	Thermocouple inputs are not ready for operation
		on	Error-free operation condition of the thermocouple inputs
TERR	rot	off	Error-free operation condition of the thermocouple inputs
		flashing	displays an error code
RUN	green	This LED indicates the state of the EtherCAT State Machine state:	
		off	INIT = Initialization of the terminal
		even flashing	PREOP = Setting for mailbox communication and variant standard settings
		single flash	SAFEOP = Channel checking of the Sync Manager and the Distributed Clocks. Outputs stay in safe operation mode.
		on	OP = Normal operation mode, mailbox- and process data communication possible
fast flashing	BOOTSTRAP = Function for firmware updates of the terminal		
L/A IN	green	Link/Activity LED EtherCAT input	
		off	no connection with the fieldbus
		on	Connection with fieldbus established
		flashing	Data exchange via the fieldbus
L/A OUT	green	Link/Activity LED EtherCAT output	
		off	no connection with the following fieldbus client
		on	connection with the following fieldbus client established
		flashing	Data exchange with the following fieldbus client

LED error codes

Blinking codes

Blinking code	Meaning
fast flashing	Start of the error code
1. slow sequence	Error code
2. slow sequence	Error code argument

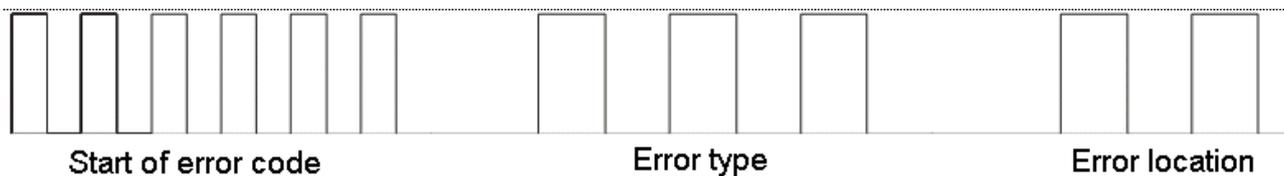


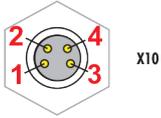
Fig. 16: *Blinking code of the "TERR"-LED*

Error codes of thermocouple inputs

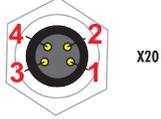
Error code	Error code argument	Comment
1 pulse	0	EEPROM checksum error
1 pulse	1	Overflow inline code buffer
1 pulse	2	Unknown data type
2 pulses	-	Programmed configuration
3 pulses	Faulty channel number. The first faulty channel is indicated	Backvoltage
4 pulses	Faulty channel number. The first faulty channel is indicated	Open circuit

4.1.2 Connection

Fieldbus/ supply voltage

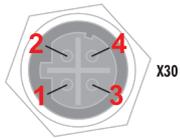


24V IN



24V OUT

Fig. 17: Connection supply voltage



ECAT IN

ECAT OUT



Fig. 18: Connection fieldbus

Conn. point	Designation	Comment
X10	24 V IN	1: Control voltage $U_s +24 V_{DC}$
X20	24 V OUT	2: Auxiliary voltage $U_p +24 V_{DC}$ 3: GNDs 4: GNDp
X30	ECAT IN	1: Tx + (Transmit Data +)
X40	ECAT OUT	2: Tx - (Transmit Data -) 3: Rx + (Receive Data +) 4: Rx - (Receive Data -)
-	PE	Screw connection - protective earth

Pin assignment for the 24-pin connection strip (FM3312-B110-0010)



Fig. 19: Connection strip FM3312-B110-0010

Pin	Meaning
1 ... 12	TC1+ ... TC12+
13 ... 24	TC1- ... TC12-

Pin assignment for the 64-pin connection strip (FM3332-B110-1010)



Fig. 20: Connection strip FM3332-B110-1010

Pin	Meaning
1 ... 16 (A..)	TC1+ ... TC16+
1 ... 16 (B..)	TC1- ... TC16-
1 ... 16 (C..)	TC17+ ... TC32+
1 ... 16 (D..)	TC17- ... TC32-

PE connection of earthed/floating thermocouples

The PE connection is responsible for the potential equalization of grounded and non-grounded thermocouples, and for thermocouples grounded at different potentials (see thermocouple measurement). The back voltage protection diverts mains voltages via the PE connection. This means that it is essential for proper earthing of the module to be guaranteed. When the module is being fitted, the PE connection should

be established first. Connection of the module to the thermocouples via the connecting strip should be performed subsequently. In this way the risk of back voltage being present at the thermocouple inputs during assembly is avoided.

 **WARNING**

Risk of injury due to electric shock!

If the earthing is inadequate the user is exposed to hazardous voltages on the housing! When the module is being dismantled, the module should, for this reason, first be disconnected from the thermocouples before the PE connection is removed.

4.2 Cabling

A list of EtherCAT cables, power cables, sensor cables, Ethernet/EtherCAT connectors and field-configurable connectors can be found under the following link: https://beckhoff.de/english/fieldbus_box/ethercat_box_accessories_overview.htm?id=25525466903389

The corresponding data sheets can be found under the following link:

https://beckhoff.de/english/ethercat-box/ethercat_box_cables.htm?id=690338951657421

EtherCAT cables



Fig. 21: ZK1090-3131-0xxx

For connecting EtherCAT devices, only use shielded Ethernet cables with a minimum specification of **category 5 (CAT5) according to EN 50173 or ISO/IEC 11801**.

● Wiring recommendations

i Detailed recommendations for EtherCAT cabling can be found in the documentation "Design recommendations for EtherCAT/Ethernet infrastructure", which is available for download from www.beckhoff.de.

EtherCAT uses four cable wires for signal transmission.

Due to automatic cable detection (auto-crossing) symmetric (1:1) or cross-over cables can be used between EtherCAT devices from Beckhoff.

Power cable



Fig. 22: ZK2020-3132-0xxx

Sensor cables



Fig. 23: Selection of Beckhoff sensor cables

4.3 Power cables

Ordering data

Order designation	Power cable	Screw-in connector	Contacts	Cross-section	Length
ZK2020-3200-0020	Straight socket, open end	M8	4-pin	0.34 mm ²	2.00 m
ZK2020-3200-0050					5.00 m
ZK2020-3200-0100					10.00 m
ZK2020-3400-0020	Angled socket, open end				2.00 m
ZK2020-3400-0050					5.00 m
ZK2020-3400-0100					10.00 m
ZK2020-3132-0001	Straight socket, straight socket				0.15 m
ZK2020-3132-0005					0.50 m
ZK2020-3132-0010					1.00 m
ZK2020-3132-0020					2.00 m
ZK2020-3132-0050					5.00 m
ZK2020-3334-0001	Angled socket, angled socket				0.15 m
ZK2020-3334-0005					0.50 m
ZK2020-3334-0010					1.00 m
ZK2020-3334-0020					2.00 m
ZK2020-3334-0050		5.00 m			

Further available power cables may be found in the Beckhoff catalog or on our internet pages (<http://www.beckhoff.com>).

Technical data

Technical data	
Rated voltage according to IEC61076-2-101	30 V _{DC}
Contamination level according to IEC 60 664-1	3/2
Insulation resistance IEC 60 512-2	>10 ⁹ Ω
Current carrying capacity according to IEC 60512-3	4 A
Volume resistance according to IEC 60512-2	< 5 mΩ
Protection class according to IEC 60529	IP65/66/67, when screwed together
Ambient temperature	-30°C to +80°C

4.4 UL notice

	<p>Application Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p>Examination 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>
	<p>For devices with Ethernet connectors Not for connection to telecommunication circuits.</p>

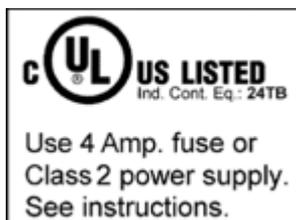
Basic principles

Two UL certificates are met in the Beckhoff EtherCAT product range, depending upon the components:

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



2. UL certification according to UL508 with limited power consumption. The current consumed by the device is limited to a max. possible current consumption of 4 A. Devices with this kind of certification are marked by this sign:



Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V_{DC} must be limited accordingly by means of supply

- from an isolated source protected by a fuse of max. 4 A (according to UL248) or
- from a voltage supply complying with *NEC class 2*.
A voltage source complying with *NEC class 2* may not be connected in series or parallel with another *NEC class 2* compliant voltage supply!

These requirements apply to the supply of all EtherCAT bus couplers, power adaptor terminals, Bus Terminals and their power contacts.

5 Commissioning and basic function principles

5.1 Basic function principles

Technology

Requirements

The FM33xx-B110-xxxx thermocouple terminals can evaluate thermocouples of the types J und K. The characteristic curves are linearized and the reference temperature determined directly within the terminal. Temperatures are output in 1/10°C. The terminal is fully configurable via the Bus Coupler or the control system. Linearization of the characteristic curve and determination and calculation of the reference temperature (temperature at the terminal connection contacts) can be switched off.

Measuring principle of a thermocouple

Thermocouples can be classified as active transducers. They exploit the thermo-electric effect (Seebeck, Peltier, Thomson). Where two electrical conductors of different materials (e.g. iron and constantan) make contact, charge is transferred across the contact surface. A contact potential develops, and is strongly dependent on temperature. This thermovoltage is a function of both the measuring temperature T_m and the reference temperature T_r at the connection contacts of the thermocouple (see fig. "Principle of the thermocouple").

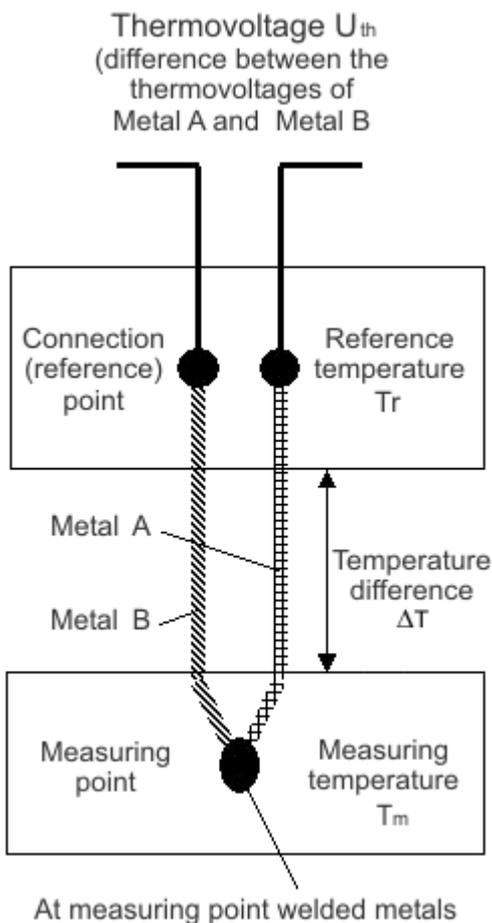


Fig. 24: Principle of the thermocouple

Thermocouple measurement

A type-K thermocouple consists of a junction of a nickel-chrome alloy and nickel, where k_{NiCr} and k_{Ni} represent the thermoelectric coefficients of the metals nickel-chrome and nickel respectively. By adapting the equation according to T_m , the sought-after temperature can be calculated from the voltage measured across the thermocouple. Based on the difference to the reference temperature, the temperature at the measurement point can be determined to an accuracy of better than one tenth of a Kelvin.

● Sensor circuit

i A modification of the sensor circuit with additional devices such as change over switches or multiplexer decreases the measure accuracy. It is strongly discouraged to implement such modifications.

As mentioned, the thermocouples consist of two different metal alloys. Voltages proportional to the temperature arise where the alloys are in contact. In the case of a type K thermocouple these voltages are typically in the range of $50 \mu\text{V}/^\circ\text{C}$. This means that the inputs for the thermocouples must be capable of measuring voltages in the μV and mV range. In addition to the signal measurement itself, the module performs functions related to error detection, interference suppression and protection from external voltages.

Input circuits

The module can contain up to 32 thermocouple inputs. Electrical isolation between the channels could not be implemented, for reasons of both cost and space. In order to exclude the possibility of mutual interference, every input has a high-impedance differential amplifier. This rejects interfering voltages that can be coupled into the thermocouple circuit. Voltage differences between individual thermocouples are also suppressed without them having any effect on one another.

Broken lead detection

Broken leads are detected as a diagnosis of the state of the thermocouple. The break is recognized through the infinite resistance of the broken thermocouple. Broken leads are detected in both grounded and ungrounded thermocouples.

Cold junction (CJ) compensation

When a thermocouple is used for temperature measurement, a voltage is thermally generated at the other end of the connection as well as at the location of the measuring junction. On both the 12-pin and the 64-pin contact strips the socket contact at the module is connected to the thermal wire. Other requirements mean that it is not possible to measure directly at the socket contacts (plugging in without contact losses, and ability to withstand mains voltage). There is another cold junction on the plug side, opposite the cold junction on the socket side. Good temperature equalization between the socket and plug contacts, and thermal insulation of the socket and plug contacts, ensure that the thermal voltages on the two sides are equalized. The voltages thus cancel each other out. To achieve this structure, the connection in the module between the plug contact and the circuit board is made with the necessary thermal material in each case (J or K type). The resulting cold junction is then on the circuit board for the module electronics. At this point it is possible to perform a simple yet effective measurement of temperature in order to implement the cold junction compensation.

Back Voltage Protection

The back voltage protection guards against external voltages applied through the thermocouple inputs. Voltages of up to $230 V_{\text{AC}}$ are tolerated, or are withstood without damage to the module. Those thermocouple inputs that are not affected remain functionally operative, or are only affected for a short time. The module switches measurements from the affected channel off and disconnects the voltage, so that the unwanted voltage can remain connected to the thermocouple input for a long period. If it is possible to manage without that particular thermocouple channel, it is not necessary to rectify the fault immediately.

How Back Voltage Protection operates

The thermocouple inputs are protected within the module against voltages that are outside the permitted signal range. The permitted range is about $-10\text{V} \dots +10\text{V}$. If voltages outside this range appear, the current is limited by an internal resistance and is diverted to a protective circuit. The current-limiting function of the input circuit protects the thermocouple wires from excessive short-circuit currents, so that they are not destroyed. A protective circuit is responsible for a group of at most eight thermocouple inputs. The central

protective circuit detects the excess voltage and switches off all the thermocouple inputs in the group. This process affects the current measurements being taken by the channels in this group, and those measurements will have to be made again. Monitoring and switching groups off means that the channel that is directly affected cannot be identified. The faulty thermocouple input is identified by switching on the channels that have been switched off again one by one. As soon as it is seen that the protective circuit for these thermocouple inputs is triggered again after a channel is switched on, this channel is recognized as faulty, and remains switched off. When all the channels in the group have been tested, and are either operating again or have been switched off because they are faulty channels, it is possible to continue with the normal measuring process. The fault in the channels that have been switched off is checked by switching on again after a configurable time, so that the faulty channels can be automatically brought back into operation.

Connecting strips for thermocouples module FM3332-B110-1010

The thermocouples are connected to the 24-pin or 64-pin connection strips. The signal assignment and polarity must be correctly observed. Cold junction compensation is implemented on the circuit board within the module. The connection between the module electronics and the connector pins is made with the corresponding thermal material. The use of thermocouples other than those listed in the documentation is not permitted. Good thermal conduction between the module's plug and socket means that no significant thermal voltage is created by the plugged connections. Thermal insulation on the socket side of the connection should be ensured in order to avoid a large temperature error. Draughts and unevenly heated areas in particular are to be avoided.

Overview of suitable thermocouples

The following thermocouples are suitable for temperature measurements with the FM33xx-B110-xxxx (see following table)

Type (according to EN60584-1)	Element	Implemented temperature range	Colour coding (sheath - plus pole - minus pole)
J	Fe-CuNi	-100°C to 1200°C	black - black - white
K	NiCr-Ni	-200°C to 1370°C	green - green - white

● **Max. cable length**

i The cable length to the module must not exceed a length of 10 m without further protective measures. A suitable surge protection is mandatory in this case.

Use of FM33xx-B110-xxxx in the TwinCAT System Manager

● **Description of FM3332-B110-1010 module**

i The following description refers to the 32 channel module FM3332-B110-1010. It also can be used for the FM3312-B110-0010 module, because the only difference is the number of the channels.

In the full configuration (all possible PDOs activated, see PDO assignment), the FM3332-B110-1010, for example, offers the following process data for use:

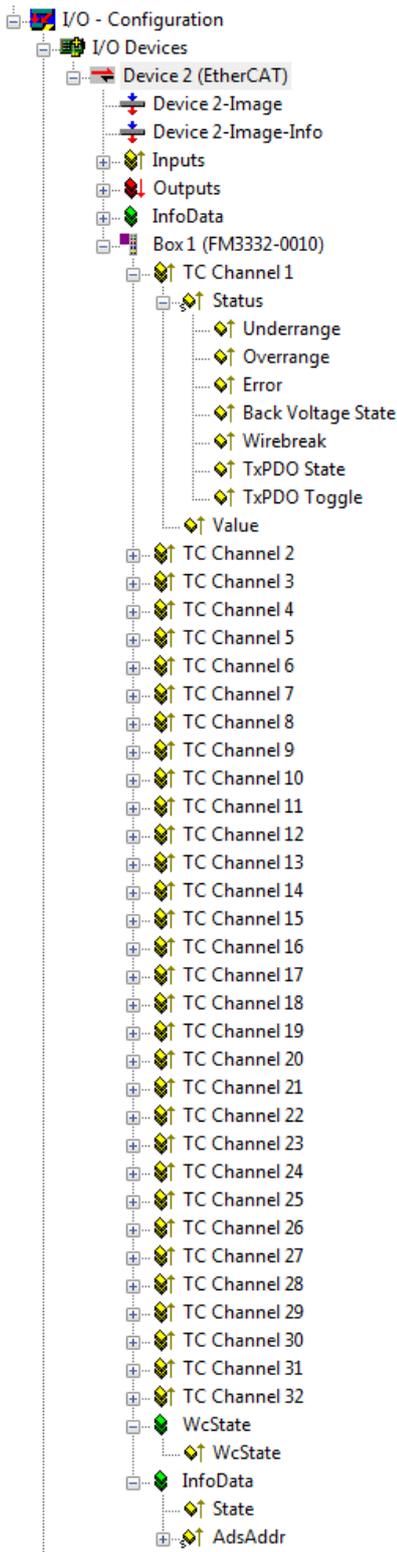


Fig. 25: Process data FM3332-B110-1010

In the case of the FM3332-B110-1010, 32 sets of process data are available, one for each measurement channel.

- **Underrange:** Measurement is below range
- **Overrange:** Range of measurement exceeded ("Cable break" together with "Error")
- **Error:** The error bit is set if the process data is invalid (cable break, over-range, under-range)
- **Back Voltage State:** A backvoltage was detected. The module tries to reset the backvoltage every 10 seconds.

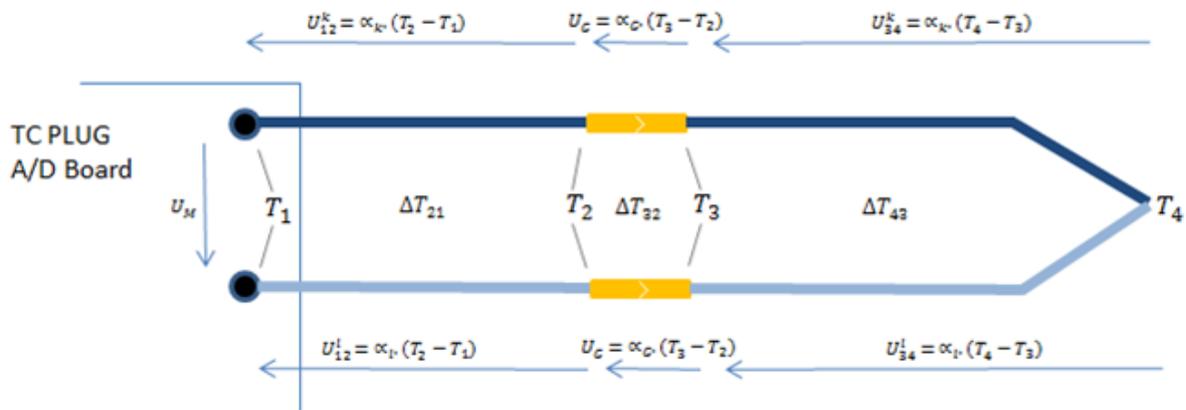
- **Wirebreak:** Detection of a wire break
- **TxPDO State:** Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).
- **TxPDO Toggle:** The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated. This allows the currently required conversion time to be derived

For detailed information on settings and operating modes, please read the chapter "Process data and operating modes".

5.2 Notices and measurement accuracy

FM33xx operating principle

The picture below shows the thermocouple circuit starting from inside the FM33xx electronic board out to the point of temperature measurement T4:



$$U_M = U_{12}^k + U_G + U_{34}^k - U_{34}^l - U_G - U_{12}^l$$

Fig. 26: Structure FM33xx

On the left side there is the A/D converter board inside the plug, then we have two thermo element wires going to the male connector (the connector consisting of male and female part is assumed as yellow/golden rectangular block in the drawing), then the two wires exit the female connector part of the mating TC plug and join at the measurement point. The indexes “l” and “k” are used to mark the different materials of the thermoelement wires.

The temperatures

- T1 is the temperature at the A/D-converter board in the TC plug,
- T2 is the temperature at the point where the thermo element wires enter the male part of the gold plated connector,
- T3 is the temperature at the point where the thermo element wire exits the female gold plated connector
- T4 is the target point where the actual temperature is measured

The thermocouple principle is based on the Seebeck-effect: if one heats one end of a conductor and cools the other end then one can observe a small voltage difference between the two ends. The voltage is assumed proportional to the temperature difference with the Seebeck-coefficient in-between:

$$U = \alpha \cdot \Delta T$$

Taking this simple equation, and assuming two different materials (Index “k” and index “l”) and the goldplated mating connector (male+female, Index “G”) in-between and put the voltage arrows across the circuit, the Kirchhoff voltage law across a closed loop results in:

$$U_M = U_{12}^k + U_G + U_{34}^k - U_{34}^l - U_G - U_{12}^l$$

Each individual voltage is a function of its individual temperature difference at the end-points of the respective conductor following the Seebeck equation.

U_M is ultimately the voltage measured by the AD-converter in the TC plug. This voltage is translated to a corresponding temperature T_4 by using internal conversion tables specific to the thermocouple material (e.g. type J is FE / CU-NI).

TC plug measurement accuracy

In the factory, the FM33xx-B110 modules are calibrated to an accuracy of

$\pm 0.5 \text{ }^\circ\text{C}$ at $546.19 \text{ }^\circ\text{C}$ with a max. temperature difference $\Delta T \leq 0.5 \text{ }^\circ\text{C}$ across all thermocouple channels

However, when measuring target temperature in real circumstances, there are several factors which will introduce an error. Here is a short explanation of the most important ones, together with their corresponding error range.

- a.) Translation table inaccuracy at different temperatures $\pm 0.1 \text{ }^\circ\text{C}$ (internal software algorithm based on segmented temp/voltage table)
- b.) Self-heating or external heat to plug: $\pm 0.4 \text{ }^\circ\text{C}$ (for ambient temperature $+10^\circ\text{C}$ to $+55^\circ\text{C}$)
- c.) Electronic ageing / burn-in $\pm 0.5 \text{ }^\circ\text{C}$ (based on statistic experience)
- d.) Temperature drop in plug mating section $\pm 0.5 \text{ }^\circ\text{C}$ (typical, but depends on mounting direction, external heat/cooling in vicinity, ...)
- e.) Quality of thermocouple wires $\pm 2.5 \text{ }^\circ\text{C}$ (or $\pm 1.5 \text{ }^\circ\text{C}$, if higher quality tc material is used)

As shown, a large error budget is introduced by the quality of thermocouple wire material.

Here is a deeper explanation for d):

An installation of TC plug at the machine consists of the plug himself and a receptacle part, typically fastened to the body of the machine. The cold junction compensation for the thermo-measurement is implemented on the controller board in the vicinity of the where the internal tc wiring is soldered to the analog printed circuit board. In theory we would need to measure the cold junction temperature at point T_3 , but this is not possible because this is across the mating section of the installation.

In the above equation (fig. 1), the voltage U_G disappears, no matter of its magnitude and this is one potential source for inaccurate measurement. If we would assume $T_3 = T_2$, then we have no issue, measurement is correct. If T_3 is different of T_2 , the associated Seebeck-voltage U_G will still disappear in the closed loop voltage equation and the delta temperature T_3-T_2 will be a direct constant fault value in the measurement of T_4 .

As a consequence, U_M is a function of ΔT_{21} and ΔT_{43} , but is not a function of ΔT_{32} . The temperature difference between T_3 and T_2 influences the measurement: it simply needs to be added to what the plug thinks it measures as a final temperature. The assumption is, that the temperature difference between T_2 and T_3 is not very high, because the metal parts of male and female connector will go into a steady and balanced thermal state with minimum temperature difference. This should be taken into account when choosing where to install the plug and it's receptacle.

5.3 QUICK START

No special measures are required for the initial commissioning of the FM33xx-B110-xxxx.

1. Connection

Install the FM33xx-B110-xxxx module as described in section connection and diagnosis

Connect 1 to 12 or 1 to 32 thermocouples according to the pin assignment.

2. Configuration

Create a configuration in the TwinCAT System Manager by manually inserting the module or scanning it online. Refer to installation chapter TwinCAT 2.x regarding this.

i EtherCAT XML Device Description

If the XML description of the FM33xx-B110-xxxx is not available in your system you can request the latest XML file from Beckhoff and install it according to the installation instructions.

Sample configuration created with online scan

1. Start the TwinCAT System Manager and click the right mouse button on "I/O Devices -> "Scan Devices". Confirm the following Pop up windows with "OK".

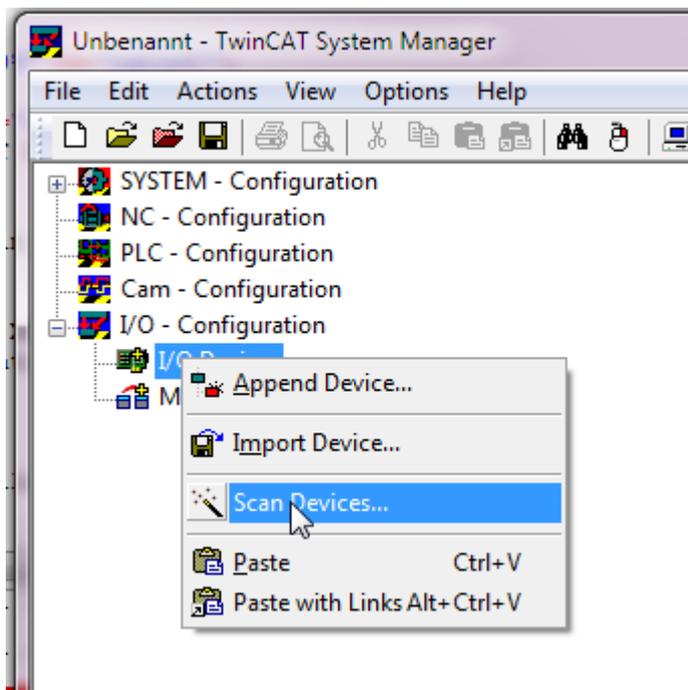


Fig. 27: Dialog "Scan devices"

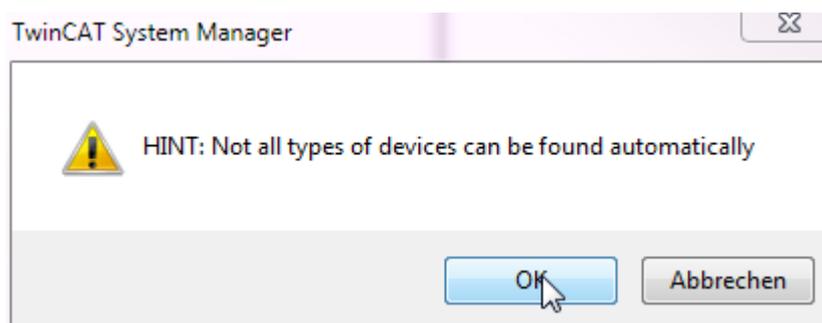


Fig. 28: Information

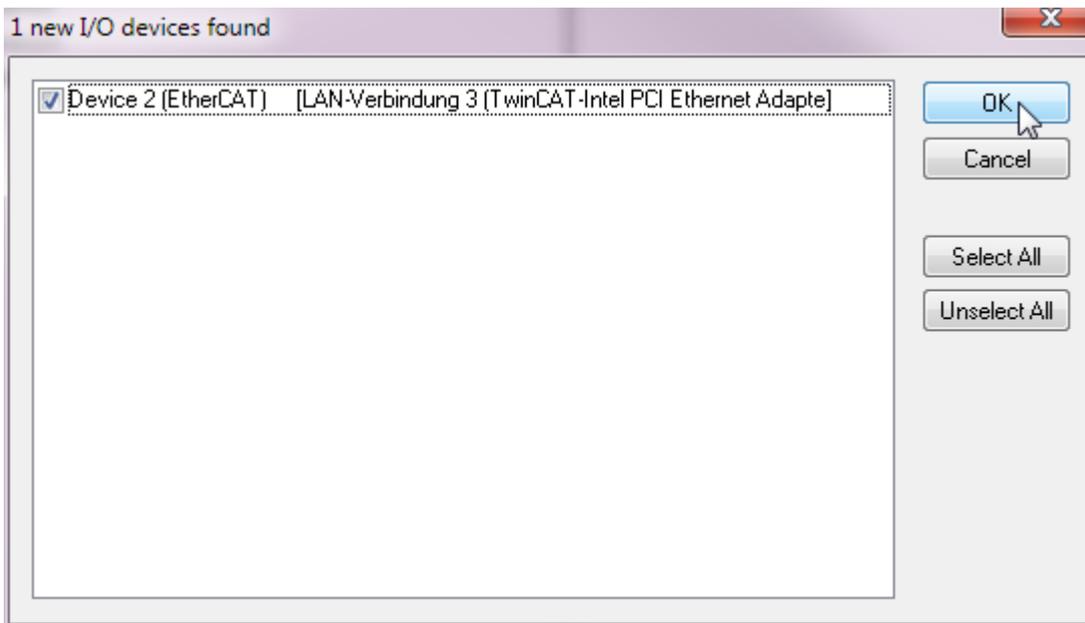


Fig. 29: Confirmation Device "EtherCAT"

2. Scan for new slaves and activate "Free Run"

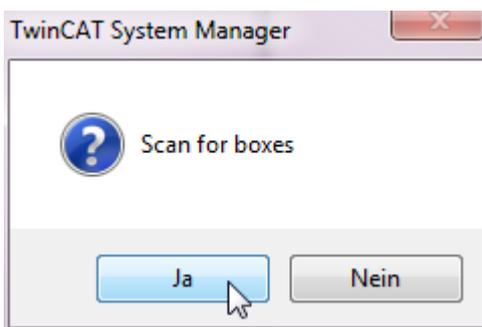


Fig. 30: Dialog "Scan for boxes"

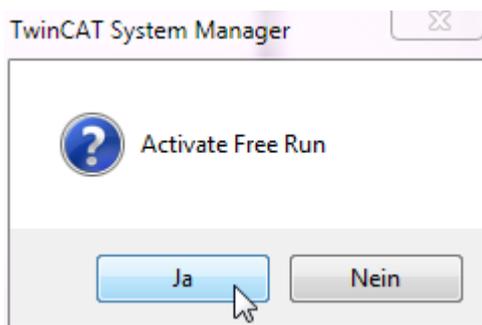


Fig. 31: Dialog "Activate Free Run"

3. The FM33xx-Modul appears in the system configuration now. Click on the "+" to see all active input channels.

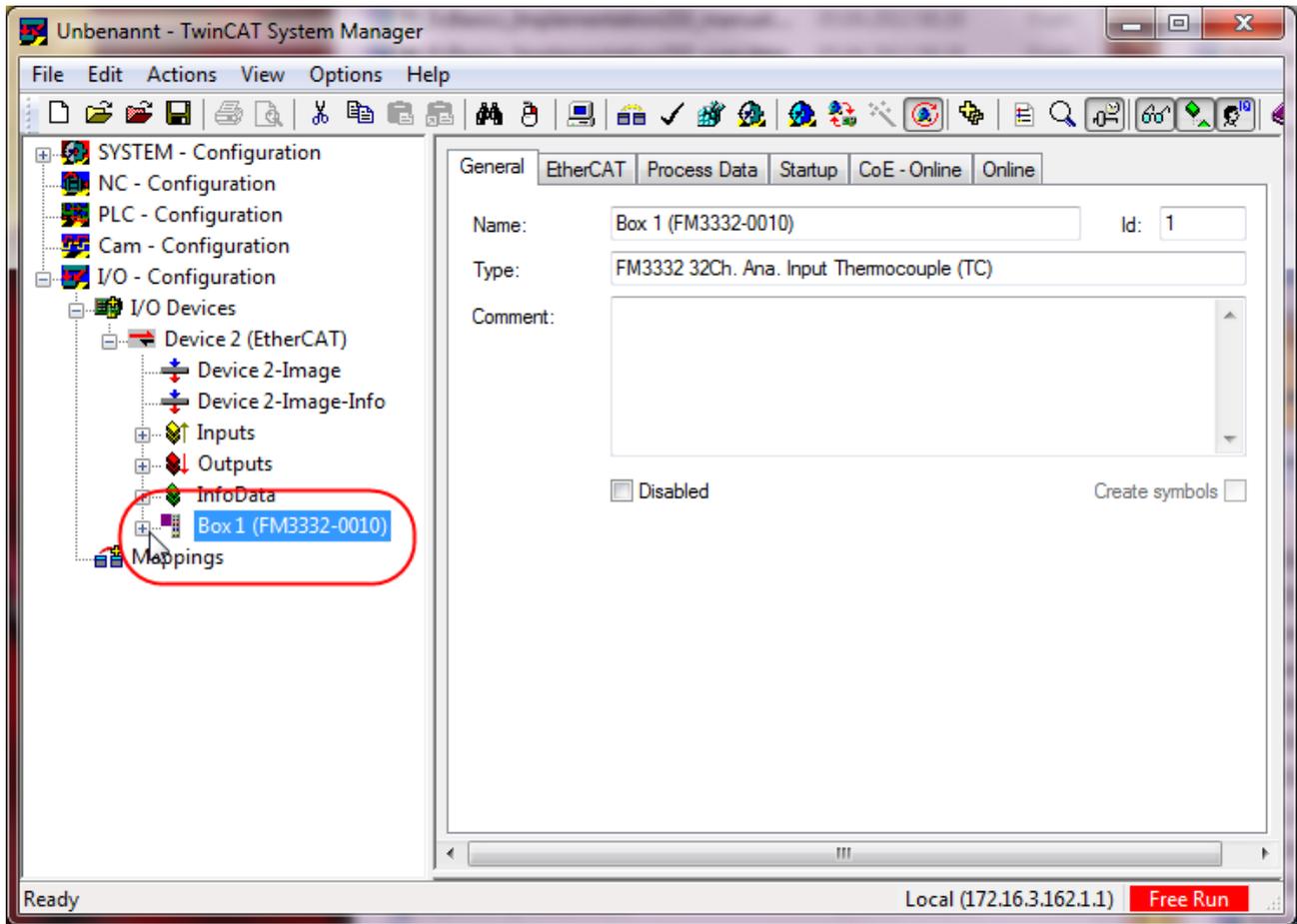
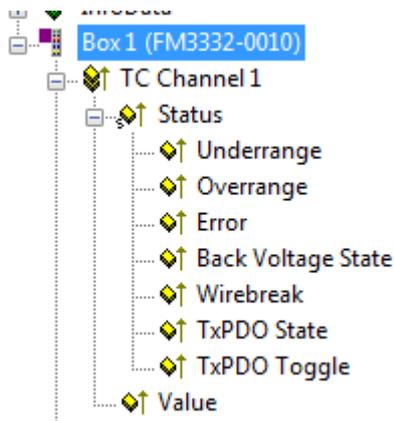


Fig. 32: Box in TwinCAT tree

4. It is possible to link up to 12 or 32 process datasets. All active process datasets are shown in the system configuration.

One process dataset consists of the variables below.



3. Online operation

Activate the EtherCAT master and start the terminal in OP state.

In the input variables the FM33xx-B110-xxxx must deliver State=OP (8) and WC=0.

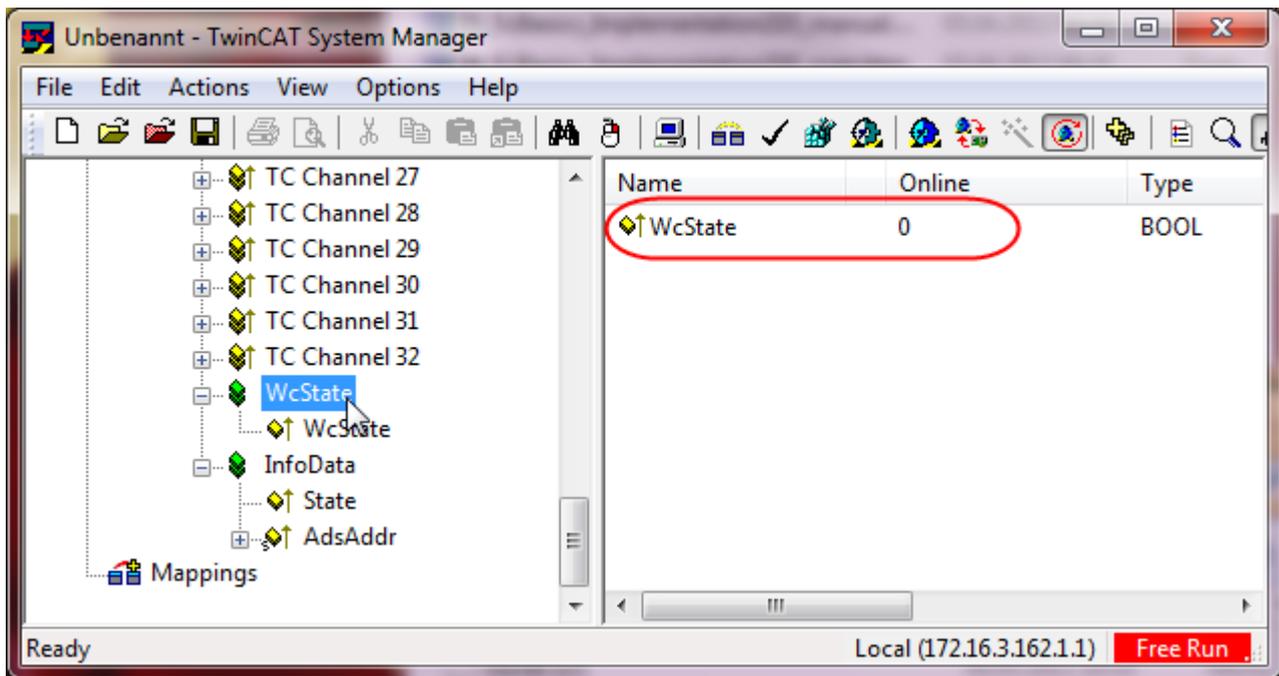


Fig. 33: WcState

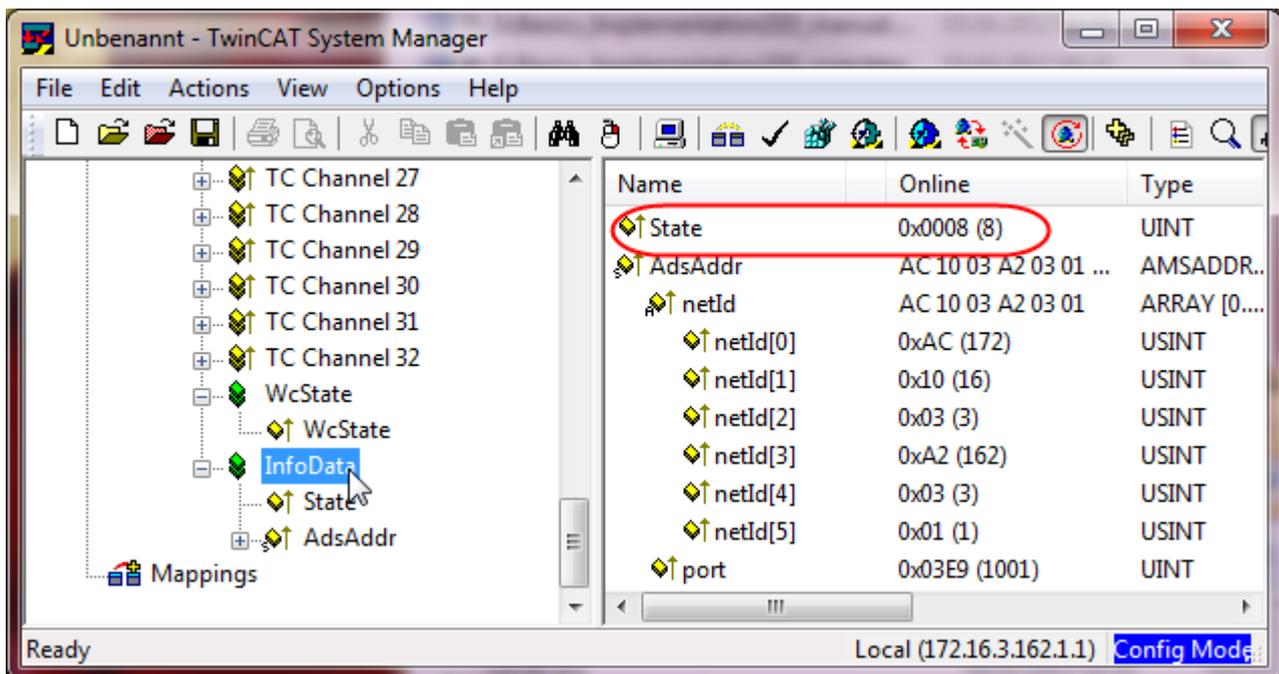


Fig. 34: InfoData

4. Delivery state

The module behaves as follows in the delivery state or after scanning in the TwinCAT System Manager:

- 12 or 32 channels active
- Internal coldjunction compensation
- Filter activated
- TC element Type J

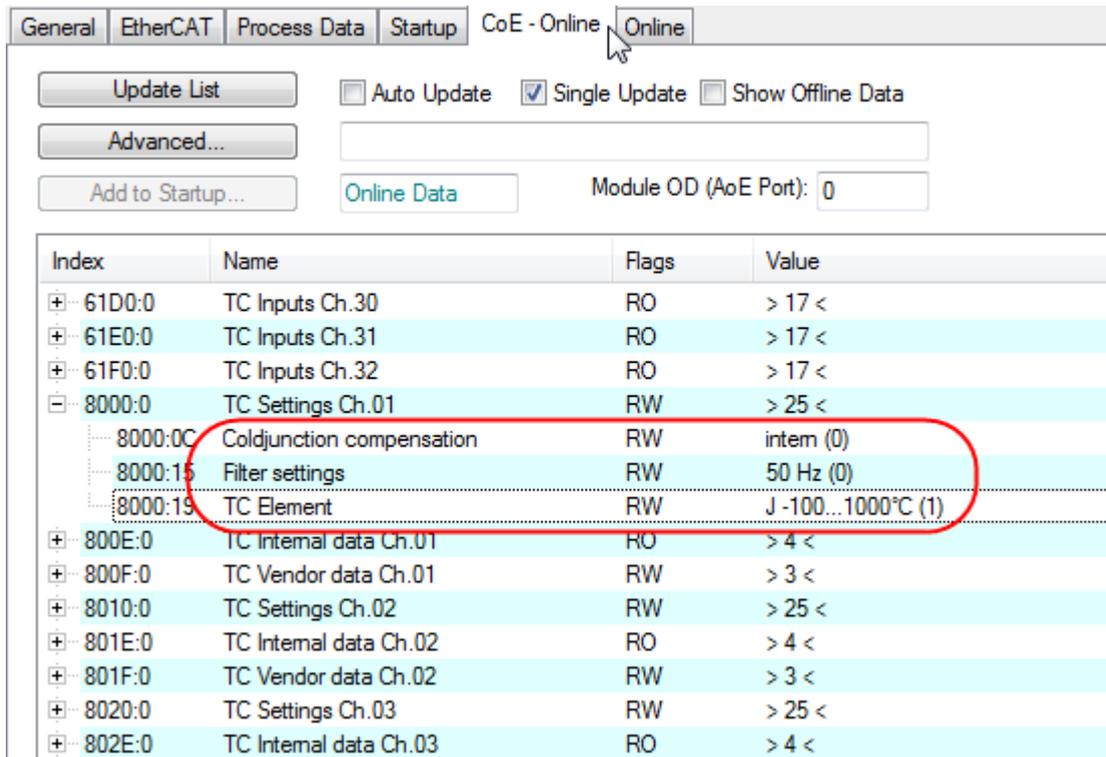


Fig. 35: CoE object 8000

This results in a conversion

5. Setting parameters and process data

Process data

Via the PDO selection (click on the "Process Data" tab (A)) 1 to 12 or 1 to 32 channels can be selected or deselected as PDOs to be cyclically transmitted (deselection of channel 2 via PDO 0x1A01 (B)). The deactivation of a channel reduces the size of the process image (in this example channel 2 (C))

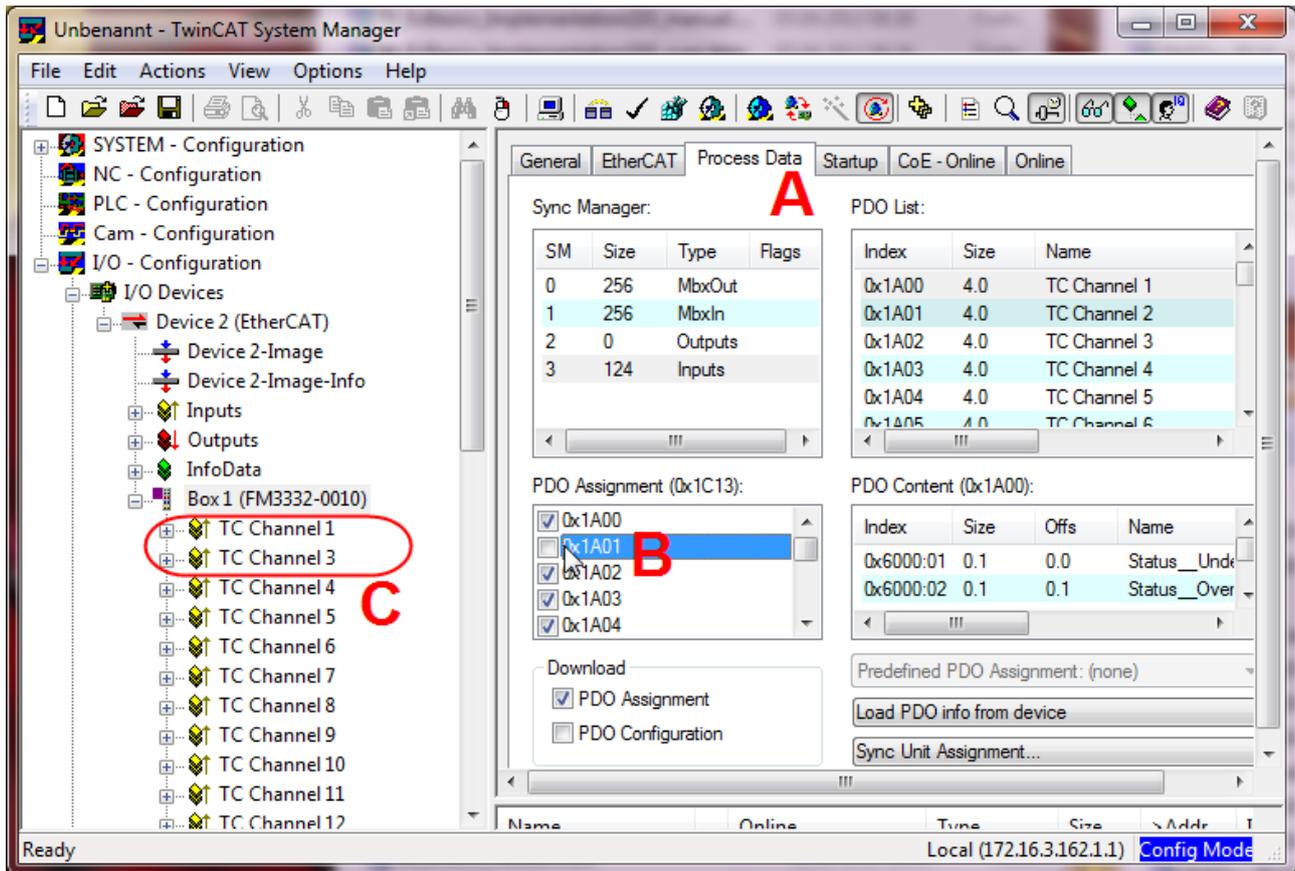


Fig. 36: Process data PDO assignment

CoE parameters

The CoE parameters can now be changed for each channel in the CoE.

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

The following settings should be made in each channel

1. Coldjunction compensation

Per default, the internal coldjunction compensation is selected (see also chapter [Basic Function Principles \[p. 31\]](#)).

In order to measure the temperature using an external reference point, the value "2" (external processdata (1/10 °C)) must be set.

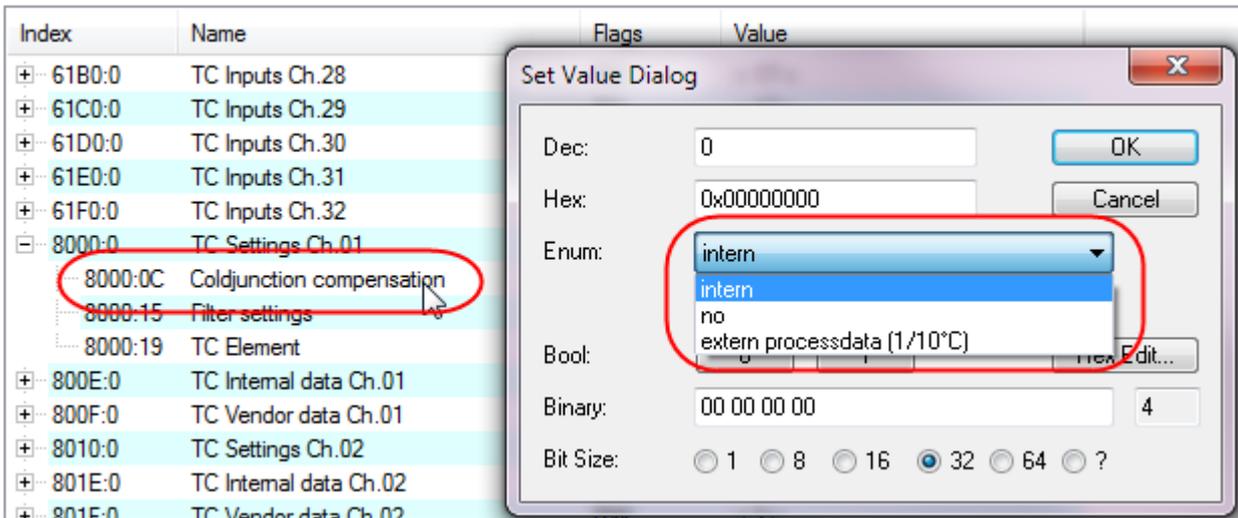


Fig. 37: Object "Coldjunction compensation"

2. Filter setting

The filter setting from channel 1 is simultaneously effective for all channels.
The selection of 50, 60, 250 or 500 Hz filter is possible

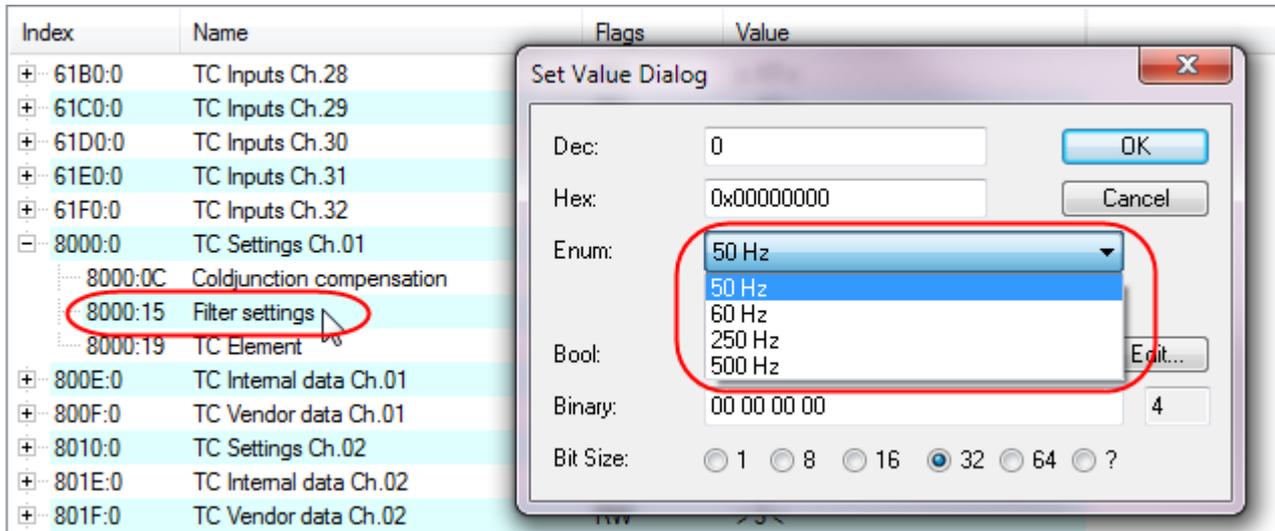


Fig. 38: Object "Filter settings"

3. TC Element

You can set the type of thermocouple (J or K) or the resolution of the respective channel here.

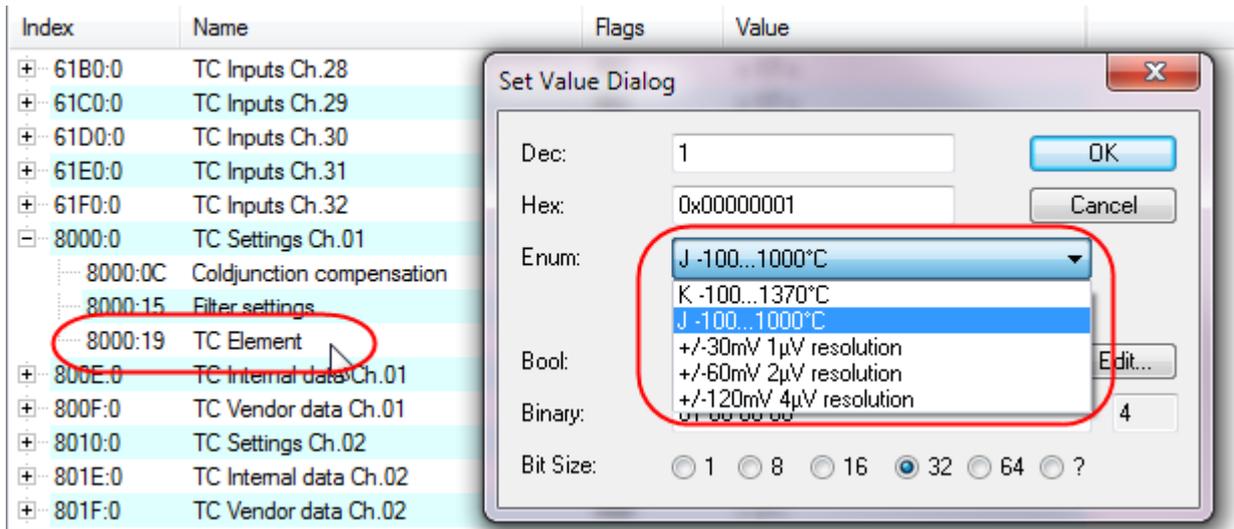


Fig. 39: Object “TC element”

With the exception of the filter setting, different values can be specified for each channel.

The settings can also be loaded via the SPS/PLC/Task at runtime, e.g. using function blocks from *TcEtherCAT.lib*.

6. Diagnostics

Both the terminal and each individual channel signal their proper function in status variables, see above. For proper operation the cyclic evaluation of the following information is recommended:

- Module FM33xx-B110-xxxx
 - WorkingCounter (*WcState*)
 - must be 0 in each cycle, otherwise no valid data acceptance has taken place
 - Status *State*
 - The EC status of the terminal must be in OP mode, i.e. the 3rd bit is set. There must not be any error.
- per channel
 - Error, Back Voltage State, Wirebreak
 - These variables must be 0/FALSE
 - TxPDO State
 - This variable indicates an internal process data error and must be 0/FALSE.
 - TxPDO Toggle
 - This variable indicates the delivery of new data by changing its value.

7. Operation

During operation the thermocouples are cyclically measured and the process value is transmitted to the master.

5.4 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- und 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>.

5.4.1 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. One option is described here.

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

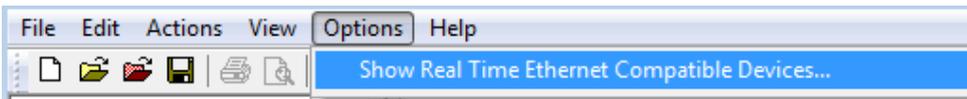


Fig. 40: System Manager “Options” (TwinCAT 2)

This has to be called up by the Menü “TwinCAT” within the TwinCAT 3 environment:

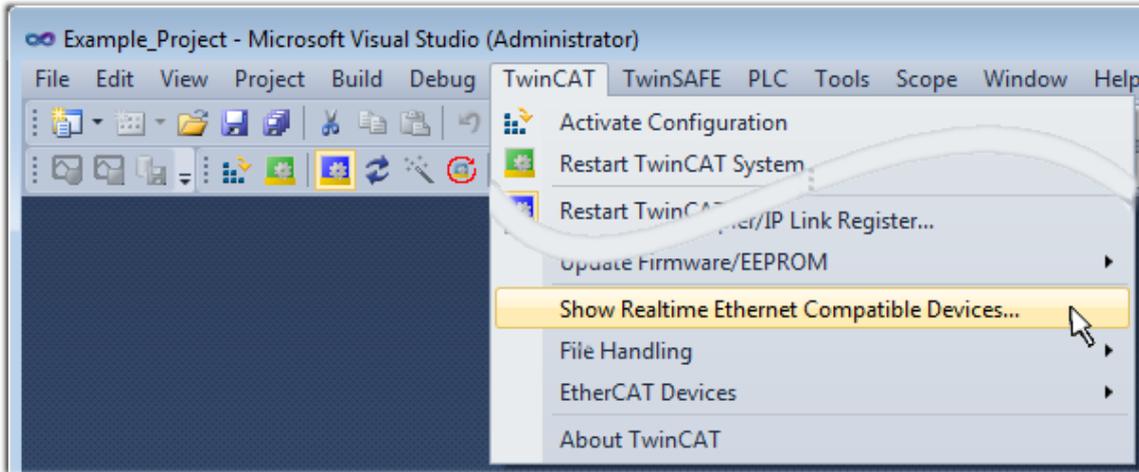


Fig. 41: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

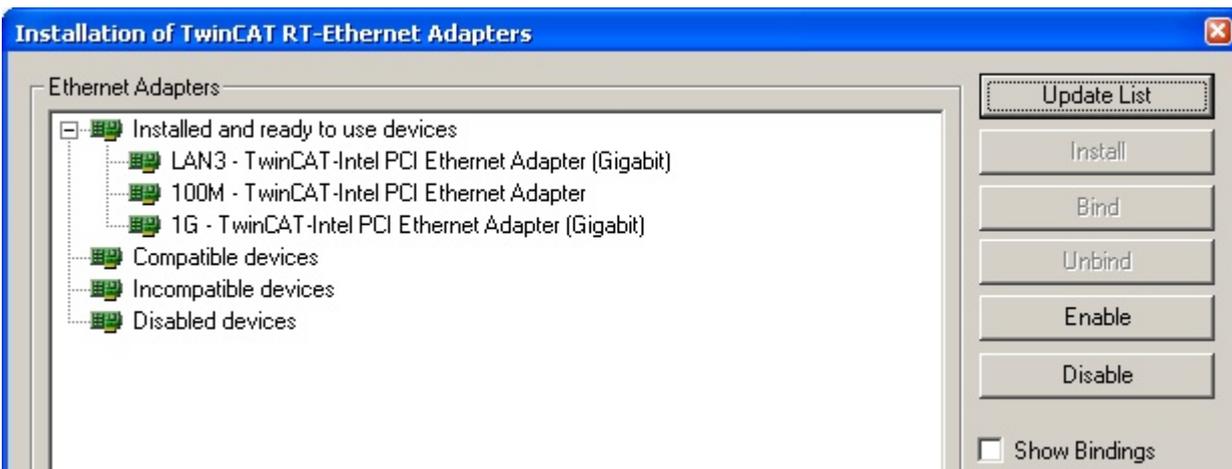


Fig. 42: Overview of network interfaces

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”](#) [▶ 56] in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 43: 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)

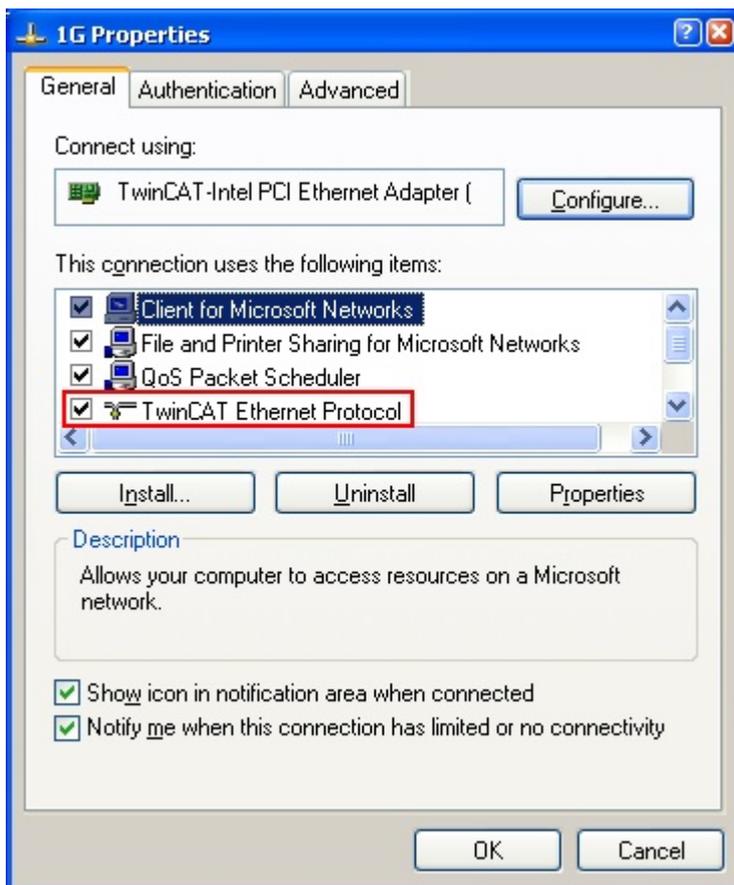


Fig. 44: Windows properties of the network interface

A correct setting of the driver could be:



Fig. 45: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

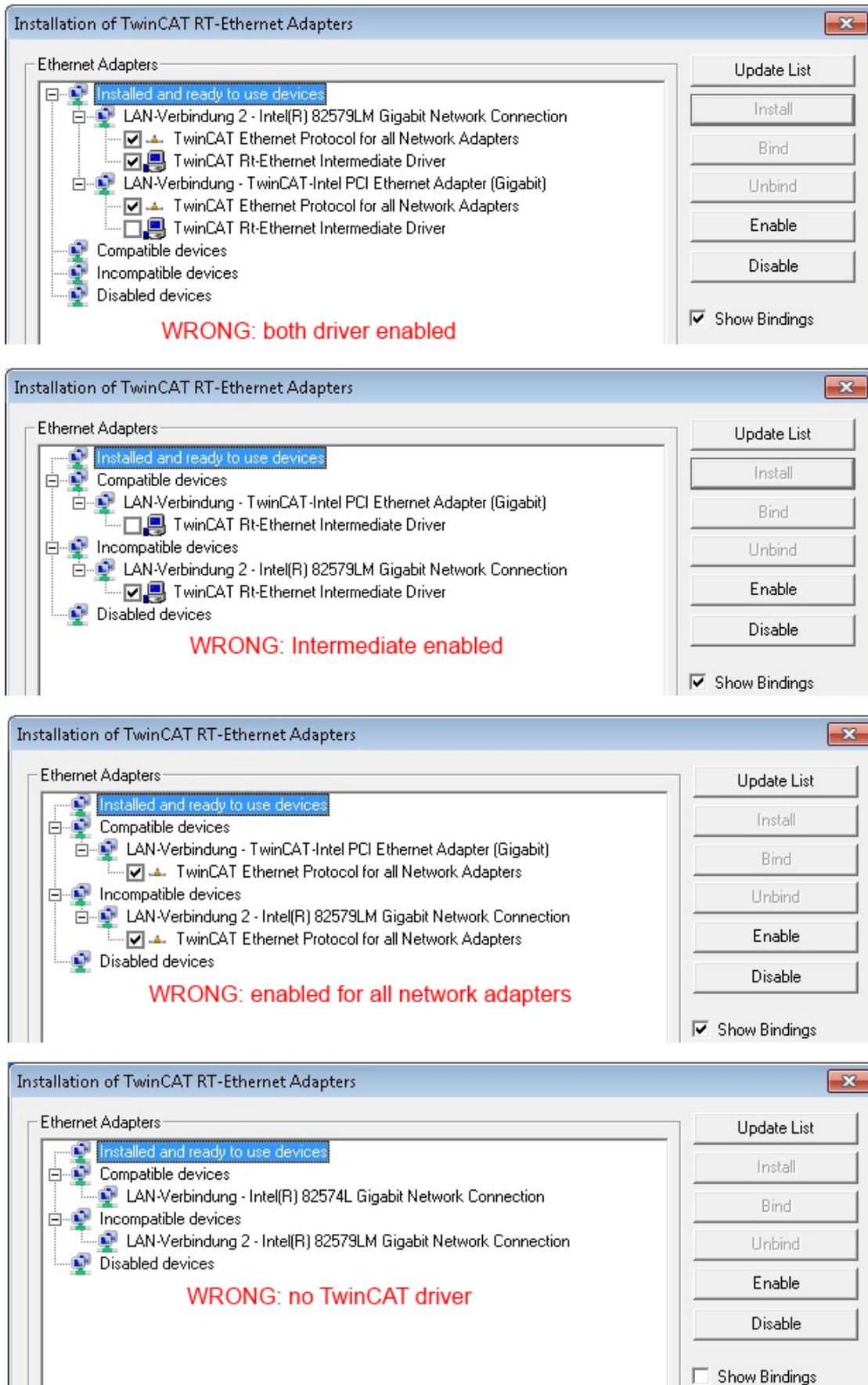


Fig. 46: Incorrect driver settings for the Ethernet port

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.

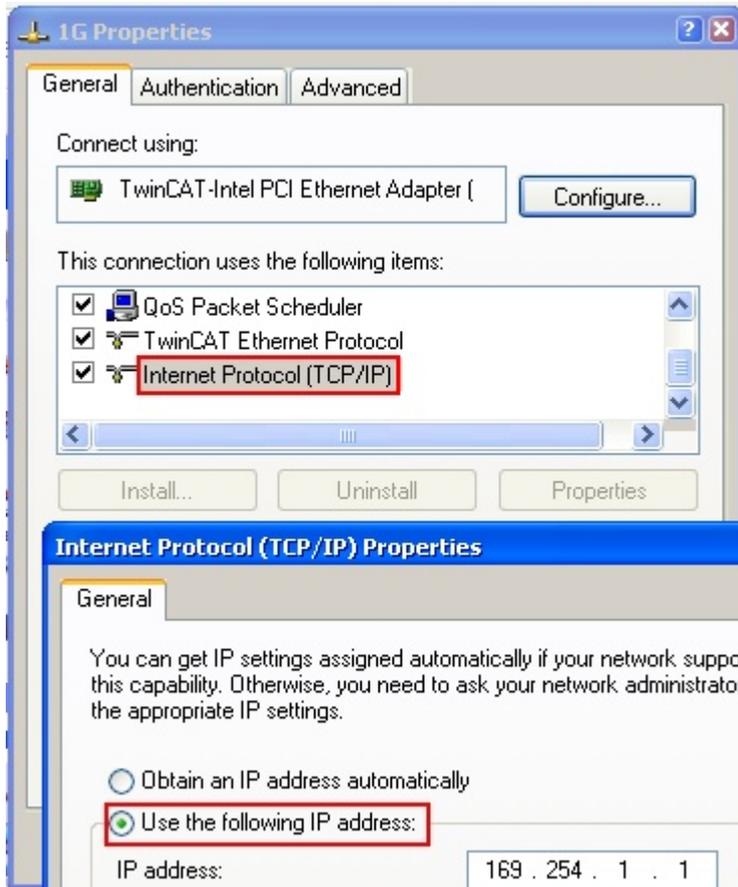


Fig. 47: TCP/IP setting for the Ethernet port

5.4.2 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 is available for this purpose.

i 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”

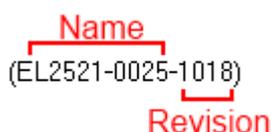


Fig. 48: Identifier structure

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](#) [► 8].

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.

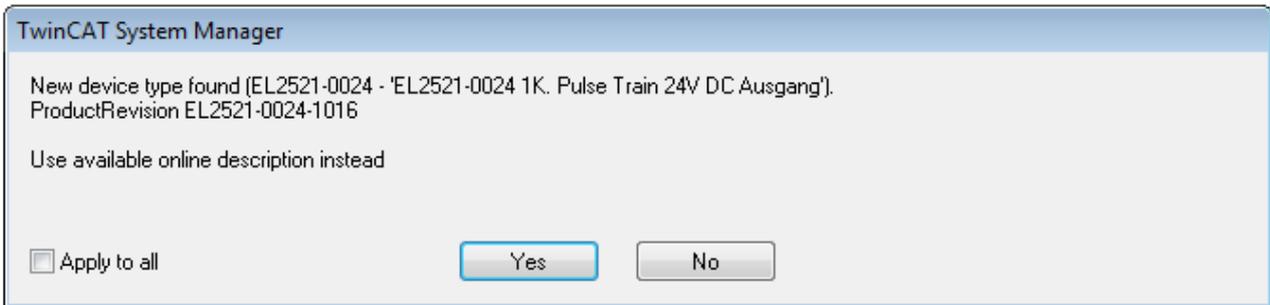


Fig. 49: *OnlineDescription information window (TwinCAT 2)*

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

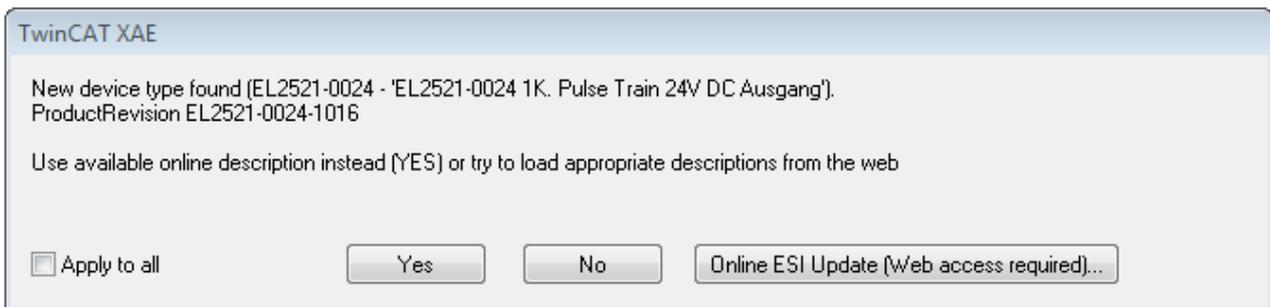


Fig. 50: *Information window OnlineDescription (TwinCAT 3)*

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
<p>Changing the ‘usual’ configuration through a scan</p> <ul style="list-style-type: none"> ✓ 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](#)’ [► 56].

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. 51: 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").

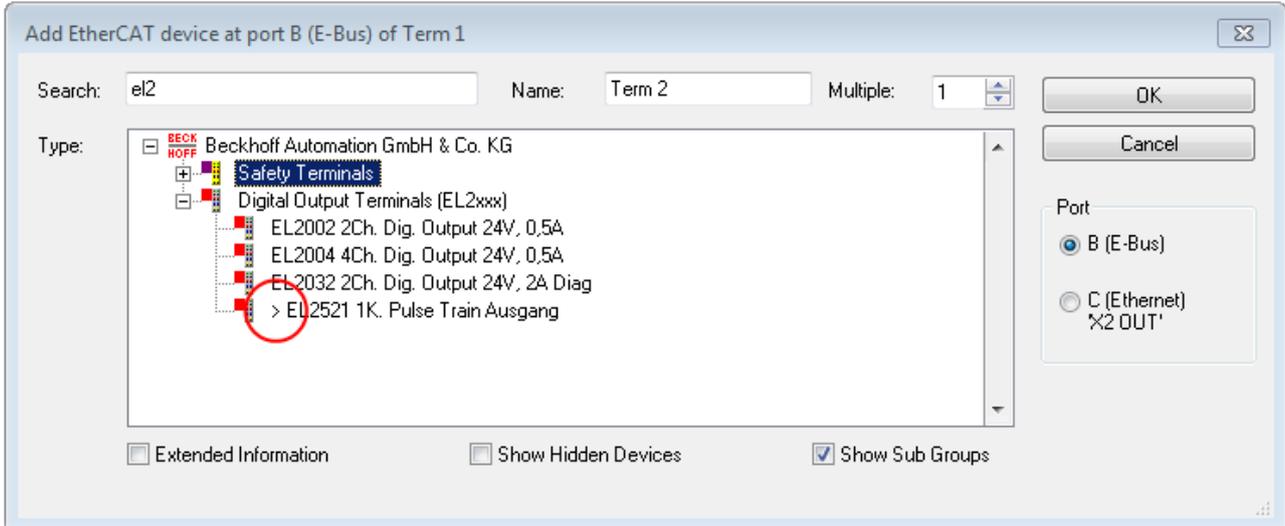


Fig. 52: Indication of an online recorded ESI of EL2521 as an example

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.

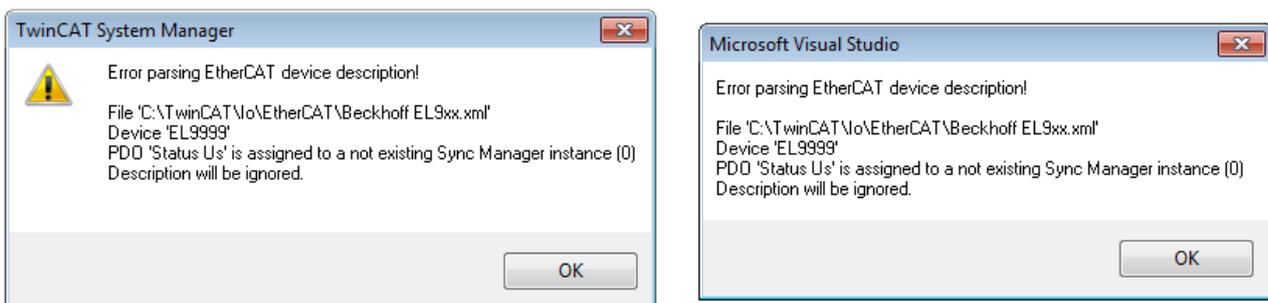


Fig. 53: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.4.3 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

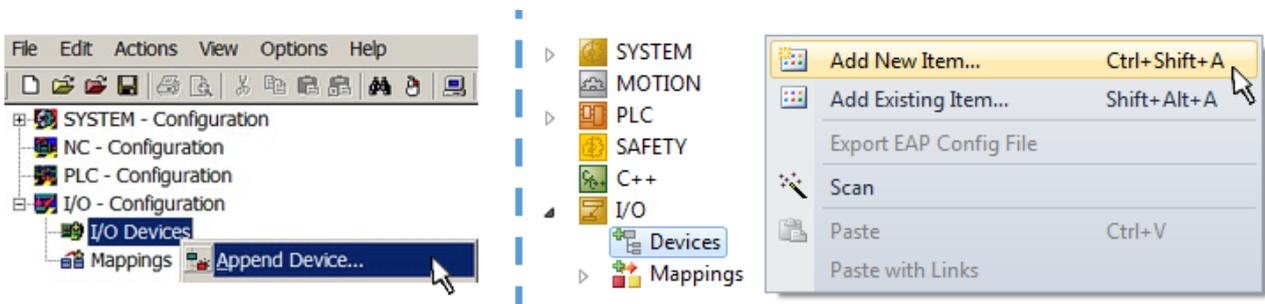


Fig. 54: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

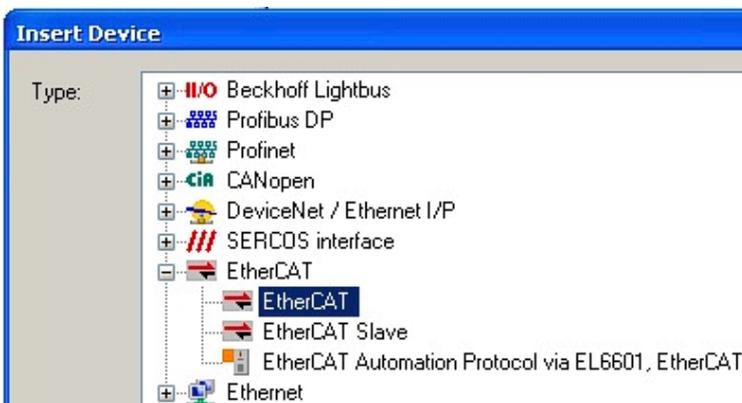


Fig. 55: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

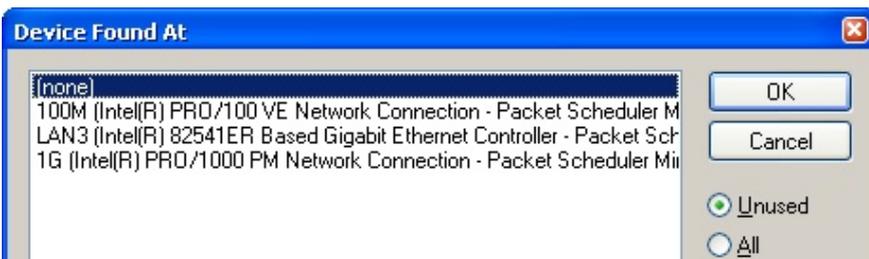


Fig. 56: Selecting the Ethernet port

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)".

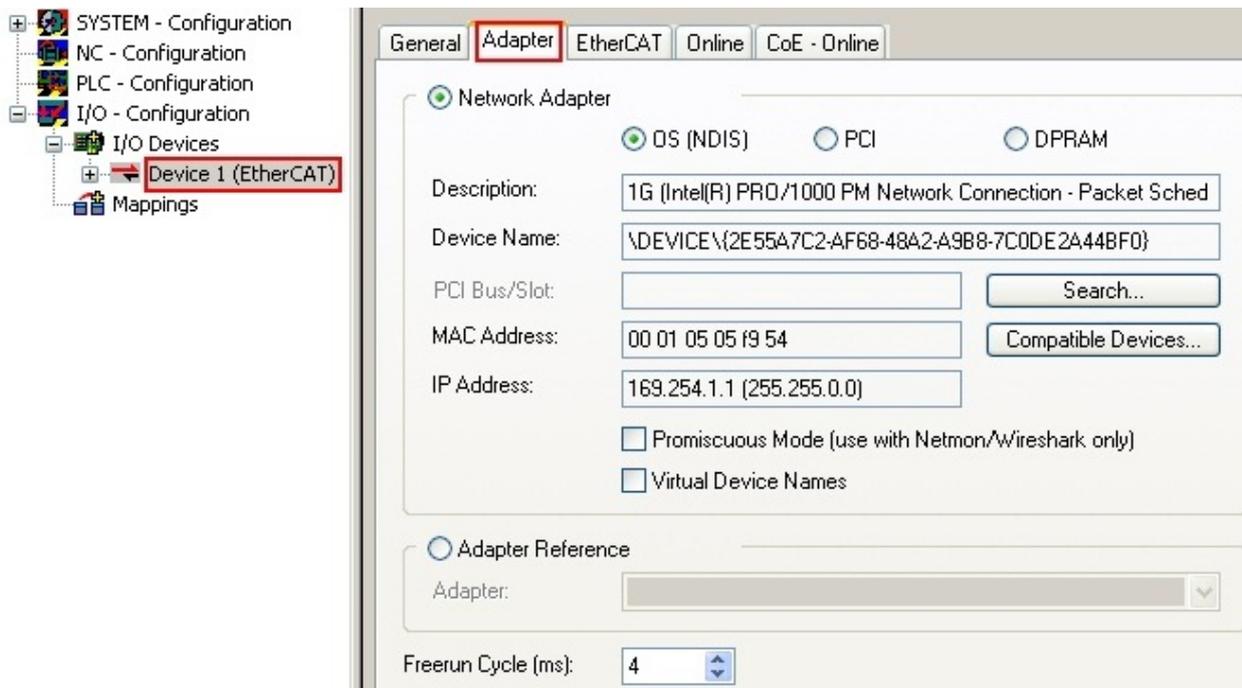


Fig. 57: EtherCAT device properties (TwinCAT 2)

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](#) [▶ 46].

Defining EtherCAT slaves

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

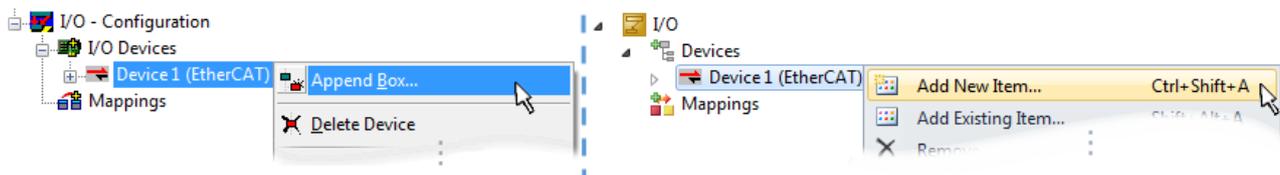


Fig. 58: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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: EK couplers, EP boxes, devices with RJ45/M8/M12 connector
- “E-Bus”: LVDS “terminal bus”, “EJ-module”: EL/ES terminals, various modular modules

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

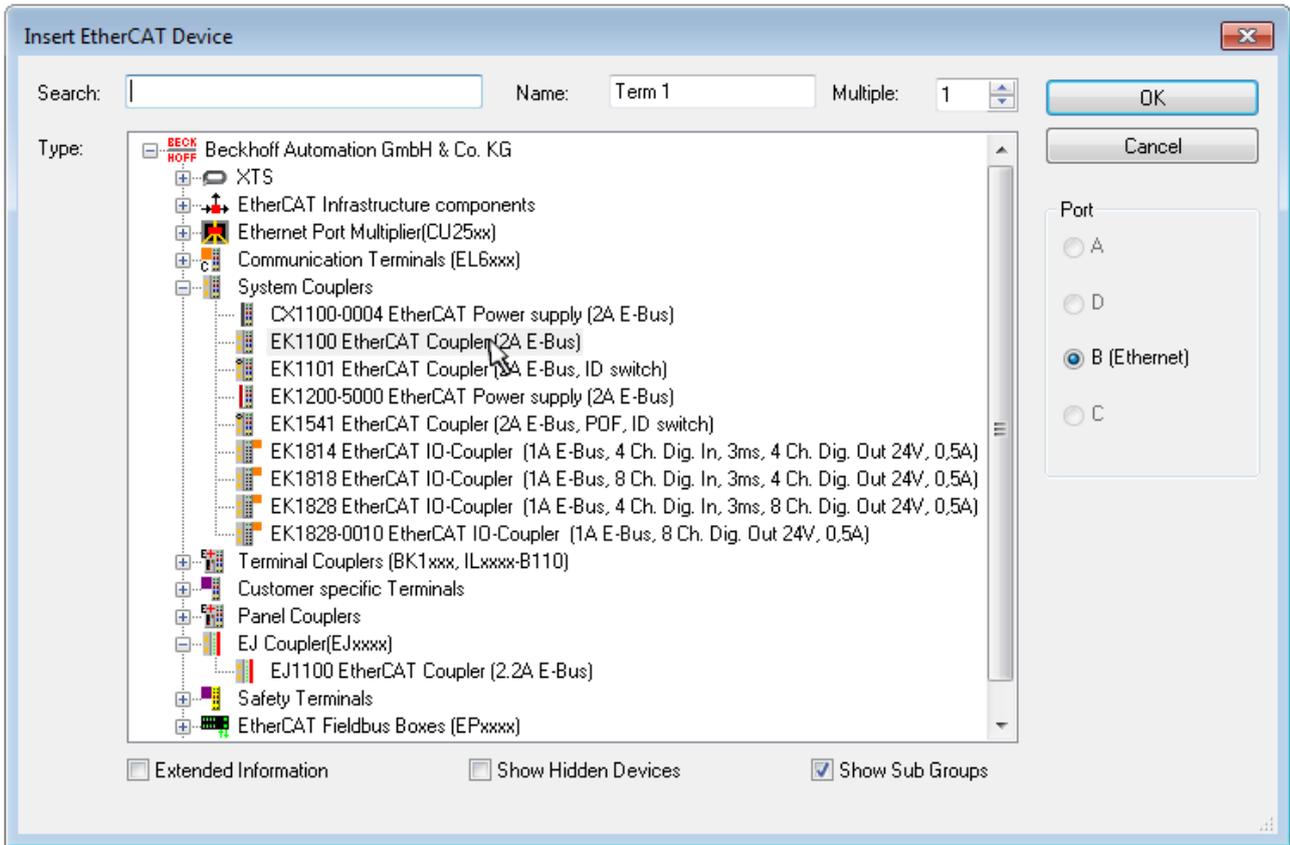


Fig. 59: Selection dialog for new EtherCAT device

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”.

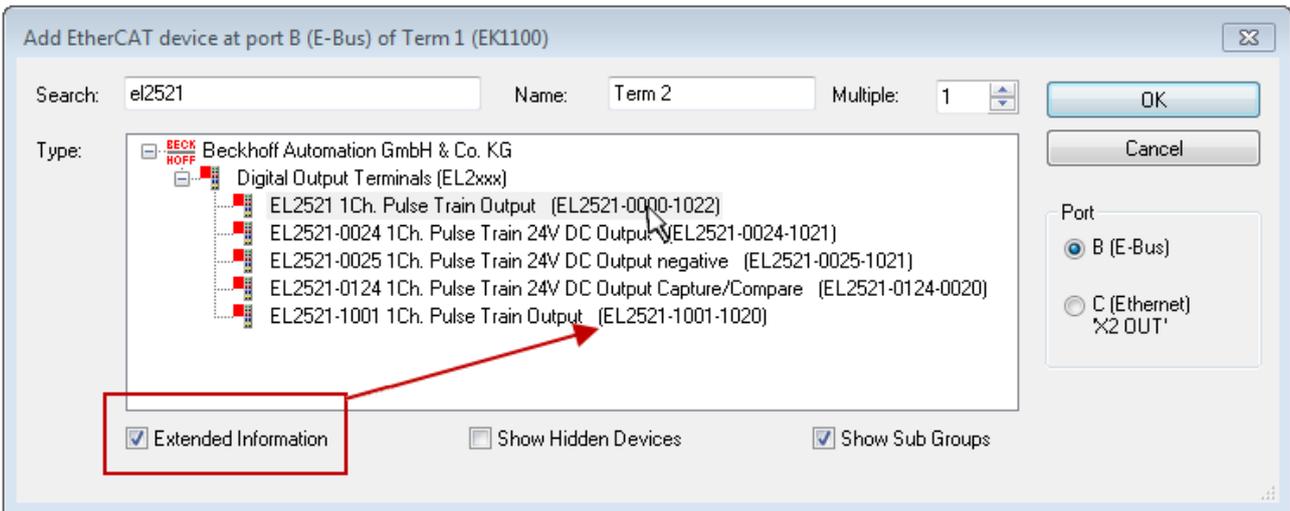


Fig. 60: Display of device revision

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”.

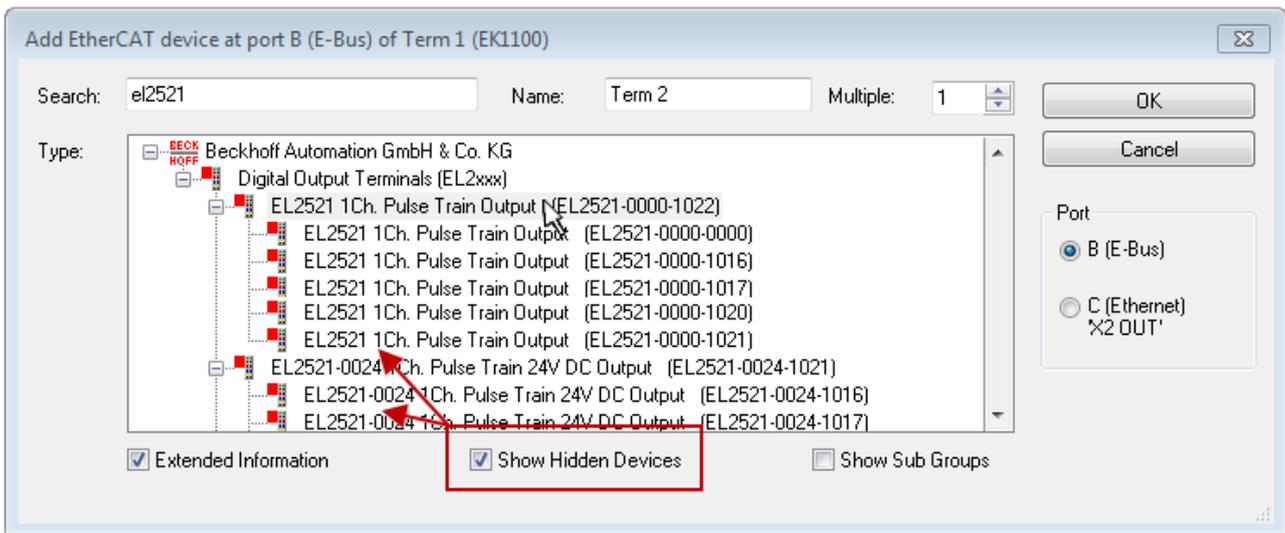


Fig. 61: Display of previous revisions

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.

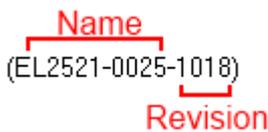


Fig. 62: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

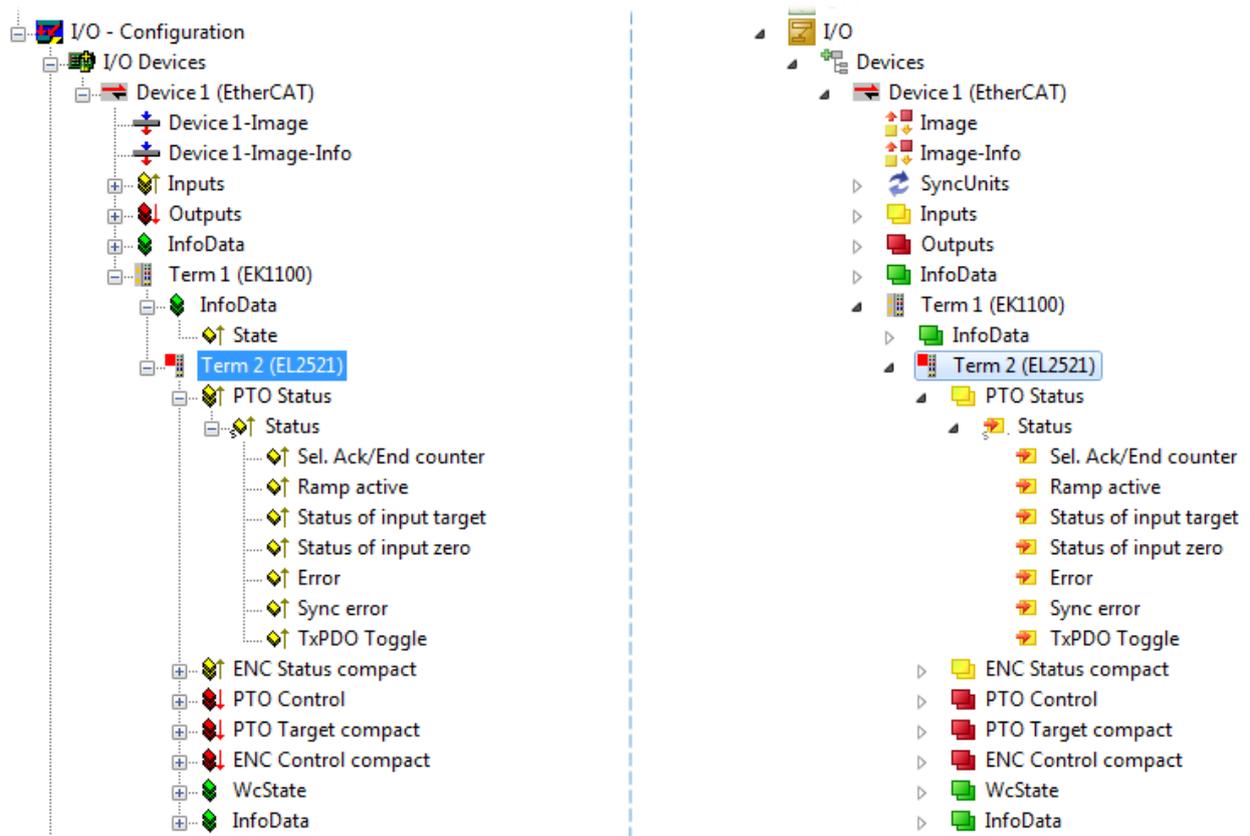


Fig. 63: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.4.4 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)”

i Online scanning in Config mode

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

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

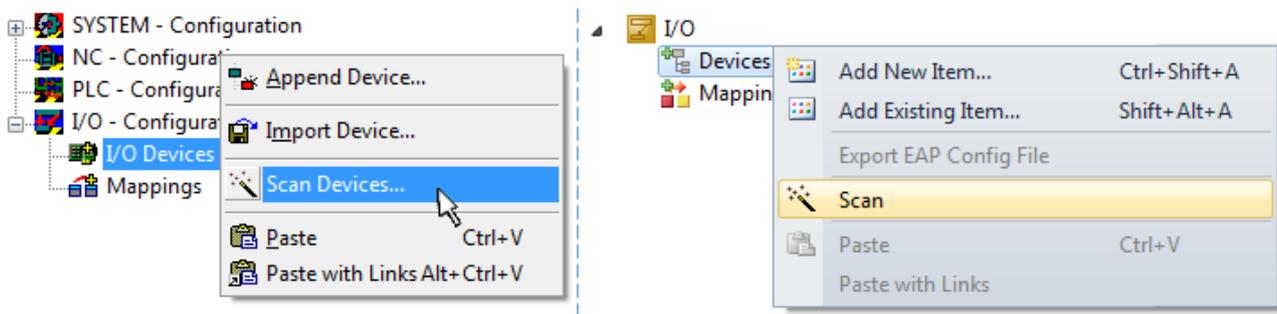


Fig. 65: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

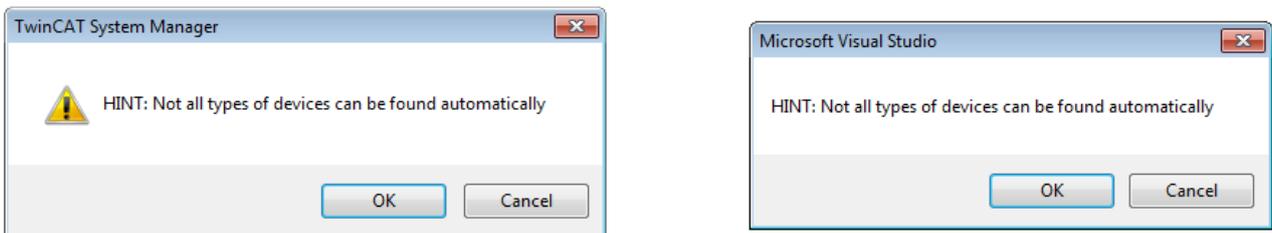


Fig. 66: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

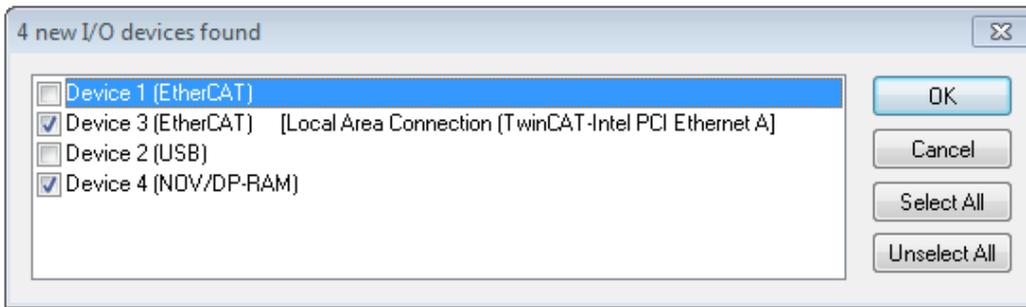


Fig. 67: Detected Ethernet devices

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](#) [▶ 46].

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.

Name
(EL2521-0025-1018)
Revision

Fig. 68: 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](#) [▶ 66] 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:

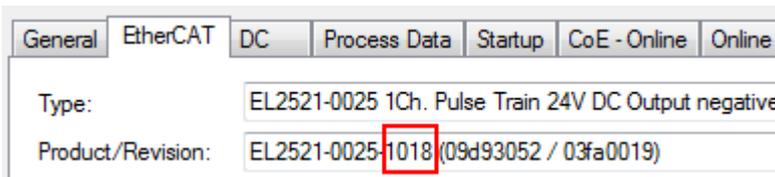


Fig. 69: Installing EtherCAT terminal with revision -1018

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 [▶ 66](#) 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:

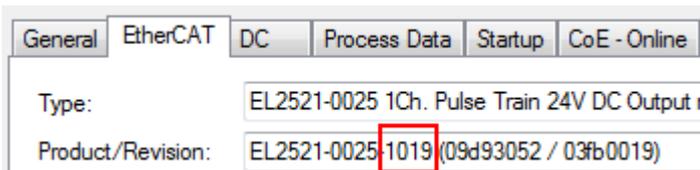


Fig. 70: Detection of EtherCAT terminal with revision -1019

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

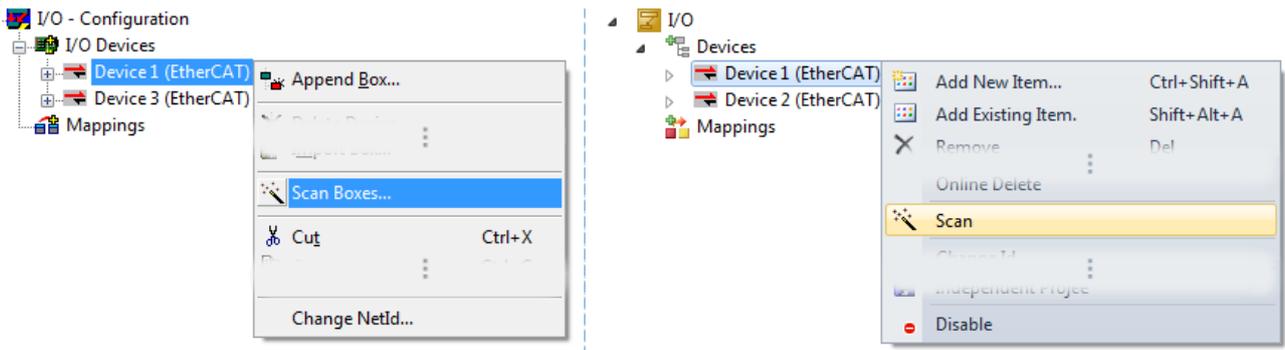


Fig. 72: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

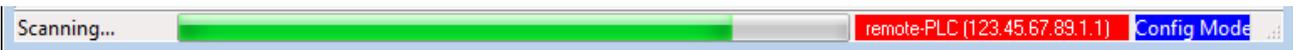


Fig. 73: Scan progress exemplified by TwinCAT 2

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



Fig. 74: 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. 75: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 76: 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”.

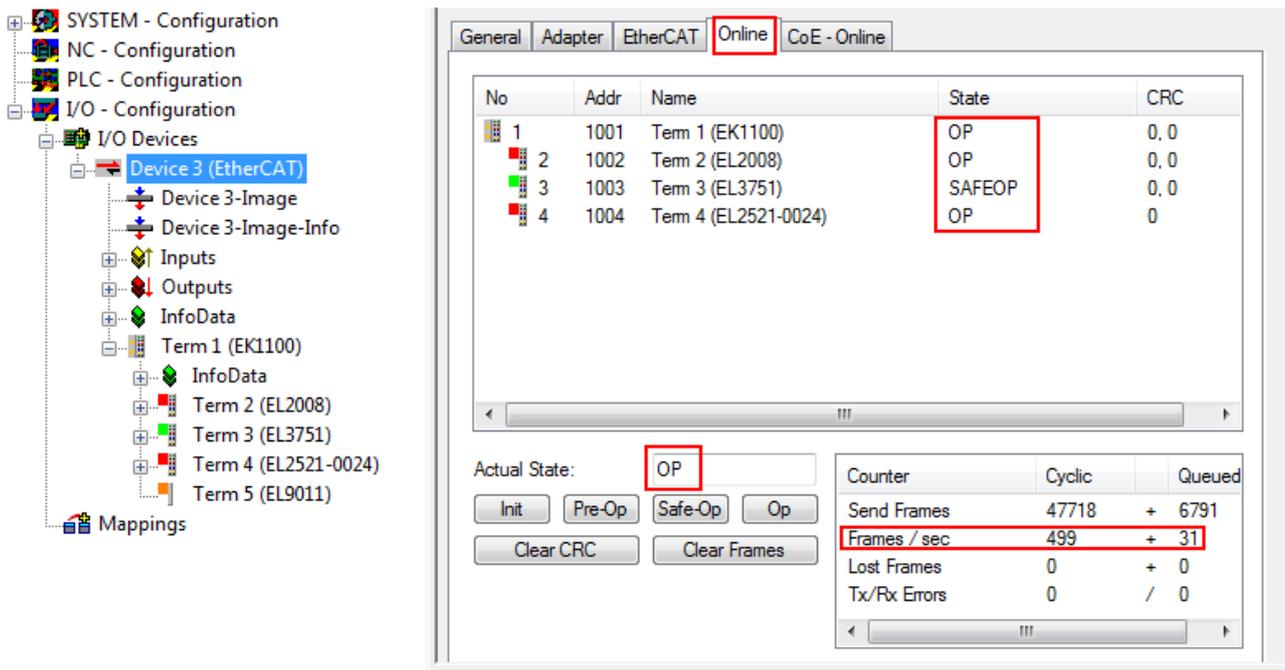


Fig. 77: Online display example

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 \[► 56\]](#).

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.

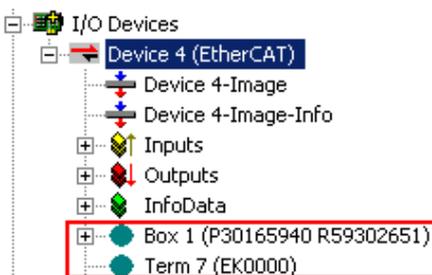


Fig. 78: Faulty identification

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. 79: 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.

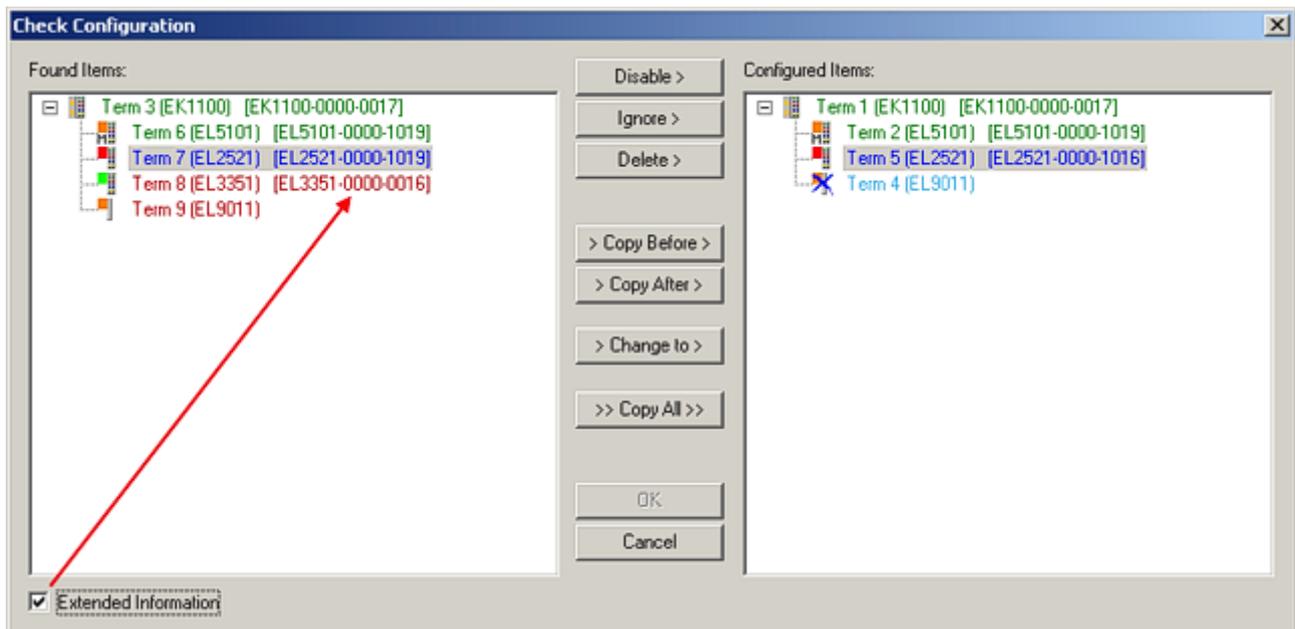


Fig. 80: Correction dialog

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

Colour	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.

● Device selection based on revision, compatibility

i 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.

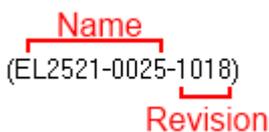


Fig. 81: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

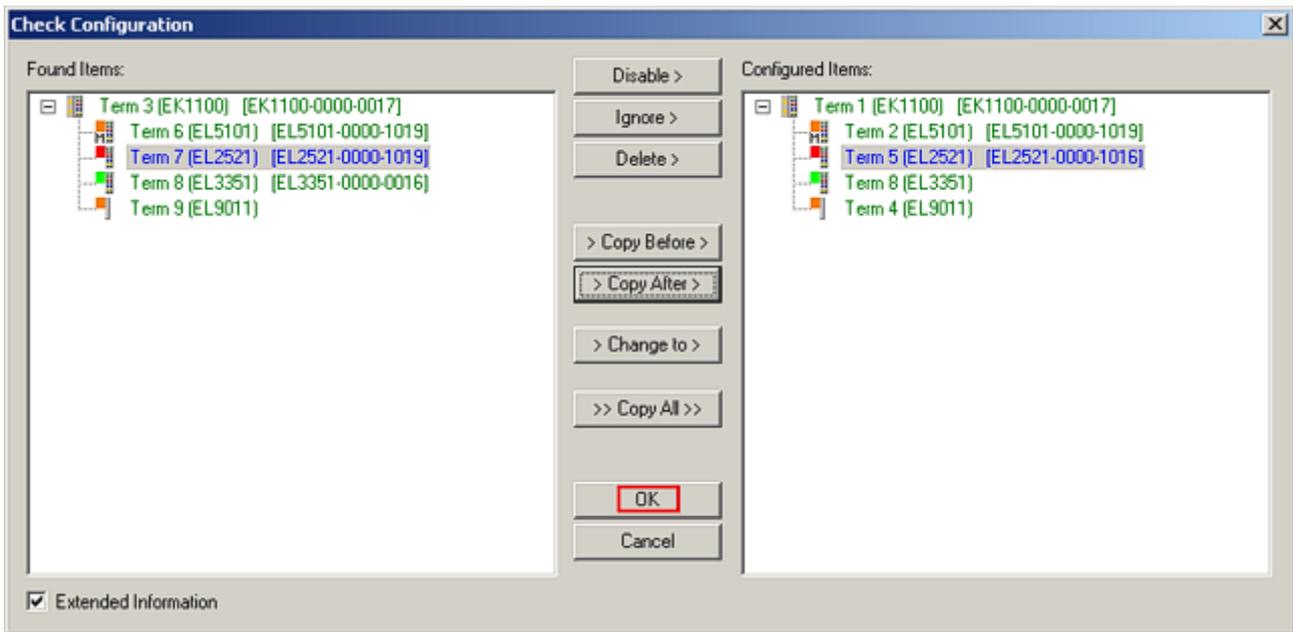


Fig. 82: Correction dialog with modifications

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.

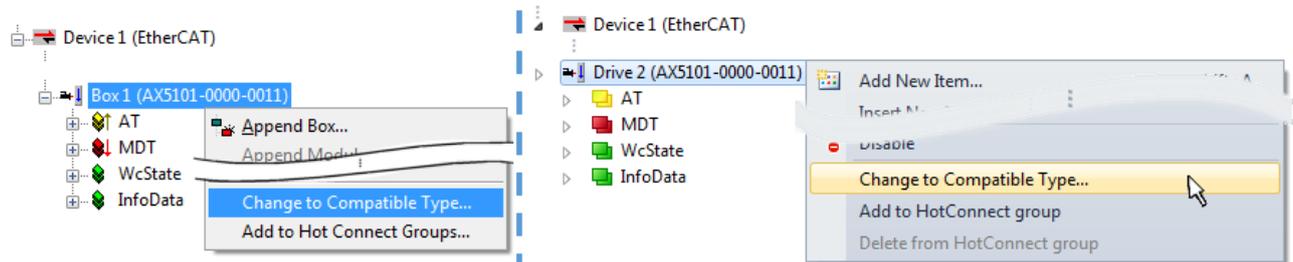


Fig. 83: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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*

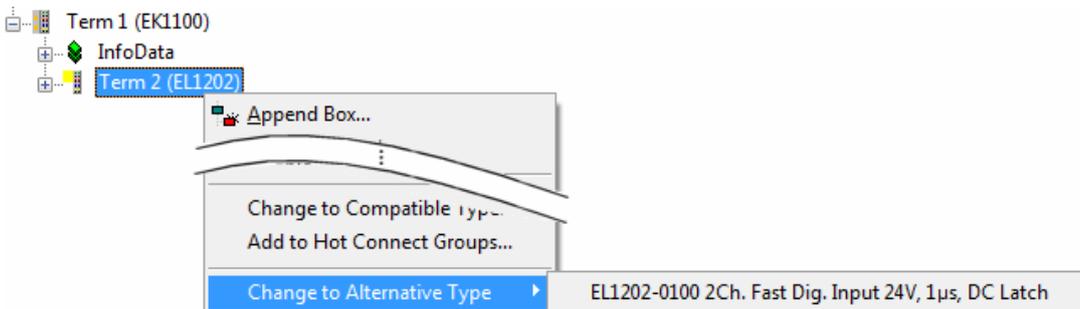


Fig. 84: TwinCAT 2 Dialog Change to Alternative Type

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).

5.4.5 EtherCAT slave process data settings

The process data transferred by an EtherCAT slave during each cycle (**Process Data Objects, PDOs**) 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 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”).

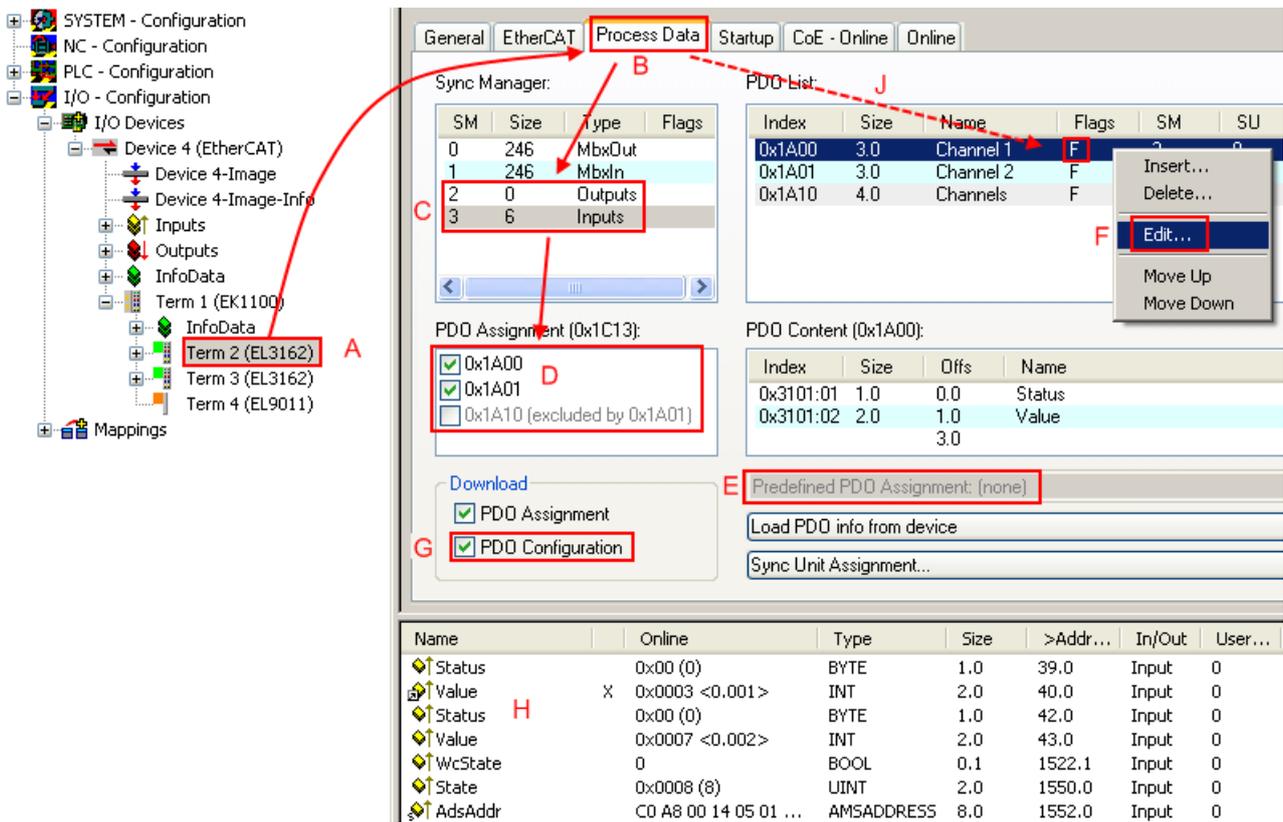


Fig. 85: Configuring the process data

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” log-ger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

5.4.6 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).

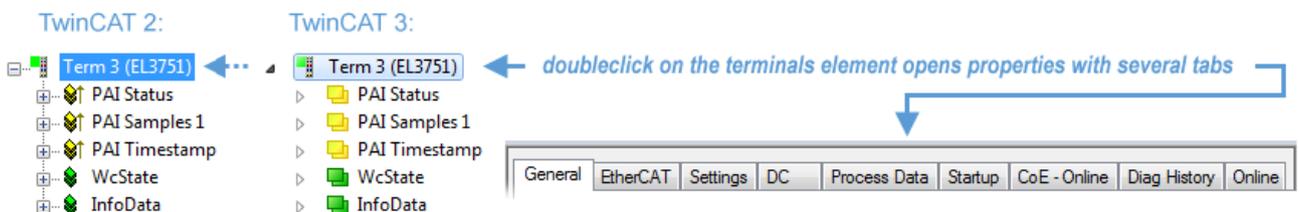


Fig. 86: Branch element as terminal EL3751

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

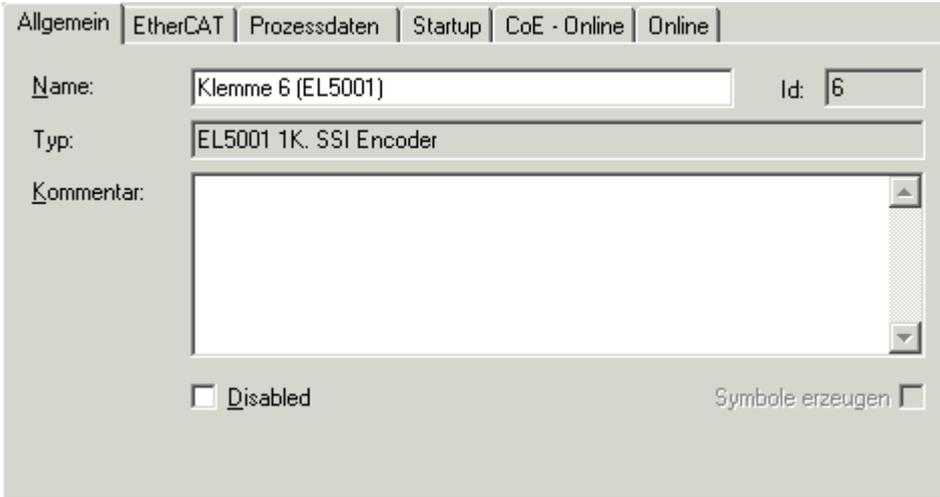


Fig. 87: "General" tab

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

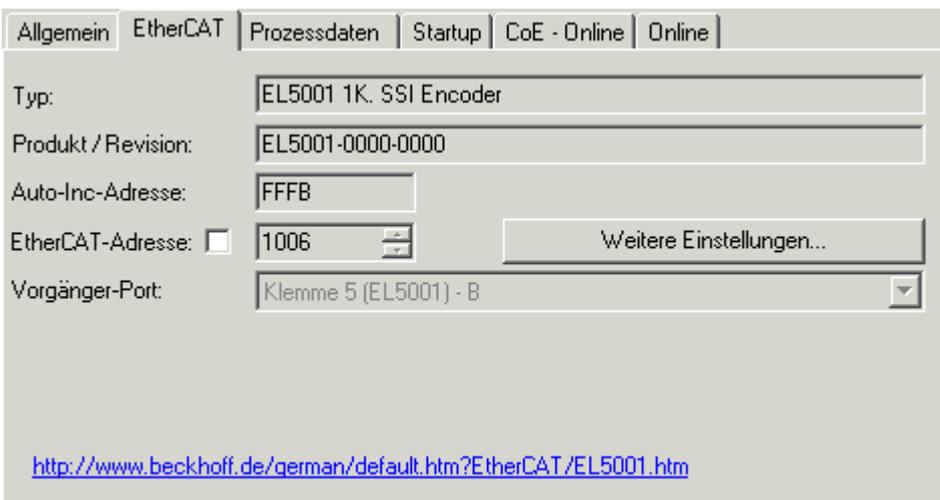


Fig. 88: „EtherCAT“ tab

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.

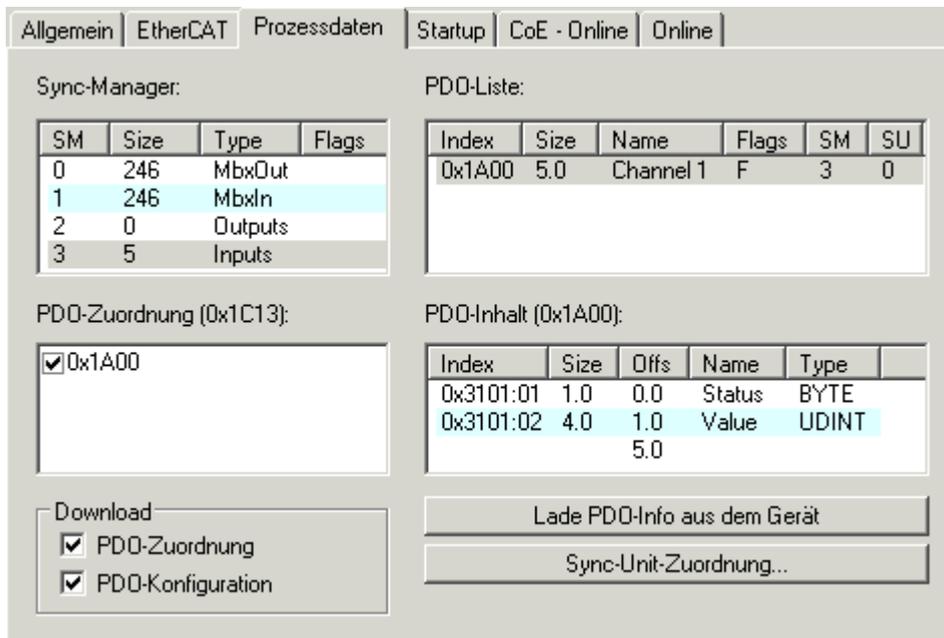


Fig. 89: “Process Data” tab

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”).

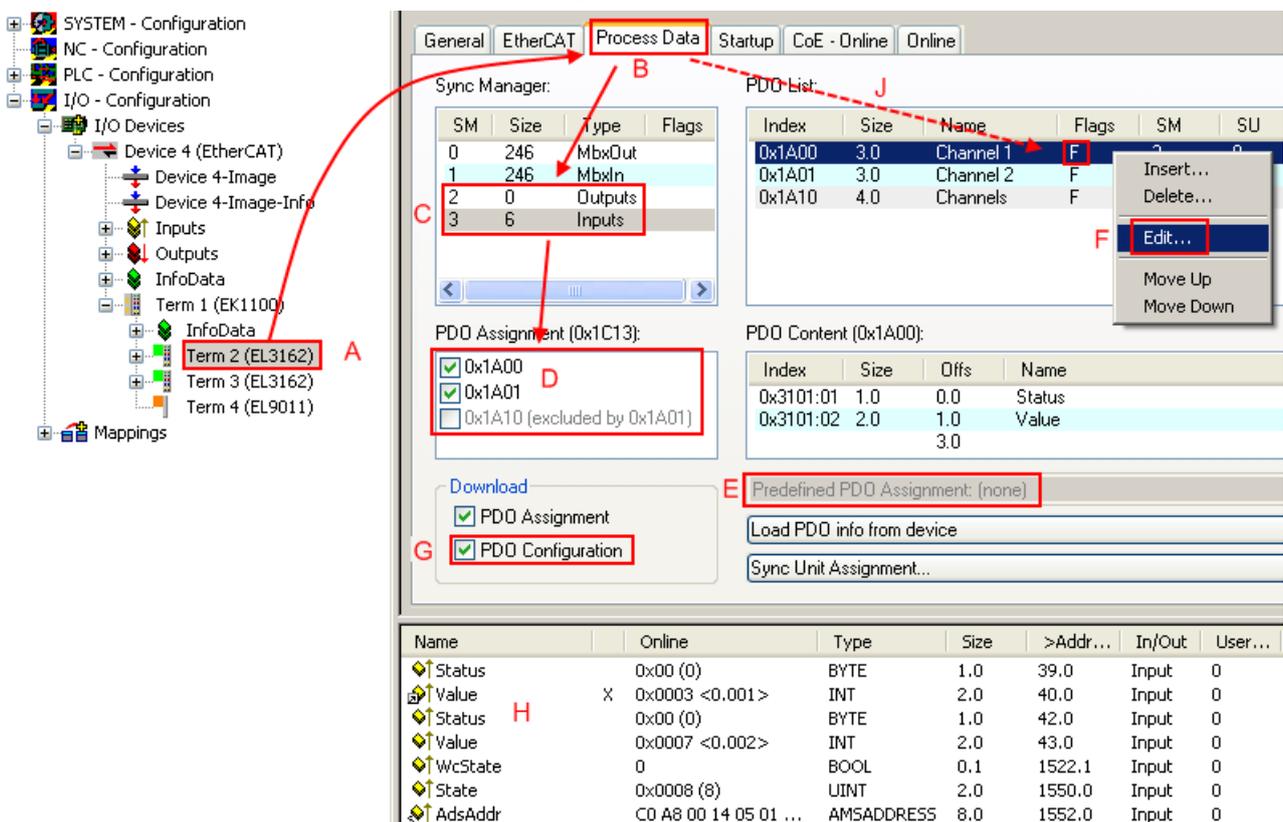


Fig. 90: Configuring the process data

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” log-ger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A [detailed description \[► 78\]](#) 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.

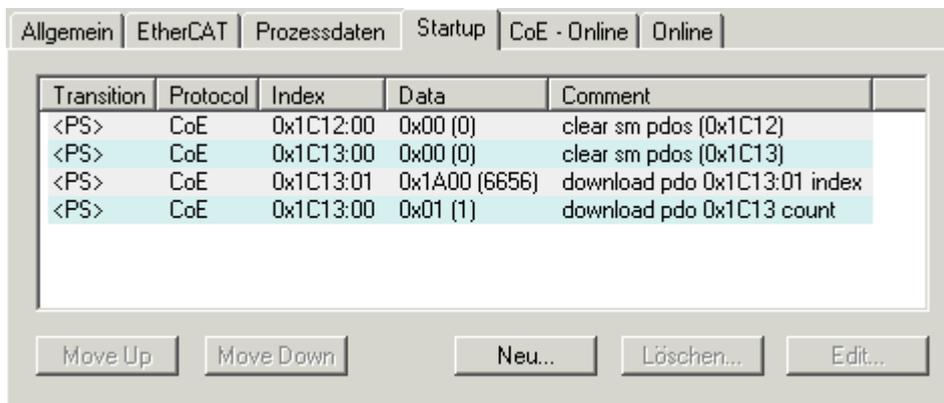


Fig. 91: „Startup“ tab

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.

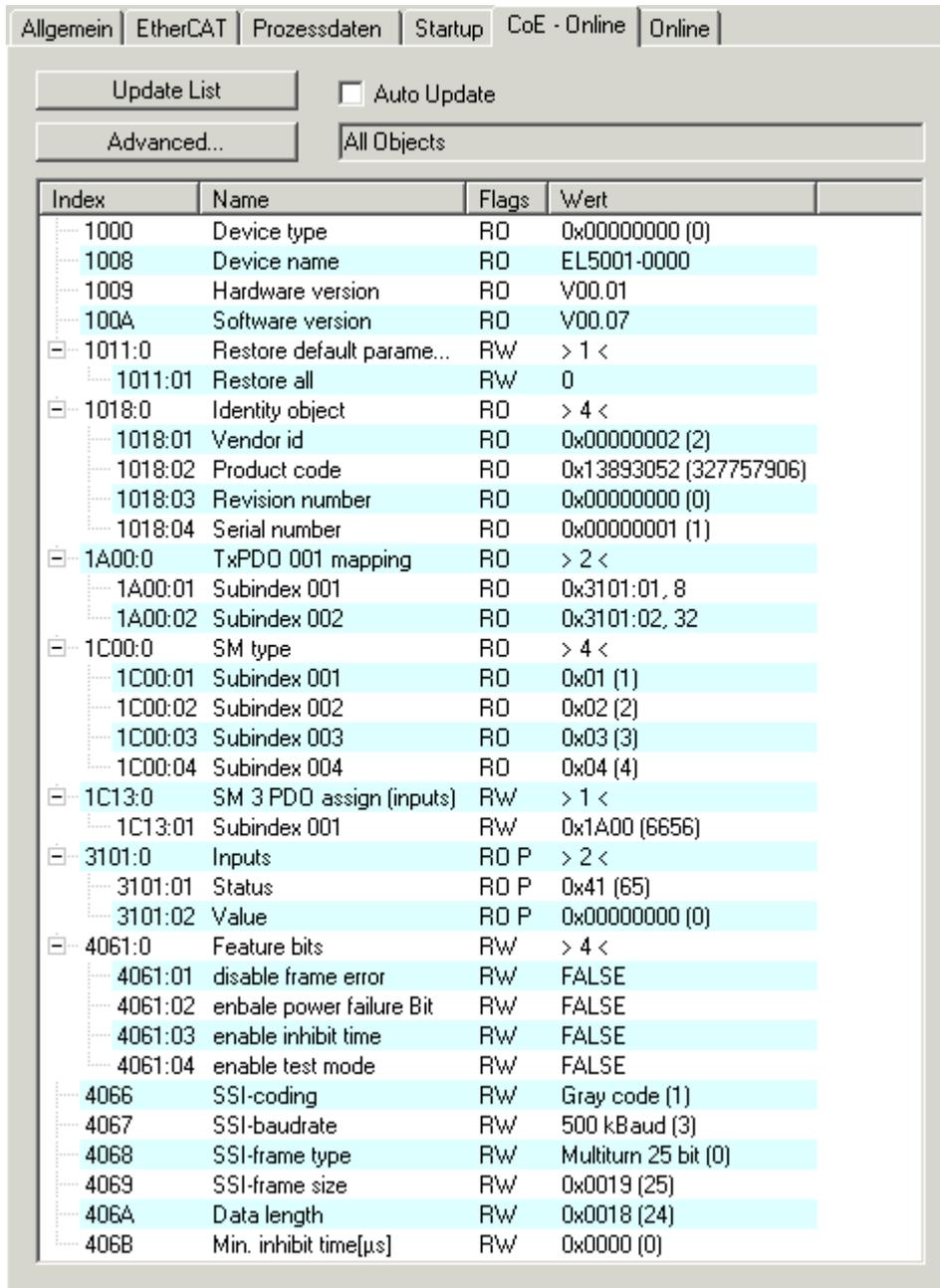


Fig. 92: “CoE – Online” tab

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.

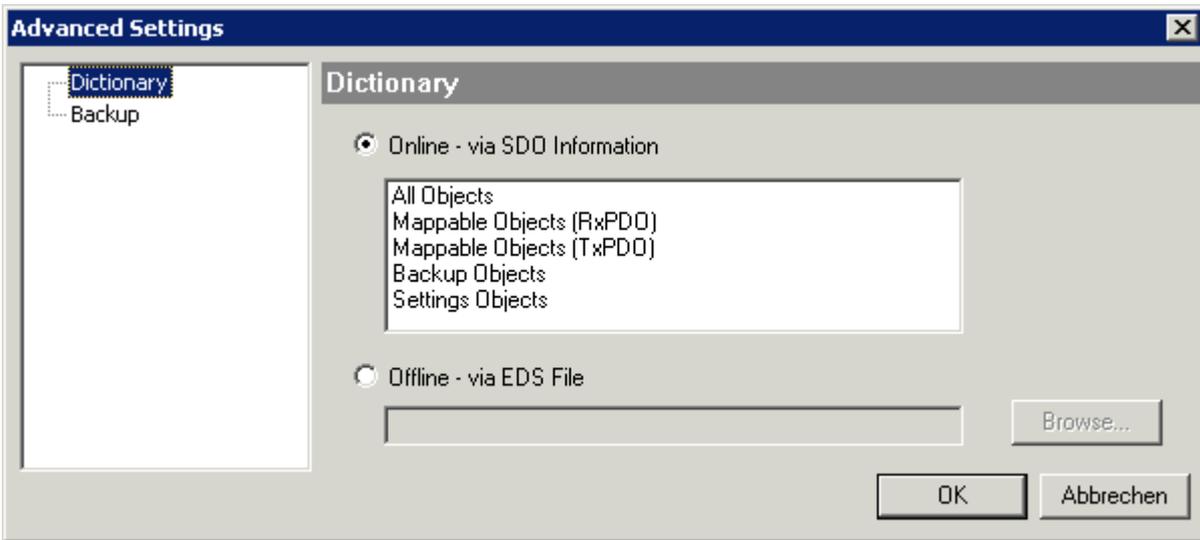


Fig. 93: Dialog "Advanced settings"

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

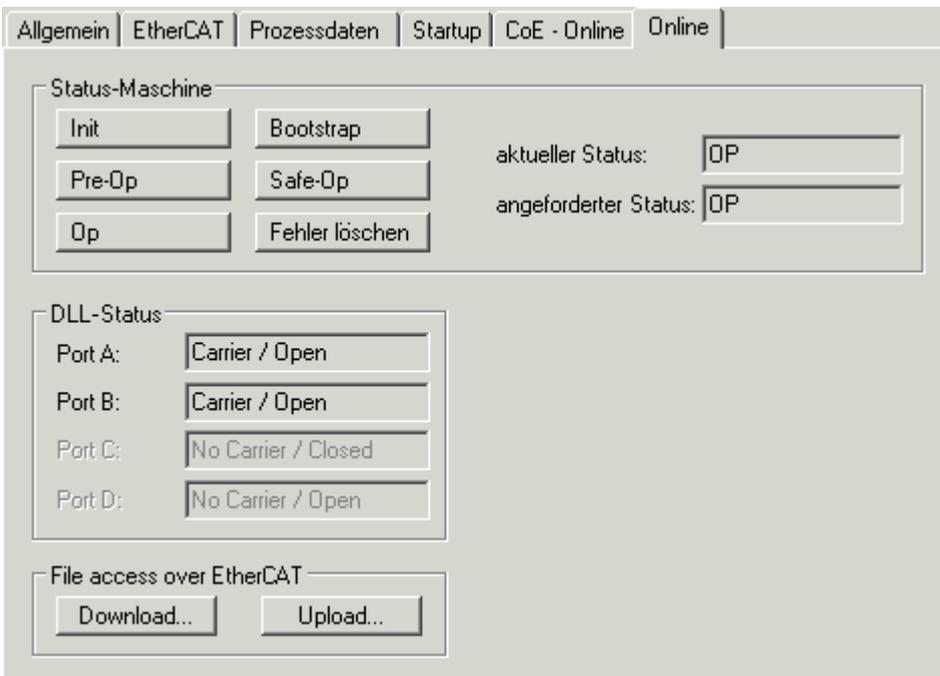


Fig. 94: „Online“ tab

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)

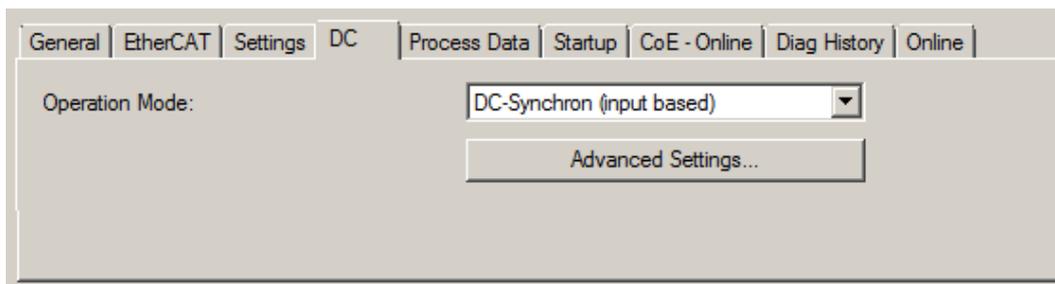


Fig. 95: "DC" tab (Distributed Clocks)

- 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 are specified on <http://infosys.beckhoff.com>:

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

5.4.6.1 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 \[▶ 76\]](#)),
 - 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 \[► 74\]](#) 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.

5.4.7 Object description and parameterization

Parameterization

i Terminals' parameterization will be conducted by the [CoE tab \[► 75\]](#) (double-click on the referring object) or the [Process data tab \[► 72\]](#) (PDO assignment)

EtherCAT XML Device Description

i If the XML description of the FM33xx-B110-xxxx is not available in your system you can request the latest [XML](#) file from Beckhoff and install it according to the installation instructions.

Restore object

Table 1: Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore the default settings in the EtherCAT slave	UINT8	RO	0x01 (1 _{dec})
1011:01	SubIndex 001	If this object is set to " 0x64616F6C " ("Set Value Dialog") all terminal-specific objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 _{dec})

Configuration data

Index 8nn0 TC Settings (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
8nn0:0	TC Settings	Max. subindex	UINT8	RO	0x19 (25 _{dec})
8nn0:0C	Coldjunction compensation	0: intern 1: no coldjunction compensation is not active 2: Extern process data coldjunction compensation is effected over process data (resolution [1/10]°C)	BIT2	RW	0x00 (0 _{dec})
8nn0:15	Filter settings	This object defines the digital filter settings	UINT16	RW	0x0000 (0 _{dec})
8nn0:19	TC Element [► 38]	Thermocouples 0: Type: K -200°C to 1370°C 1: Type: J -100°C to 1200°C 100: ± 30 mV (1 µV resolution) 101: ± 60 mV (2 µV resolution) 102: ± 75 mV (4 µV resolution)	UINT16	RW	0x0000 (0 _{dec})

Index F080 Channel Enable (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
F080:0	Channel Enable	Max. channel number	UINT8	RO	0x20 (32 _{dec})
F080:[nn+1]	SubIndex 001 .. 032	FALSE: Channel disabled TRUE: Channel enabled	BOOLEAN	RW	0x01 (0 _{dec})

Configuration data (vendor specific)

Index 8nnF TC Vendor data (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
8nnF:0	TC Vendor data	Max. subindex	UINT8	RO	0x03 (3 _{dec})
8nnF:01	Calibration offset TC	Calibration offset thermocouple	INT16	RW	0x0000 (0 _{dec})
8nnF:02	Calibration gain TC	Calibration gain thermocouple	UINT16	RW	0x1E88 (7816 _{dec})
8nnF:03	Calibration offset CJ	Calibration offset coldjunction	INT16	RW	0x0000 (0 _{dec})

Input data

Index 6nn0 TC Inputs (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
6nn0:0	TC Inputs	Max. Subindex	UINT8	RO	0x11 (17 _{dec})
6nn0:01	Underrange	Measurement underrange	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:02	Overrange	Measurement overrange ("open circuit" detection if also "Error" [Index 0x6nn0:07])	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:07	Error	Error bit is set, if data is invalid (open circuit, overrange, underrange)	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:08	Back Voltage State	A backvoltage was detected. The module tries to reset the Backvoltage every 10 seconds.	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:09	Wirebreak	Detection of a wirebreak	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:0F	TxPDO State	Valid of data of assigned TxPDO (0=valid, 1=invalid).	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:10	TxPDO Toggle	The TxPDO Toggle is toggled by the slave, if the data of the corresponding PDO was updated.	BOOLEAN	RO	0x00 (0 _{dec})
6nn0:11	Value	Analog input data (resolution 1/10 °C)	INT16	RO	0x0000 (0 _{dec})

Information / diagnosis data

Index 8nnE TC Internal data (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
8nnE:0	TC Internal data	Max. subindex	UINT8	RO	0x04 (4 _{dec})
8nnE:01	ADC raw value TC	ADC raw value thermocouple	UINT32	RO	0x00000000 (0 _{dec})
8nnE:02	ADC raw value TMP36	ADC raw value CJ sensor	UINT32	RO	0x00000000 (0 _{dec})
8nnE:03	CJ temperature	Coldjunction temperature (resolution [1/10]°C)	INT16	RO	0x0000 (0 _{dec})
8nnE:04	CJ voltage	Coldjunction voltage (resolution 1 µV)	INT16	RO	0x0000 (0 _{dec})

Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device Type of the EtherCAT slave: The Lo-Word contains the supported CoE Profile (5001). The Hi-Word contains the Module Profile corresponding to the Modular Device Profile.	UINT32	RO	0x014A1389 (21631881 _{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	FM3312-0010 FM3332-1010

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slaves	STRING	RO	03

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slaves	STRING	RO	01

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Contains information to identify the EtherCAT 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	0x0CF03462 (217068642 _{dec})
1018:03	Revision	Revision number of the EtherCAT-Slave, the Lo-Word (Bit 0-15) indicates the special functions terminal number; the Hi-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 Lo-Byte (Bit 0-7) of the Lo-Word contains the year of manufacturing, the Hi-Byte (Bit 8-15) of the Lo-Word contains the week of manufacturing, the Hi-Word (Bit 16-31) is 0.	UINT32	RO	0x00000006 (110 _{dec})

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Contains information for the standardized Upload and Download of the Backup Entries	UINT8	RO	0x01 (1 _{dec})
10F0:01	Checksum	Checksum over all backup entries	UINT32	RO	0x00000000 (0 _{dec})

Index 10F2 Backup parameter storage

Index (hex)	Name	Meaning	Data type	Flags	Default
10F2:0	Backup parameter storage	reserved	OCTET-STRING[4]	RW	{0}

Index 1Ann TxPDO-Map (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
1Ann:0	TxPDO-Map	PDO Mapping TxPDO	UINT8	RW	0x0A (10 _{dec})
1Ann:01	SubIndex 001	1. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x01 (Underrange))	UINT32	RO	0x6nn0:01, 1
1Ann:02	SubIndex 002	2. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x02 (Overrange))	UINT32	RO	0x6nn0:02, 1
1Ann:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1Ann:04	SubIndex 004	4. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x07 (Error))	UINT32	RO	0x6nn0:07, 1
1Ann:05	SubIndex 005	5. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x08 (Back Voltage State))	UINT32	RO	0x6nn0:08, 1
1Ann:06	SubIndex 006	6. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x09 (Wirebreak))	UINT32	RO	0x6nn0:09, 1
1Ann:07	SubIndex 007	7. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1Ann:08	SubIndex 008	8. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x0F (TxPDO State))	UINT32	RO	0x6nn0:0F, 1
1Ann:09	SubIndex 009	9. PDO Mapping entry (object 0x18nn, entry 0x09)	UINT32	RO	0x18nn:09, 1
1Ann:0A	SubIndex 010	10. PDO Mapping entry (object 0x6nn0 (TC Inputs Ch.01), entry 0x11 (Value))	UINT32	RO	0x6nn0:11, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

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

Index 1C13 TxPDO assign (for Ch. 1 - 32 (00 ≤ nn ≤ 1F))

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x1F (32 _{dec})
1C13:[nn]	Subindex 001 .. 032	nn. assigned TxPDO (contains the index of the corresponding TxPDO Mapping object)	UINT16	RW	0x1Ann

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameter of the inputs	UINT8	RO	0x20 (32 _{dec})
1C33:01	Sync mode	actual 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	Cycle time (in ns): <ul style="list-style-type: none"> • Free Run: cycle time of the local timer • Synchron with SM 2 Event: Cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle time 	UINT32	RW	0x000F4240 (1000000 _{dec})
1C33:03	Shift time	time between SYNC0-Event and Input Latch (in ns, only in DC-Mode)	UINT32	RW	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: Synchron with SM 2 Event is supported (Outputs available) • Bit 1: Synchron with SM 3 Event is supported (no Outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input Shift with local event (Outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no Outputs available) • Bit 14 = 1: dynamic times (could be measured by writing 0x1C33:08) 	UINT16	RO	0xC007 (49159 _{dec})
1C33:05	Minimum cycle time	Minimum cycle time supported (in ns)	UINT32	RO	0x00002710 (10000 _{dec})
1C33:06	Calc and copy time	time between Input Latch and the availability of the inputs for the master (in ns, only in DC-Mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:07	Minimum delay time	Minimum delay time	UINT32	RO	0x00000000 (0 _{dec})
1C33:08	Command	<ul style="list-style-type: none"> • 0: Measurement of the times will be stopped • 1: Measurement of the times will be started <p>The Entries 0x1C33:03, 0x1C33:06, 0x1C33:09 will be updated with the maximum measured values.</p>	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Delay time	Time between SYNC1 Event and Outputs Valid (in ns, only in DC-Mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	Number of the missed SM-Events in state OPERATIONAL (only in DC Mode)	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	Number of the exceeded cycles in state OPERATIONAL	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	Number of the too short distances between SYNC0 and SYNC1 Event (only in DC Mode)	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	TRUE: In the last cycle the synchronization was not correct (only in DC Mode)	BOOLEAN	RO	0x00 (0 _{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information about the Modular Device Profile	UINT8	RO	0x02 (2 _{dec})
F000:01	Module index distance	Index distance between the objects of two channels	UINT16	RO	0x0010 (16 _{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0020 (32 _{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word	The manufacturer reserves the authority for the basic calibration of the terminals. The Producer code-word is therefore at present reserved.	UINT32	RW	0x00000000 (0 _{dec})

Index F010 Module list (00 ≤ nn ≤ 1F)

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Max. subindex	UINT32	RW	0x020 (32 _{dec})
F010:[nn]	SubIndex 001 .. 032	TC Profil	UINT32	RW	0x000014A (330 _{dec})

5.5 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.

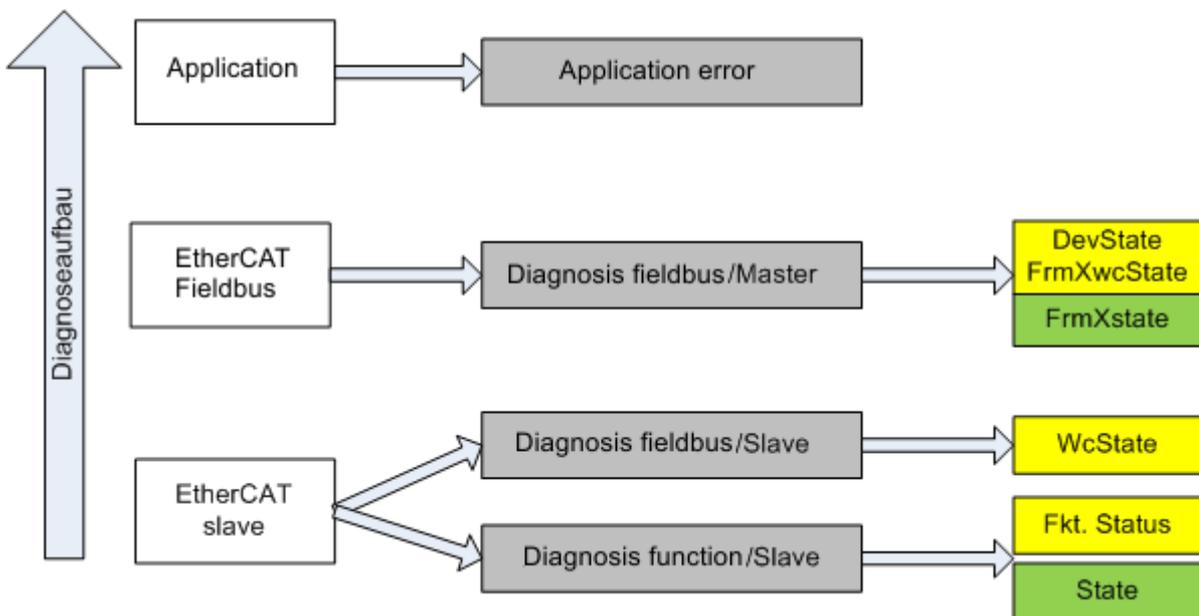


Fig. 96: Selection of the diagnostic information of an EtherCAT Slave

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.

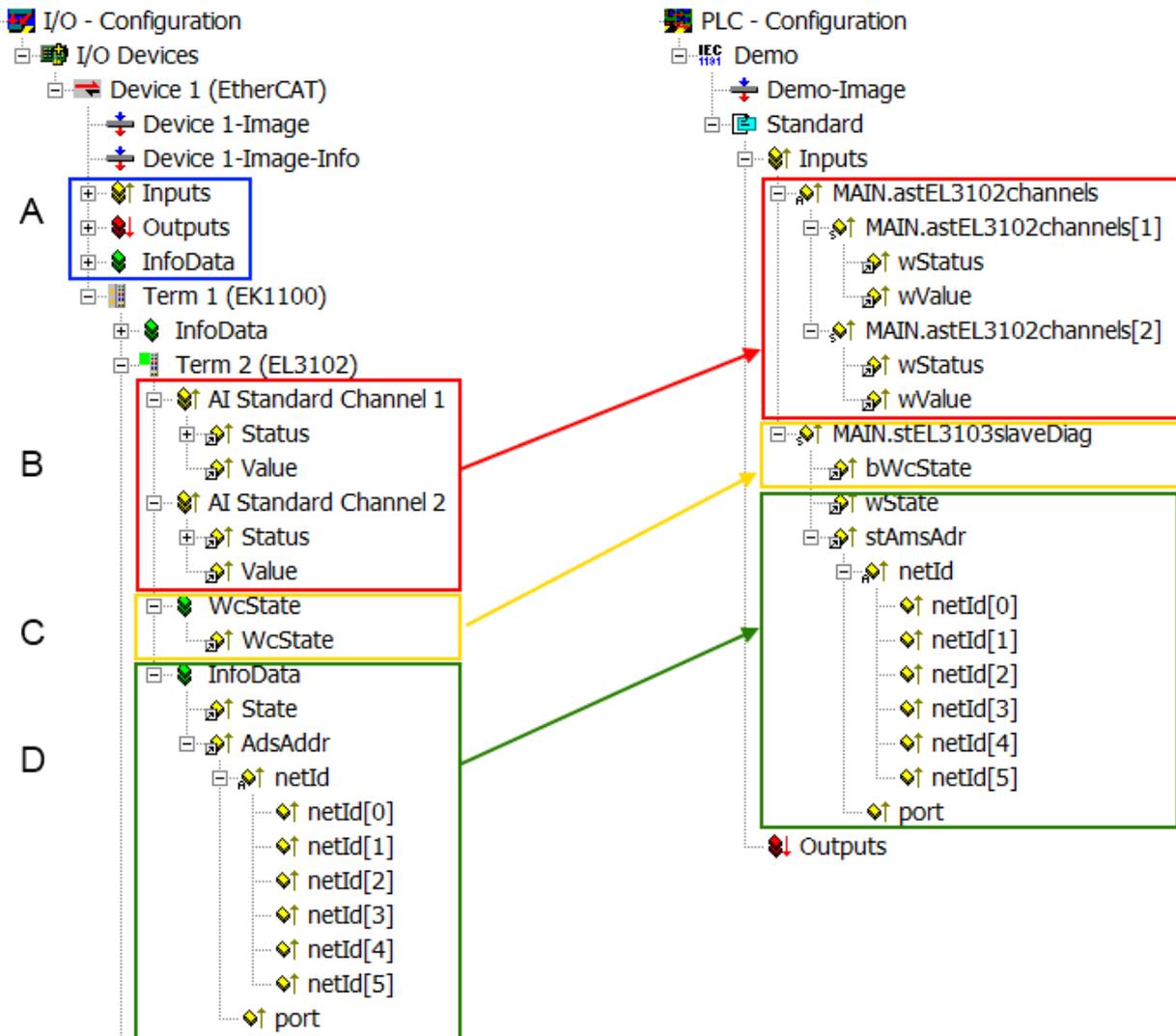


Fig. 97: Basic EtherCAT Slave Diagnosis in the PLC

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".

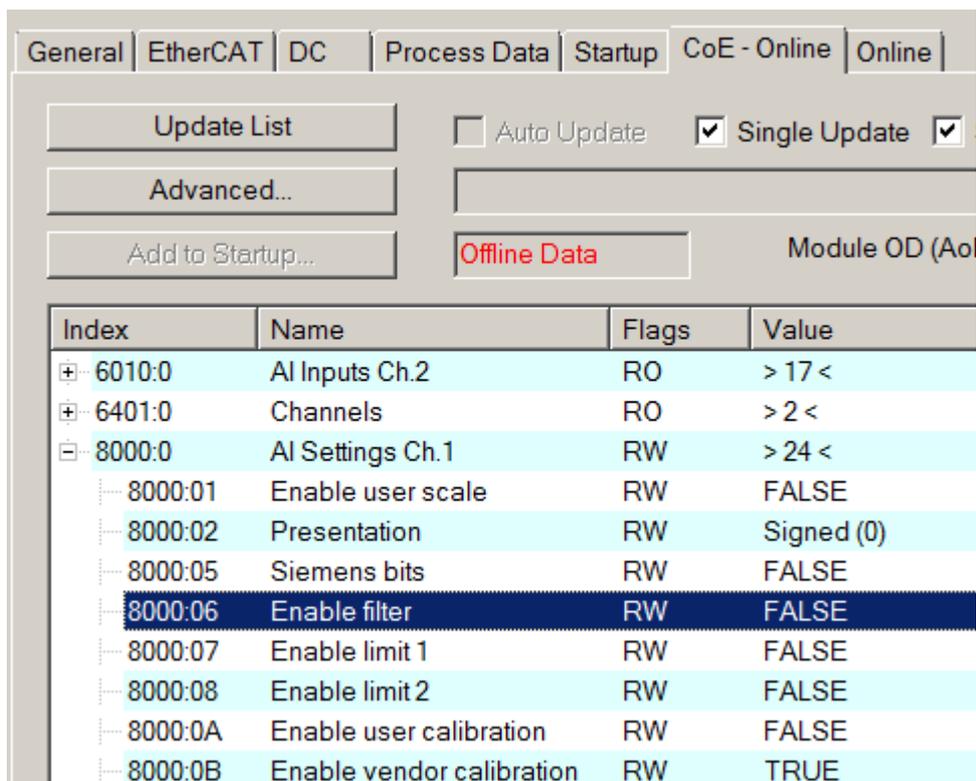


Fig. 98: EL3102, CoE directory

● EtherCAT System Documentation

i 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.

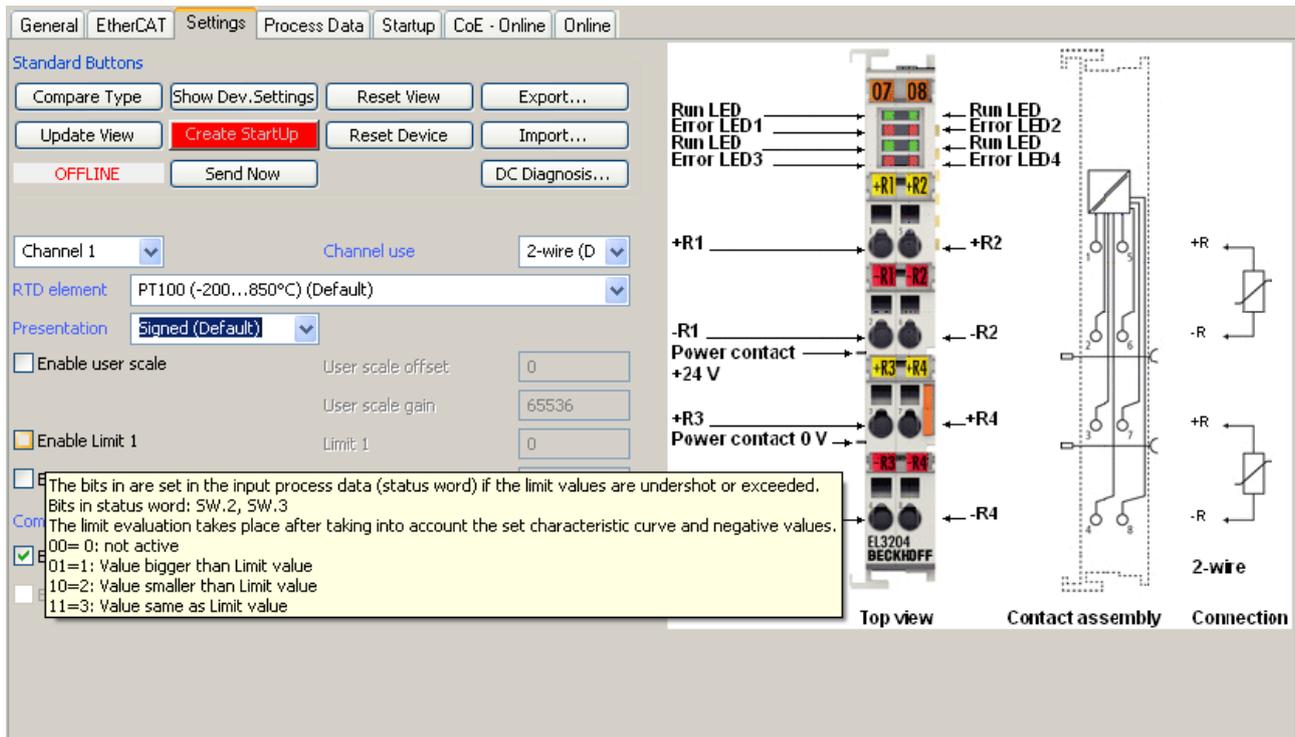


Fig. 99: Example of commissioning aid for a EL3204

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 \[15\]](#)" 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.

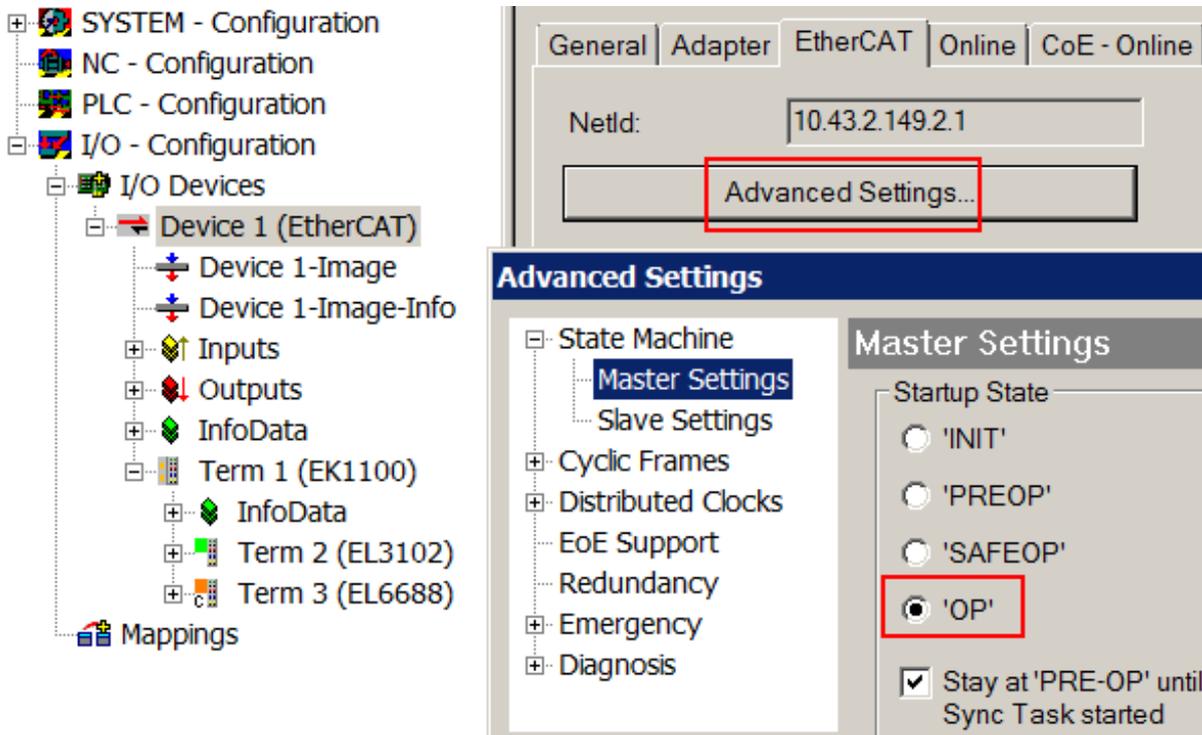


Fig. 100: Default behaviour of the System Manager

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

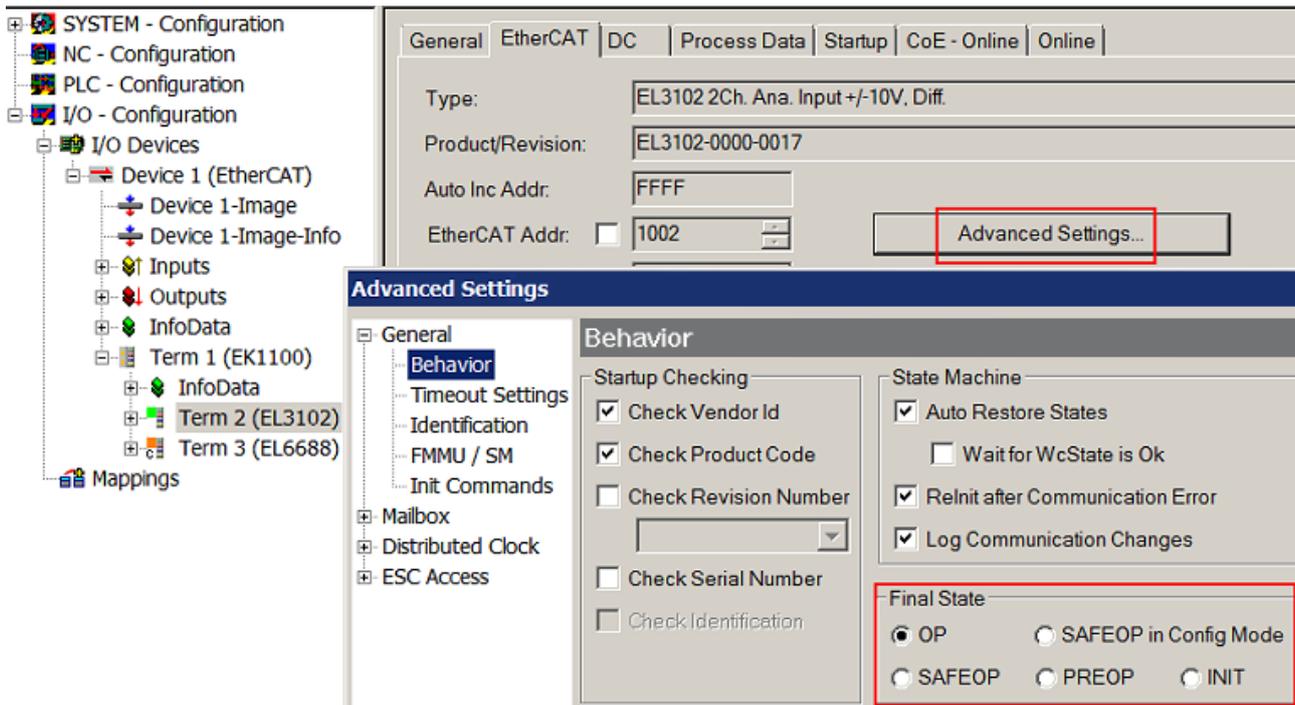


Fig. 101: Default target state in the Slave

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.

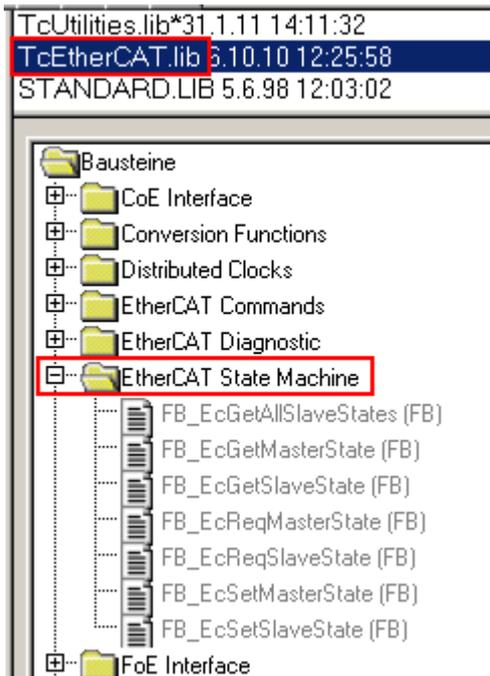


Fig. 102: PLC function blocks

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. 103: 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:

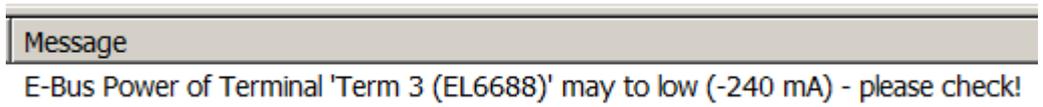


Fig. 104: Warning message for exceeding E-Bus current

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

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 will be equipped with the last available firmware on delivery. There are certain dependencies between firmware and hardware, not all combinations are compatible. The overview below shows which firmware is compatible to a certain hardware status.

Notice

- It is strictly recommended to apply the last available firmware on the particular hardware
- There is no customer right to get a firmware update at no charge for already delivered products by the Beckhoff Company.

NOTE

Damage of devices possible!

Please note the details concerning the firmware update on the special page. If you run a device in BOOT-STRAP mode to initiate a firmware update, it is possible that there is no checking of compatibility of the firmware. A damage of the device is possible! Please always make sure, that the firmware is suitable for the hardware status of the device!

FM3312-B110-x010			
Hardware (HW)	Firmware	Type	Prod. date
08*	07*	J	29.10.2015

FM3332-B110-x010			
Hardware (HW)	Firmware	Type	Prod. date
08*	07*	J	12.10.2015

*) At the time of creation of this documentation this is the current compatible hardware status. Please check the Beckhoff website for the latest [documentation](#).

6.3 Firmware Update EL/ES/EM/EPxxxx

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

Storage locations

An EtherCAT slave stores operating data in up to 3 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.de>). 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 3 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

- Firmware downloads to an EtherCAT device must not be interrupted
- Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
- 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:

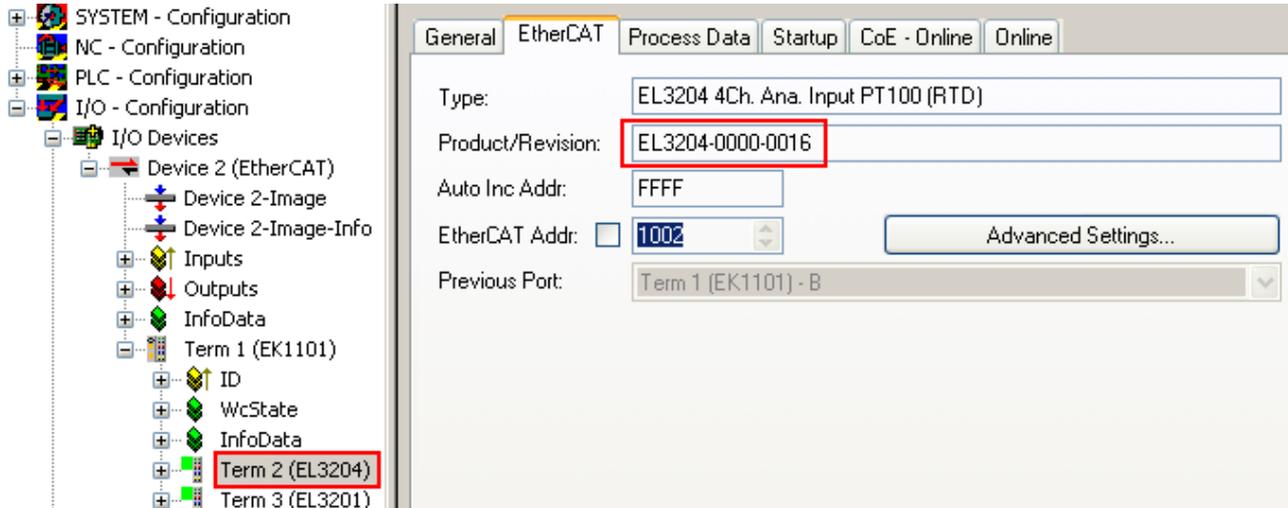


Fig. 105: Device identifier consisting of name EL3204-0000 and revision -0016

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:

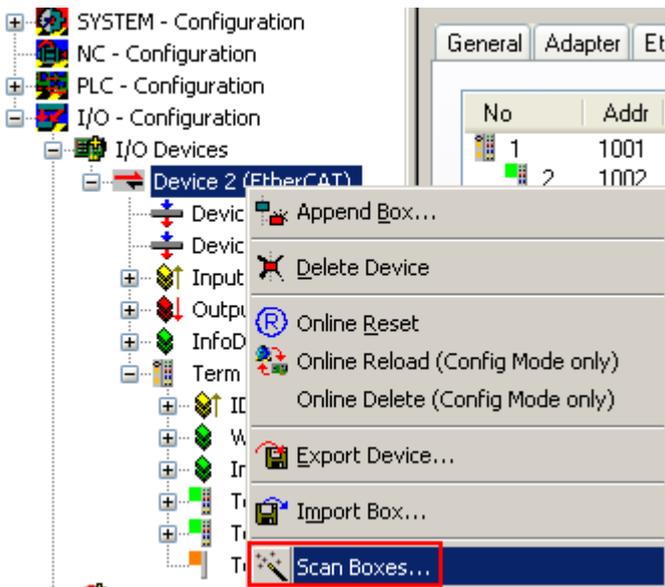


Fig. 106: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 107: Configuration is identical

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

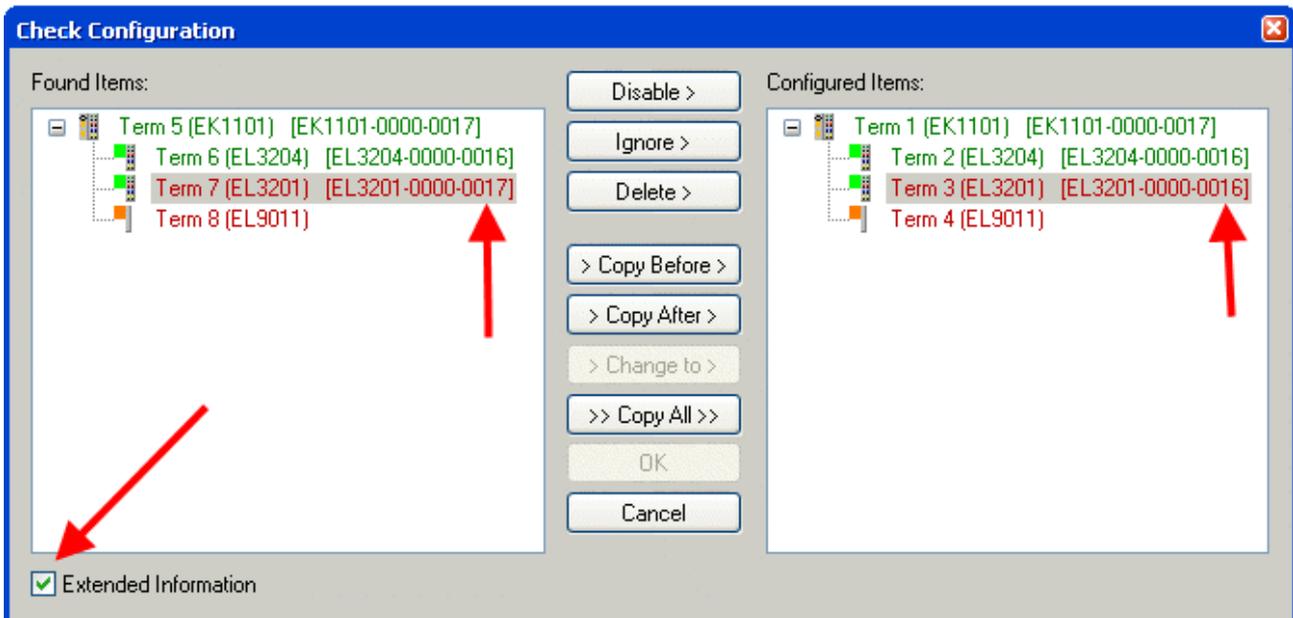


Fig. 108: Change dialog

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*

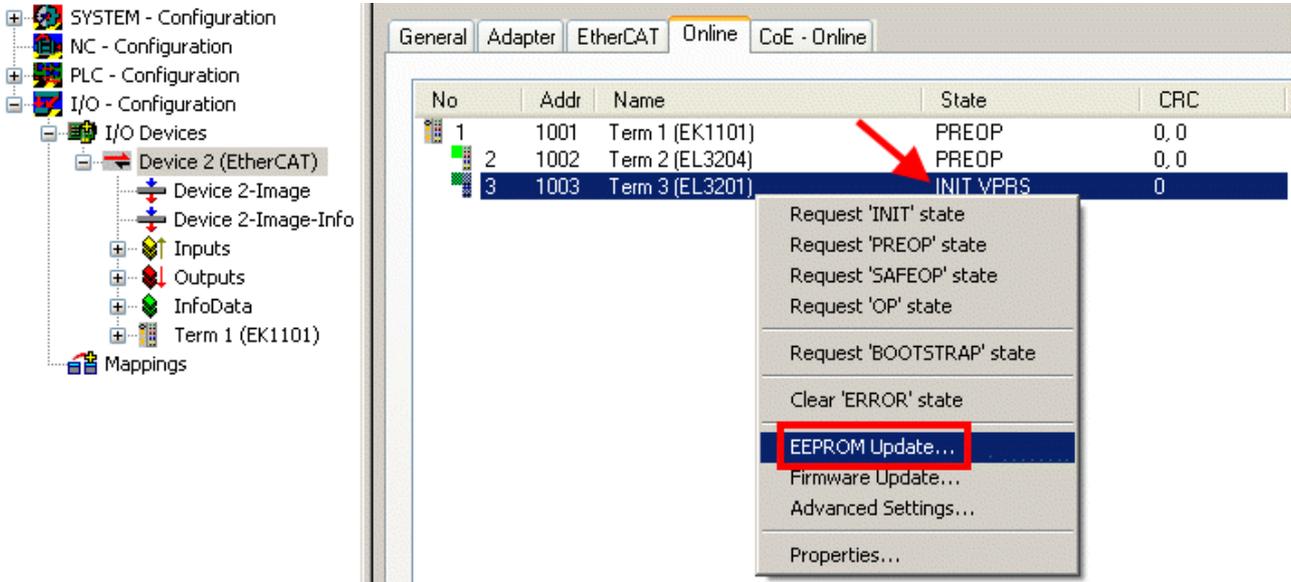


Fig. 109: EEPROM Update

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.

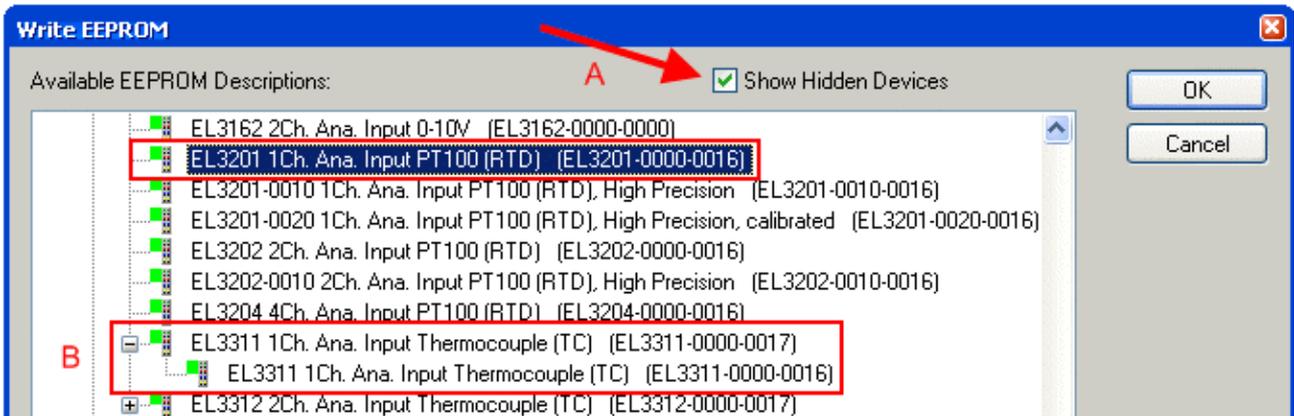


Fig. 110: Selecting the new ESI

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

● The change only takes effect after a restart.

i 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 on laser inscription

Beckhoff EtherCAT slaves feature serial numbers applied by laser. The serial number has the following structure: **KK YY FF HH**

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

Example with ser. no.: 12 10 03 02:

- 12 - week of production 12
- 10 - year of production 2010
- 03 - firmware version 03
- 02 - hardware version 02

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).

● CoE Online and Offline CoE

- i** 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.

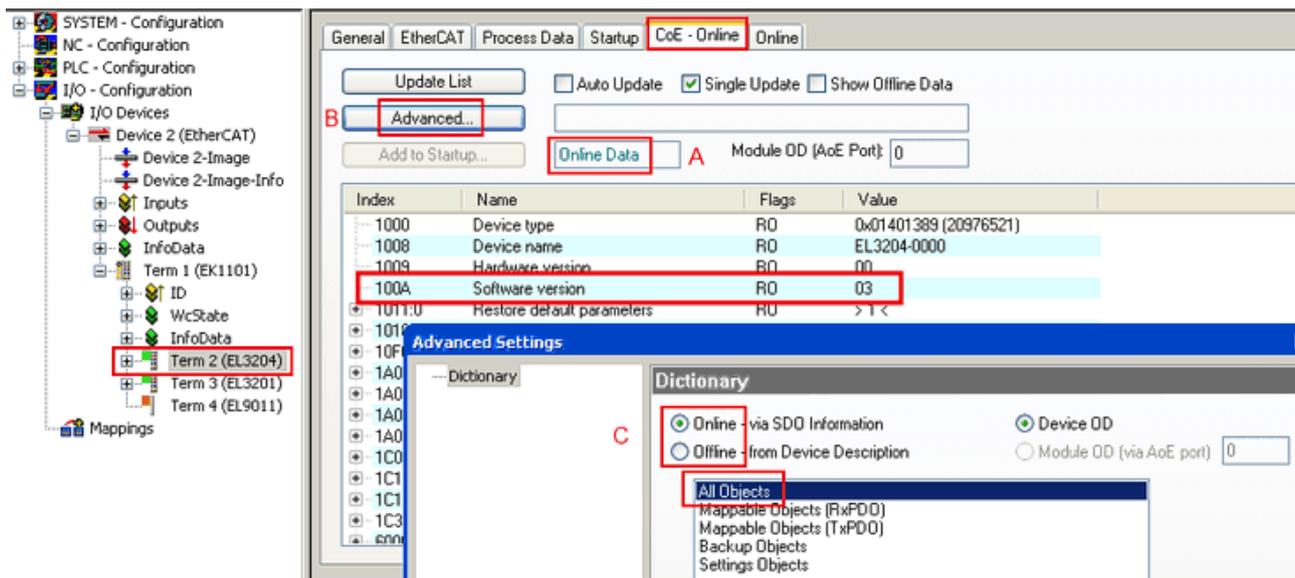


Fig. 111: Display of EL3204 firmware version

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 *All Objects*.

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*.

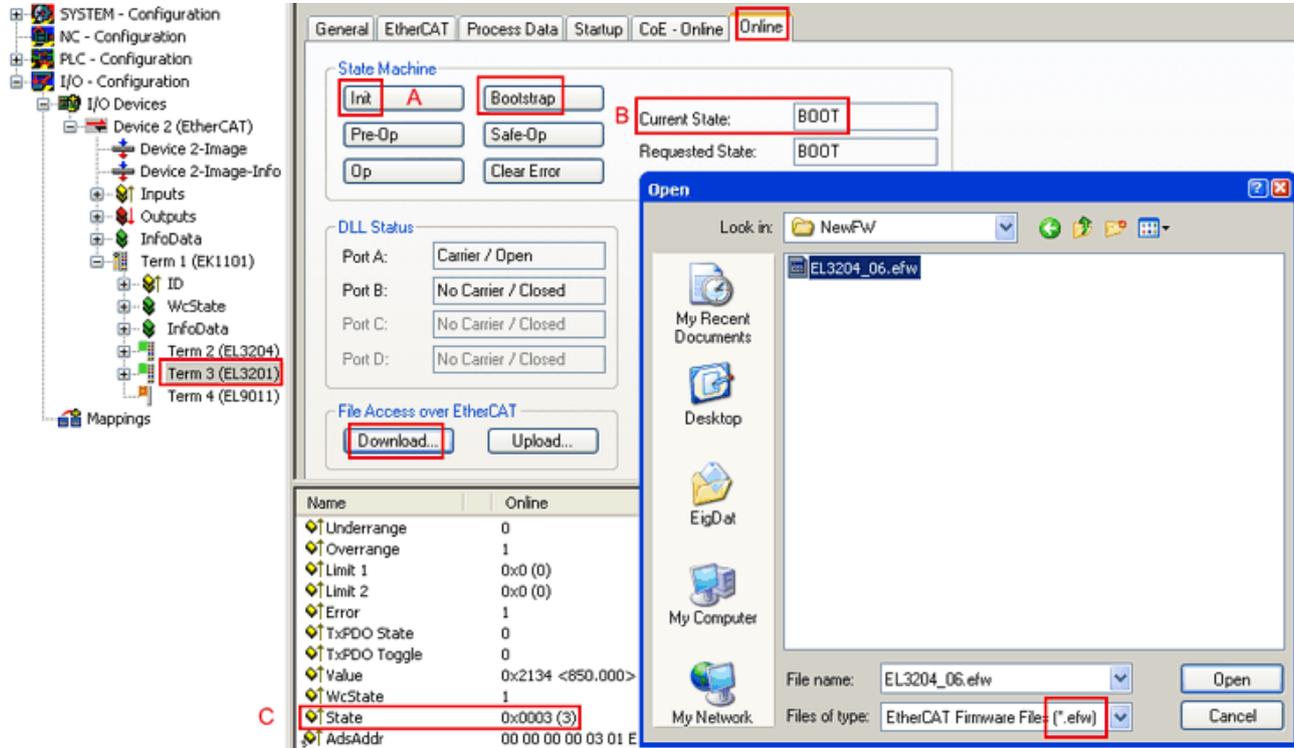
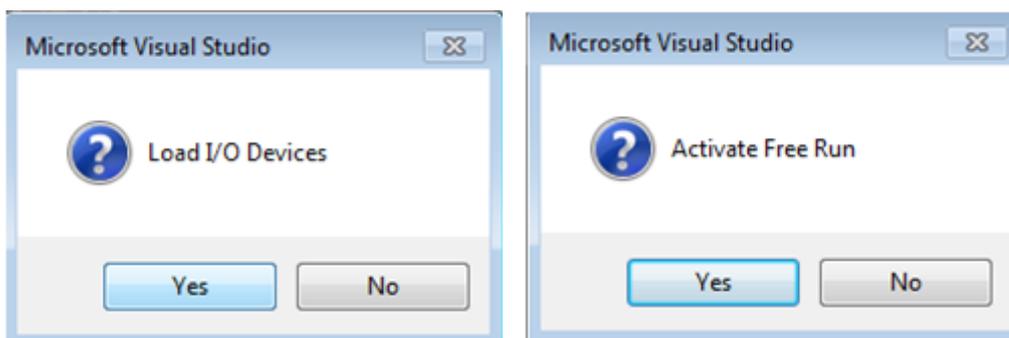


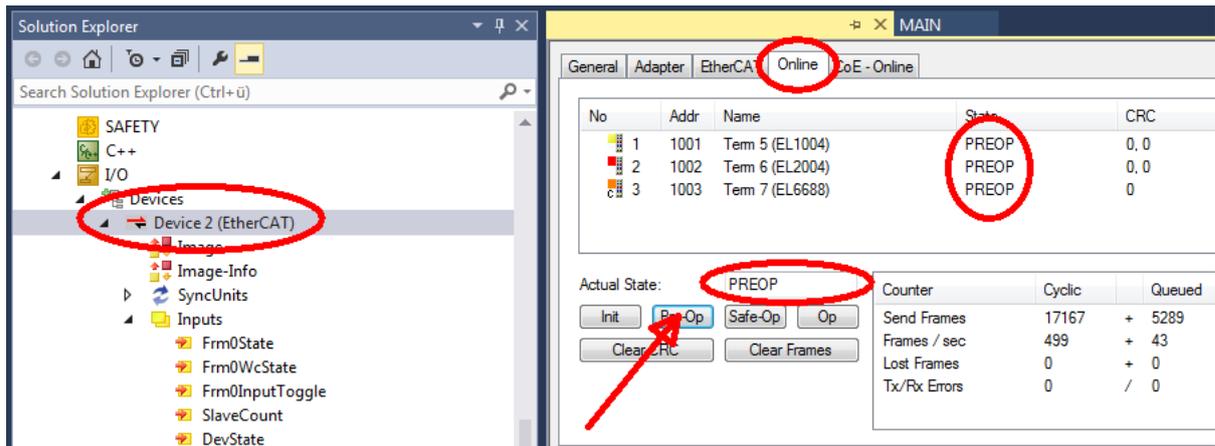
Fig. 112: Firmware Update

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

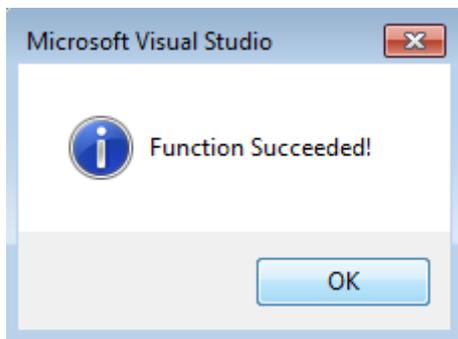
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



- 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 pass word 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.

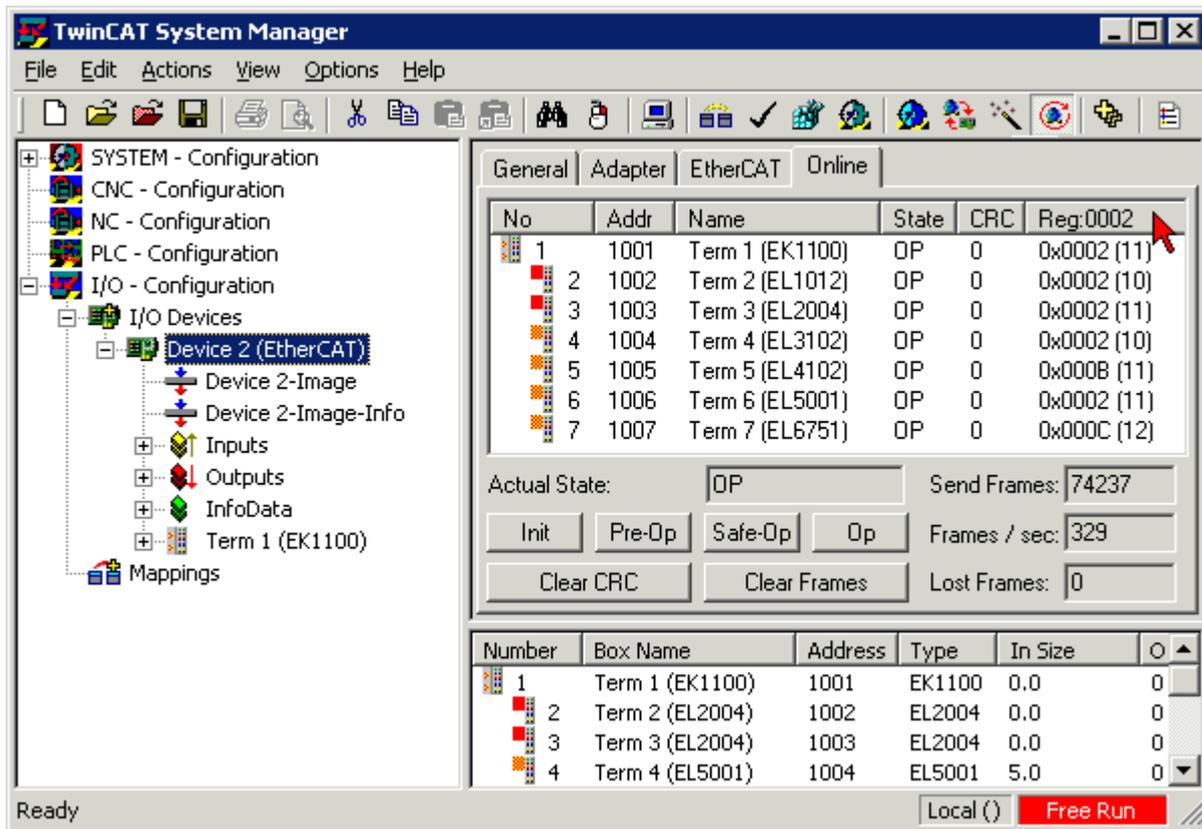
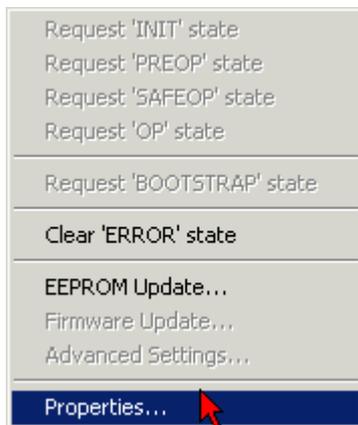


Fig. 113: FPGA firmware version definition

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

Fig. 114: 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.

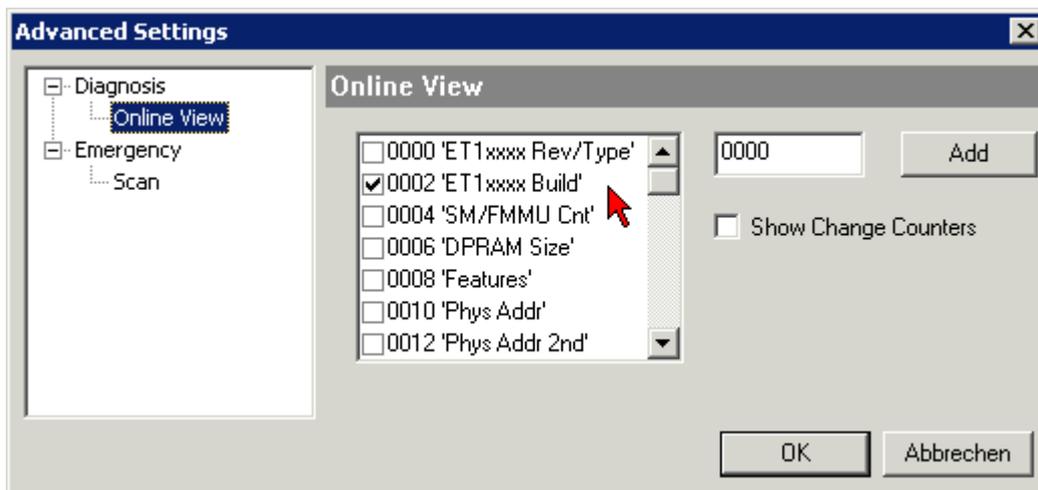


Fig. 115: Dialog *Advanced Settings*

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.

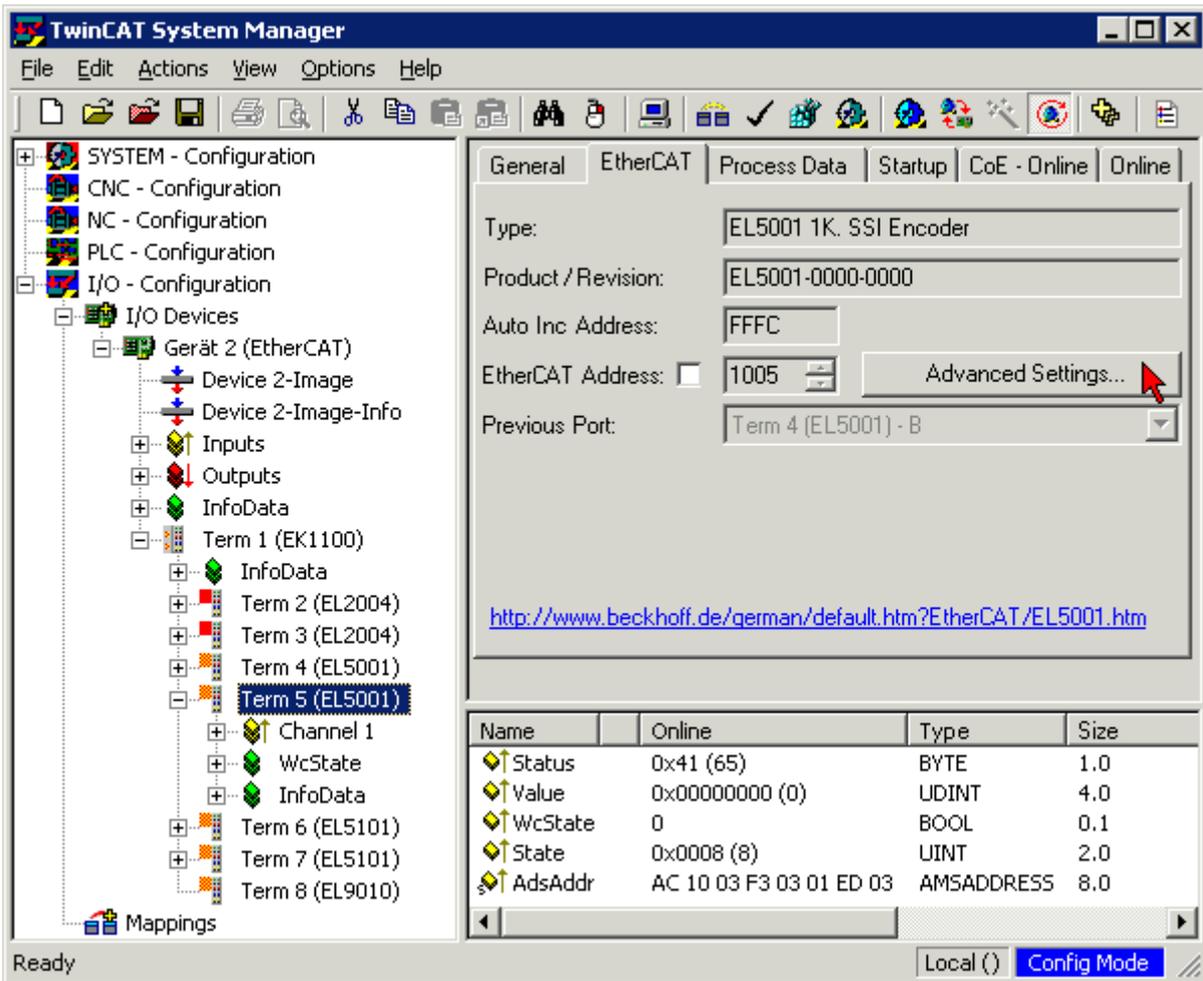
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

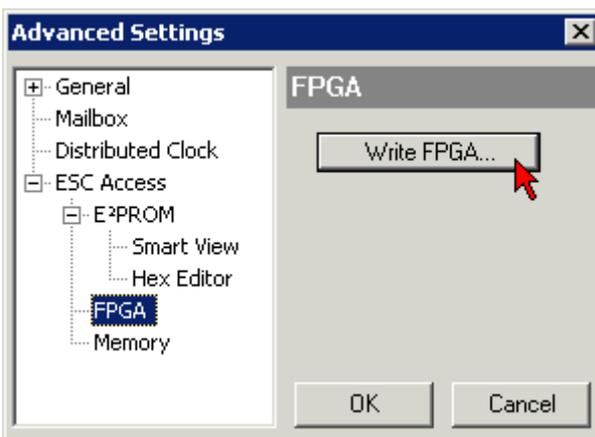
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

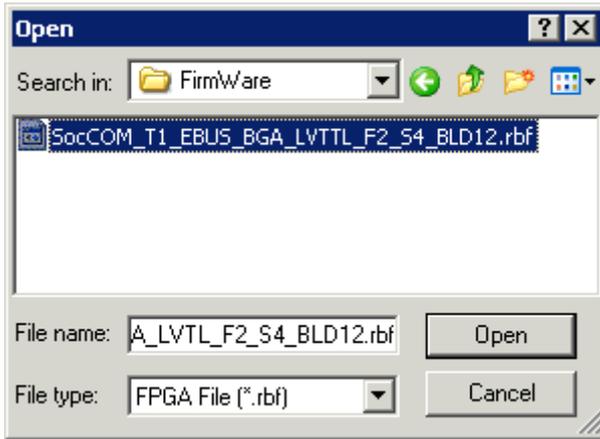
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- 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.

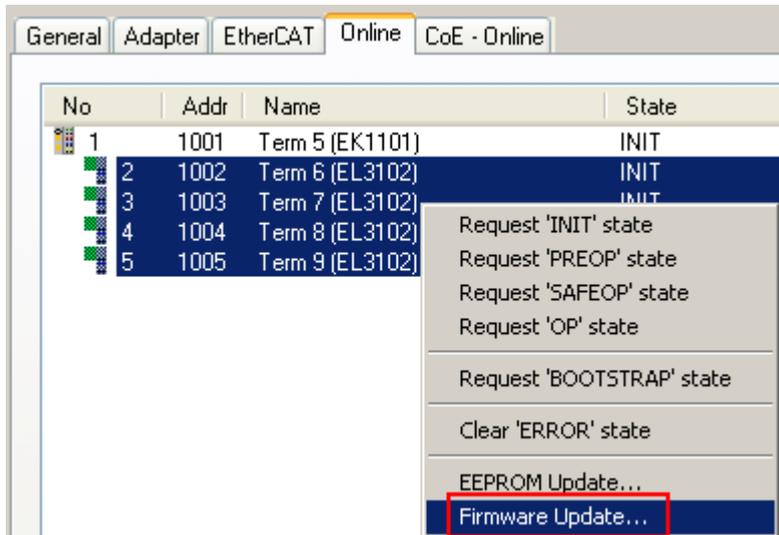


Fig. 116: Multiple selection and firmware update

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 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*)

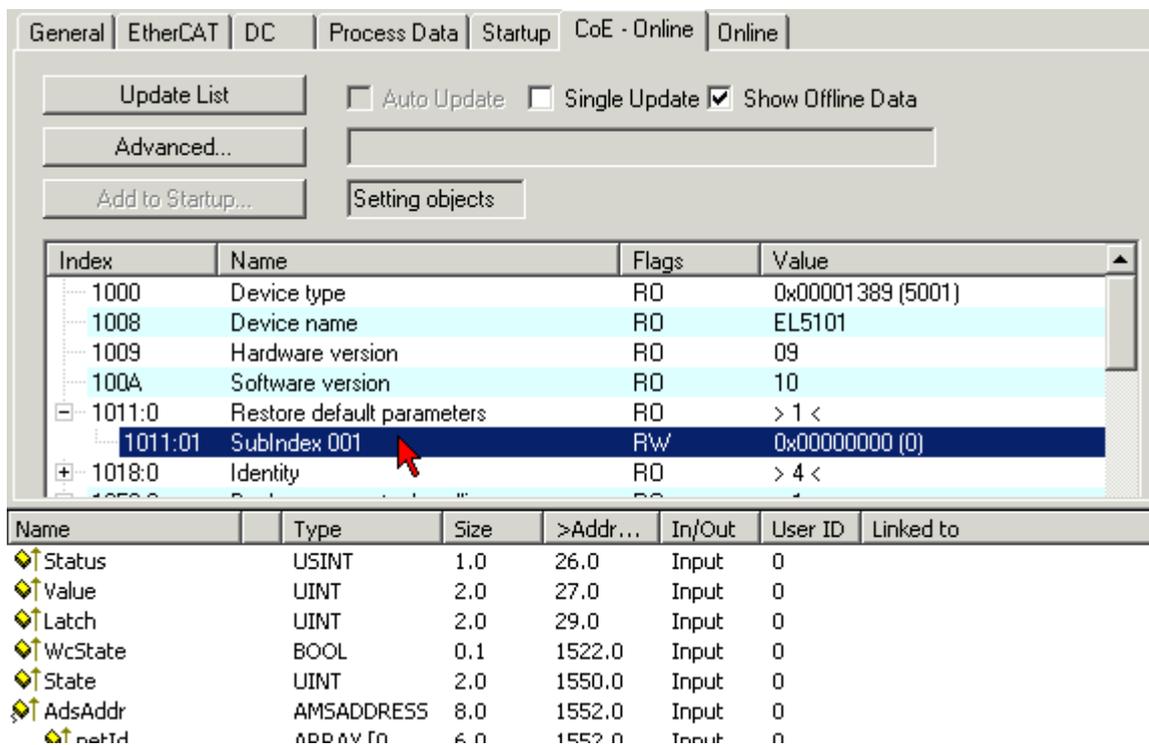


Fig. 117: Selecting the "Restore default parameters" PDO

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.

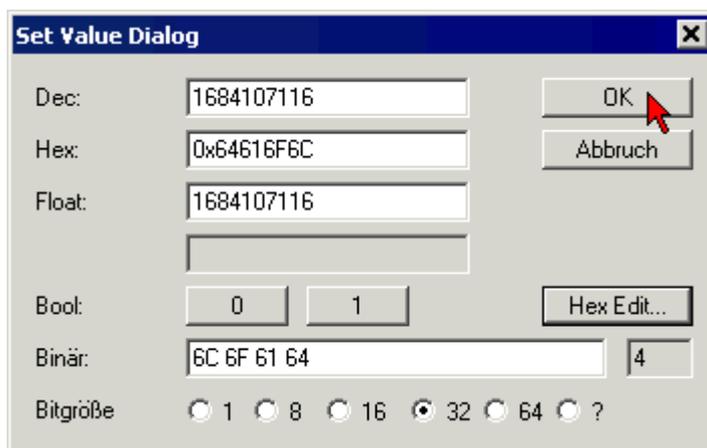


Fig. 118: Entering a restore value in the Set Value dialog

● 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:

<http://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

Beckhoff Support

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

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

Hotline:	+49(0)5246/963-157
Fax:	+49(0)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(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

Table of figures

Fig. 1	EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01).....	9
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number.....	10
Fig. 3	CU2016 switch with serial/ batch number.....	10
Fig. 4	EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418	10
Fig. 5	EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102.....	11
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070	11
Fig. 7	EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701.....	11
Fig. 8	ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201.....	11
Fig. 9	Thermocouple input plug	12
Fig. 10	States of the EtherCAT State Machine.....	15
Fig. 11	"CoE Online " tab	17
Fig. 12	Startup list in the TwinCAT System Manager	18
Fig. 13	Offline list.....	19
Fig. 14	Online list	20
Fig. 15	Top view TC Modul	22
Fig. 16	Blinking code of the "TERR"-LED	23
Fig. 17	Connection supply voltage.....	24
Fig. 18	Connection fieldbus	24
Fig. 19	Connection strip FM3312-B110-0010	25
Fig. 20	Connection strip FM3332-B110-1010	25
Fig. 21	ZK1090-3131-0xxx	27
Fig. 22	ZK2020-3132-0xxx	28
Fig. 23	Selection of Beckhoff sensor cables.....	28
Fig. 24	Principle of the thermocouple	31
Fig. 25	Process data FM3332-B110-1010	34
Fig. 26	Structure FM33xx	36
Fig. 27	Dialog "Scan devices".....	38
Fig. 28	Information.....	38
Fig. 29	Confirmation Device "EtherCAT"	39
Fig. 30	Dialog "Scan for boxes"	39
Fig. 31	Dialog "Activate Free Run.....	39
Fig. 32	Box in TwinCAT tree.....	40
Fig. 33	WCState	41
Fig. 34	InfoData	41
Fig. 35	CoE object 8000	42
Fig. 36	Process data PDO assignment.....	43
Fig. 37	Object "Coldjunction compensation".....	44
Fig. 38	Object "Filter settings".....	44
Fig. 39	Object "TC element"	45
Fig. 40	System Manager "Options" (TwinCAT 2).....	47
Fig. 41	Call up under VS Shell (TwinCAT 3)	47

Fig. 42	Overview of network interfaces	47
Fig. 43	EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter”	48
Fig. 44	Windows properties of the network interface.....	48
Fig. 45	Exemplary correct driver setting for the Ethernet port	49
Fig. 46	Incorrect driver settings for the Ethernet port	50
Fig. 47	TCP/IP setting for the Ethernet port	51
Fig. 48	Identifier structure	52
Fig. 49	OnlineDescription information window (TwinCAT 2)	53
Fig. 50	Information window OnlineDescription (TwinCAT 3).....	53
Fig. 51	File OnlineDescription.xml created by the System Manager	54
Fig. 52	Indication of an online recorded ESI of EL2521 as an example.....	54
Fig. 53	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3).....	54
Fig. 54	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	56
Fig. 55	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3).....	56
Fig. 56	Selecting the Ethernet port	56
Fig. 57	EtherCAT device properties (TwinCAT 2)	57
Fig. 58	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3).....	57
Fig. 59	Selection dialog for new EtherCAT device	58
Fig. 60	Display of device revision	58
Fig. 61	Display of previous revisions	59
Fig. 62	Name/revision of the terminal.....	59
Fig. 63	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3).....	60
Fig. 64	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	61
Fig. 65	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	61
Fig. 66	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	61
Fig. 67	Detected Ethernet devices	62
Fig. 68	Example default state	62
Fig. 69	Installing EthetCAT terminal with revision -1018	63
Fig. 70	Detection of EtherCAT terminal with revision -1019.....	63
Fig. 71	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	63
Fig. 72	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	64
Fig. 73	Scan progressexemplary by TwinCAT 2	64
Fig. 74	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	64
Fig. 75	Displaying of “Free Run” and “Config Mode” toggling right below in the status bar	64
Fig. 76	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)	64
Fig. 77	Online display example	65
Fig. 78	Faulty identification	65
Fig. 79	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	66
Fig. 80	Correction dialog	66
Fig. 81	Name/revision of the terminal.....	67
Fig. 82	Correction dialog with modifications	68
Fig. 83	Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3).....	68
Fig. 84	TwinCAT 2 Dialog Change to Alternative Type	68
Fig. 85	Configuring the process data	70

Fig. 86	Branch element as terminal EL3751	70
Fig. 87	“General” tab	71
Fig. 88	„EtherCAT“ tab	71
Fig. 89	“Process Data” tab	72
Fig. 90	Configuring the process data	73
Fig. 91	„Startup“ tab	74
Fig. 92	“CoE – Online” tab	75
Fig. 93	Dialog “Advanced settings”	76
Fig. 94	„Online“ tab	76
Fig. 95	“DC” tab (Distributed Clocks)	77
Fig. 96	Selection of the diagnostic information of an EtherCAT Slave	86
Fig. 97	Basic EtherCAT Slave Diagnosis in the PLC	87
Fig. 98	EL3102, CoE directory	89
Fig. 99	Example of commissioning aid for a EL3204	90
Fig. 100	Default behaviour of the System Manager	91
Fig. 101	Default target state in the Slave	91
Fig. 102	PLC function blocks	92
Fig. 103	Illegally exceeding the E-Bus current	93
Fig. 104	Warning message for exceeding E-Bus current	93
Fig. 105	Device identifier consisting of name EL3204-0000 and revision -0016	96
Fig. 106	Scan the subordinate field by right-clicking on the EtherCAT device	97
Fig. 107	Configuration is identical	97
Fig. 108	Change dialog	97
Fig. 109	EEPROM Update	98
Fig. 110	Selecting the new ESI	98
Fig. 111	Display of EL3204 firmware version	99
Fig. 112	Firmware Update	100
Fig. 113	FPGA firmware version definition	102
Fig. 114	Context menu Properties	102
Fig. 115	Dialog Advanced Settings	103
Fig. 116	Multiple selection and firmware update	105
Fig. 117	Selecting the "Restore default parameters" PDO	106
Fig. 118	Entering a restore value in the Set Value dialog	106