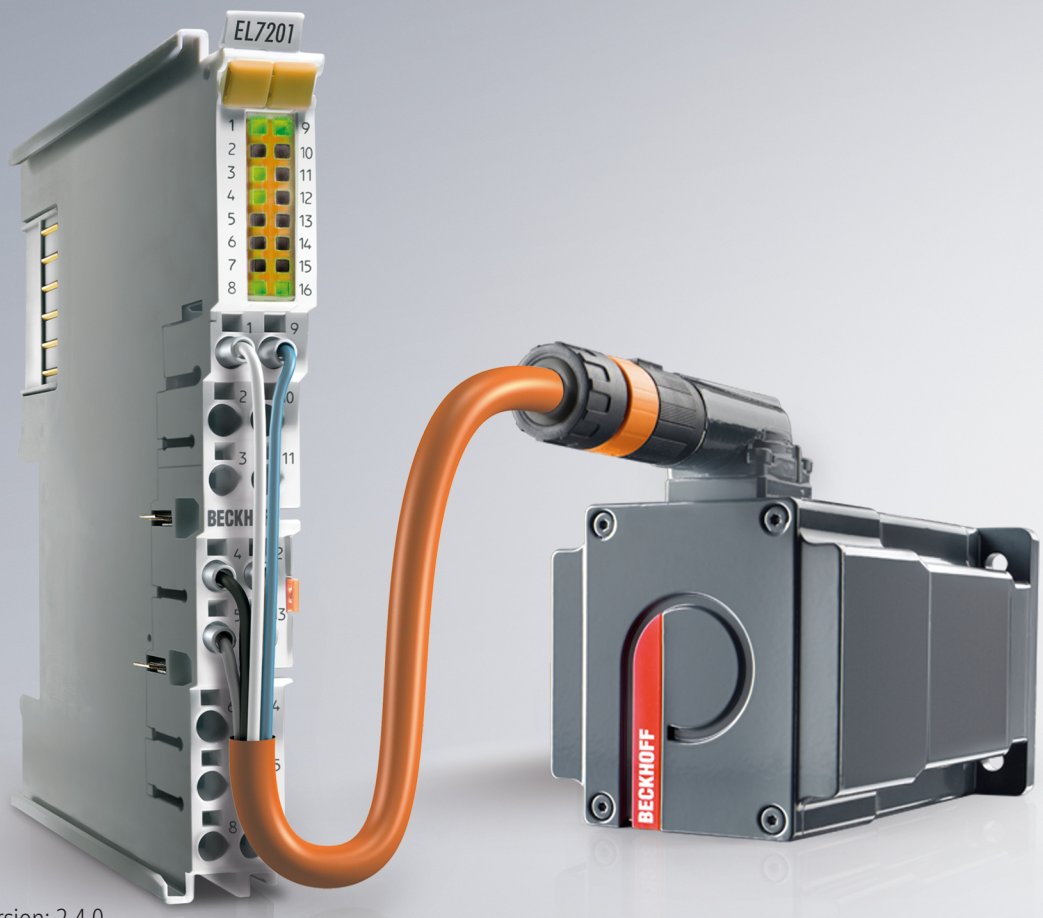
BECKHOFF New Automation Technology

Documentation | EN

EL72x1-901x

Servo Motor Terminals with OCT and STO, 48 V DC



2024-10-07 | Version: 2.4.0

Table of contents

1 Foreword 7
1.1 Notes on the documentation 7
1.2 Guide through documentation 9
1.3 Safety instructions 10
1.3.1 Delivery state 10
1.3.2 Operator's obligation to exercise diligence 10
1.3.3 Description of instructions 11
1.4 Documentation issue status 12
1.5 Version identification of EtherCAT devices 13
1.5.1 General notes on marking 13
1.5.2 Version identification of EL Terminals 14
1.5.3 Beckhoff Identification Code (BIC) 14
1.5.4 "Electronic access to the BIC (eBIC)" 16

2 Product overview 18
2.1 Product overview Servomotor terminal with OCT and STO 18
2.2 Introduction 18
2.3 Technical data 21
2.4 Technology 23
2.5 Start-up 25

3 Basics communication 26
3.1 EtherCAT basics 26
3.2 EtherCAT cabling – wire-bound 26
3.3 General notes for setting the watchdog 28
3.4 EtherCAT State Machine 30
3.5 CoE Interface 32
3.6 Distributed Clock 38

4 Installation 39
4.1 Example configuration for temperature measurement 39
4.2 Safety instructions 39
4.3 Environmental conditions 39
4.4 Transport / storage 40
4.5 Control cabinet / terminal box 40
4.6 Notes on ESD protection 40
4.7 Mounting rail installation 42
4.8 Installation position for operation with or without fan 45
4.9 Positioning of passive terminals 48
4.10 Installation instructions for enhanced mechanical load capacity 49
4.11 Connection 50
4.11.1 Connection system 50
4.11.2 Wiring 52
4.11.3 Shielding 53
4.12 Note on power supply 54
4.13 Note on load voltage supply 54

4.14 Shielding, earthing ... 55
4.15 UL notice - Compact Motion ... 56
4.16 Notes on current measurements using Hall sensors ... 57
4.17 EL72x1-9014 - LEDs and connection ... 58
4.17.1 EL7201-901x - LEDs and connection ... 58
4.17.2 EL7211-901x, EL7221-901x - LEDs and connection ... 60
4.18 Disposal ... 62

5 Commissioning ... 63
5.1 TwinCAT Quick Start ... 63
5.1.1 TwinCAT 2 ... 66
5.1.2 TwinCAT 3 ... 76
5.2 TwinCAT Development Environment ... 89
5.2.1 Installation of the TwinCAT real-time driver ... 90
5.2.2 Notes regarding ESI device description ... 96
5.2.3 TwinCAT ESI Updater ... 100
5.2.4 Distinction between Online and Offline ... 102
5.2.5 OFFLINE configuration creation ... 102
5.2.6 ONLINE configuration creation ... 108
5.2.7 EtherCAT subscriber configuration ... 116
5.3 Start-up and parameter configuration ... 126
5.3.1 Integration into the NC configuration ... 126
5.3.2 Settings with the Drive Manager ... 129
5.3.3 Settings in the CoE register ... 135
5.3.4 NC settings ... 138
5.3.5 Application example ... 144
5.3.6 Commissioning without NC, status word/control word ... 150
5.3.7 Settings for the automatic configuration ... 153
5.3.8 Configuring the limit switch ... 155
5.3.9 Homing ... 156
5.3.10 Touch Probe ... 159
5.4 Operation modes ... 162
5.4.1 Overview ... 162
5.4.2 CSV ... 163
5.4.3 CST ... 166
5.4.4 CSTCA ... 169
5.4.5 CSP ... 172
5.5 Profile MDP742 or DS402 ... 175
5.6 MDP742 process data ... 177
5.7 DS402 Process data ... 183

6 Integrated safety ... 187
6.1 Safety regulations ... 187
6.2 Description of product and function ... 187
6.2.1 Intended use ... 187
6.2.2 Dimensions ... 188
6.2.3 TwinSAFE reaction times ... 189

6.2.4 Application example for STO function (Cat. 3, PL d) ... 190
6.3 Maintenance ... 197

7 Object description and parameterization ... 198
7.1 EL72x1-9014 (MDP742) ... 198
7.1.1 Restore object ... 198
7.1.2 Configuration data ... 199
7.1.3 Configuration data (vendor-specific) ... 206
7.1.4 Command object ... 206
7.1.5 Input data ... 206
7.1.6 Output data ... 210
7.1.7 Information / diagnosis data ... 213
7.1.8 Standard objects ... 216
7.2 EL72x1-9014 (DS402) ... 232
7.2.1 Configuration data ... 233
7.2.2 Configuration data (vendor-specific) ... 239
7.2.3 Command object ... 239
7.2.4 Input/output data ... 239
7.2.5 Information / diagnosis data ... 244
7.2.6 Standard objects ... 248

8 Error correction ... 255
8.1 Diagnostics - basic principles of diag messages ... 255

9 Appendix ... 265
9.1 EtherCAT AL Status Codes ... 265
9.2 Firmware compatibility ... 266
9.3 Firmware Update ... 268
9.3.1 Device description ESI file/XML ... 269
9.3.2 Firmware explanation ... 272
9.3.3 Updating controller firmware *.efw ... 273
9.3.4 FPGA firmware *.rbf ... 275
9.3.5 Simultaneous updating of several EtherCAT devices ... 279
9.4 Restoring the delivery state ... 280
9.5 Certificates ... 283
9.6 Support and Service ... 284

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 following notes and explanations are followed when installing and commissioning these components.

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.

Origin of the document

This is a translation of the original instructions which are written in German. All other languages are derived from the German original.

Product features

Only the product features specified in the current user documentation are valid. Further information given on the product pages of the Beckhoff homepage, in emails or in other publications is not authoritative.

Disclaimer

The documentation has been prepared with care. The products described are subject to cyclical revision. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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

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

Patent Pending

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



EtherCAT® and Safety over EtherCAT® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.

Copyright

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

Delivery conditions

In addition, the general delivery conditions of the company Beckhoff Automation GmbH & Co. KG apply.

Currentness

Check whether you are using the current and valid version of this document. The current version can be downloaded from the Beckhoff homepage at http://www.beckhoff.com/twinsafe. In case of doubt, contact Support and Service.

1.2 Guide through documentation

NOTICE



Further components of documentation

This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the device / devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
TwinSAFE application example (PDF)	Application example EP7211-0034 - STO Function
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

1.3 Safety instructions

1.3.1 Delivery state

All of the components are supplied in specific hardware and software configurations depending on the application requirements. 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.

This manual and the application examples it contains are intended solely for the purpose of conducting the acceptance testing of the delivered control system. The application examples included in the manual do not represent customer-specific solutions. They are intended solely to verify the functionality of the delivered application. The provision of these application examples does not relieve you of your responsibility for the proper operation of the delivered machine. These acceptance instructions do not in any way replace an operating manual or documentation. In the event of any conflict between these instructions and the published documentation, the documentation shall take precedence.

We make no warranties regarding the information provided in this document.

We assume no liability for damages resulting from the information contained in this document. This does not apply in cases of liability under the Product Liability Act, in cases of willful misconduct or gross negligence, liability for injury to life, body, or health, or liability for a breach of an essential contractual obligation. In the event of liability for a breach of material contractual obligations, liability is limited to the typically foreseeable damages.

1.3.2 Operator's obligation to exercise diligence

The operator must ensure that

- the TwinSAFE products are only used as intended (see chapter Product description);
- the TwinSAFE products are only operated in sound condition and in working order.
- the TwinSAFE products are operated only by suitably qualified and authorized personnel.
- the personnel is instructed regularly about relevant occupational safety and environmental protection aspects, and is familiar with the operating instructions and in particular the safety instructions contained herein.
- the operating instructions are in good condition and complete, and always available for reference at the location where the TwinSAFE products are used.
- none of the safety and warning notes attached to the TwinSAFE products are removed, and all notes remain legible.

1.3.3 Description of instructions

In these operating instructions 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.

NOTICE

Damage to the 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.

1.4 Documentation issue status

Version	Comment
2.4.0	• Update chapter “LEDs and connection” • Update revision status • Update structure
2.3	• Update chapter “Technical data” • Update revision status • Update structure
2.2	• Update chapter “Commissioning” • Update structure
2.1	• Update chapter “Version identification of EtherCAT devices” • Update chapter “Technical data” • Update chapter “Installation” • Update chapter “Process data” and “Object description” • Update revision status • Update structure
0.1 - 2.0	*archived*

1.5 Version identification of EtherCAT devices

1.5.1 General notes on marking

Name

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

- Family key
- Type
- Version
- Revision

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

Notes

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

1.5.2 Version identification of EL Terminals

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

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



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

1.5.3 Beckhoff Identification Code (BIC)

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

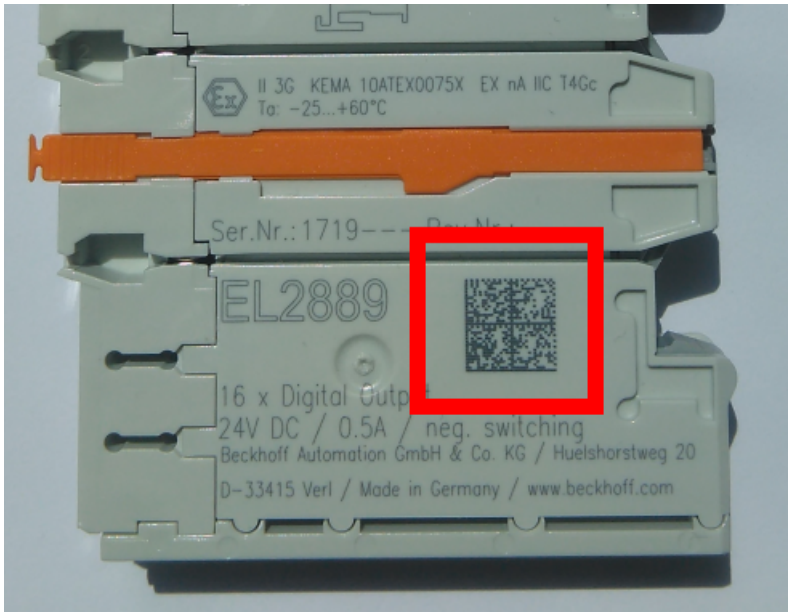


Fig. 2: BIC as a Data Matrix code (DMC, ECC200 code scheme)

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

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

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

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

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

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

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

Structure of the BIC

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

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

The same number as a DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

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

NOTICE

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

1.5.4 "Electronic access to the BIC (eBIC)"

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

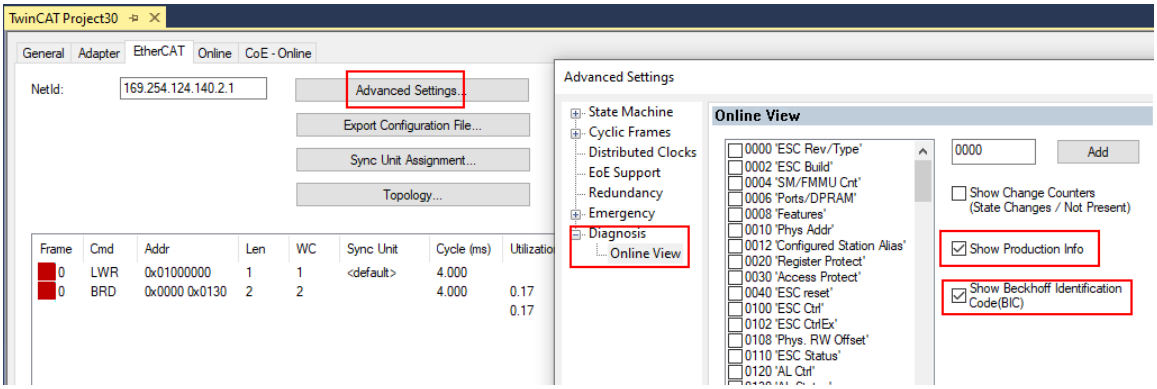
EtherCAT devices (IP20, IP67)

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

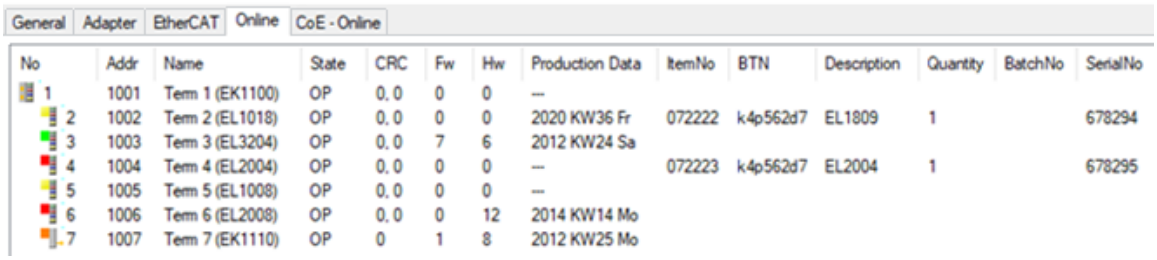
Beckhoff also writes the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

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

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

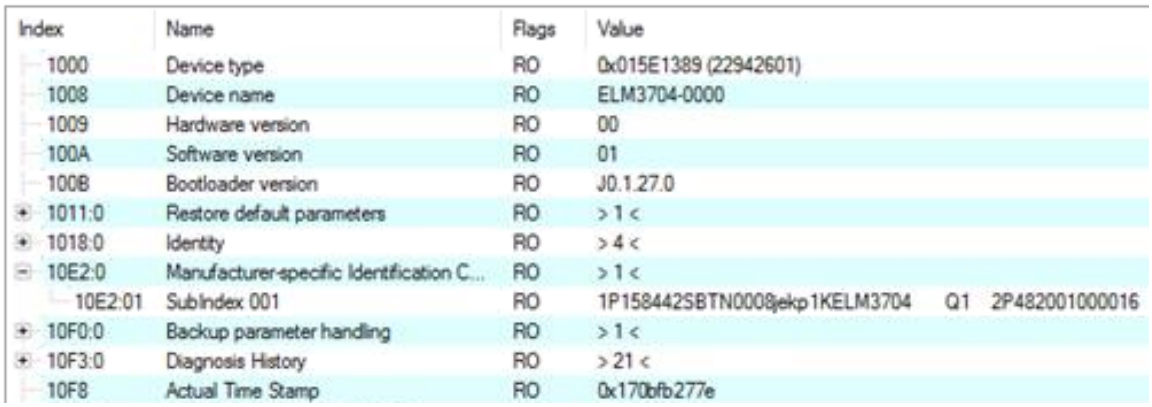


  - The BTN and its contents are then displayed:



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

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



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

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

PROFIBUS, PROFINET, DeviceNet® devices, etc.

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

2 Product overview

2.1 Product overview Servomotor terminal with OCT and STO

EL7201-9014 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 2.8 A_{rms}, MDP742 profile

EL7201-9015 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 2.8 A_{rms}, DS402 profile

EL7211-9014 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 4.5 A_{rms}, MDP742 profile

EL7211-9015 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 4.5 A_{rms}, DS402 profile

EL7221-9014 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 7…8 A_{rms}, MDP742 profile

EL7221-9015 [▶ 18] servo motor terminal with OCT and STO, 48 V_{DC}, 7…8 A_{rms}, DS402 profile

2.2 Introduction

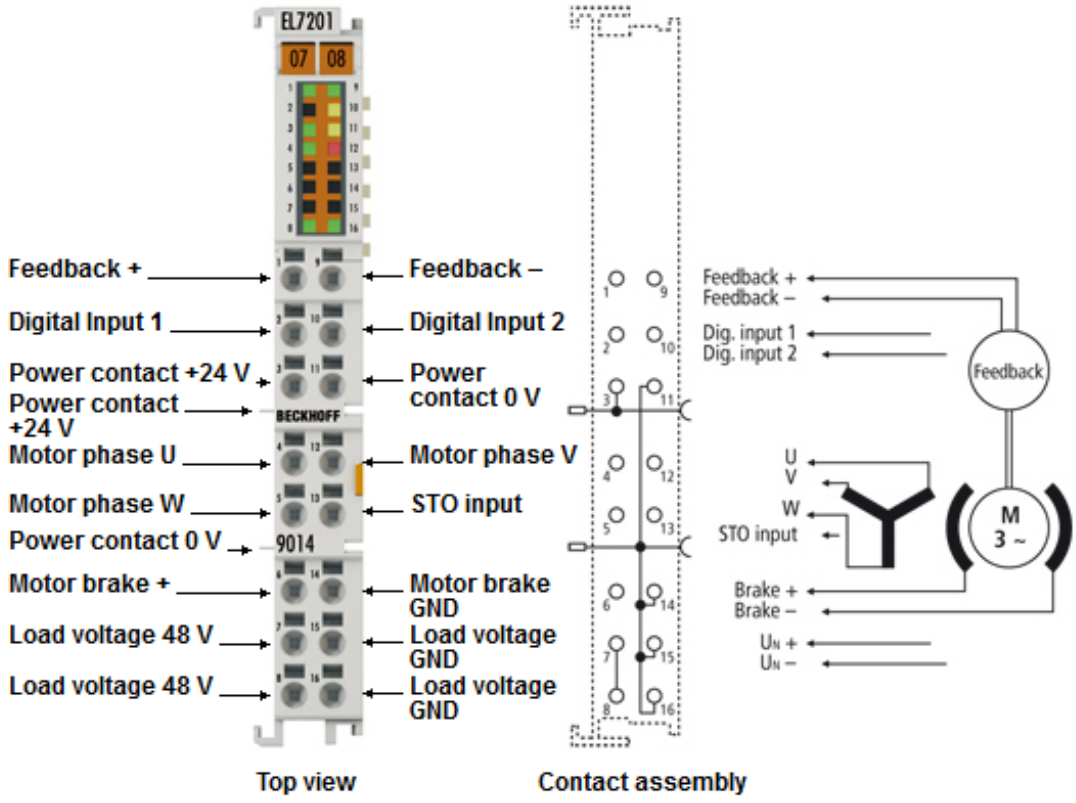


Fig. 4: EL7201-901x

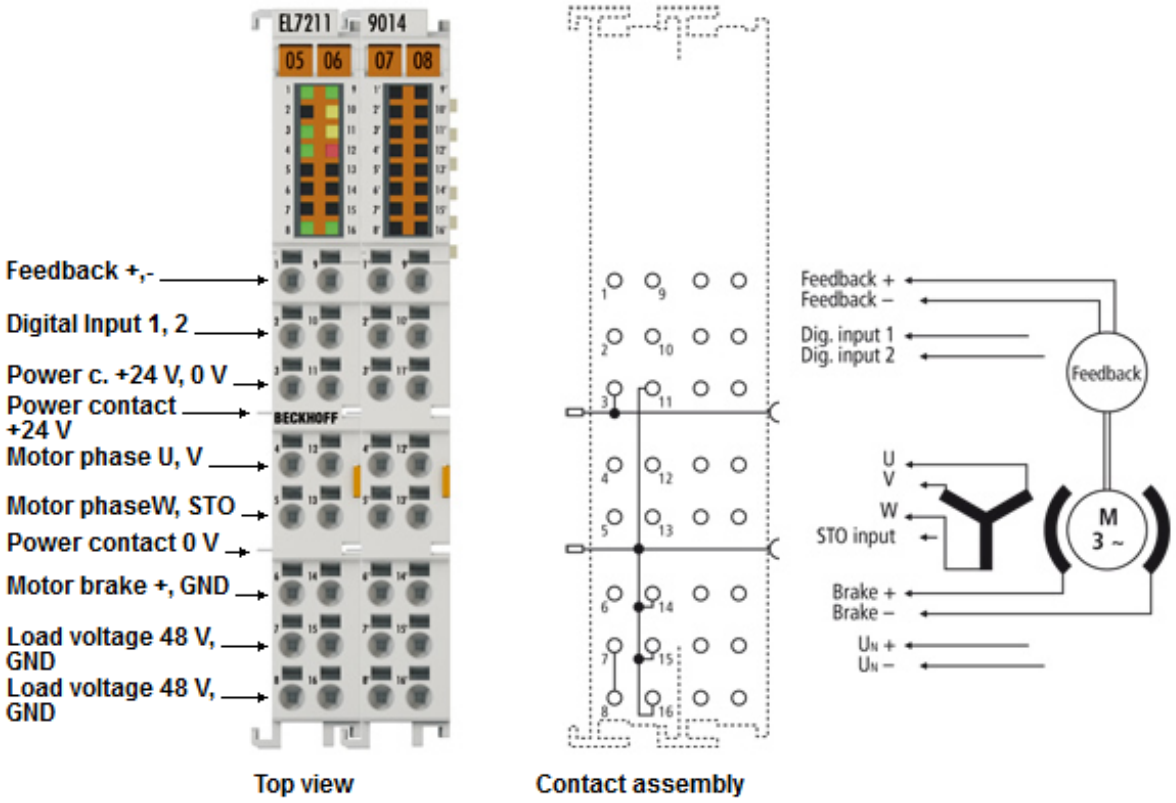


Fig. 5: EL7211-901x, EL7221-901x

Servo motor terminals with OCT and STO input

The servo-motor EtherCAT terminals EL7201-901x (48 V_{DC}, 2.8 A_{rms}), EL7211-901x (48 V_{DC}, 4.5 A_{rms}) and EL7221-901x (48 V_{DC}, 7 … 8 A_{rms}) with integrated absolute value interface, offer high servo performance in a very compact design. The EL72x1-901x were designed for the motor types of the AM81xx series from Beckhoff Automation.

The fast control technology, based on field-orientated current and PI speed control, supports fast and highly dynamic positioning tasks. The monitoring of numerous parameters, such as overvoltage and undervoltage, overcurrent, terminal temperature or motor load via the calculation of an I²T model, offers maximum operational reliability.

EtherCAT, as a high-performance system communication, and CAN-over-EtherCAT (CoE), as the application layer, enable ideal interfacing with PC-based control technology.
The latest power semiconductors guarantee minimum power loss and enable feedback into the DC link when braking.

The LEDs indicate status, warning and error messages as well as possibly active limitations.

With the One Cable Technology (OCT) the encoder cable is omitted by transmitting the signals of the encoder digitally via the existing motor cable. The option to read the electronic type plates of suitable motors from the AM81xx series enables a plug-and-play solution for maximum convenience during commissioning.

The EL72x1-901x provides an STO input with which the motor connected to the terminal can be switched torque-free. This STO input is connected to a safe output of an EL2904.

Performance Level d, Category 3 according to DIN EN ISO 13849-1:2015 is attained for the SFO safety function of the EL72x1-901x together with an EL2904.

Recommended TwinCAT version

In order to be able to utilize the full power of the EL72x1-901x, we recommend using the EL72x1-901x with TwinCAT 2.11 R3 or higher!

Mandatory hardware

The EL72x1-901x must be operated with a real-time capable computer and distributed clocks!

Approved motors

The EL72x1-901x may be operated only with the following Beckhoff motors.

- AM8111-xF1x, AM8112-xF1x, AM8113-xF1x, AM8121-xF1x, AM8122-xF1x, AM8131-xF1x, AM8132-xJ1x, AM8133-xJ1x, AM8141-xJ1x

- AM8111-xF2x, AM8112-xF2x, AM8113-xF2x, AM8121-xF2x, AM8122-xF2x, AM8131-xF2x, AM8132-xJ2x, AM8133-xJ2x, AM8141-xJ2x

Operation of the EL7221-901x with fan cartridge ZB8610

Due to the increased thermal load, the EL7221-091x must only be operated in conjunction with the fan cartridge ZB8610 in order to avoid malfunctions.

Quick links

Connection instructions

- Chapter "Mounting and wiring",
  - LEDs and pin assignment [▶ 58]
  - Shielding concept [▶ 55]
  - Notes on current measurement via Hall sensor [▶ 57]

Configuration instructions

- Chapter "Commissioning",
  - Configuration of the main parameters [▶ 126]
- Chapter "Configuration with the TwinCAT System Manager",
  - Object description and parameterization [▶ 198]

Application example

- Chapter "Commissioning",
  - Application example [▶ 144]

2.3 Technical data

Technical data	EL7201-901x	EL7211-901x	EL7221-901x
Number of outputs	3 motor phases, 2 motor holding brake		
Number of inputs	2 (4) DC link voltage, 2 absolute feedback, 2 digital inputs. 1 STO input		
DC link supply voltage	8 ... 48 V_{DC}		
STO input response time	10 ms		
Supply voltage	24 V_{DC} via the power contacts / via the E-bus		
Output current	2.8 A_{rms} (without fan cartridge ZB8610) 4.5 A_{rms} (with fan cartridge ZB8610)	4.5 A_{rms}	7 A_{rms} up to 55°C (with fan cartridge ZB8610) 8 A_{rms} up to 45°C (with fan cartridge ZB8610)
Peak current	5.7 A_{rms} for 1 second (without fan cartridge ZB8610) 9 A_{rms} for 1 second (with fan cartridge ZB8610)	9 A_{rms} for 1 second	16 A_{rms} for 1 second (with fan cartridge ZB8610)
Rated power	170 W (without fan cartridge ZB8610) 276 W (with fan cartridge ZB8610)	276 W	428 W up to 55°C (with fan cartridge ZB8610) 490 W up to 45°C (with fan cartridge ZB8610)
Motor holding brake output voltage	24 V (+ 6 %, - 10 %)		
Max. motor holding brake output current	max. 0.5 A		
Load type	permanently excited synchronous motors, inductive (series AM81xx)		
PWM switching frequency	16 kHz		
Current controller frequency	double PWM switching frequency		
Velocity controller frequency	16 kHz		
Diagnostic LED	Status, warning, errors and limits		
Power loss	typ. 1.6 W		
Current consumption via E-bus	typ. 120 mA		
Current consumption from the 24 V	typ. 100 mA + holding brake		
Supports NoCoeStorage [▶ 32] function	Yes		
Reverse voltage protection	24 V power supply yes, through the body diode of the overvoltage protection device 50 V power supply yes, through the body diode of the overvoltage protection device		
Fuse protection (to be carried out by the user)	24 V power supply 10 A 50 V power supply 10 A		
Electrical isolation	500 V (E-bus/signal voltage)		
Possible EtherCAT cycle times	Multiple of 125 µs		
Configuration	no address setting required configuration via TwinCAT System Manager		

General data	EL7201-901x	EL7211-901x	EL7221-901x
Weight	approx. 60 g	approx. 95 g	approx. 95 g
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	approx. 27 mm x 100 mm x 70 mm (width aligned: 24 mm)	approx. 27 mm x 100 mm x 70 mm (width aligned: 24 mm)
Mounting [▶ 42]	on 35 mm mounting rail conforms to EN 60715		
Installation position	without fan cartridge ZB8610: standard installing position with fan cartridge ZB8610: standard installing position, other installing positions (example 1 and 2) see notice [▶ 45]!		

Environmental conditions	EL7201-901x	EL7211-901x	EL7221-901x
Permissible ambient temperature range during operation	0°C ... + 55°C		
Permissible ambient temperature range during storage	-25°C ... + 85°C		
Permissible relative humidity	95 %, no condensation		

Standards and approvals	EL7201-901x	EL7211-901x	EL7221-901x
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 49] for enhanced mechanical load capacity		
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4 according to IEC/EN 61800-3		
EMC category	Category C3 - standard Category C2, C1 - auxiliary filter required		
Protection class	IP20		
Approvals / markings*	CE, EAC, UKCA, cULus [▶ 56] TÜV-Süd [▶ 283]		

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

2.4 Technology

The very compact EL72x1-xxxx servomotor terminal integrates a complete servo drive for servomotors up to 276 W.

Servomotor

The servomotor is an electrical motor. Together with a servo amplifier the servomotor forms a servo drive. The servomotor is operated in a closed control loop with position, torque or speed control.

The servo terminal EL72x1-xxxx supports control of permanent magnet synchronous motors. These consist of three coils which are offset by 120° and a permanent magnet rotor.

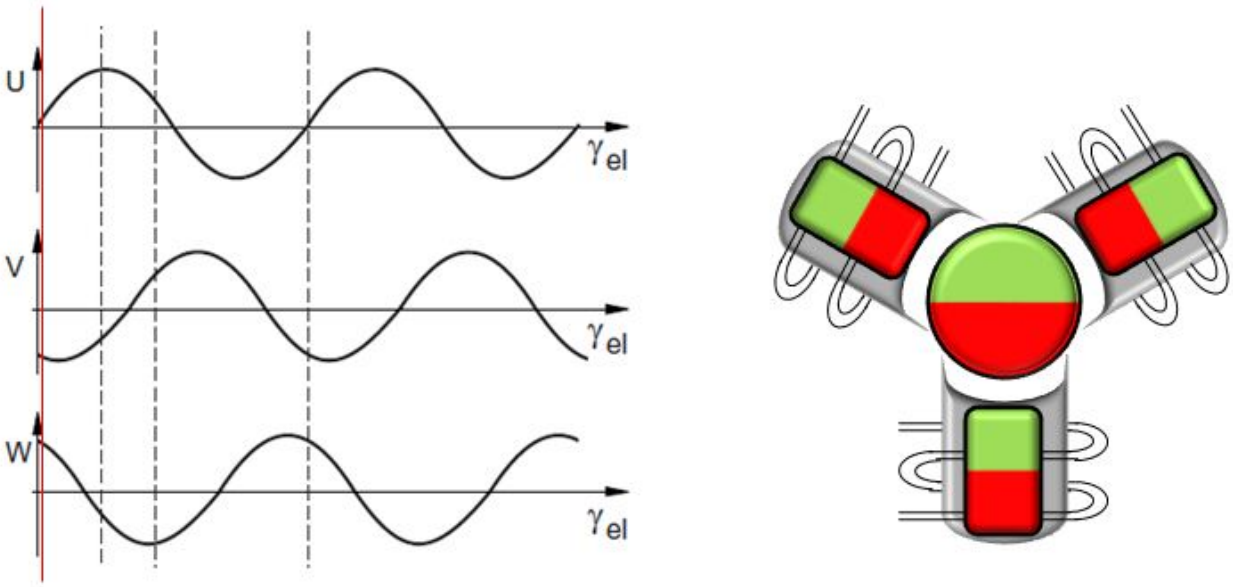


Fig. 6: Three synchronous motor coils, each offset by 120°

Servomotors particularly demonstrate their advantages in highly dynamic and precise positioning applications:

- very high positioning accuracy in applications where maximum precision is required through integrated position feedback
- high efficiency and high acceleration capacity
- servomotors are overload-proof and therefore have far greater dynamics than stepper motors, for example.
- load-independent high torque right up to the higher speed ranges
- maintenance requirements reduced to a minimum

The EtherCAT servomotor terminal offers users the option to configure compact and cost-effective systems without having to give up the benefits of a servomotor.

The Beckhoff servo terminal

The EL72x1-xxxx is a fully capable servo drive for direct connection to servomotors in the lower performance range. There is no need for further modules or cabling to make a connection to the control system. This results in a very compact control system solution. The E-Bus connection of the EL72x1-xxxx makes the full functionality of EtherCAT available to the user. This includes in particular the short cycle time, low jitter, simultaneity and easy diagnostics provided by EtherCAT. With this performance from EtherCAT the dynamics that a servomotor can achieve can be used optimally.

With a rated voltage up to 48 V_{DC} and a rated current of up to 4.5 A, this enables the user to operate a servomotor with a power of up to 276 W. Permanent magnet synchronous motors with a rated current of up to 4.5 A can be connected as loads. The monitoring of numerous parameters, such as overvoltage and

undervoltage, overcurrent, terminal temperature or motor load, offers maximum operational reliability. Modern power semiconductors guarantee minimum power loss and enable feedback into the DC link when braking.

With the integration of a complete servo drive into a standard EL7201 EtherCAT Terminal only 12 mm wide, Beckhoff is setting new standards in matters of size. This small manufactured size is possible thanks to the latest semiconductor technology and the resulting very high power factor. And yet, despite the small dimensions, nothing has to be sacrificed.

The integrated fast control technology, with a field-orientated current and PI speed control, supports highly dynamic positioning tasks. Apart from the direct connection of motor and resolver, the connection of a motor holding brake is also possible.

The EL72x1-xx1x EtherCAT terminal has two digital inputs that can be used for the “Touch Probe” function. The status of the inputs can be read by “Select Info Data” (MDP742 profile and DS402 profile).

Connection to the control system

A further big advantage of the EL72x1-xxxx is the easy incorporation into the control solution. The complete integration into the control system simplifies commissioning and parameterization. As with all the other Beckhoff terminals, the EL72x1-xxxx is simply inserted into the terminal network. Then the full terminal network can be scanned by the TwinCAT System Manager or manually added by the application engineer. In the System Manager the EL72x1-xxxx can be linked with the TwinCAT NC and parameterized.

Scalable motion solution

The servo terminal complements the product range of compact drive technology for Beckhoff I/O systems that are available for stepper motors, AC and DC motors. With the EL72x1-xxxx, the range of servo drives becomes even more finely scalable: from the miniature servo drive up to 170 W in the EtherCAT Terminal through to the AX5000 servo drive with 118 KW, Beckhoff offers a wide range including the servomotors. The AM81xx series was specially developed for the servomotor terminal EL72x1-xxxx.

One Cable Technology (OCT)

In the servomotors from the AM8100-xF**2** x series the feedback signals are transmitted directly via the power supply cable, so that power and feedback system are combined in a single motor connection cable. With the use of the One Cable technology, the information is sent reliably and without interference through a digital interface. Since a cable and plug are omitted at both the motor and controller end, the component and commissioning costs are reduced.

Thermal I²T motor model

The thermal I²T motor model represents the thermal behavior of the motor winding taking into account the absolute thermal resistance R_{th} and the thermal capacity C_{th} of motor and the stator winding.

The model assumes that the motor reaches its maximum continuous operating temperature T_{nom} during continuous operation with rated current I_{nom}. This temperature corresponds to 100% motor load. During operation at rated current the motor model reaches a load of 63% after a time of $\tau_{th}=R_{th}\cdot C_{th}$ and slowly reaches its continuous operating temperature.

If the motor is operated with a current that is greater than the rated current, the model reaches 100% load more quickly.

If the load of the I²T model exceeds 100%, the requested set current is limited to the rated current, in order to protect the motor winding thermally. The load reduces to a maximum of 100%. If the current falls below the rated current, the load falls below 100% and the set current limitation is cancelled.

For a motor that has been cooled to ambient temperature, the time for reaching 100% load with a set current that exceeds the rated current can be estimated with $\tau_{th}\cdot I_{nom}^2/I_{actual}^2$.

The actual load must be known for exact calculation of the time when the 100% load threshold is exceeded.

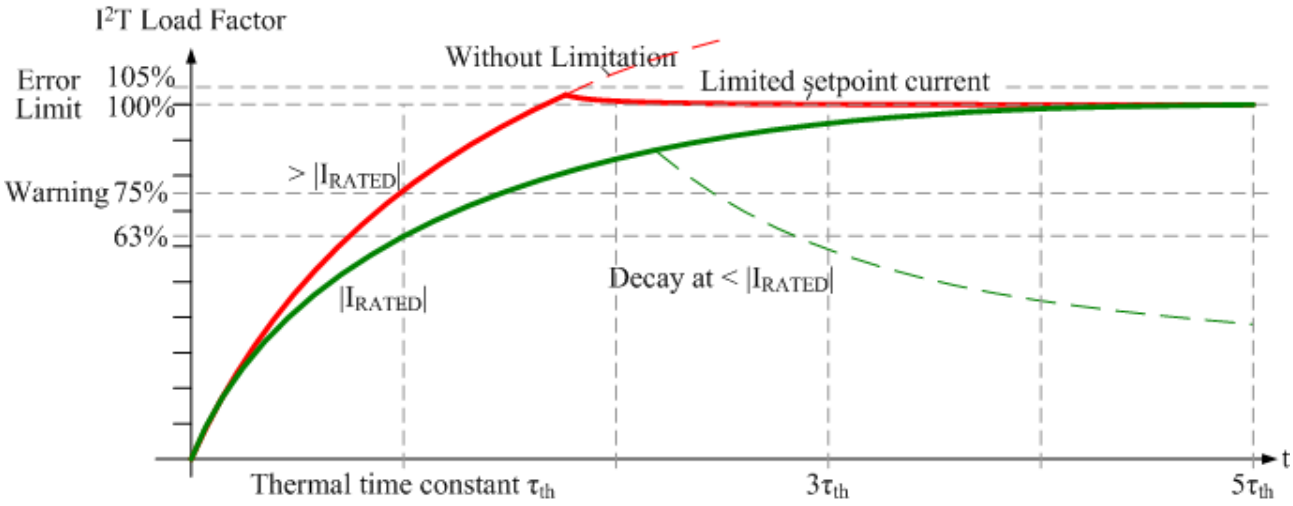


Fig. 7: Limitation to the rated motor current

2.5 Start-up

For commissioning:

- mount the EL72x1-901x as described in the chapter Installation [▶ 39].
- configure the EL72x1-901x in TwinCAT as described in the chapter Commissioning [▶ 63].

3 Basics communication

3.1 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics.

3.2 EtherCAT cabling – wire-bound

The cable length between two EtherCAT devices must not exceed 100 m. This results from the FastEthernet technology, which, above all for reasons of signal attenuation over the length of the cable, allows a maximum link length of 5 + 90 + 5 m if cables with appropriate properties are used. See also the Design recommendations for the infrastructure for EtherCAT/Ethernet.

Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAt5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

EtherCAT uses RJ45 plug connectors, for example. The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Pin	Color of conductor	Signal	Description
1	yellow	TD +	Transmission Data +
2	orange	TD -	Transmission Data -
3	white	RD +	Receiver Data +
6	blue	RD -	Receiver Data -

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

Recommended cables

It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff website!

E-Bus supply

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 (see details in respective device documentation).
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. 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.

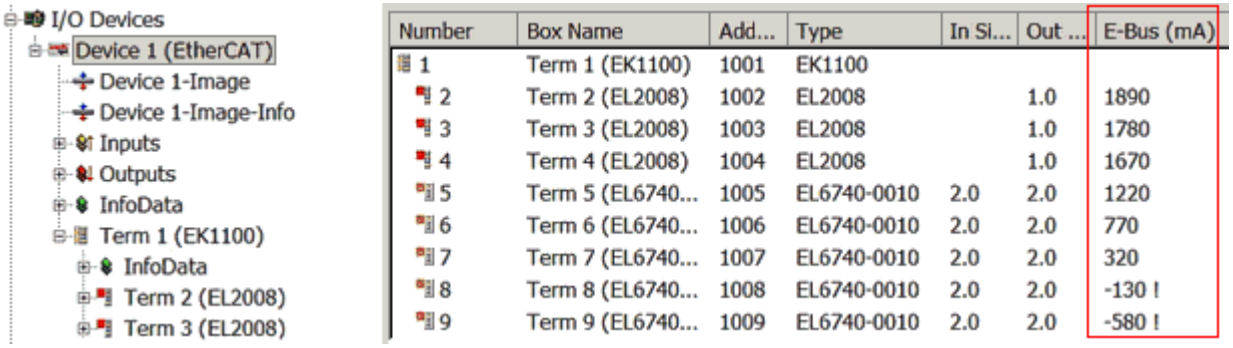


Fig. 8: System manager current calculation

NOTICE

Malfunction possible!

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

3.3 General notes for setting the watchdog

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

The EtherCAT slave controller features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

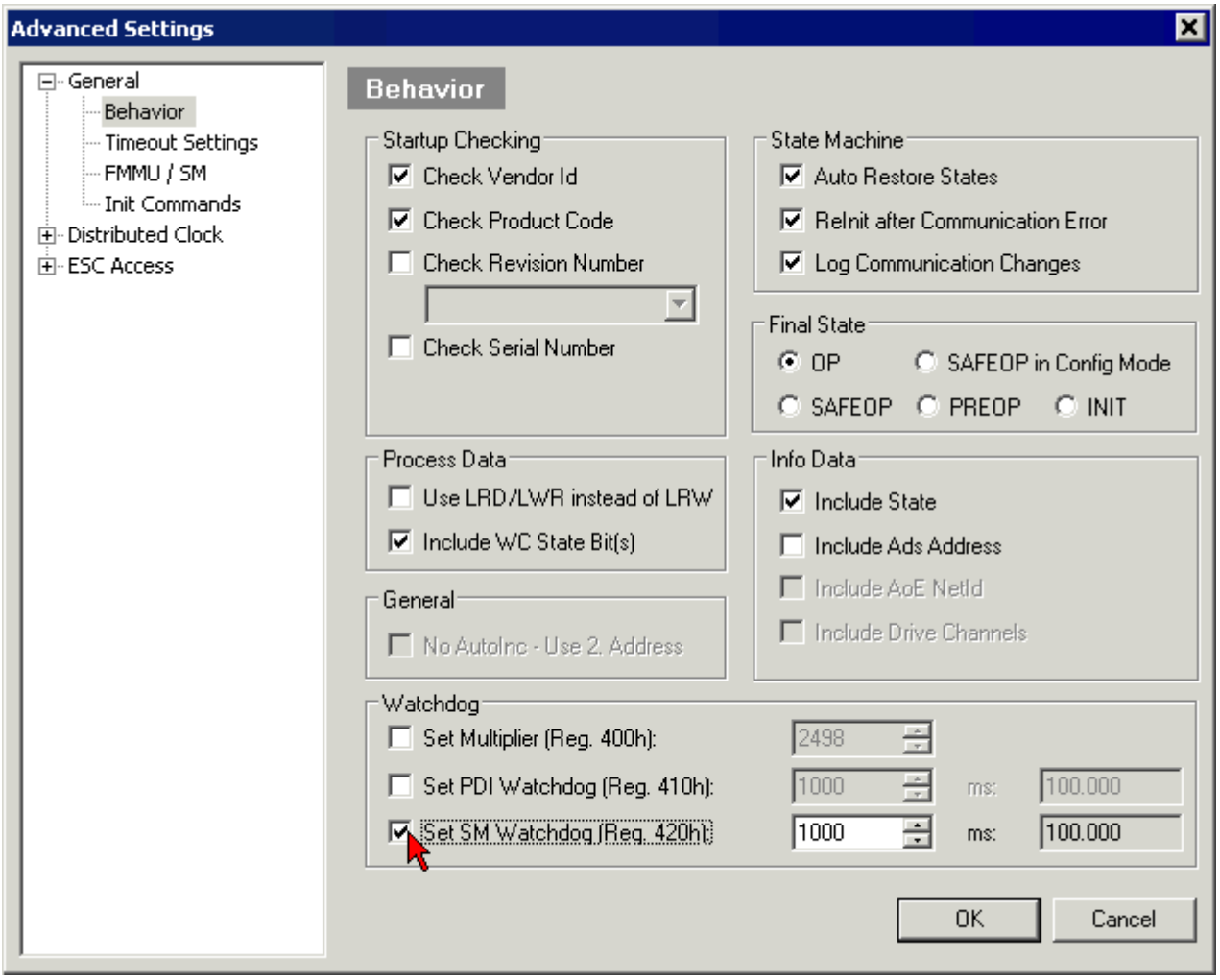


Fig. 9: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers 400h, 410h and 420h: ESC Access -> Memory

SM watchdog (Sync Manager Watchdog)

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

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

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

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

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

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

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * SM/PDI watchdog

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

The value in “Watchdog multiplier + 2” in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 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
- Operational
- Bootstrap

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

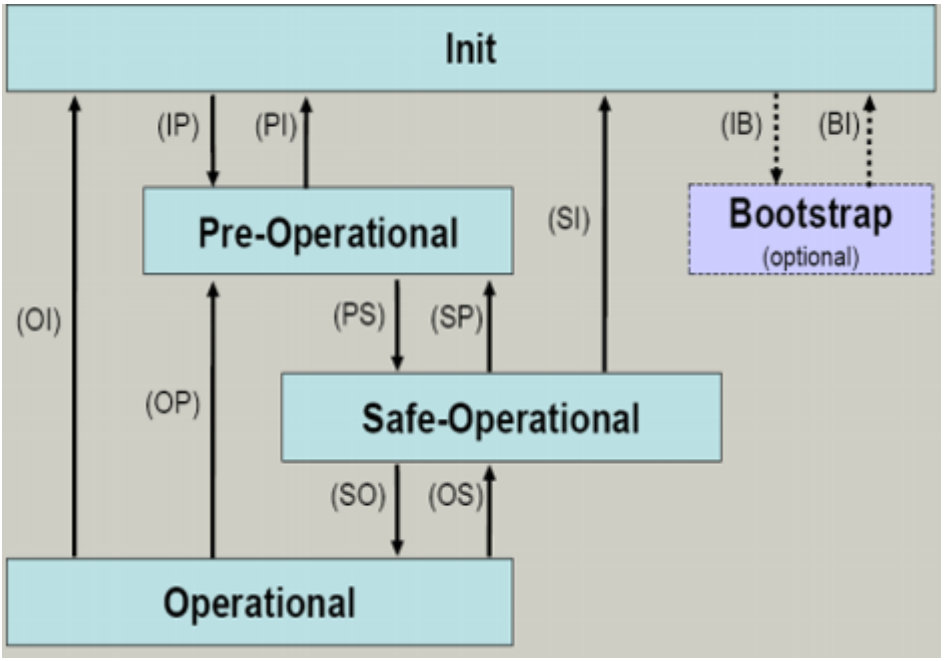


Fig. 10: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

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 ESC 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 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.

3.5 CoE Interface

General description

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

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

The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

Availability

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

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

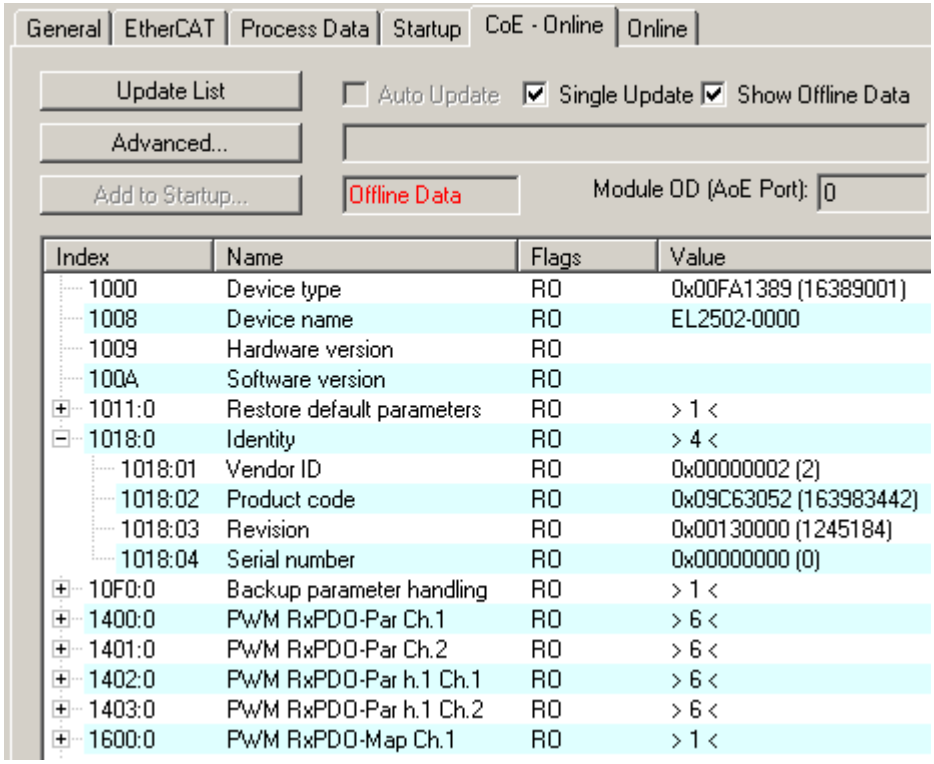


Fig. 11: “CoE Online” tab

The figure “’CoE Online’ tab” shows the CoE objects available in device “EL2502”, ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

NOTICE

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

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT 3 | PLC Library: "Tc2_EtherCAT" and Example program R/W CoE)

Data management and function “NoCoeStorage”

Some parameters, particularly the setting parameters of the slave, are configurable and writeable,

- via the System Manager (Fig. “CoE Online” tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.
- from the control system or 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.

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 index 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.
- If the function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

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.

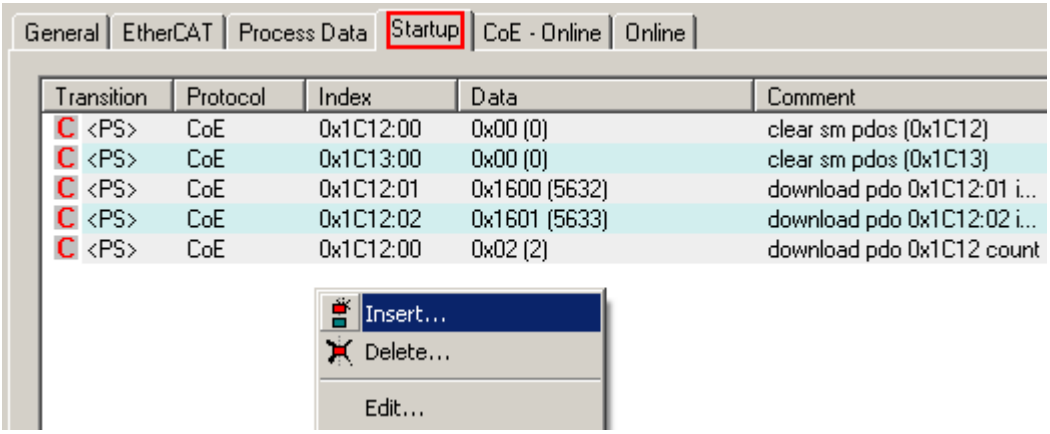


Fig. 12: Startup list in the TwinCAT System Manager

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

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

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 Data** is shown in red.

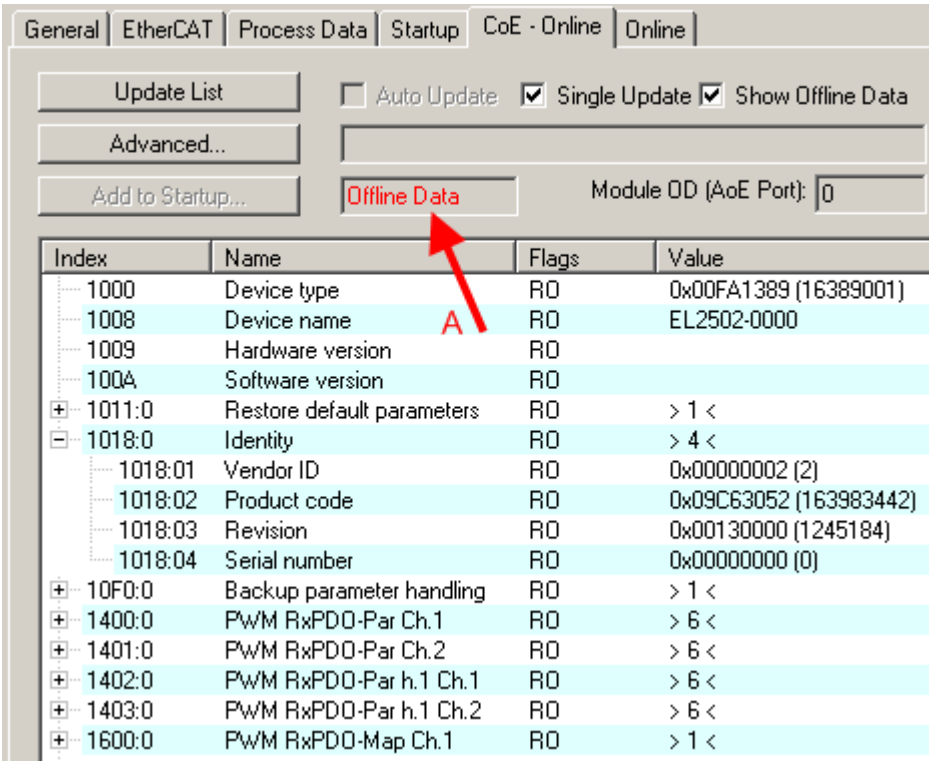


Fig. 13: Offline list

- 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 status of the device is displayed in the CoE.
  - **Online Data** is shown in green.

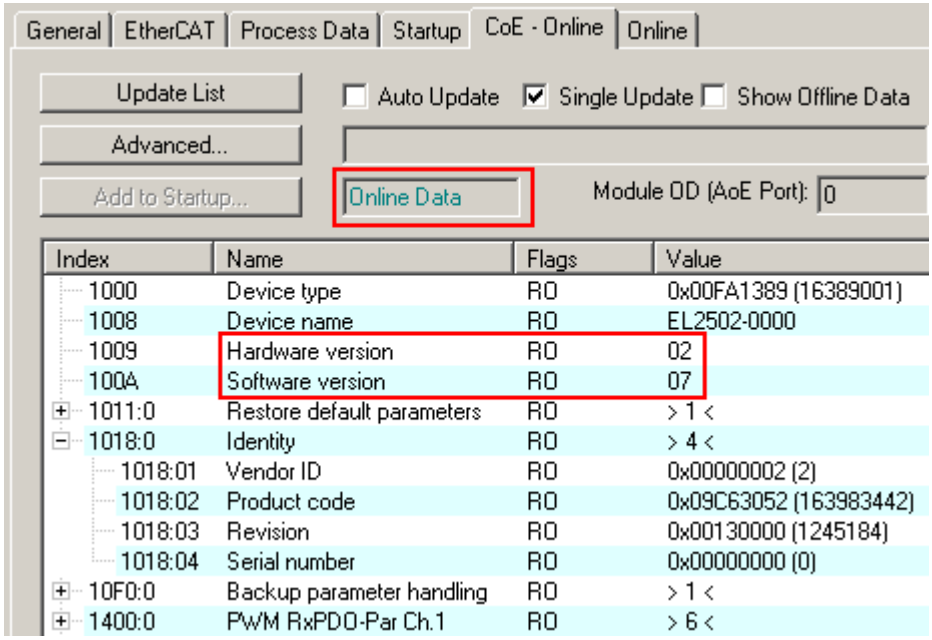


Fig. 14: Online list

Channel-based order

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

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in 16_{dec} or 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.

BackUp objects and checksum 0x10F0:01

The following object types, among others, are defined for the CoE parameter/object directory with regard to information retention, but they do not all have to appear in every device at the same time:

- Vendor objects
  - Are stored persistently (fail-safe) in the device.
  - Technically with the property ReadWrite (RW).
  - Can only be modified/deleted if the corresponding vendor password is known.
  - Used for vendor-specific adjustment or identity data.
- BackUp objects
  - These are objects that are stored persistently in the device, even after changes.
  - Technically with the property ReadWrite (RW).
  - They can be modified/deleted at any time via CoE access from the EtherCAT master

    - In particular, they are reset to their initial state, which is stored permanently in the firmware, by selecting "Restore Default Parameters" (see chapter "Restoring the delivery state"). Since this reset to a previous value resembles the restoration of a backup, these are referred to as “BackUp objects”.
    - They are used for regular function parameters of the device that determine the behavior.
- BackUp objects with write protection option.
    - The same applies as for the BackUp objects.
    - In addition, the user can activate write protection for these objects using a code word in xF009, thereby preventing accidental changes. Details can be found in the device documentation of the devices that contain these objects.
- Volatile objects
    - These are objects that are not stored persistently in the device.
    - They are used to display internal information (process data, states, temperatures ...) and are available as ReadOnly (RO) or as function parameters (ReadWrite). However, the latter must be written by the EtherCAT master for each PowerOn if they are to have a value other than the default value.

The device indicates a 16-bit CRC in the 32-bit object 0x10F0:01 CheckSum, subindex 01 of Backup Parameter Handling by means of the current state of the so-called BackUp objects:

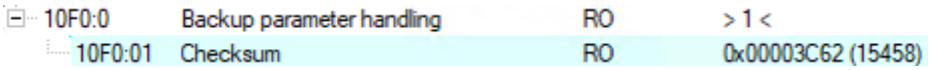


Fig. 15: CoE index 10F0

If a BackUp object is changed, the firmware calculates a new checksum accordingly. This can be used to detect changes to the BackUp objects.

Note: The initial value of the checksum may vary depending on the firmware version, as function extensions can add additional objects to the CRC’s detection range.

3.6 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 Installation

4.1 Example configuration for temperature measurement

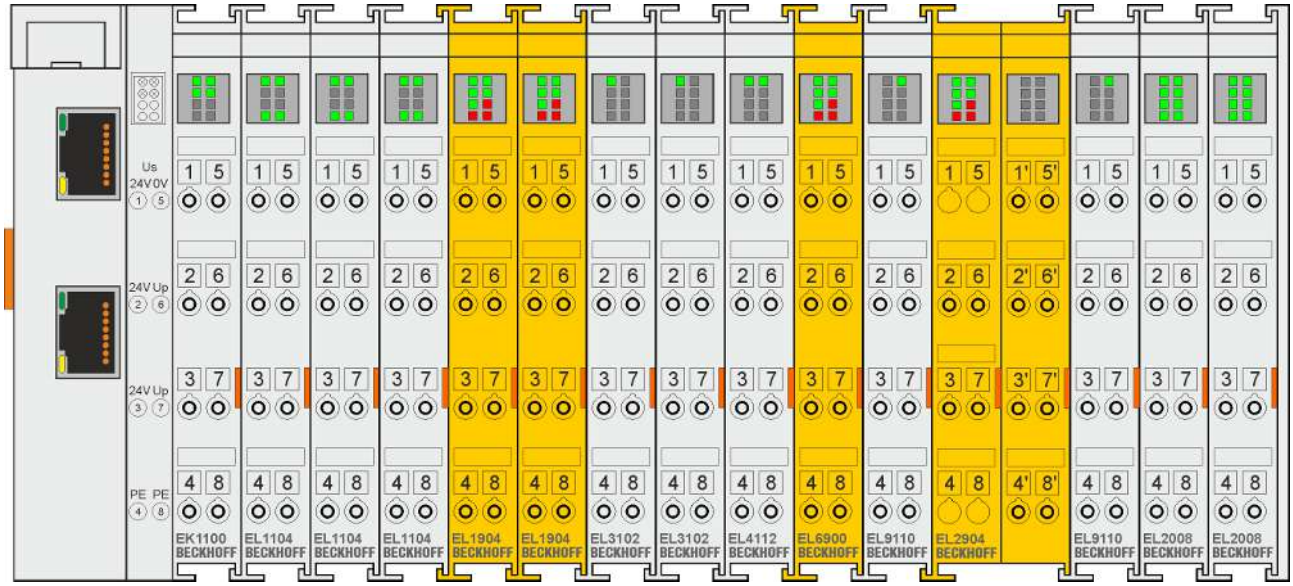


Fig. 16: Example configuration for temperature measurement

The example configuration for the temperature measurement consists of an EK1100 EtherCAT coupler with connected terminals that match the typical distribution of digital and analog signal types at a machine. On the EL6900 a safety project is active, which reads safe inputs and enables all 4 safe outputs during the measurement.

External heat sources / radiant heat / impaired convection

The maximum permissible ambient temperature of 55°C was checked with the above example configuration. Impaired convection, an unfavorable location near heat sources or an unfavorable configuration of the EtherCAT Terminals may result in overheating of the terminals.
The key parameter is always the maximum permitted internally measured temperature of 95°C, above which the TwinSAFE terminals switch to safe state and report an error. The internal temperature can be read from the TwinSAFE components via CoE (see chapter Diagnose).

4.2 Safety instructions

Before installing and commissioning the TwinSAFE components please read the safety instructions in the foreword of this documentation.

4.3 Environmental conditions

Please ensure that the TwinSAFE components are only transported, stored and operated under the specified conditions (see technical data)!

⚠ WARNING

Risk of injury!

The TwinSAFE components must not be used under the following operating conditions.

- under the influence of ionizing radiation (that exceeds the level of the natural environmental radiation)
- in corrosive environments[1]
- in an environment that leads to unacceptable soiling of the TwinSAFE component

[1] A corrosive environment exists when corrosion damage becomes apparent.

NOTICE

Electromagnetic compatibility

The TwinSAFE components comply with the current standards on electromagnetic compatibility with regard to spurious radiation and immunity to interference in particular.
However, in cases where devices such as mobile phones, radio equipment, transmitters or high-frequency systems that exceed the interference emissions limits specified in the standards are operated near TwinSAFE components, the function of the TwinSAFE components may be impaired.

4.4 Transport / storage

Use the original packaging in which the components were delivered for transporting and storing the TwinSAFE components.

⚠ CAUTION

Note the specified environmental conditions

Please ensure that the digital TwinSAFE components are only transported and stored under the specified environmental conditions (see technical data).

4.5 Control cabinet / terminal box

The TwinSAFE terminals must be installed in a control cabinet or terminal box with IP54 protection class according to IEC 60529 as a minimum.

4.6 Notes on ESD protection

NOTICE

Devices can be destroyed by electrostatic charging!

The devices contain electrostatically sensitive components which can be damaged by improper handling.

- When handling the components, take care to avoid electrostatic discharge; also, prevent direct contact with the spring-loaded contacts (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when simultaneously handling components.
- When handling the components, ensure proper earthing of the environment (workplace, packaging, and persons).
- Each bus station must be terminated with the EL9011 or EL9012 end cap on its right side to guarantee the protection rating and ESD protection.

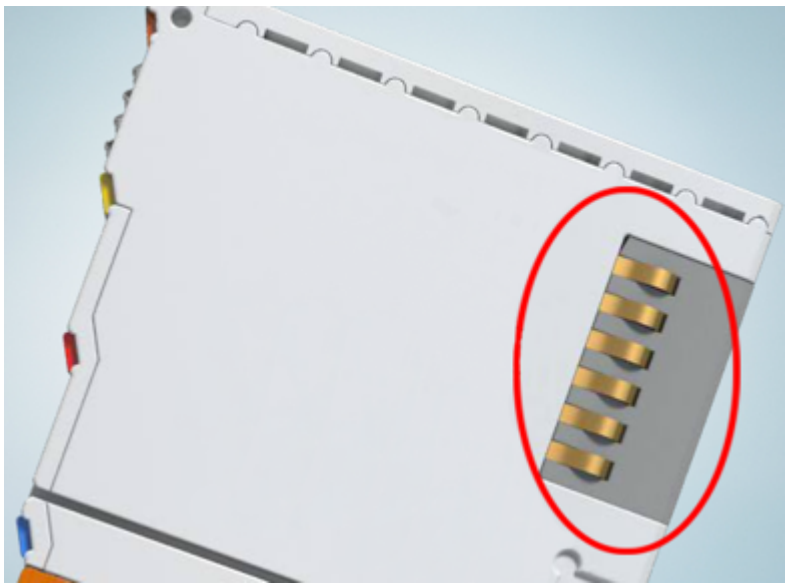


Fig. 17: Spring contacts of Beckhoff I/O components

4.7 Mounting rail installation

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

The bus terminal system is designed for installation in a control cabinet or junction box.

Installation

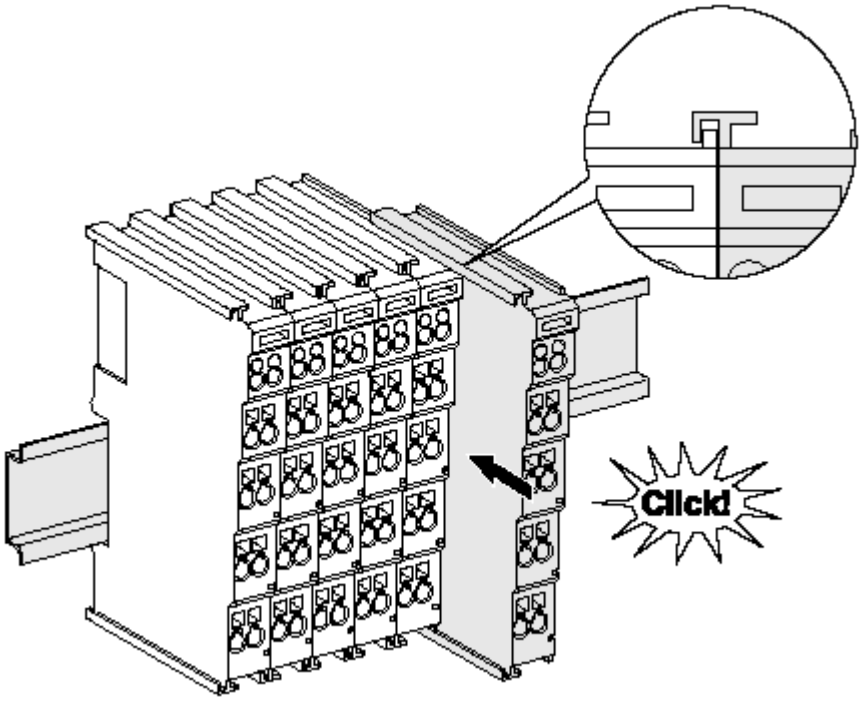


Fig. 18: Installation on the mounting rail

The bus couplers and bus terminals are attached to commercially available 35 mm mounting rails (DIN rail according to EN 60715) by applying slight pressure:

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. To do this, connect the components with the slot and key and push the terminals against the mounting rail until the latch audibly engages on the mounting rail.
   If you snap the terminals onto the mounting rail first and then push them next to each other without the slot and key engaging, no functional connection will be established! When correctly assembled, no significant gap should be visible between the housing units.

Fastening the mounting rails

The locking mechanism of the terminals and couplers protrudes into the profile of the mounting rail. When installing the components, make sure that the locking mechanism doesn't come into conflict with the fixing bolts of the mounting rail. When fastening mounting rails with a height of 7.5 mm under the terminals and couplers, use flat fastening components such as countersunk head screws or blind rivets.

NOTICE

Eath the mounting rail!

Make sure the mounting rail is properly earthed.

Connections within a bus terminal block

The electric connections between the bus coupler and the bus terminals are made automatically by joining the components:

- The six spring-loaded contacts of the E-bus/K-bus deal with the data transfer and the supply of the bus terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminal points on the bus coupler (up to 24 V) or via power supply terminals for higher voltages.

Power contacts

When configuring a bus terminal block, note the contact assignments of the individual bus terminals, as some types

- do not loop through the power contacts or do not loop them through completely (e. g. analog bus terminals or digital 4-channel bus terminals),
- the power contacts are disconnected, thus marking the start of a new supply rail (power supply terminals).

Power contact ⏚

The power contact marked ⏚ (ground connection in accordance with IEC 60417-5017) can be used as a ground. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

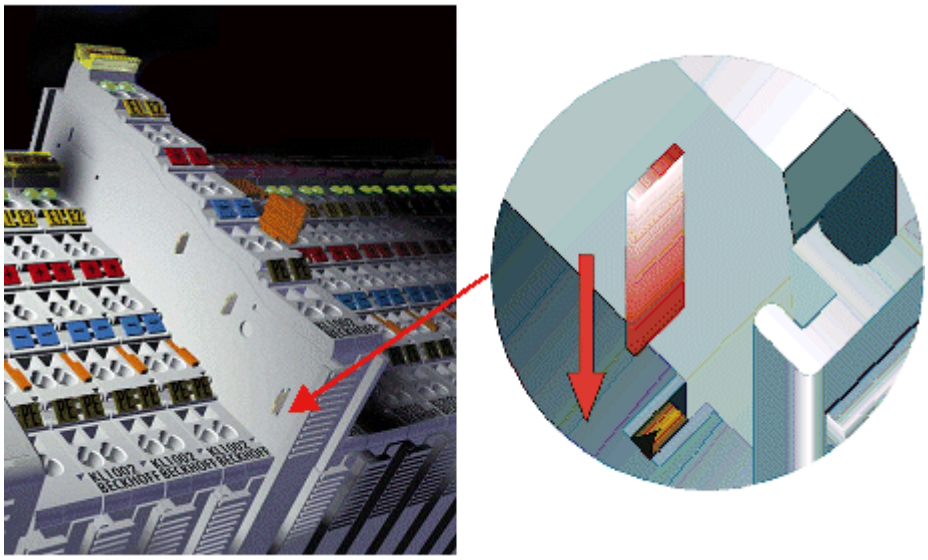


Fig. 19: Power contact on the left

⚠ WARNING

Risk of injury from electric shock!

The power contact marked ⏚ may not be used for other potentials!

NOTICE

Risk of damage to the device

Note that, for electromagnetic compatibility, the earth contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage to the terminal (e.g. disruptive discharge to the earth line during insulation testing of a consumer with a rated voltage of 230 V). For insulation testing, disconnect the earth supply line at the bus coupler or the power supply terminal! In order to decouple further supply points for testing, these power supply terminals can be released and pulled at least 10 mm from the group of terminals.

Disassembly

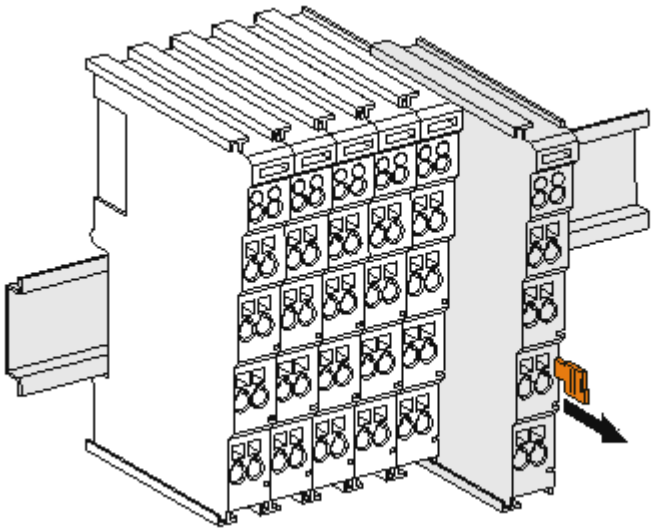


Fig. 20: Removal from mounting rail

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Pull down the terminal at its orange-colored straps from the mounting rail by approx. 1 cm. The rail locking of this terminal is automatically released, and you can now pull the terminal out of the bus terminal block with little effort.
2. To do this, grasp the unlocked terminal simultaneously at the top and bottom of the housing surfaces with your thumb and index finger and pull it out of the bus terminal block.

4.8 Installation position for operation with or without fan

NOTICE

Constraints regarding installation position and operating temperature range

When installing the terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Prescribed installation position for operation without fan

The prescribed installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. "*Recommended distances of installation position for operating without fan*").

The terminals are ventilated from below, which enables optimum cooling of the electronics through convection.

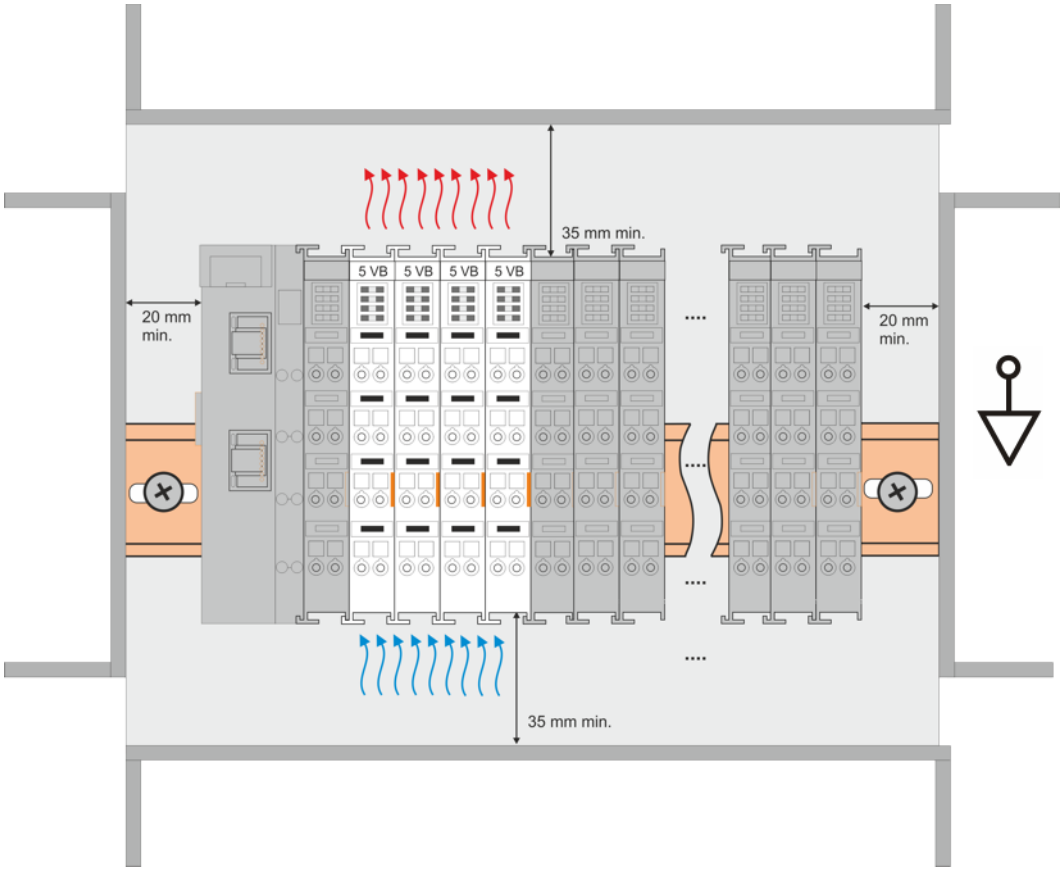


Fig. 21: Recommended distances of installation position for operating without fan

Compliance with the distances shown in Fig. "*Recommended distances of installation position for operating without fan*" is recommended.

For further information regarding the operation without fan refer to the Technical Data of the terminal.

Standard installation position for operation with fan

The standard installation position for operation with fan requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *Recommended distances for installation position for operation with fan*).
The terminals are ventilated fan supported (e.g. with ZB8610 fan cartridge) from below.

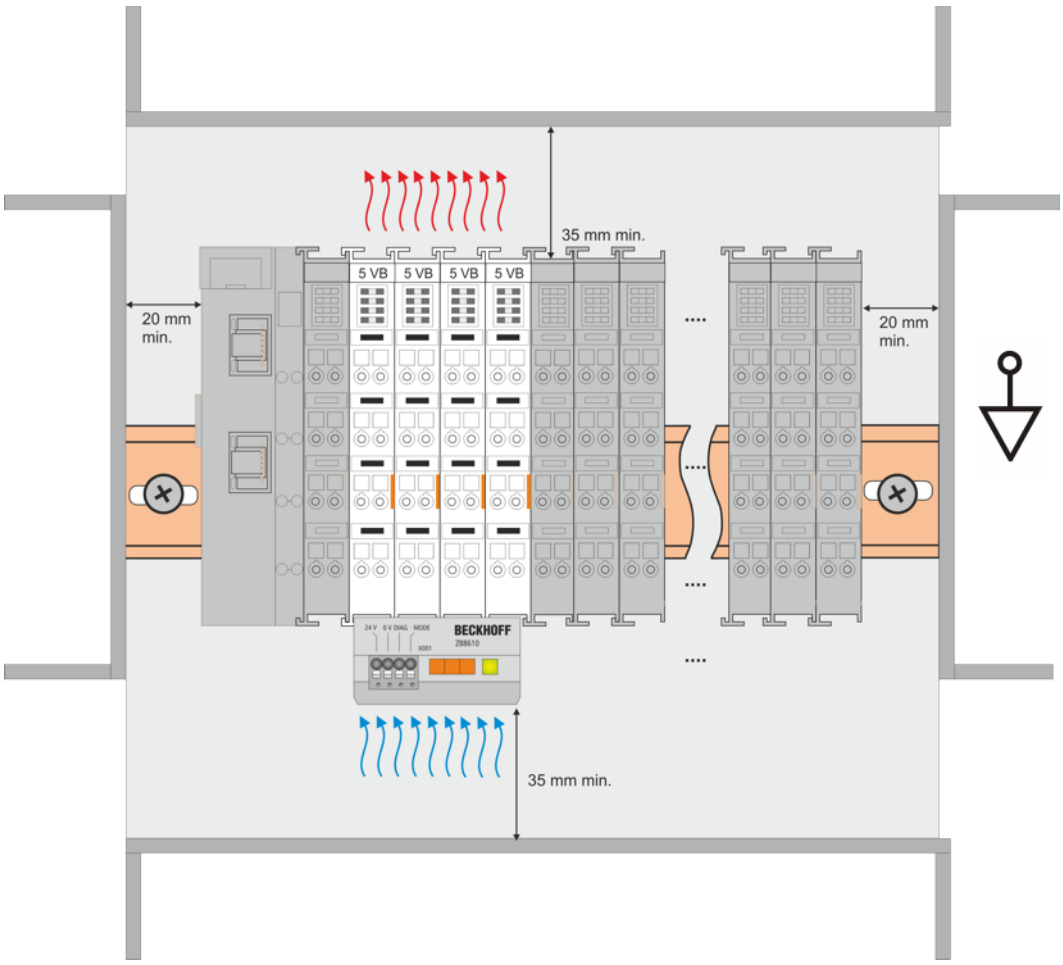


Fig. 22: Recommended distances for installation position for operation with fan

Other installation positions

Due to the enforced effect of the fan on the ventilation of the terminals, other installation positions (see Fig. "*Other installation positions, example 1 + 2*") may be permitted where appropriate.

See corresponding notes in the Technical Data of the terminal.

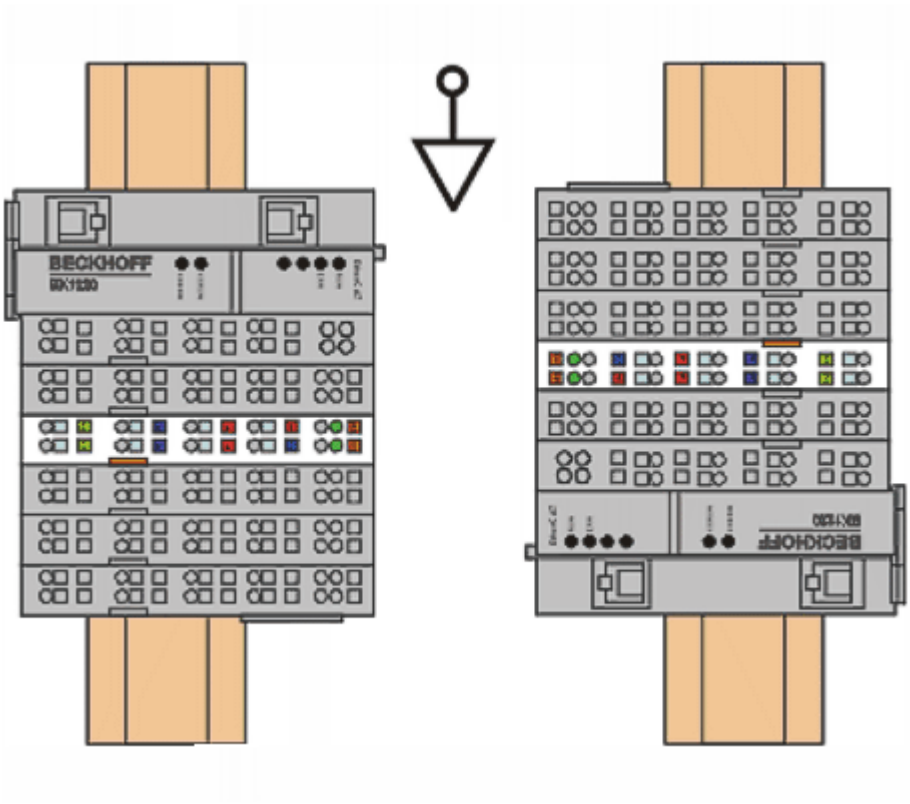


Fig. 23: Other installation positions, example 1

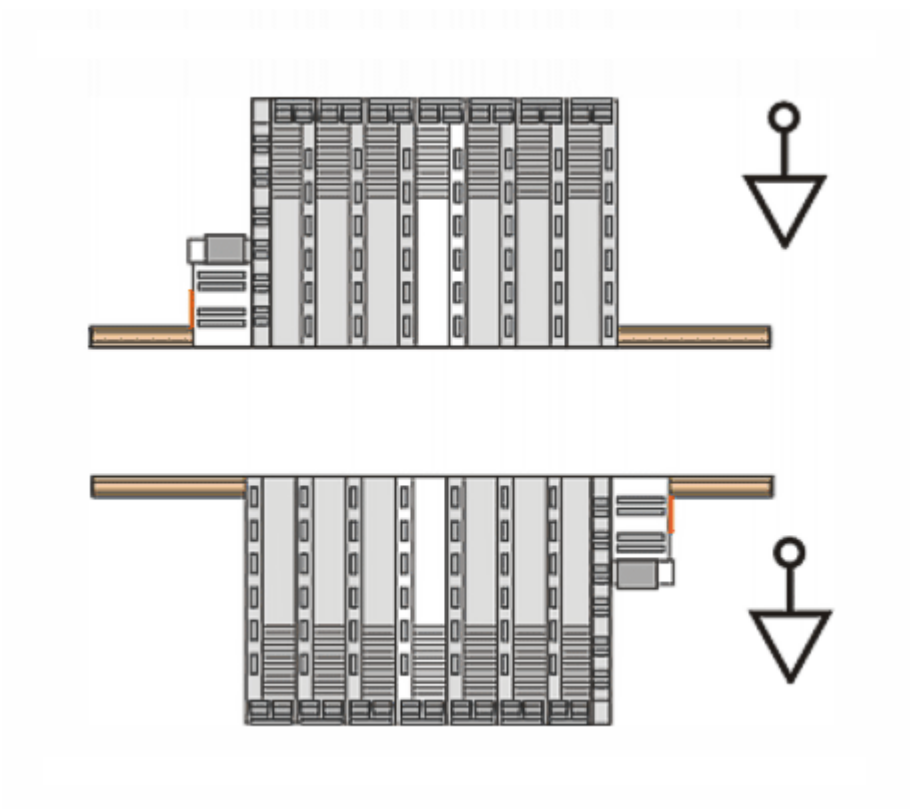


Fig. 24: Other installation positions, example 2

4.9 Positioning of passive terminals

Note on the placement of passive terminals in the bus terminal block

EtherCAT Terminals that do not take part in active data exchange within the bus terminal block are referred to as passive terminals. These terminals can be recognized because they do not consume power from the E-bus. To ensure optimal data exchange, no more than two passive terminals may be connected directly to each other!

Examples of the placement of passive terminals (highlighted in light color)

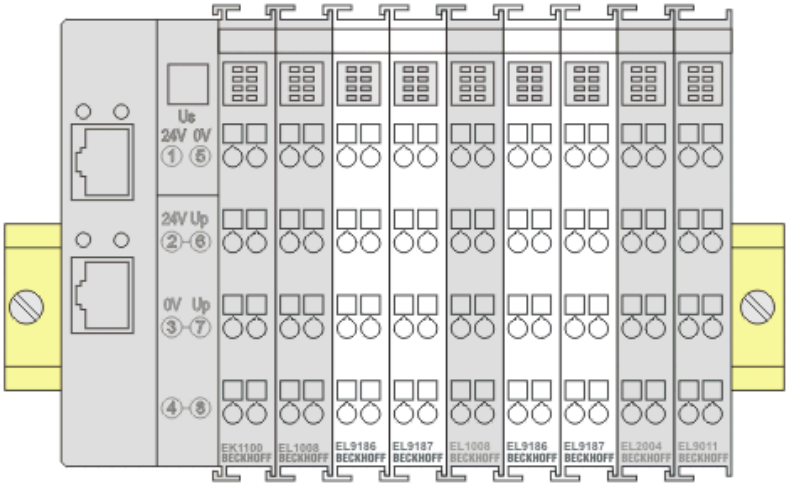


Fig. 25: Proper positioning

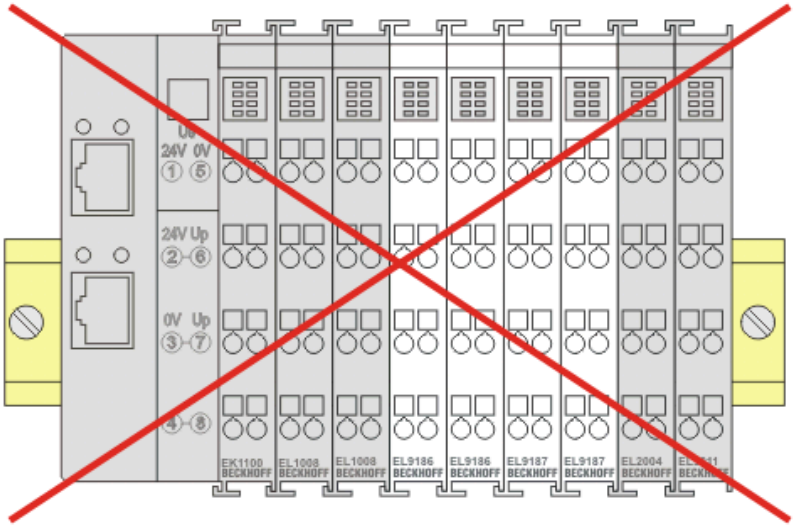


Fig. 26: Incorrect positioning

4.10 Installation instructions for enhanced mechanical load capacity

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 *g*, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 *g*, 6 ms

Additional installation instructions and notes

For terminals with enhanced mechanical load capacity, the following additional installation instructions and notes apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions.
- Use a mounting rail according to EN 60715 TH35-15.
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp.
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting width) or 32 terminals (24 mm mounting width)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail.
- The mounting points of the mounting rail must be set at 5 cm intervals.
- Use countersunk head screws to fasten the mounting rail.
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

4.11 Connection

4.11.1 Connection system

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

Overview

The bus terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of ELxxxx and KLxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of ESxxxx and KSxxxx series feature a pluggable connection level and enable steady wiring while replacing.
- The High Density Terminals (HD Terminals) include electronics and connection level in a single enclosure and have advanced packaging density.

Standard wiring (ELxxxx / KLxxxx)

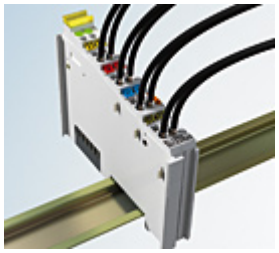


Fig. 27: Standard wiring

The terminals of the ELxxxx and KLxxxx series integrate screwless spring-cage technology for quick and easy wiring.

Pluggable wiring (ESxxxx / KSxxxx)

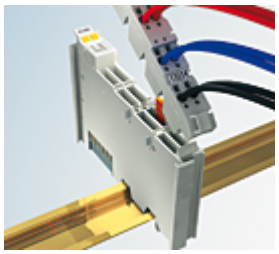


Fig. 28: Pluggable wiring

The terminals of ESxxxx and KSxxxx series feature a pluggable connection level.
The assembly and wiring procedure is the same as for the ELxxxx and KLxxxx series.
The pluggable connection level enables the complete wiring to be removed as a plug connector from the top of the housing for servicing.
The lower section can be removed from the terminal block by pulling the unlocking tab.
Insert the new component and plug in the connector with the wiring. This reduces the installation time and eliminates the risk of wires being mixed up.

The familiar dimensions of the terminal only had to be changed slightly. The new connector adds about 3 mm. The maximum height of the terminal remains unchanged.

A tab for strain relief of the cable simplifies assembly in many applications and prevents tangling of individual connection wires when the connector is removed.

Conductor cross sections between 0.08 mm^2 and 2.5 mm^2 can continue to be used with the proven spring force technology.

The overview and nomenclature of the product names for ESxxxx and KSxxxx series has been retained as known from ELxxxx and KLxxxx series.

High Density Terminals (HD Terminals)



Fig. 29: High Density Terminals

The terminals from these series with 16/32 terminal points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm bus terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Wiring HD Terminals

The High Density Terminals of the ELx8xx and KLx8xx series doesn't support pluggable wiring.

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands can also be connected to the standard and high-density terminals. In this case, please note the tables concerning the wire-size width [▶ 53]!

4.11.2 Wiring

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

Terminals for standard wiring ELxxxx/KLxxxx and for pluggable wiring ESxxxx/KSxxxx

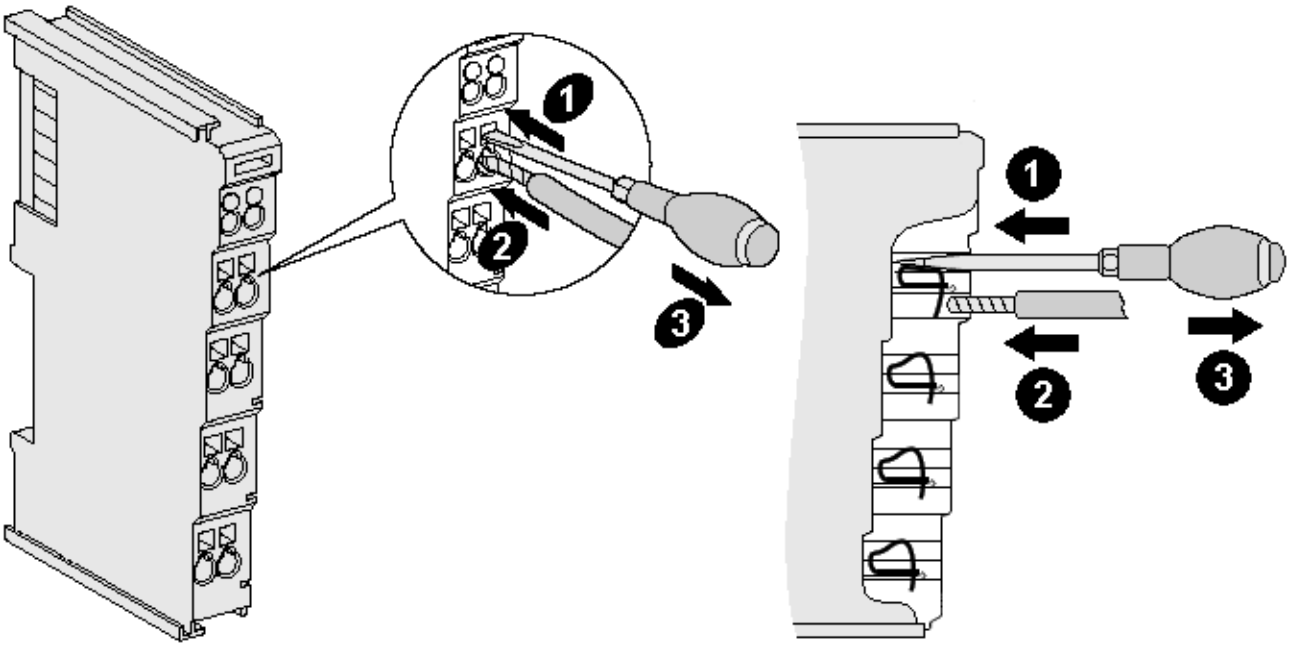


Fig. 30: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the bus terminal. The terminal points are implemented in spring force technology. Connect the cables as follows (see fig. “Connecting a cable on a terminal point”:

1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
2. The wire can now be inserted into the round terminal opening without any force.
3. When the screwdriver is removed, the terminal point closes automatically and holds the wire securely and permanently in place

See the following table for the suitable wire size width:

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm²	0.14 ... 1.5 mm²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▶ 51]) with 16/32 terminal points

The conductors of the HD Terminals are connected without tools for single-wire conductors using the direct plug-in technique, i.e. after stripping the wire is simply plugged into the terminal point. The cables are released, as usual, using the contact release with the aid of a screwdriver. See the following table for the suitable wire size width.

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm²
Wire size width (ultrasonically compacted [ultrasonically welded] strands)	only 1.5 mm² (see notice [▶ 51])
Wire stripping length	8 ... 9 mm

4.11.3 Shielding

Shielding

Feedback signal, sensors and actuators should always be connected with shielded, twisted paired wires.

4.12 Note on power supply

⚠ WARNING

Power supply from SELV / PELV power supply unit!

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

Notes:

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

4.13 Note on load voltage supply

⚠ WARNING

Load voltage supply

Some devices permit an additional load voltage, e.g. 48 V DC, to be connected for the operation of a motor. In order to avoid stray currents on the protective conductor during operation, EN 60204-1:2018 provides for the possibility that the negative pole of the load voltage does not necessarily have to be connected to the protective conductor system (SELV).
Therefore, the load voltage supply should be designed as an SELV supply.

4.14 Shielding, earthing

Please refer to the further information in the I/O analog manual, chapter “Analog information – shield and earth”, see chapter Guide through documentation [▶ 9].

4.15 UL notice - Compact Motion

⚠ CAUTION

Application

Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

⚠ CAUTION

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).

⚠ CAUTION

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

⚠ CAUTION

Notes on motion devices

- *Motor overtemperature*
  Motor overtemperature sensing is not provided by the drive.
- *Application for compact motion devices*
  The modules are intended for use only within Beckhoff's Programmable Controller system Listed in File E172151.
- *Galvanic isolation from the supply*
  The modules are intended for operation within circuits not connected directly to the supply mains (galvanically isolated from the supply, i.e. on transformer secondary).
- *Requirement for environmental conditions*
  For use in Pollution Degree 2 Environment only.

Basic principles

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

4.16 Notes on current measurements using Hall sensors

The device described in this documentation features one or several integrated Hall sensor for the purpose of current measurements.

During this process, the Hall sensor monitors the magnetic field generated by a current flowing through a conductor.

In order to prevent compromising the measurement we recommend screening exterior magnetic fields from the device, or to keep such fields at an adequate distance.



Fig. 31: Note

Background

A current-carrying conductor generates a magnetic field around it according to

$B = \mu_0 * I / (2\pi * d)$

with

B [Tesla] magnetic field

$\mu 0 = 4*\pi*10^{-7}$ [H/m] (assumption: no magnetic shielding)

I [A] current

d [m] distance to conductor

Interference from external magnetic fields

The magnetic field strength should not exceed a permitted level all around the device.
In practice this equates to a recommended minimum distance between a conductor and the device surface as follows:

- Current 10 A: 12 mm
- Current 20 A: 25 mm
- Current 40 A: 50 mm

Unless specified otherwise in the device documentation, stringing together modules (e.g. terminal blocks based on a 12 mm grid) of same type (e.g. EL2212-0000) is permitted.

4.17 EL72x1-9014 - LEDs and connection

4.17.1 EL7201-901x - LEDs and connection

EL7201-901x - LEDs

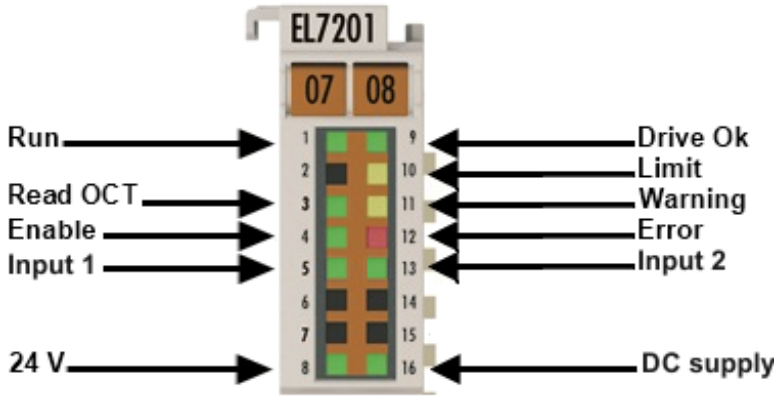


Fig. 32: EL7201-901x - LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates [▶ 268]
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
Drive OK	green	on	Driver stage ready for operation
Limit	orange	on	The LED is linked with bit 11 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (internal limit active) Limit reached (e.g. torque or speed limit)
Read OCT	green	flashing	The electronic type plate is being read
		off	The reading of the electronic type plate has been completed
Warning	orange	flashing	Error while reading the type plate
		on	The LED is linked with bit 7 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (warning) The “Warning” threshold value is exceeded. I²T model Voltage missing at STO input Temperature (80°C) exceeded Voltage
Enable	green	on	The LED is linked with the bits 1 and 2 of status word (MDP742 [▶ 208] / DS402 [▶ 240]) (if "Switched on" or "Operation enabled") Driver stage enabled
Input 1 Input 2	green	on	Signal is present at digital inputs (e.g. end position, reference or latch sensors/switches)
Error	red	on	The LED is linked with bit 3 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (fault) The “Error” threshold value is exceeded. Overcurrent STO triggered with active axis Voltage not available Resolver not connected Max. temperature (100°C) exceeded
+24 V via power contacts	green	on	24 V voltage supply for the terminal is present.
DC link supply	green	on	Voltage for the DC link supply is present.

NOTICE

Fuse protection of the supply voltage

The electrical protection of the load voltage must be selected in such a way that the maximum flowing current is limited to 3 times the rated current (max. 1 second)!

Connection

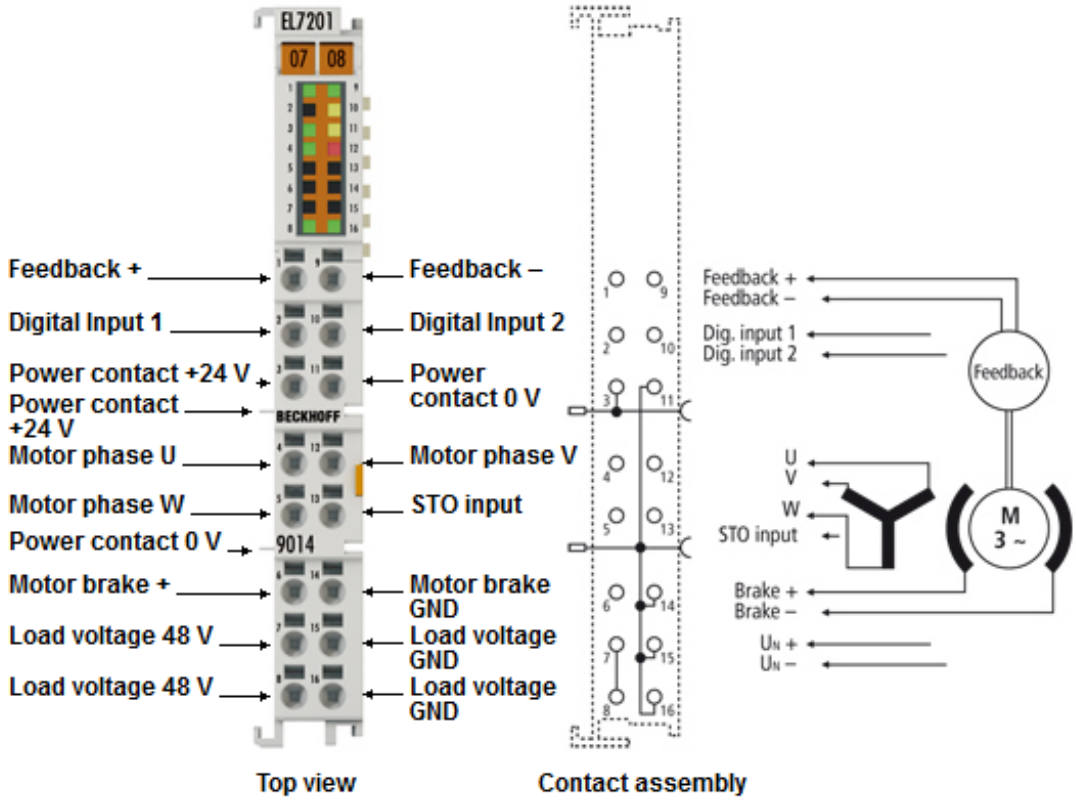


Fig. 33: EL7201-901x Connection

Terminal point	Name	Comment
1	OCT +	Positive input of the absolute feedback
2	Input 1	Digital input 1
3	+24 V	Power contact +24 V
4	U	Motor phase U
5	W	Motor phase W
6	Brake +	Motor brake +
7	48 V	DC link supply + (8 ... 48 V)
8		
9	OCT -	Negative input of the absolute feedback
10	Input 2	Digital input 2
11	0 V	Power contact 0 V
12	V	Motor phase V
13	STO input	Input for STO signal (24 V)
14	Brake GND	Motor brake 0 V
15	0 V	DC link 0 V supply
16		

4.17.2 EL7211-901x, EL7221-901x - LEDs and connection

EL7211-901x, EL7221-901x - LEDs

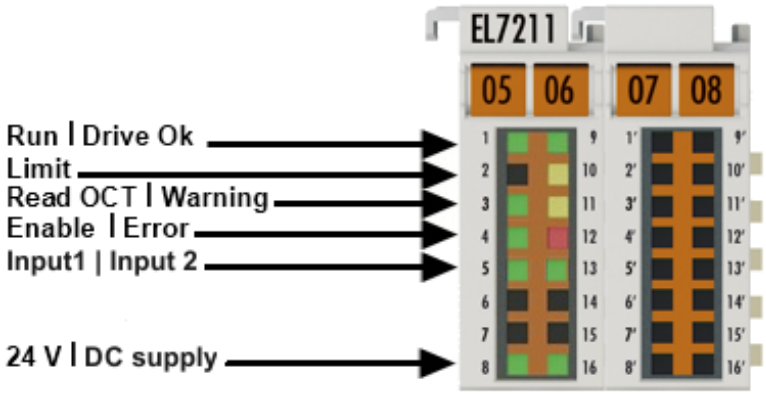


Fig. 34: EL7211-901x, EL7221-901x - LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates [▶ 268]
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
Drive OK	green	on	Driver stage ready for operation
Limit	orange	on	The LED is linked with bit 11 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (internal limit active) Limit reached (e.g. torque or speed limit)
Read OCT	green	flashing	The electronic type plate is being read
		off	The reading of the electronic type plate has been completed
Warning	orange	flashing	Error while reading the type plate
		on	The LED is linked with bit 7 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (warning) The “Warning” threshold value is exceeded. I²T model Voltage missing at STO input Temperature (80°C) exceeded Voltage
Enable	green	on	The LED is linked with the bits 1 and 2 of status word (MDP742 [▶ 208] / DS402 [▶ 240]) (if "Switched on" or "Operation enabled") Driver stage enabled
Input 1 Input 2	green	on	Signal is present at digital inputs (e.g. end position, reference or latch sensors/switches)
Error	red	on	The LED is linked with bit 3 of the status word (MDP742 [▶ 208] / DS402 [▶ 240]) (fault) The “Error” threshold value is exceeded. Overcurrent STO triggered with active axis Voltage not available Resolver not connected Max. temperature (100°C) exceeded
+24 V via power contacts	green	on	24 V voltage supply for the terminal is present.
DC link supply	green	on	Voltage for the DC link supply is present.

NOTICE

Fuse protection of the supply voltage

The electrical protection of the load voltage must be selected in such a way that the maximum flowing current is limited to 3 times the rated current (max. 1 second)!

EL7211-901x, EL7221-901x - Connection

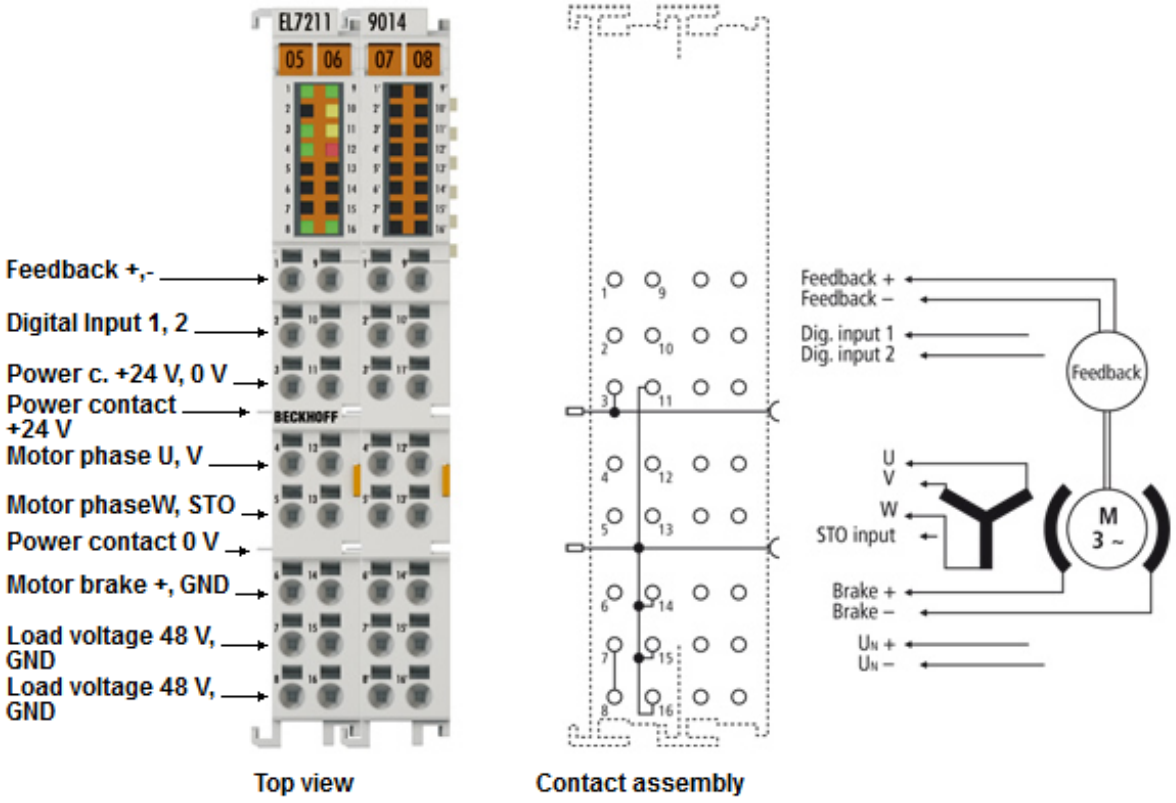


Fig. 35: EL7211-901x, EL7221-901x - Connection

Terminal point	Name	Comment
1	OCT +	Positive input of the absolute feedback
2	Input 1	Digital input 1
3	+24 V	Power contact +24 V
4	U	Motor phase U
5	W	Motor phase W
6	Brake +	Motor brake +
7	48 V	DC link supply + (8 ... 48 V)
8		
9	OCT -	Negative input of the absolute feedback
10	Input 2	Digital input 2
11	0 V	Power contact 0 V
12	V	Motor phase V
13	STO input	Input for STO signal (24 V)
14	Brake GND	Motor brake 0 V
15	0 V	DC link 0 V supply
16		
1' - 16'		n.c.

4.18 Disposal



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

5 Commissioning

5.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including a multi PLC system, NC axis control, programming and operation. The whole system is mapped through this environment and enables access to a programming environment (including compilation) for the controller. Individual digital or analog inputs or outputs can also be read or written directly, in order to verify their functionality, for example.

For further information, please refer to http://infosys.beckhoff.com:

- **EtherCAT System Manual:**
  Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O Configuration
- In particular, for TwinCAT – driver installation:
  Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the relevant terminals for the actual configuration. All configuration data can be entered directly via editor functions (offline) or via the `scan function (online):

- **“offline”**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.
  - The procedure for the offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO Configuration → Add an I/O device
- **“online”**: The existing hardware configuration is read
  - See also http://infosys.beckhoff.com:
    Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged between the user PC and individual control elements:

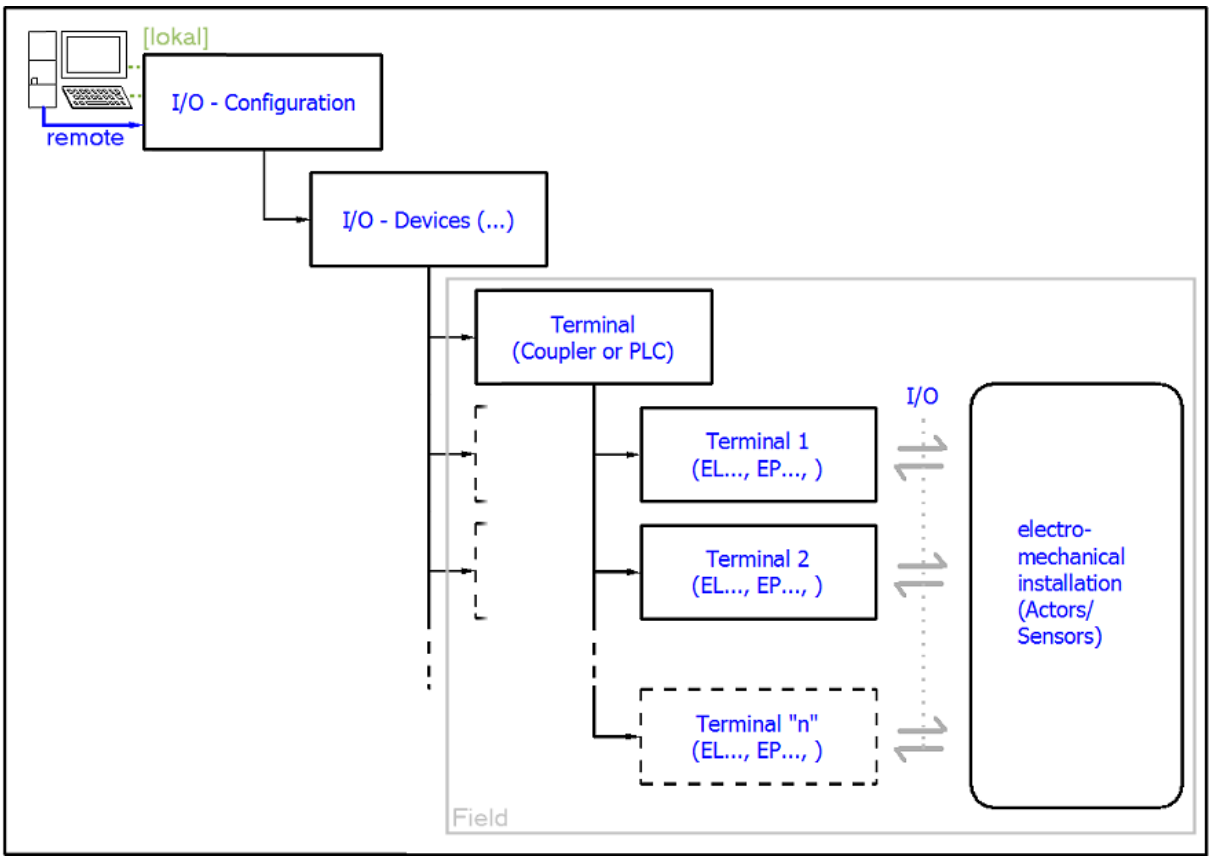


Fig. 36: Relationship between user side (commissioning) and installation

Insertion of certain components (I/O device, terminal, box...) by users functions the same way as in TwinCAT 2 and TwinCAT 3. The descriptions below relate solely to the online procedure.

Example configuration (actual configuration)

Based on the following example configuration, the subsequent subsections describe the procedure for TwinCAT 2 and TwinCAT 3:

- **CX2040** control system (PLC) including **CX2100-0004** power supply unit
- Connected to CX2040 on the right (E-bus):
  EL1004 (4-channel digital input terminal 24 V DC)
- Linked via the X001 port (RJ-45): **EK1100** EtherCAT Coupler
- Connected to the EK1100 EtherCAT Coupler on the right (E-bus):
  EL2008 (8-channel digital output terminal 24 V DC; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)

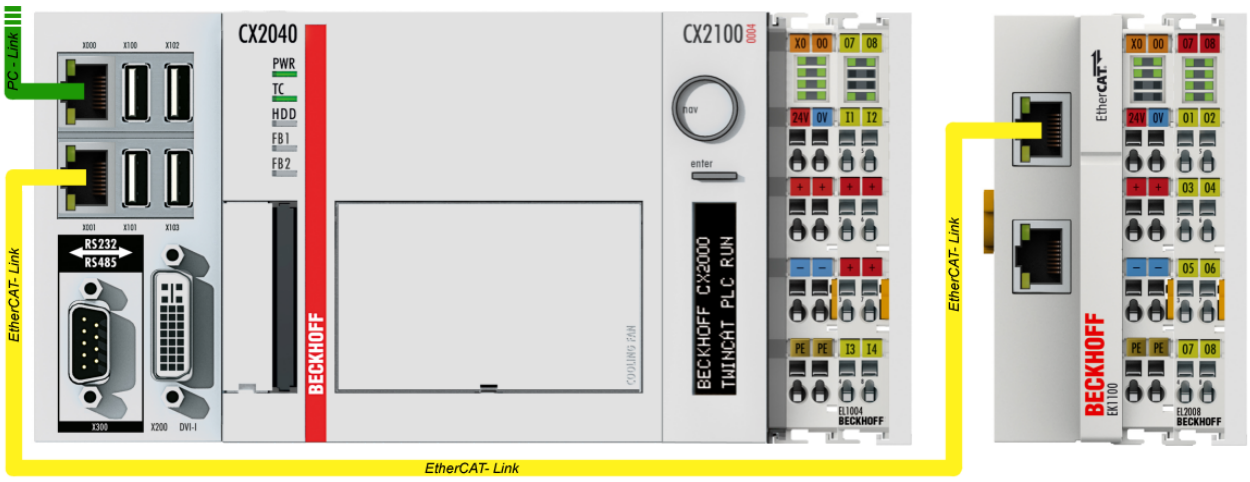


Fig. 37: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn't be necessary.

5.1.1 TwinCAT 2

Startup

TwinCAT 2 basically uses two user interfaces: the TwinCAT System Manager for communication with the electromechanical components and TwinCAT PLC Control for the development and compilation of a controller. The starting point is the TwinCAT System Manager.

After successful installation of the TwinCAT system on the PC to be used for development, the TwinCAT 2 System Manager displays the following user interface after startup:

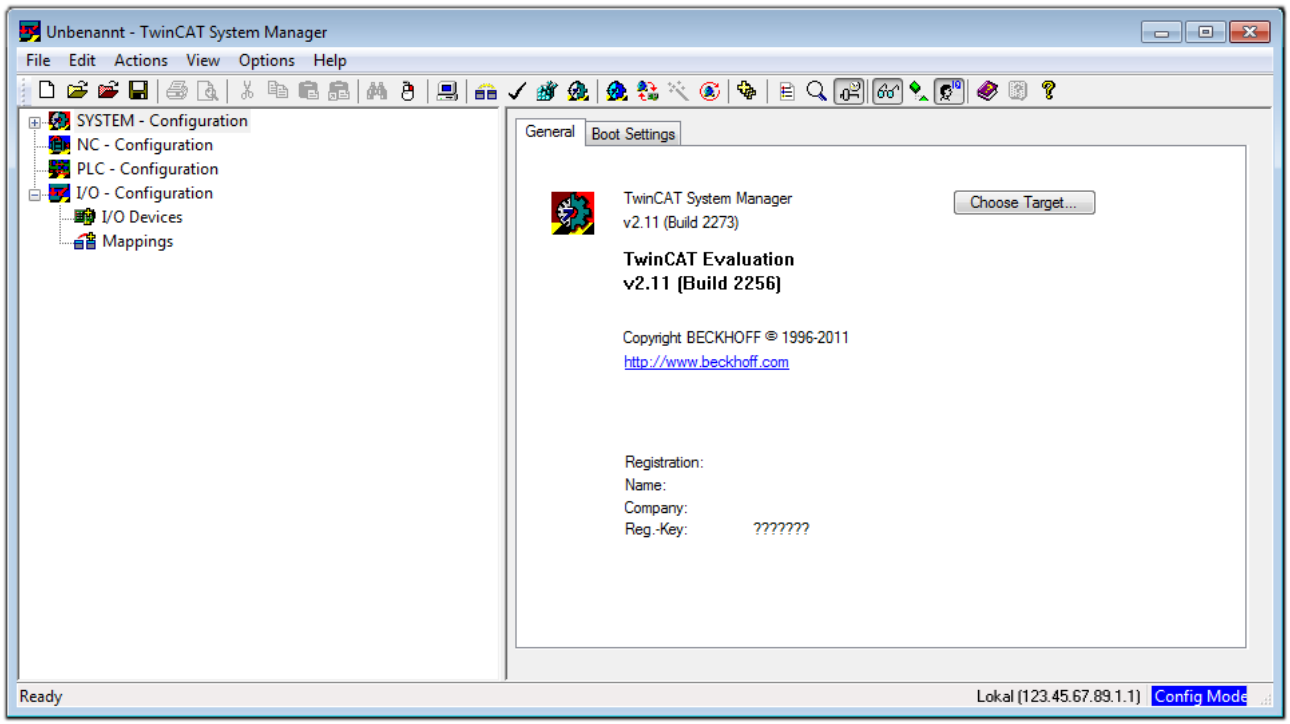


Fig. 38: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is “Insert Device [▶ 68]”.

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. In the menu under “Actions” → “Choose Target System...”, the following window is opened for this via the symbol “  ” or the “F8” key:

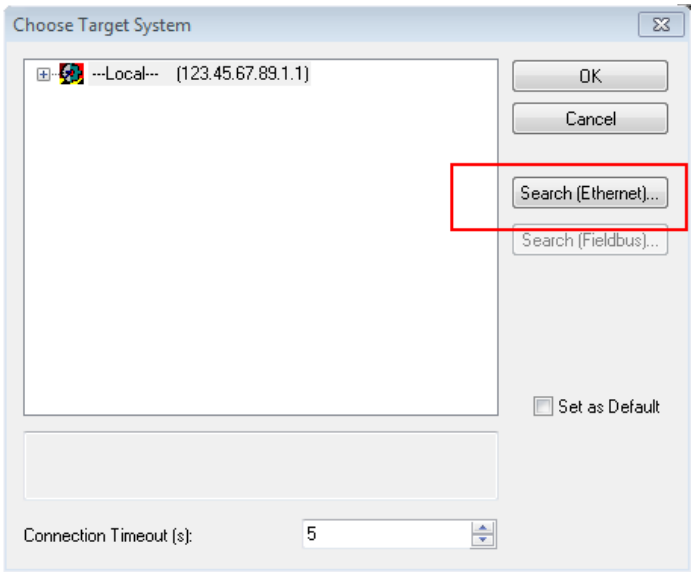


Fig. 39: Selection of the target system

Use “Search (Ethernet)...” to enter the target system. Thus another dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID

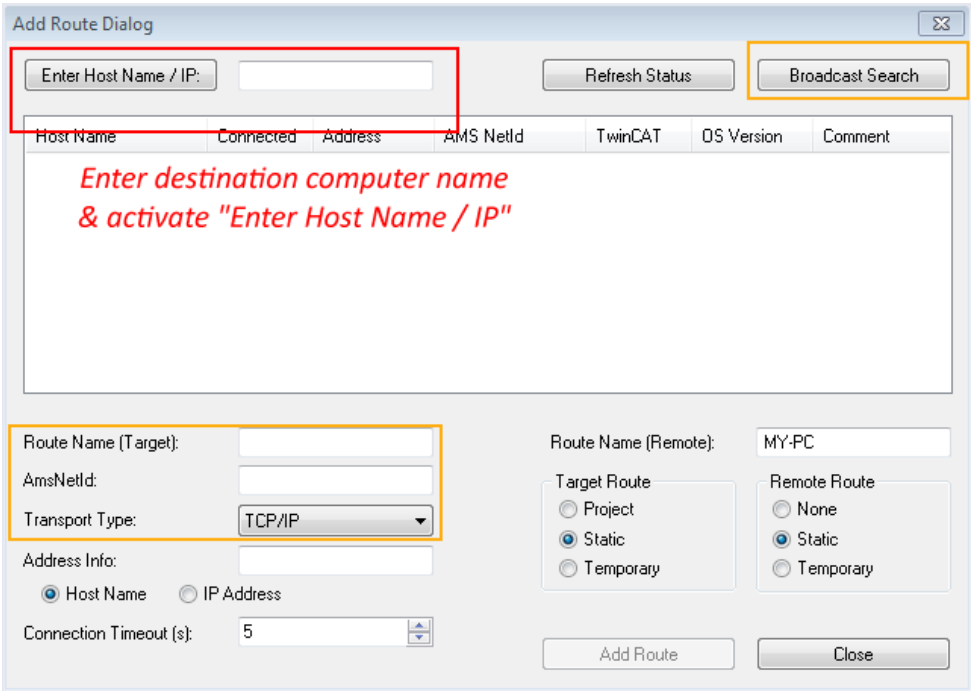
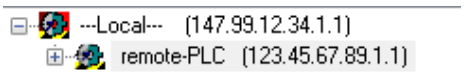


Fig. 40: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (a correct password may have to be entered before this):



After confirmation with “OK”, the target system can be accessed via the System Manager.

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select “I/O Devices” and then right-click to open a context menu and select “Scan Devices…”, or start the action in the menu bar via  . The TwinCAT System Manager may first have to be set to “Config Mode” via  or via the menu
“Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

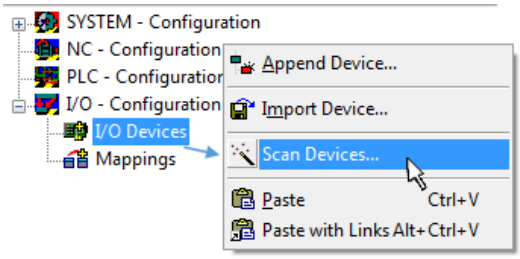


Fig. 41: Select “Scan Devices...”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

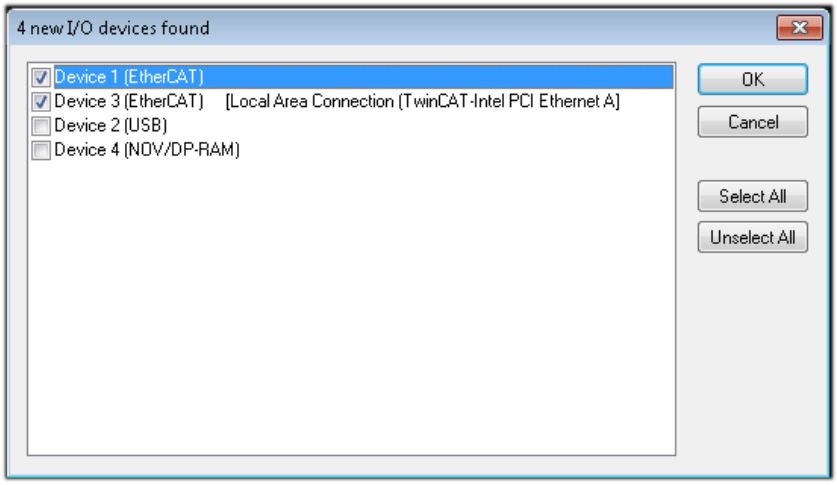


Fig. 42: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config Mode” and should also be acknowledged.

Based on the example configuration [▶ 64] described at the beginning of this section, the result is as follows:

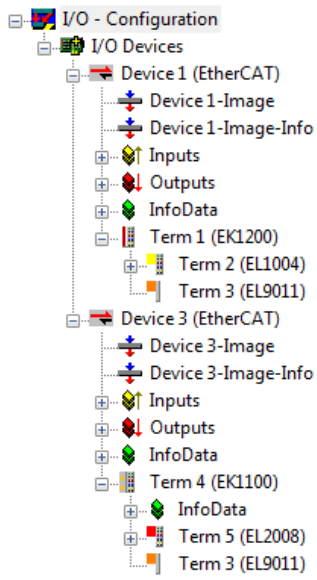


Fig. 43: Mapping of the configuration in the TwinCAT 2 System Manager

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting “Device ...” from the context menu, which then only reads the elements below which are present in the configuration:

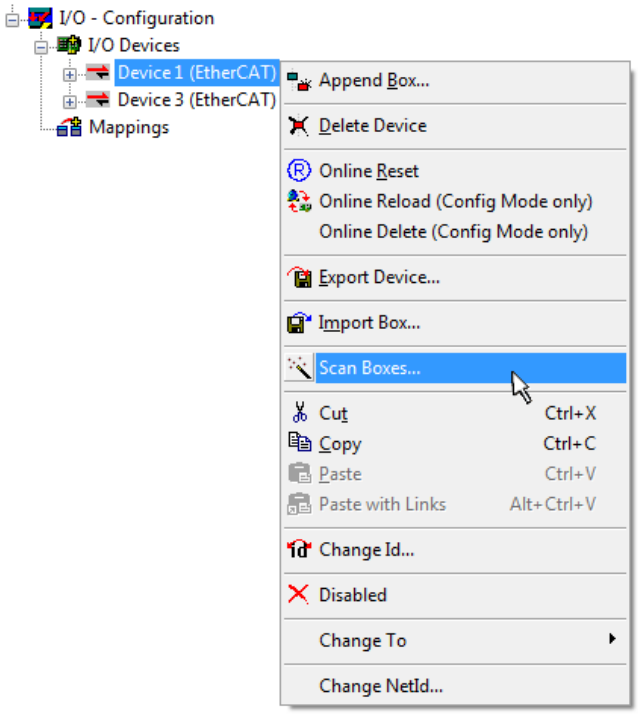


Fig. 44: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming and integrating the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)

- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:

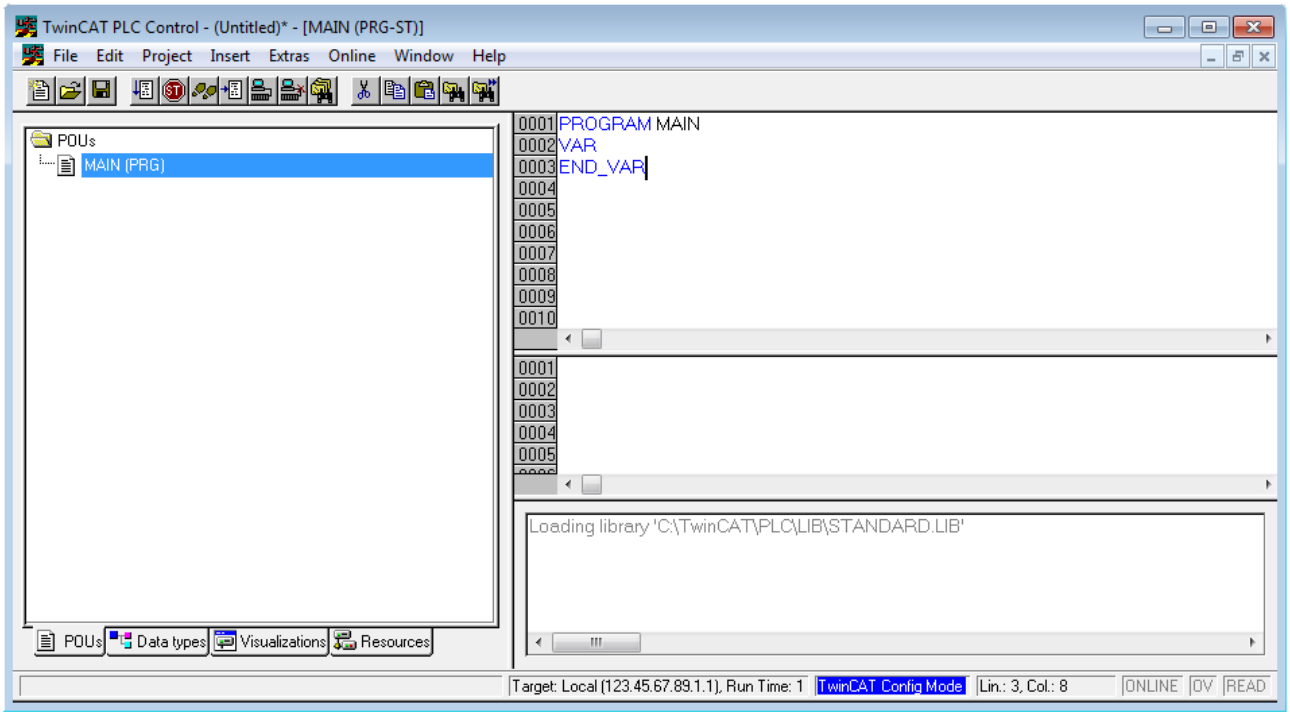


Fig. 45: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name “PLC_example.pro”:

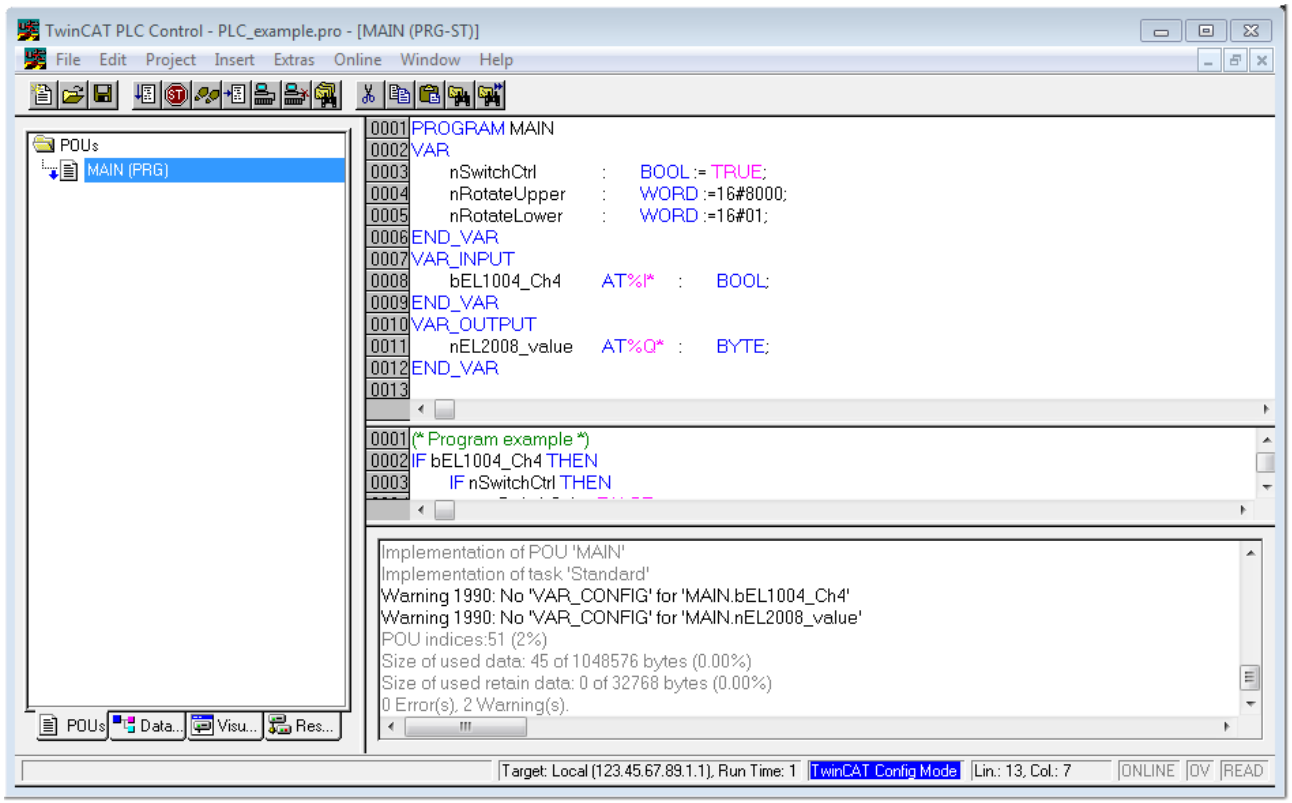


Fig. 46: Example program with variables after a compile process (without variable integration)

Warning 1990 (missing “VAR_CONFIG”) after a compile process indicates that the variables defined as external (with the ID “AT%I*” or “AT%Q*”) have not been assigned. After successful compilation, TwinCAT PLC Control creates a “*.tpy” file in the directory in which the project was stored. This file (“*.tpy”) contains variable assignments and is not known to the System Manager, hence the warning. Once the System Manager has been notified, the warning no longer appears.

First, integrate the TwinCAT PLC Control project in the **System Manager**. This is performed via the context menu of the PLC configuration (right-click) and selecting “Append PLC Project…”:

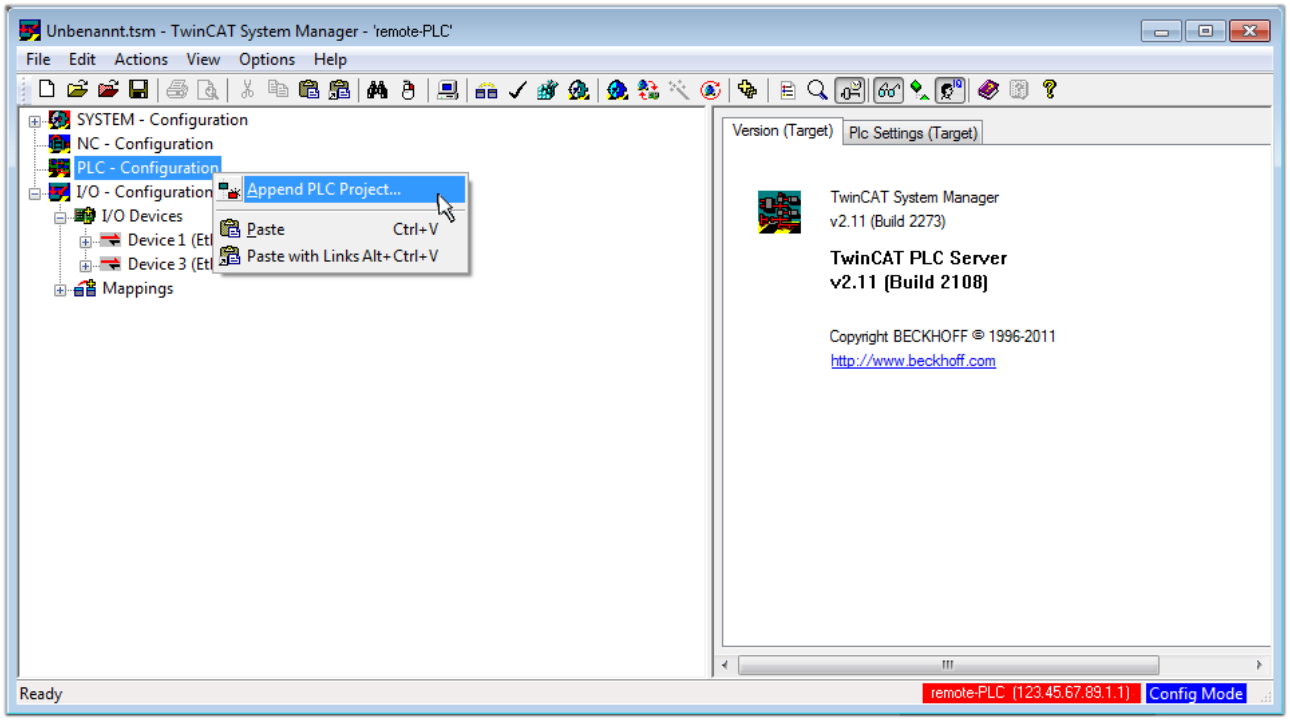


Fig. 47: Appending the TwinCAT PLC Control project

Select the PLC configuration “PLC_example.tpy” in the browser window that opens. The project including the two variables identified with “AT” are then integrated in the configuration tree of the System Manager:

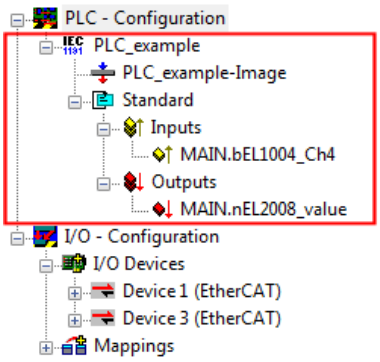


Fig. 48: PLC project integrated in the PLC configuration of the System Manager

The two variables “bEL1004_Ch4” and “nEL2008_value” can now be assigned to certain process objects of the I/O configuration.

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project “PLC_example” and via “Modify Link...” “Standard”:

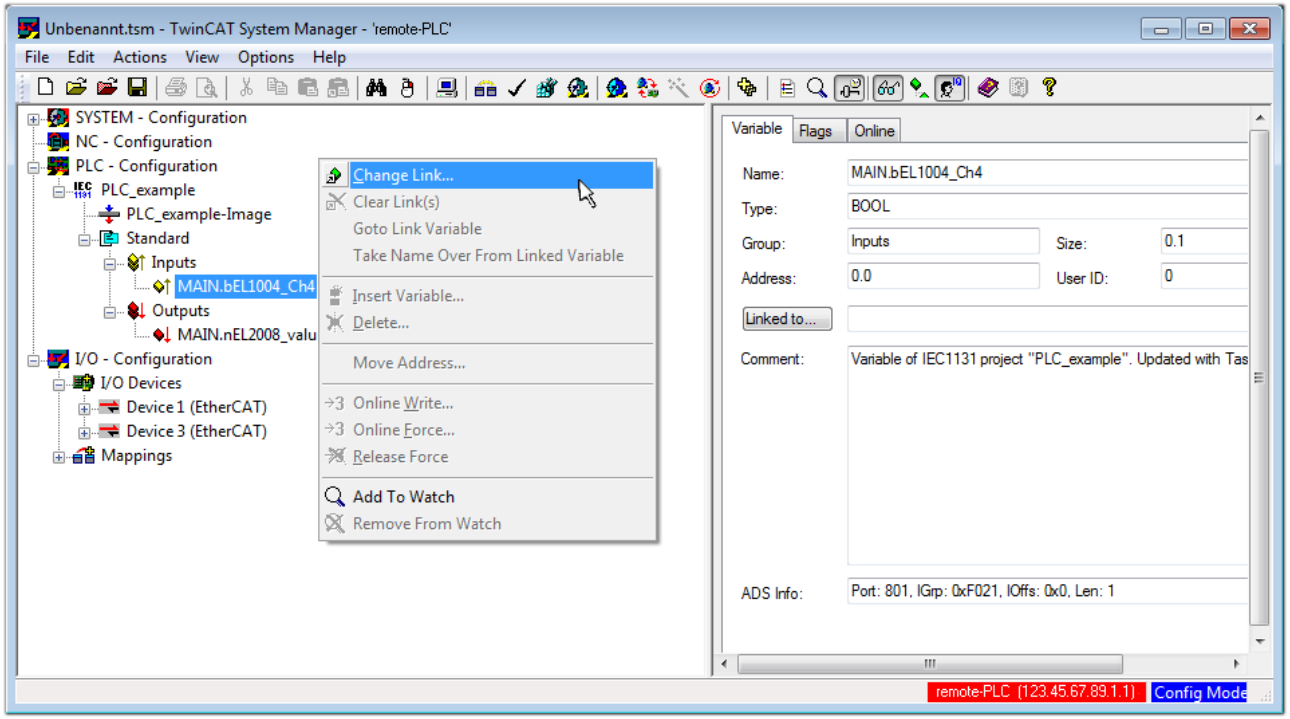


Fig. 49: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

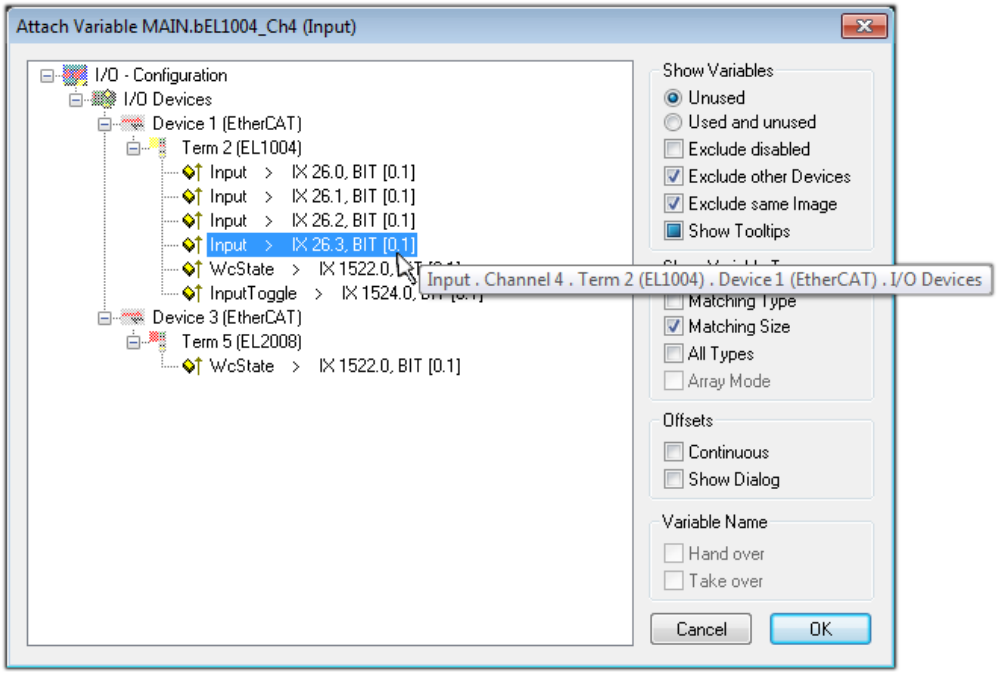


Fig. 50: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

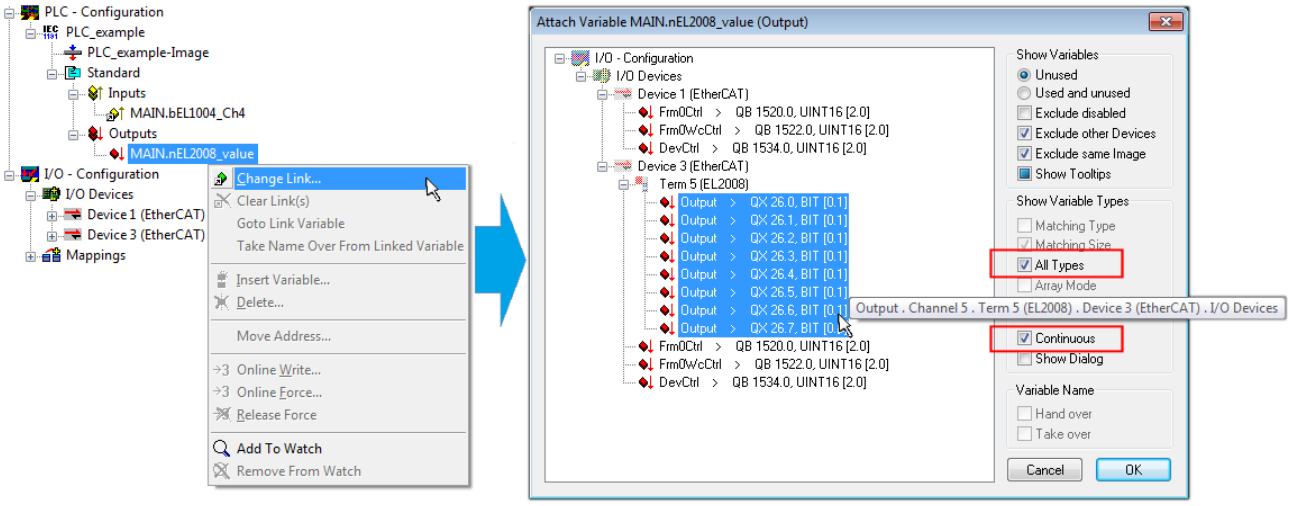


Fig. 51: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

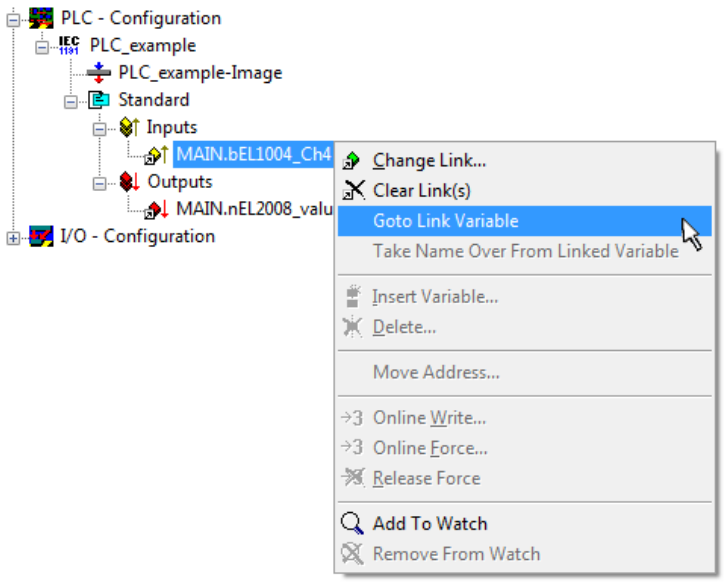
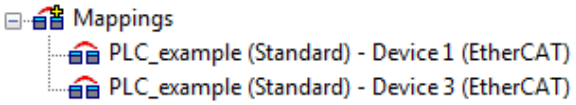


Fig. 52: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of assigning variables to the PDO is completed via the menu option “Actions” → “Create assignment”, or via  .

This can be visualized in the configuration:



The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word, integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified via  (or via “Actions” → “Check Configuration”). If no error is present, the configuration can be activated via  (or via “Actions” → “Activate Configuration…”) to transfer the System Manager settings to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”.

A few seconds later, the real-time status 

is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

Starting from a remote system, the PLC control has to be linked with the embedded PC over the Ethernet via “Online” → “Choose Runtime System…”:

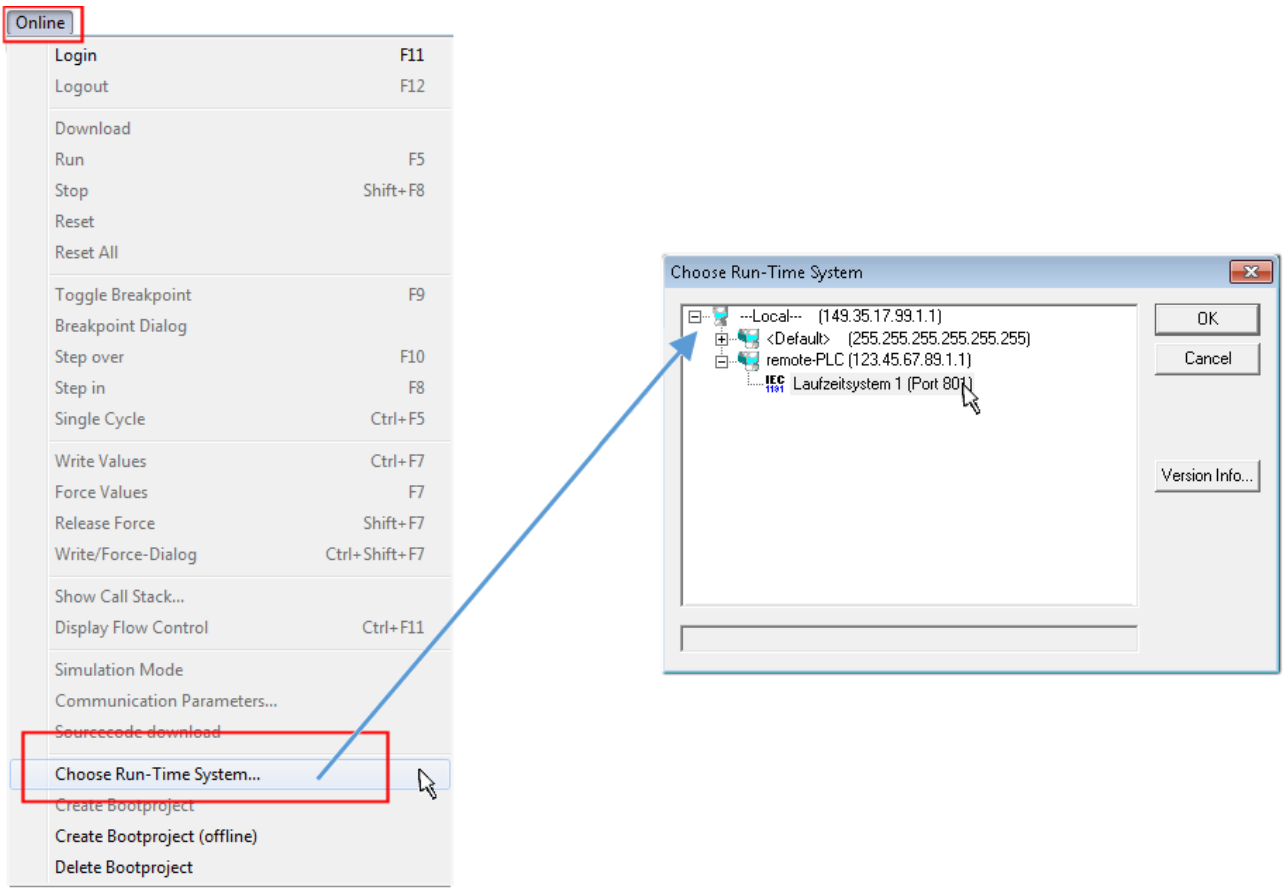


Fig. 53: Choose target system (remote)

In this example, “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time system via the menu option “Online” → “Login”, the F11 key or by clicking on the symbol . The control program can then be loaded for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be confirmed with “Yes”. The runtime environment is ready for the program start:

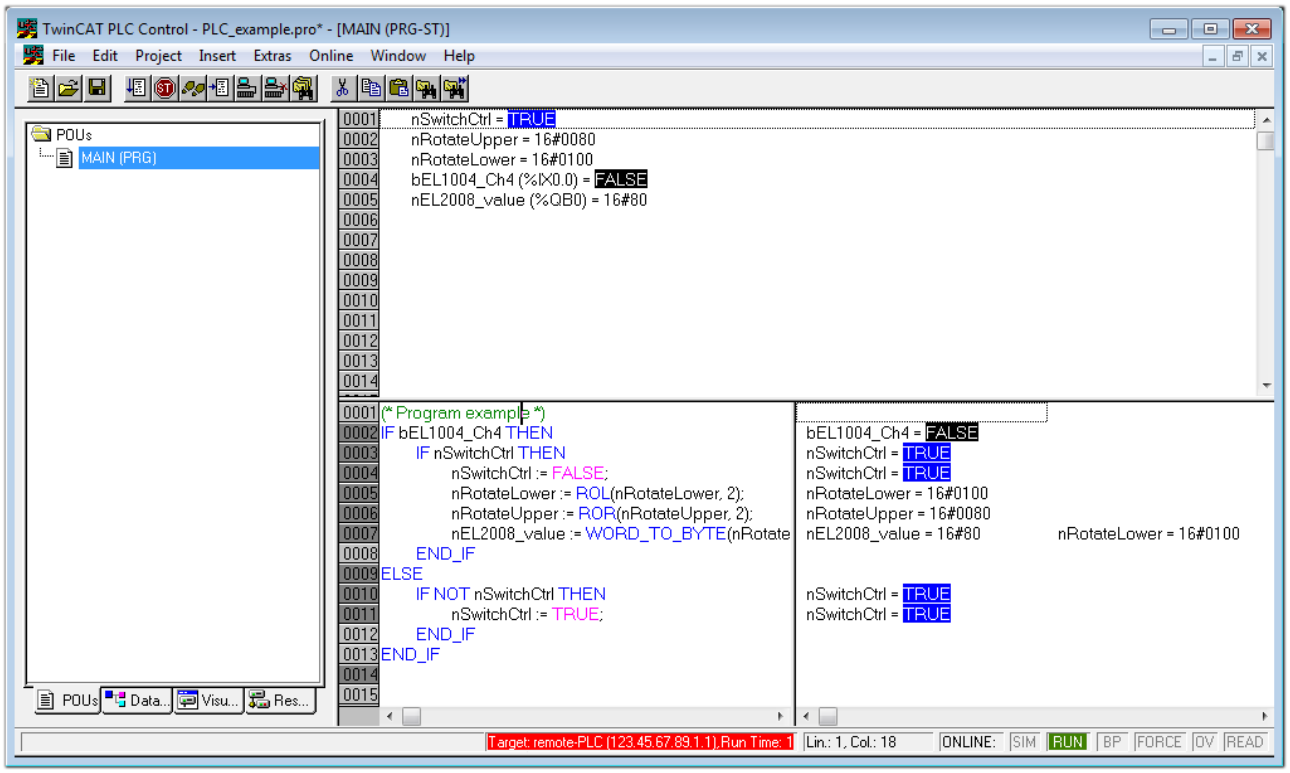


Fig. 54: PLC Control logged in, ready for program startup

The PLC can now be started via “Online” → “Run”, F5 key or .

5.1.2 TwinCAT 3

Startup

TwinCAT 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see “TwinCAT System Manager” of TwinCAT 2) for communication with the electromechanical components.

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:

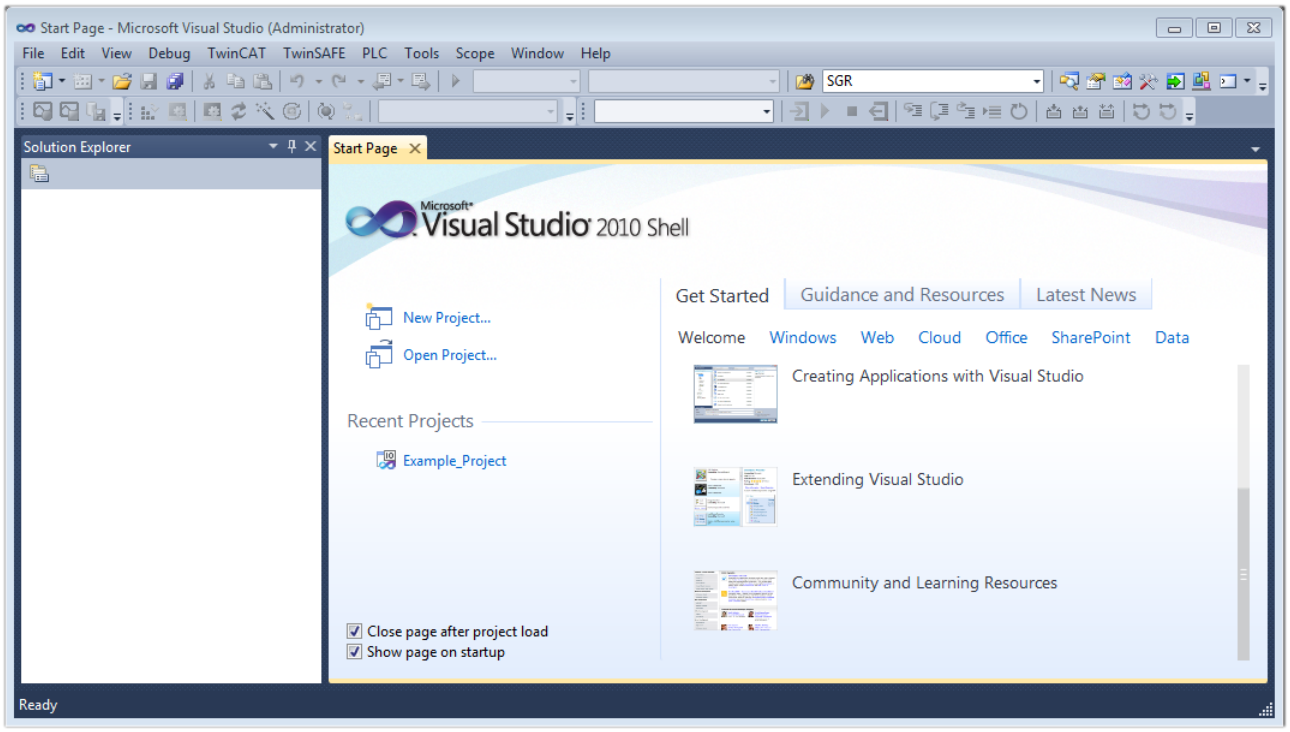


Fig. 55: Initial TwinCAT 3 user interface

First create a new project via  New TwinCAT Project... (or under “File”→“New”→ “Project…”). In the following dialog, make the corresponding entries as required (as shown in the diagram):

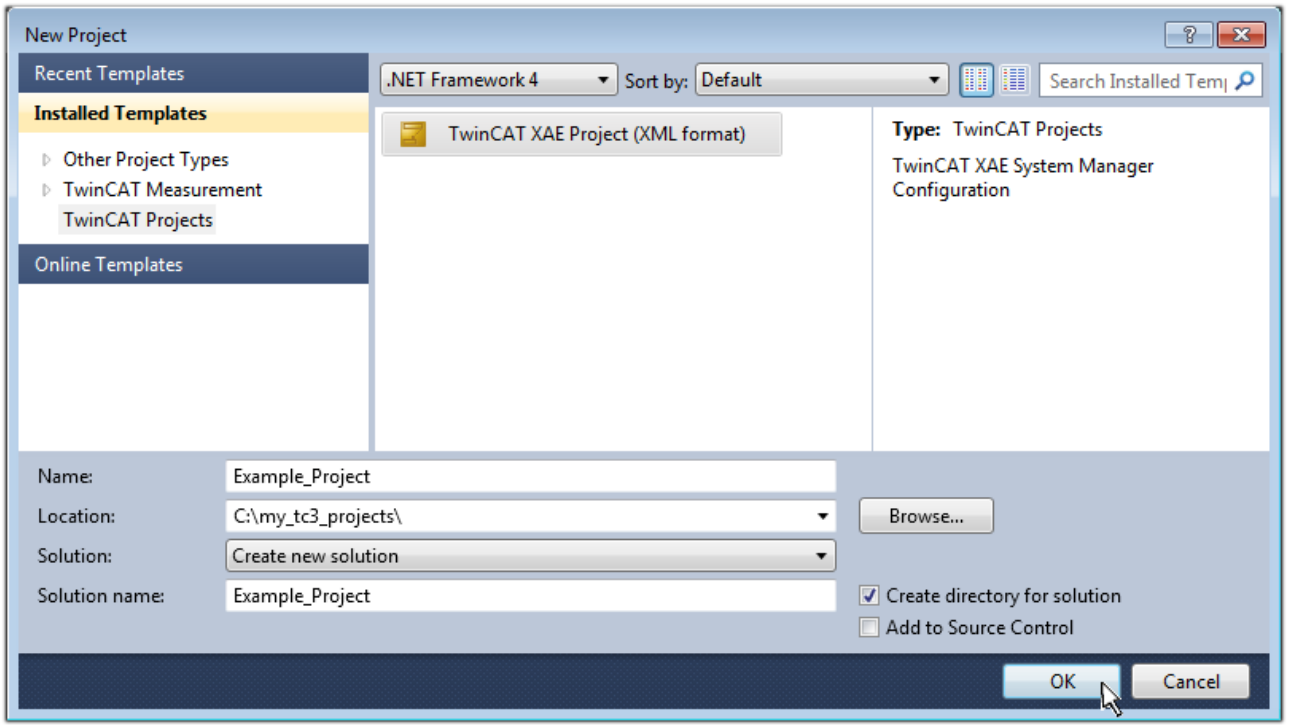


Fig. 56: Create new TwinCAT 3 project

The new project is then available in the project folder explorer:

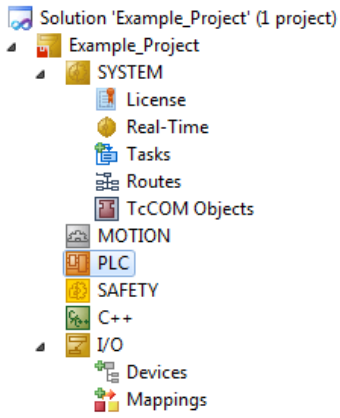
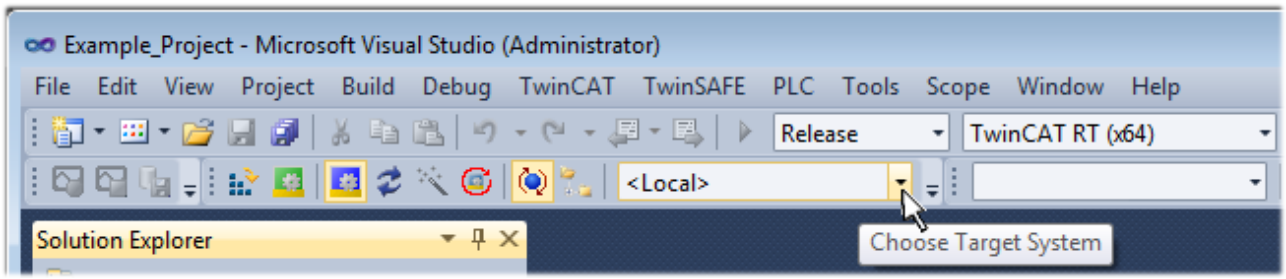


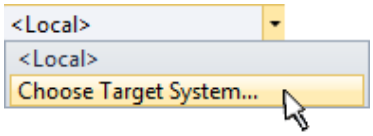
Fig. 57: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 79]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

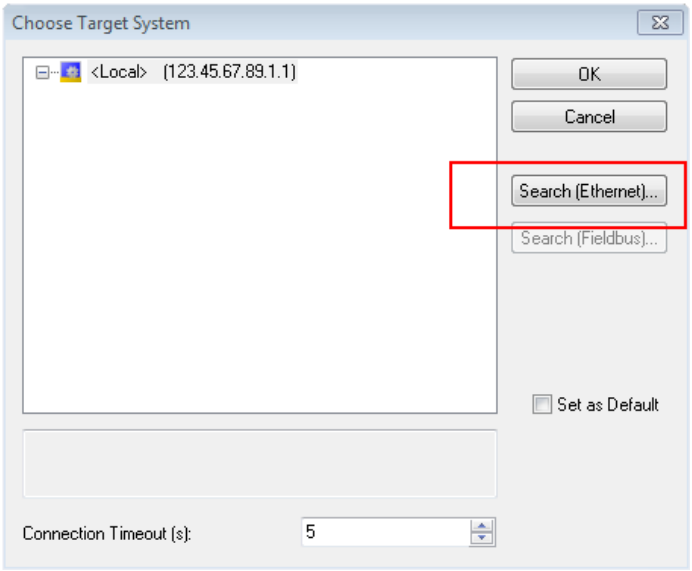


Fig. 58: Selection dialog: Choose the target system

Use “Search (Ethernet)...” to enter the target system. Thus another dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID

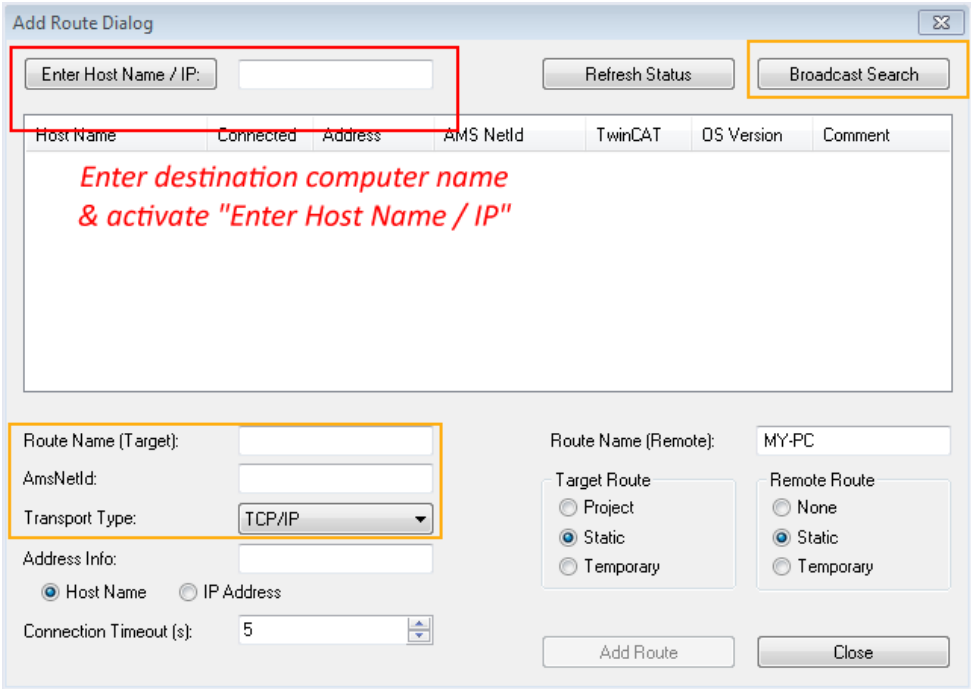
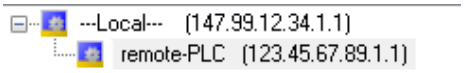


Fig. 59: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (the correct password may have to be entered beforehand):



After confirmation with “OK” the target system can be accessed via the Visual Studio shell.

Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select “Devices” within the element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  in the menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via  or via the menu “TwinCAT” → “Restart TwinCAT (Config Mode)”.

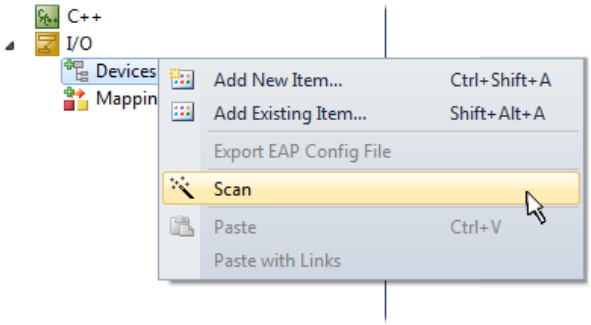


Fig. 60: Select “Scan”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

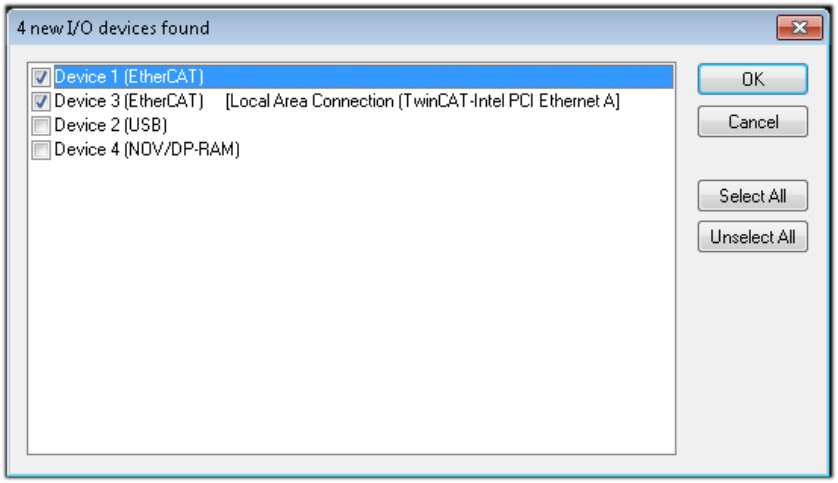


Fig. 61: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config Mode” and should also be acknowledged.

Based on the example configuration [▶ 64] described at the beginning of this section, the result is as follows:

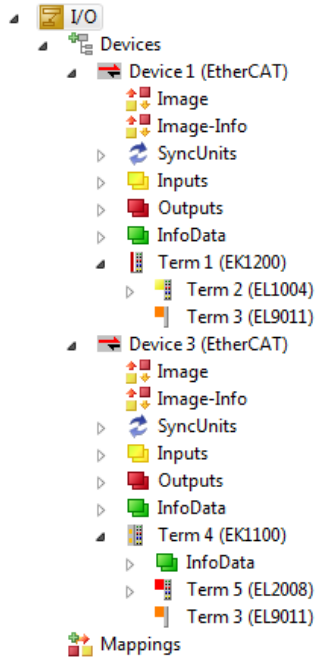


Fig. 62: Mapping of the configuration in VS shell of the TwinCAT 3 environment

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting “Device ...” from the context menu, which then only reads the elements below which are present in the configuration:

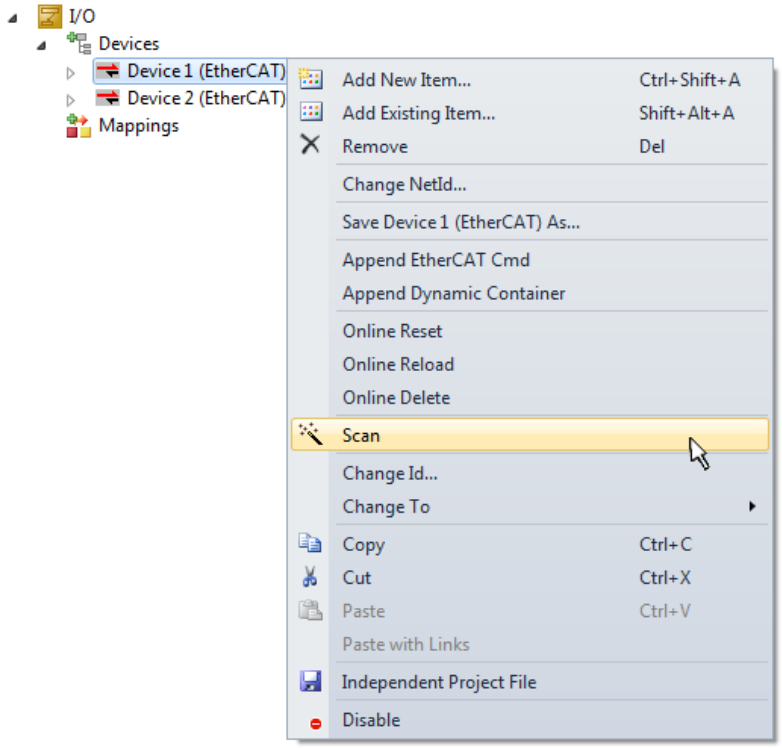


Fig. 63: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)
- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

In order to create a programming environment, a PLC subproject is added to the example project via the context menu of the “PLC” in the project folder explorer by selecting “Add New Item….”:

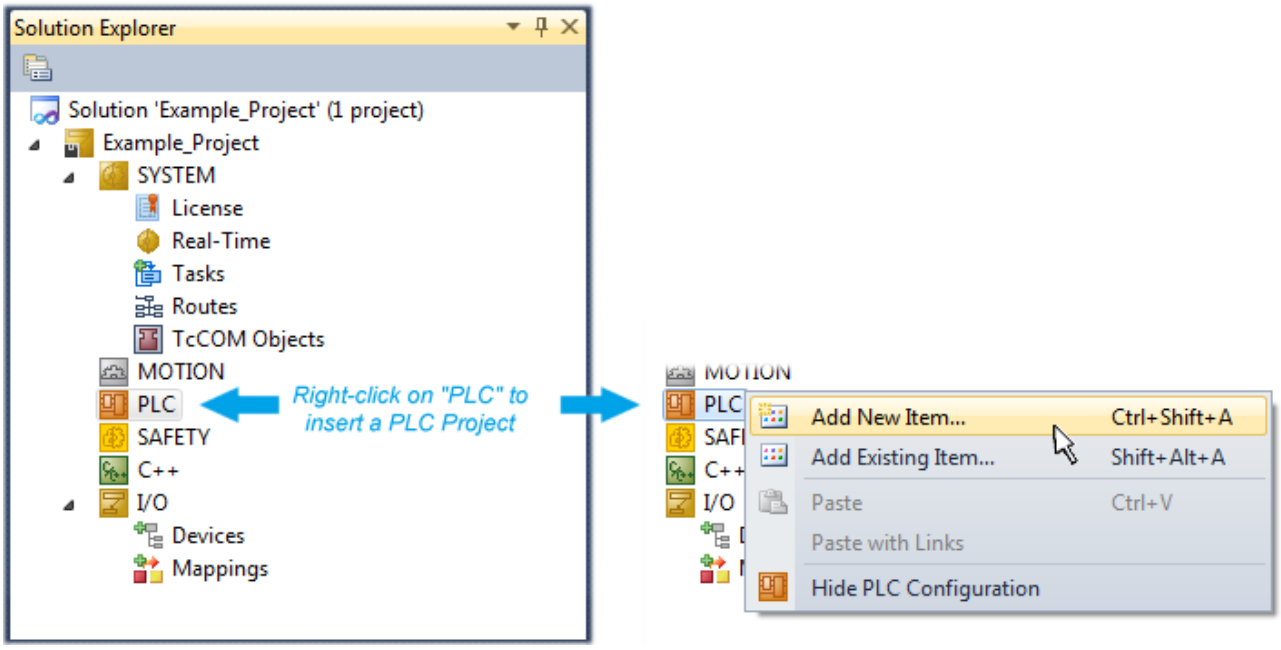


Fig. 64: Adding the programming environment in “PLC”

In the dialog that opens, select “Standard PLC project” and enter “PLC_example” as project name, for example, and select a corresponding directory:

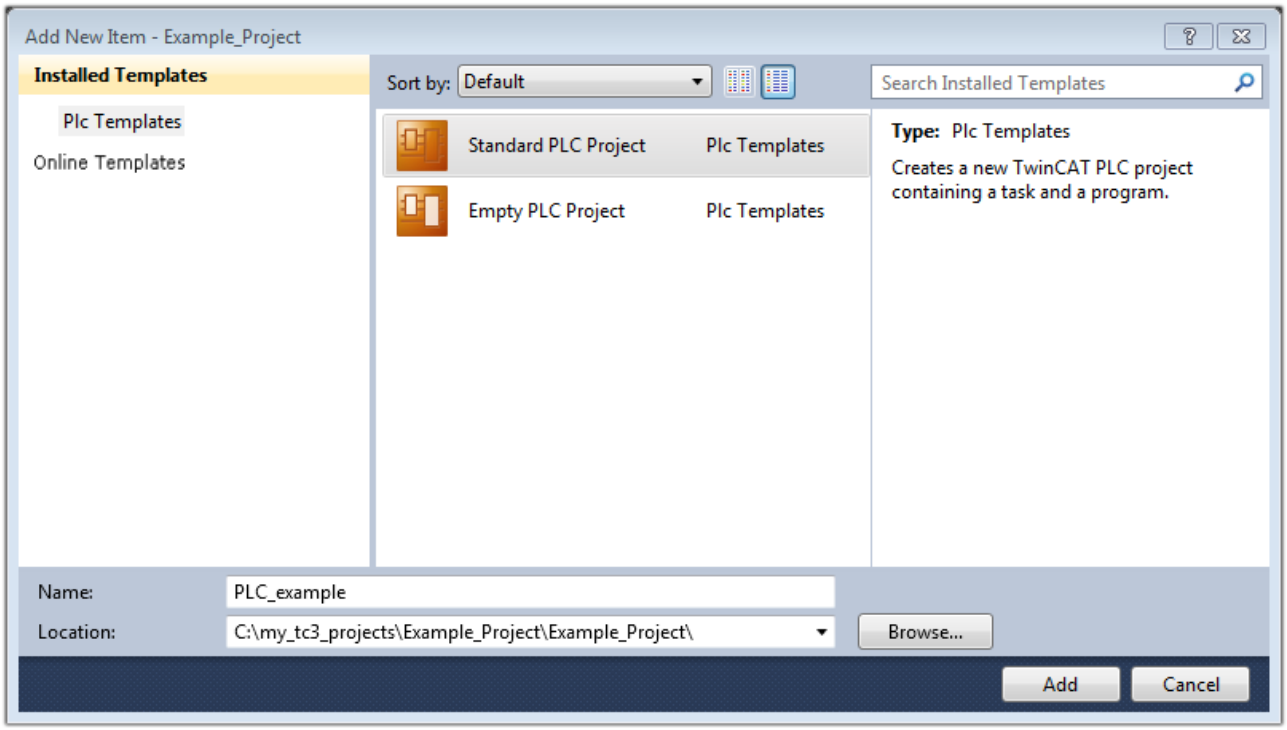


Fig. 65: Specifying the name and directory for the PLC programming environment

The “Main” program, which already exists due to selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:

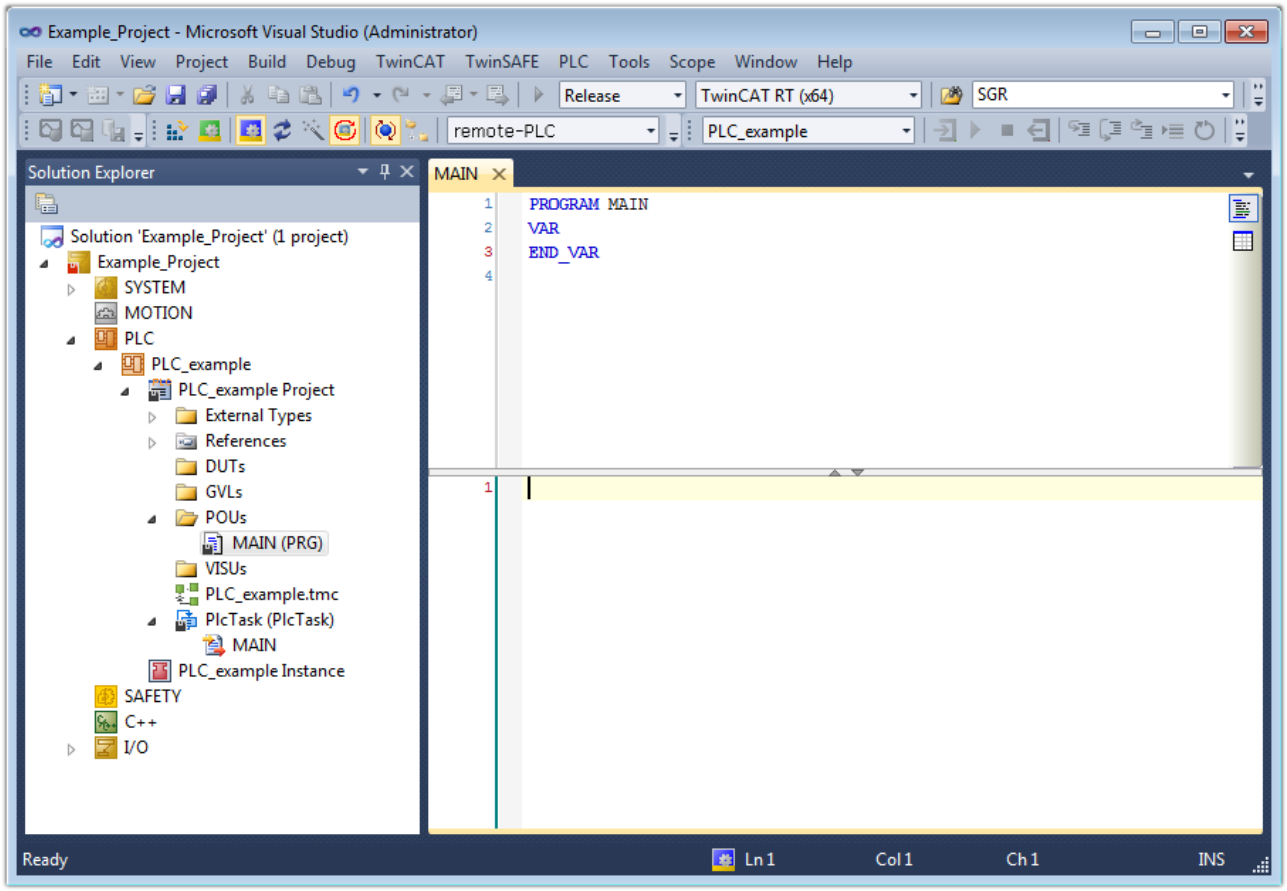


Fig. 66: Initial “Main” program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:

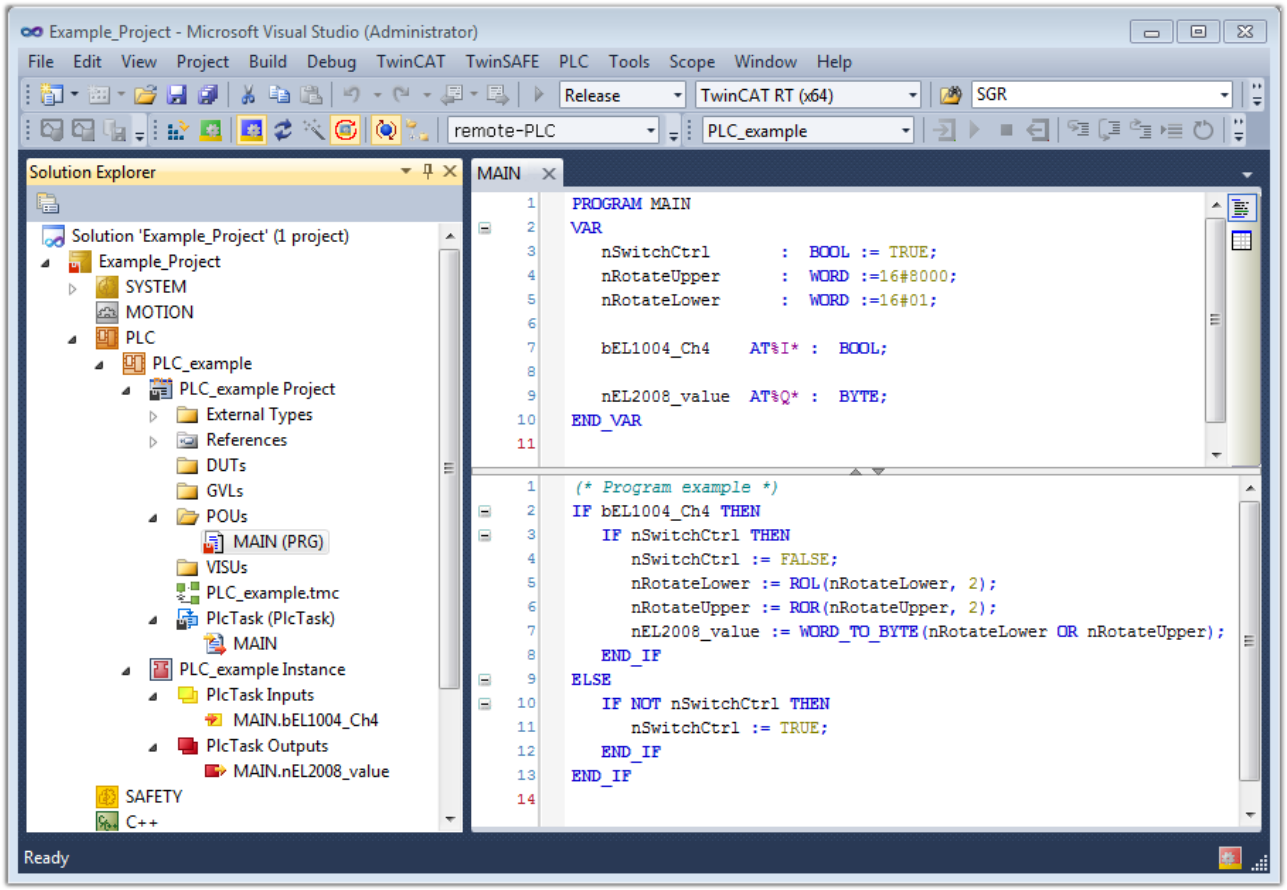


Fig. 67: Example program with variables after a compile process (without variable integration)

The control program is now created as a project folder, followed by the compile process:

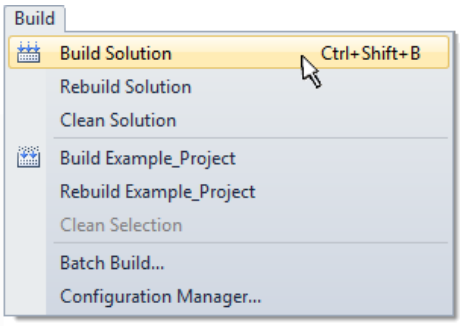
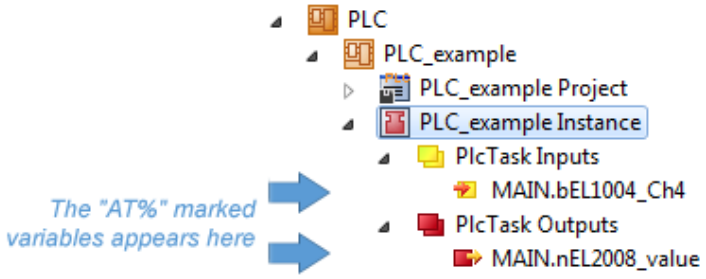


Fig. 68: Start program compilation

The following variables, identified in the ST/PLC program with “AT%”, are then available under “Assignments” in the project folder explorer:



Assigning variables

Via the menu of an instance – variables in the “PLC” context, use the “Modify Link…” option to open a window to select a suitable process object (PDO) for linking:

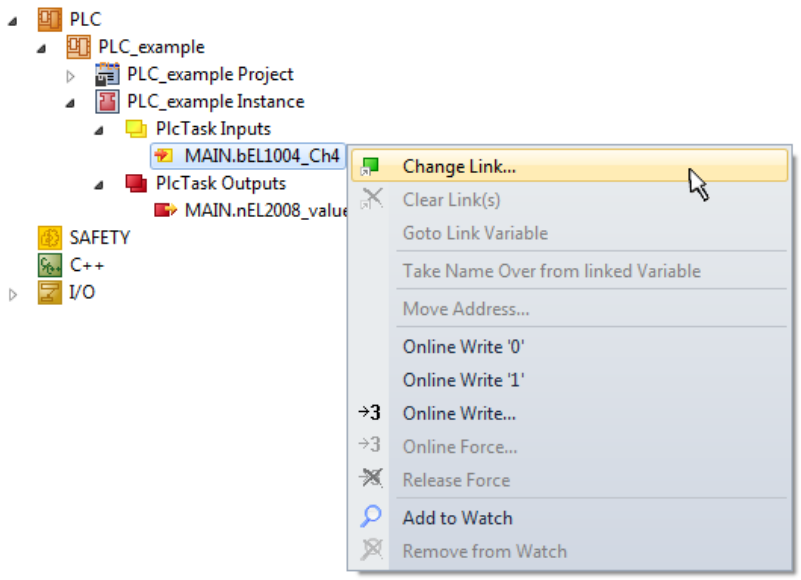


Fig. 69: Creating the links between PLC variables and process objects

In the window that opens, the process object for the “bEL1004_Ch4” BOOL-type variable can be selected from the PLC configuration tree:

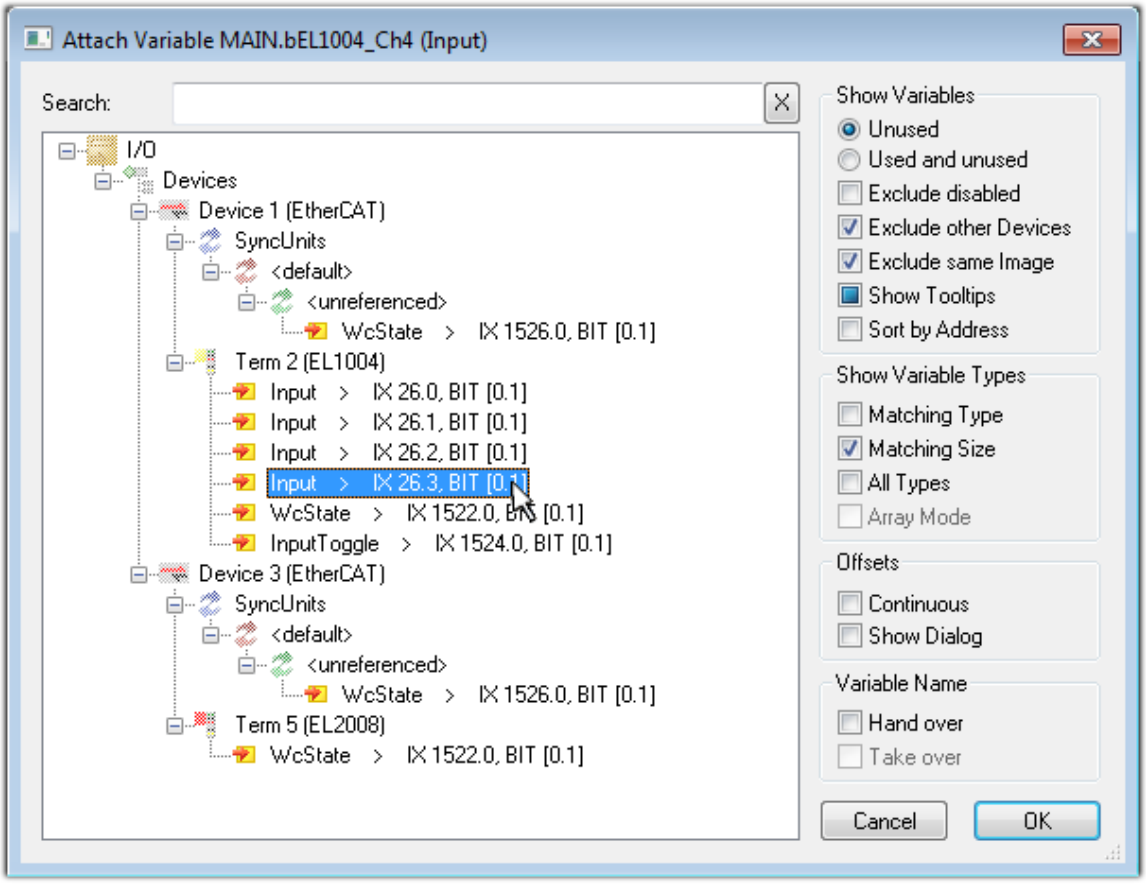


Fig. 70: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

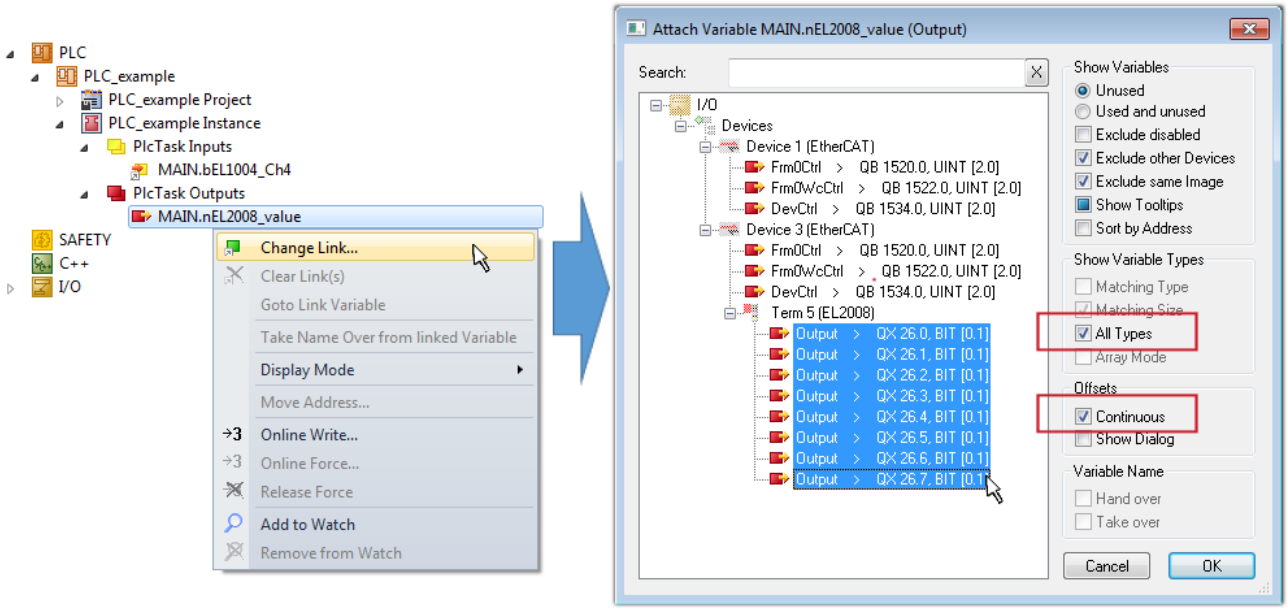


Fig. 71: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

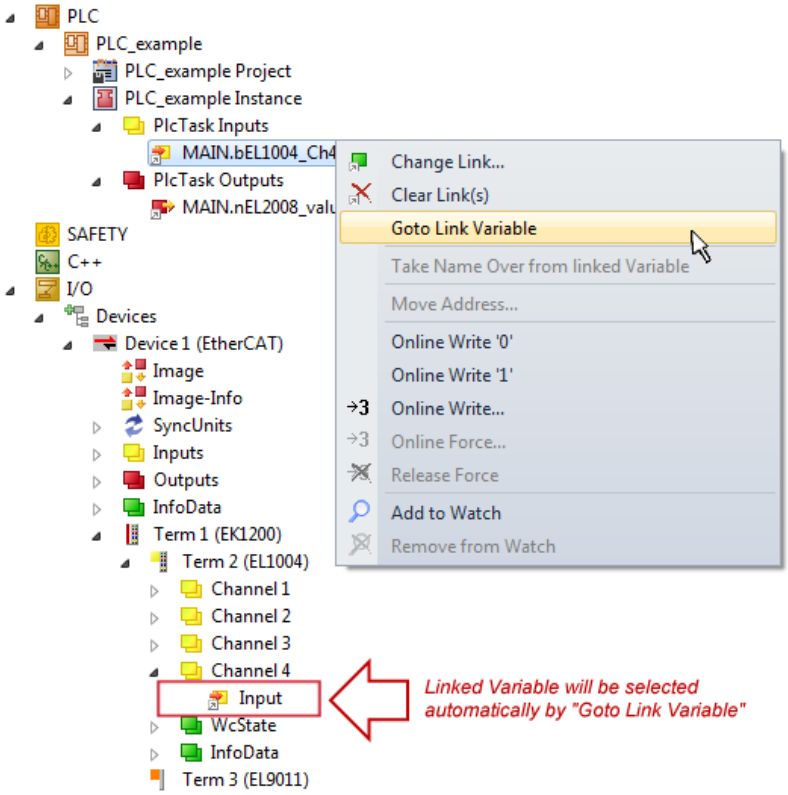


Fig. 72: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word,

integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on type of variable assignment

The following type of variable assignment can only be used from TwinCAT version V3.1.4024.4 onwards and is only available for terminals with a microcontroller.

In TwinCAT, a structure can be created from the mapped process data of a terminal. An instance of this structure can then be created in the PLC, so it is possible to access the process data directly from the PLC without having to declare own variables.

The procedure for the EL3001 1-channel analog input terminal -10...+10 V is shown as an example.

1. First, the required process data must be selected in the “Process data” tab in TwinCAT.
2. After that, the PLC data type must be generated in the “PLC” tab via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.

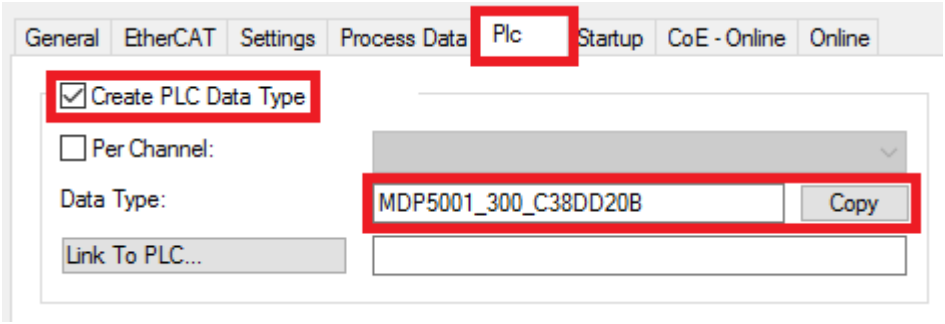


Fig. 73: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.

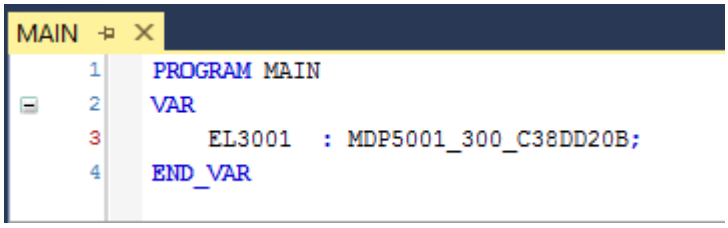


Fig. 74: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination “CTRL + Shift + B” or via the “Build” tab in TwinCAT.
6. The structure in the “PLC” tab of the terminal must then be linked to the created instance.

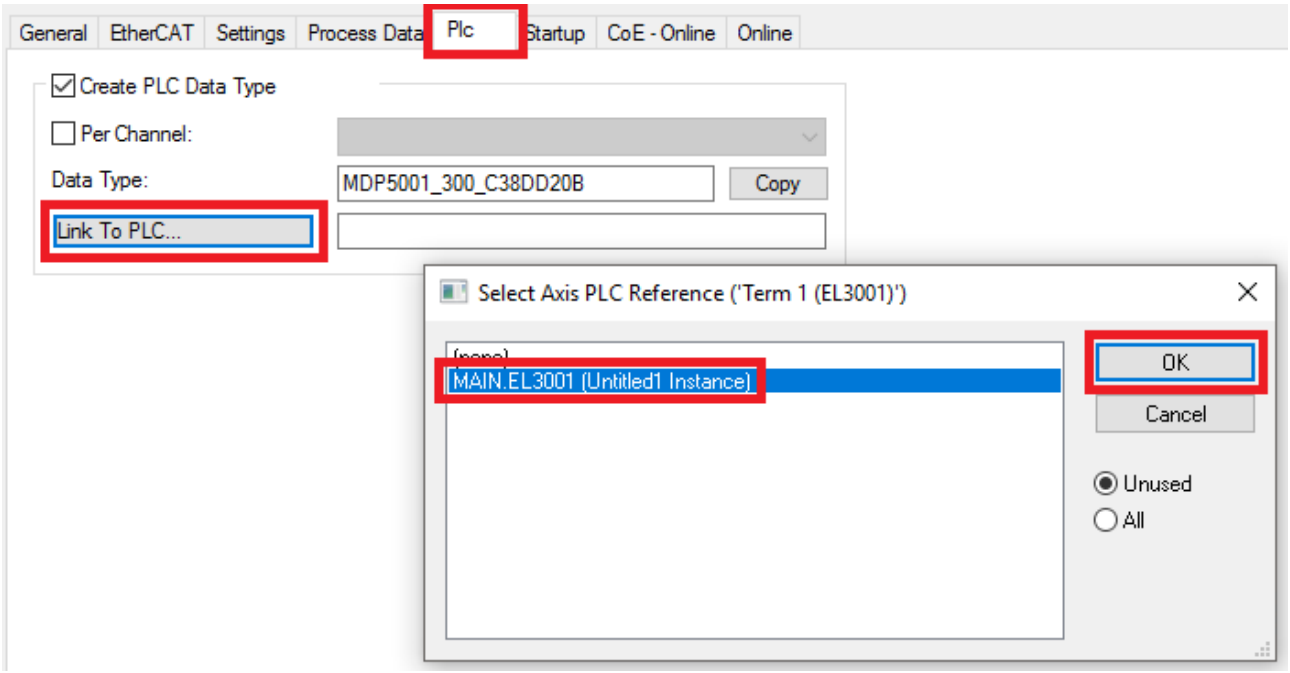


Fig. 75: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.

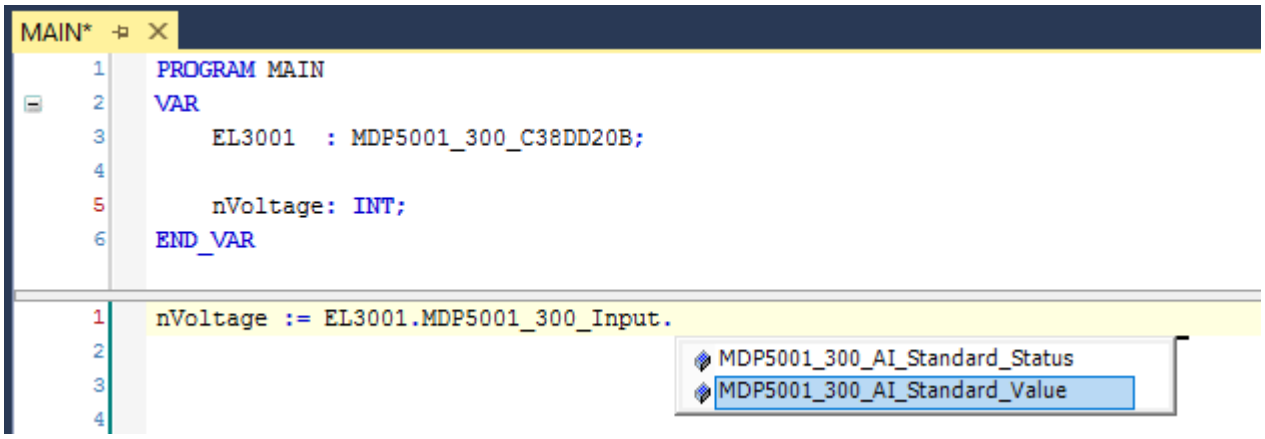
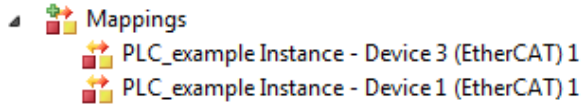


Fig. 76: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with  or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:



A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol  at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option “PLC” → “Login” or click on to link the PLC with the real-time system and load the control program for execution. This results in the message ”*No program on the controller! Should the new program be loaded?*”, which should be acknowledged with “Yes”. The runtime environment is ready for the program to be started by clicking on symbol , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:

Fig. 77: TwinCAT 3 development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping and logout result in the required action (also, “Shift + F5” can be used for stop, or both actions can be selected via the PLC menu).

5.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
  - Connects I/O devices to tasks in a variable-oriented manner
  - Connects tasks to tasks in a variable-oriented manner
  - Supports units at the bit level
  - Supports synchronous or asynchronous relationships
  - Exchange of consistent data areas and process images

- Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/ 2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**
  - Visual Studio® integration
  - Choice of the programming language
  - Supports object orientated extension of IEC 61131-3
  - Usage of C/C++ as programming language for real time applications
  - Connection to MATLAB®/Simulink®
  - Open interface for expandability
  - Flexible run-time environment
  - Active support of multi-core- and 64 bit operating system
  - 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.

5.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

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

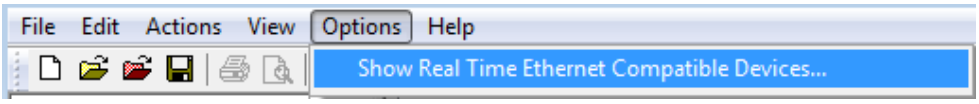


Fig. 78: System Manager “Options” (TwinCAT 2)

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

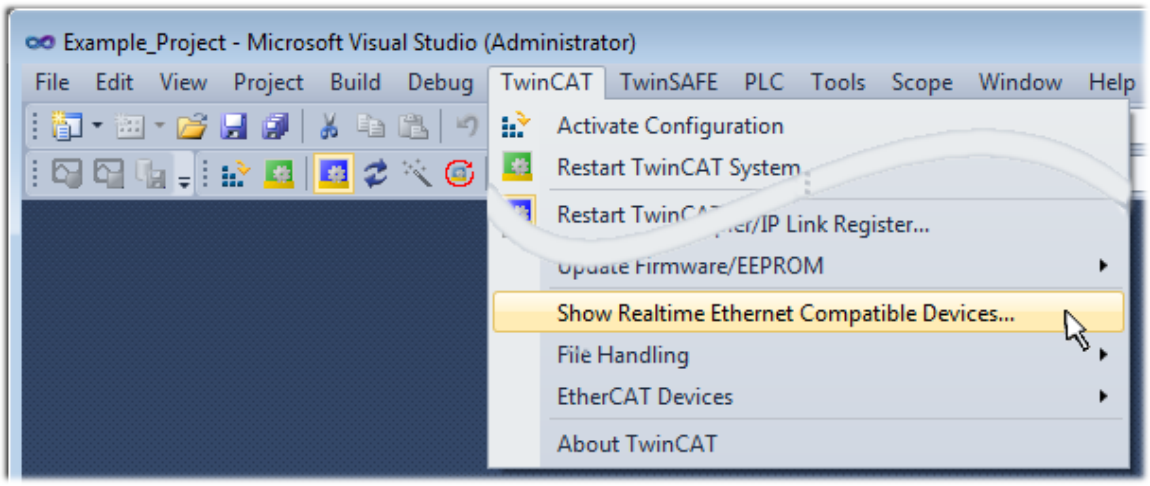


Fig. 79: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

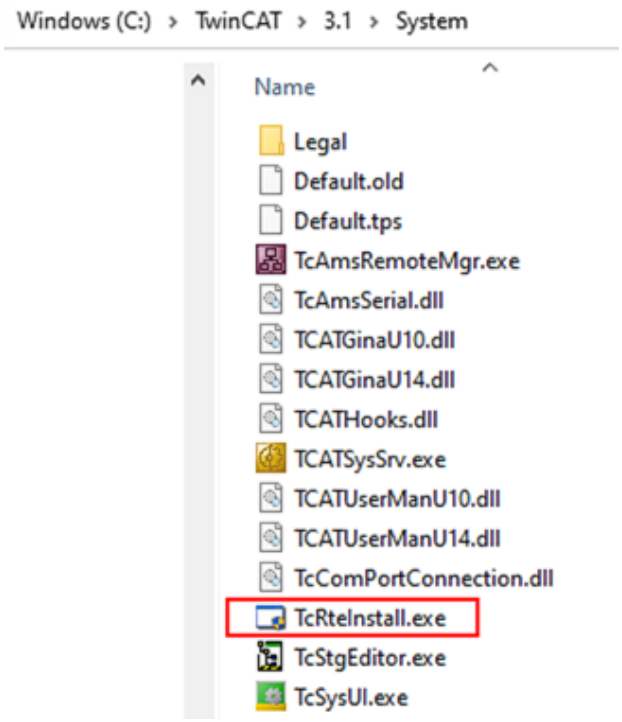


Fig. 80: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

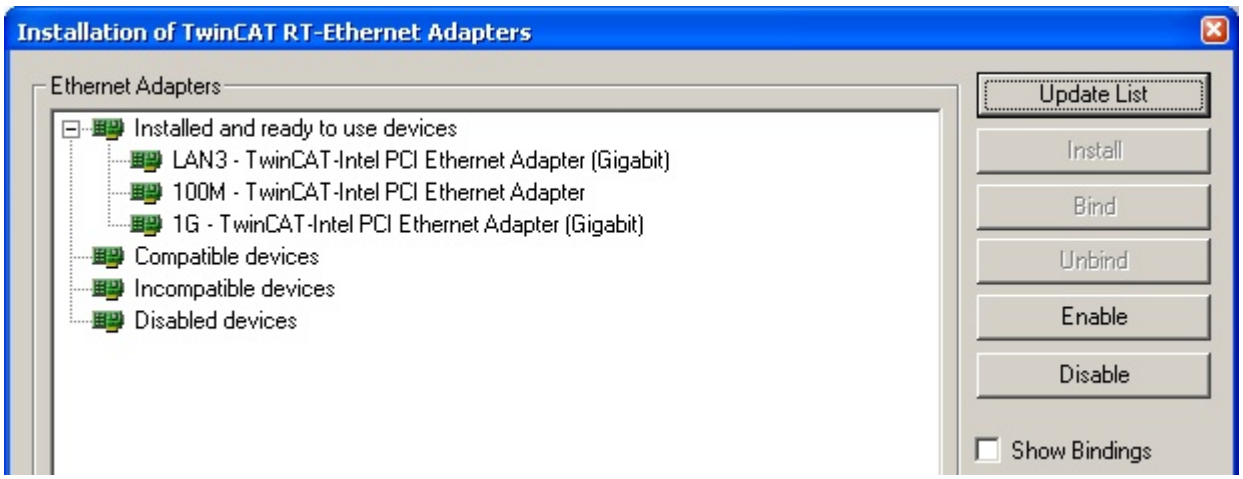


Fig. 81: Overview of network interfaces

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

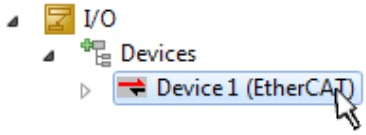
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 102] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



Fig. 82: 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)

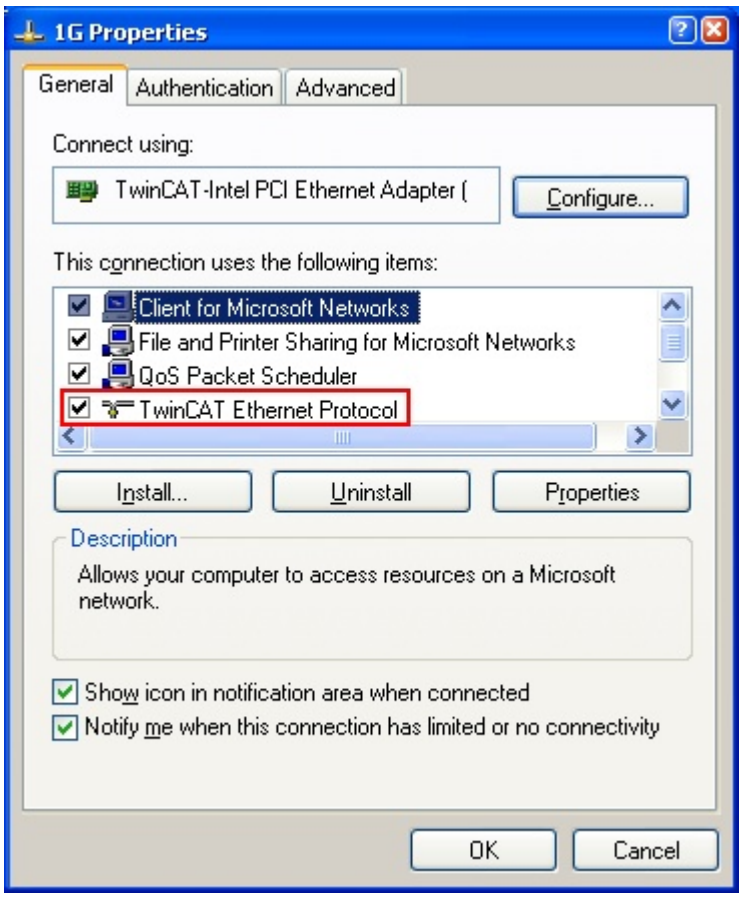


Fig. 83: Windows properties of the network interface

A correct setting of the driver could be:

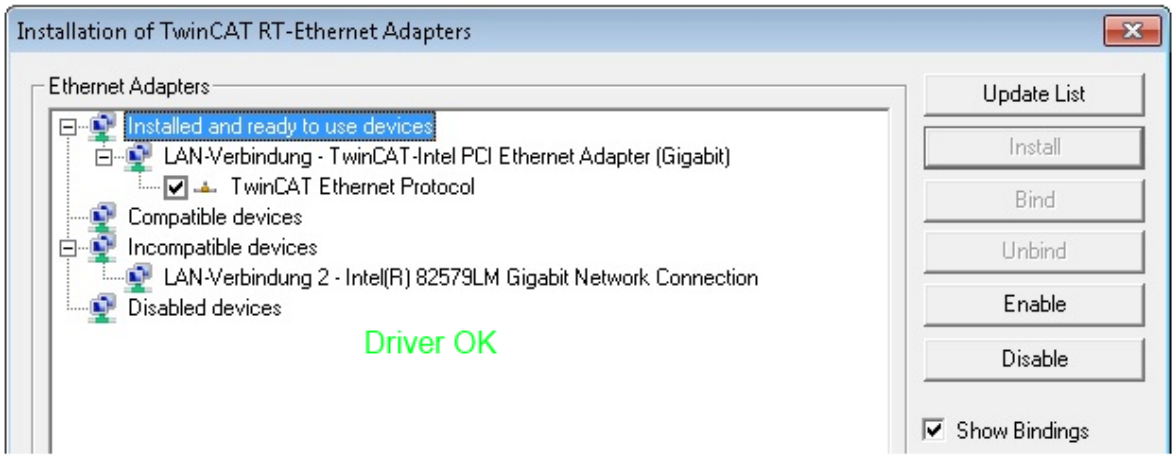


Fig. 84: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

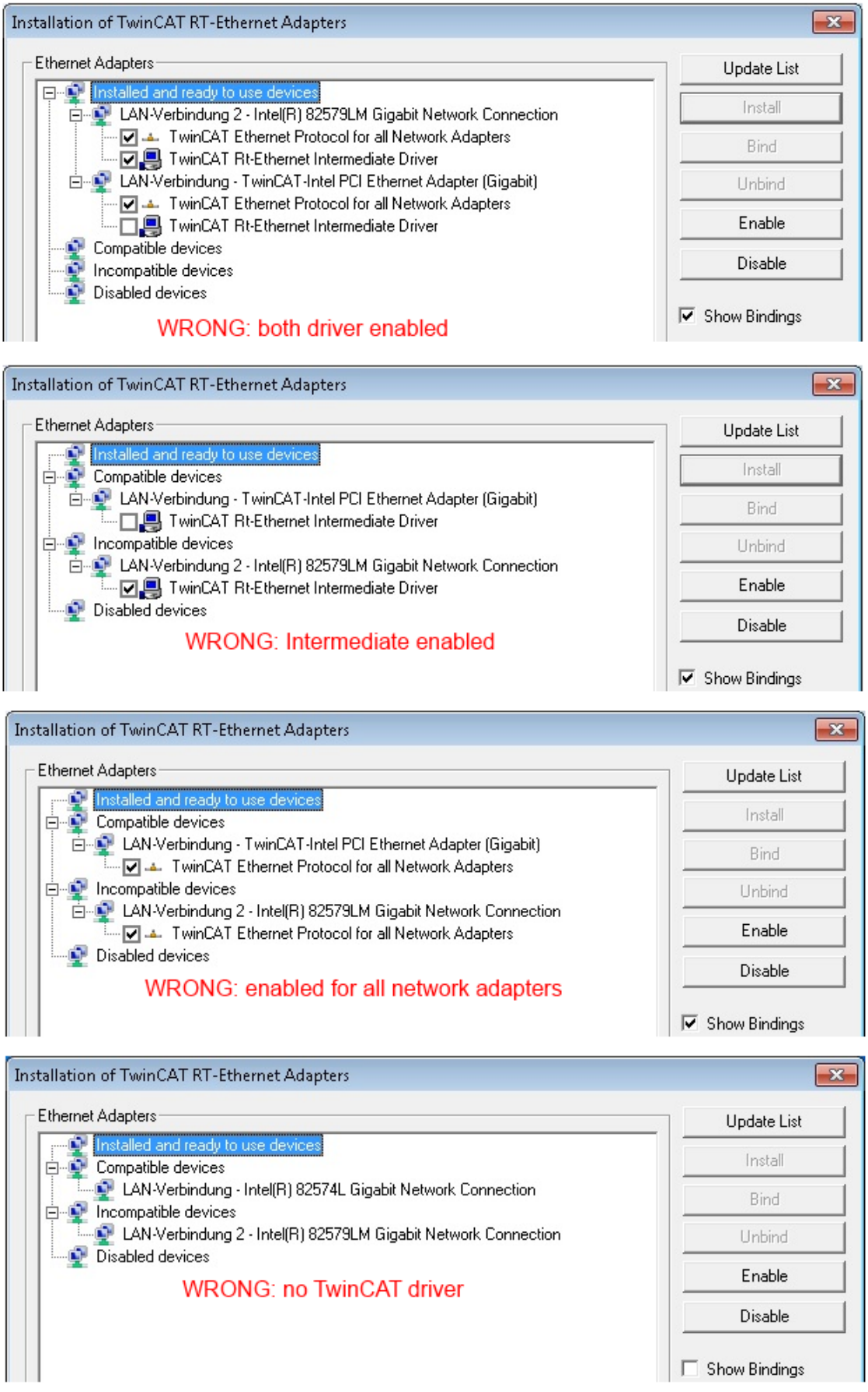


Fig. 85: Incorrect driver settings for the Ethernet port

IP address of the port used

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.

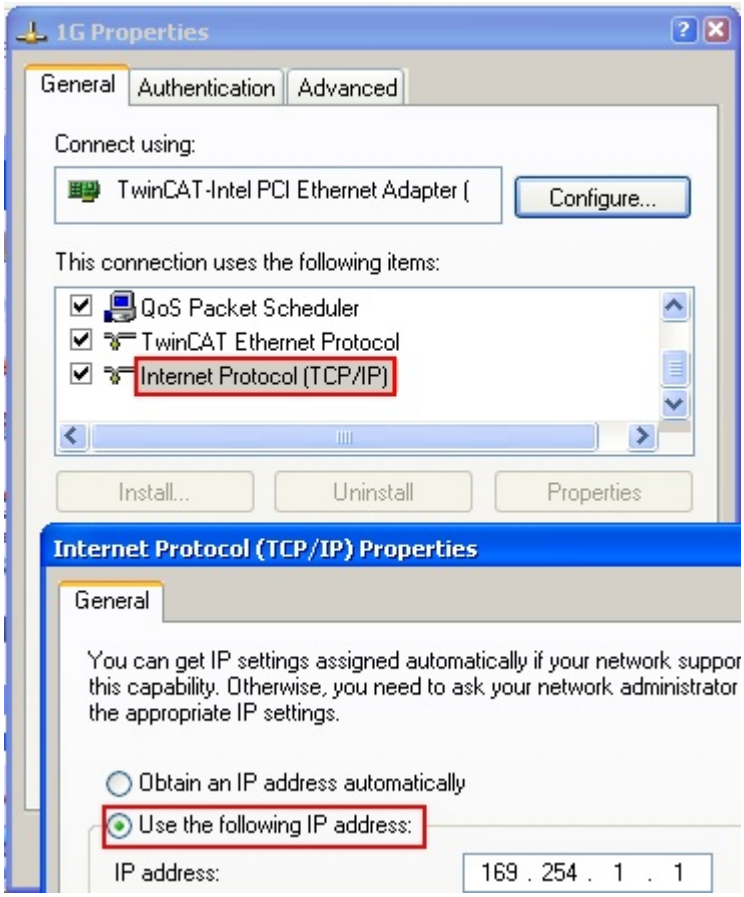


Fig. 86: TCP/IP setting for the Ethernet port

5.2.2 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 [▶ 100] is available for this purpose.

ESI

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

Device differentiation

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

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

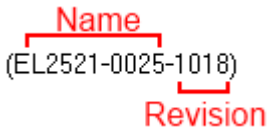


Fig. 87: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) 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 [▶ 13].

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.

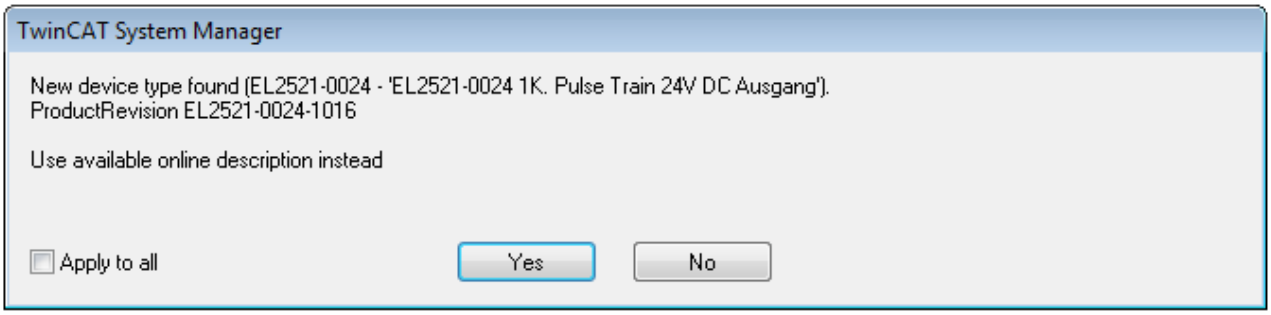


Fig. 88: OnlineDescription information window (TwinCAT 2)

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

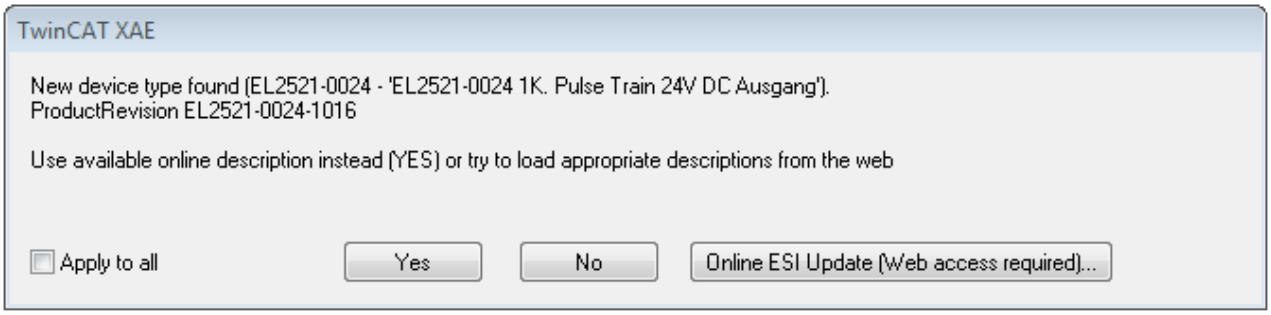


Fig. 89: Information window OnlineDescription (TwinCAT 3)

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.

NOTICE

Changing the "usual" configuration through a scan

✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019

a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).

b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 102]".

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. 90: File OnlineDescription.xml created by the System Manager

Is 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)*.

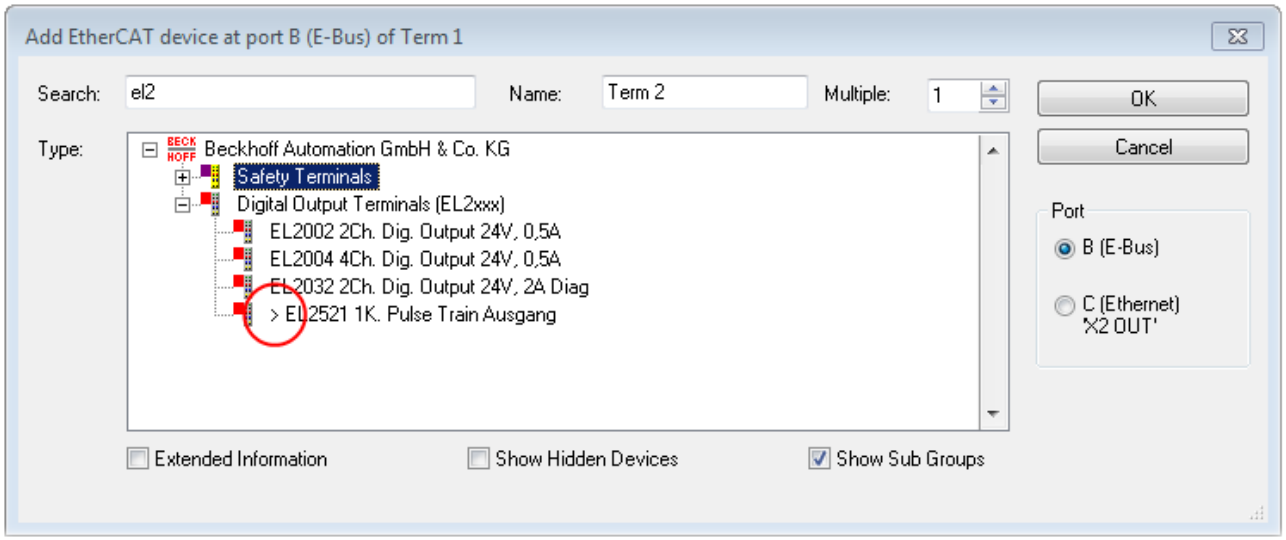


Fig. 91: Indication of an online recorded ESI of EL2521 as an example

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

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.

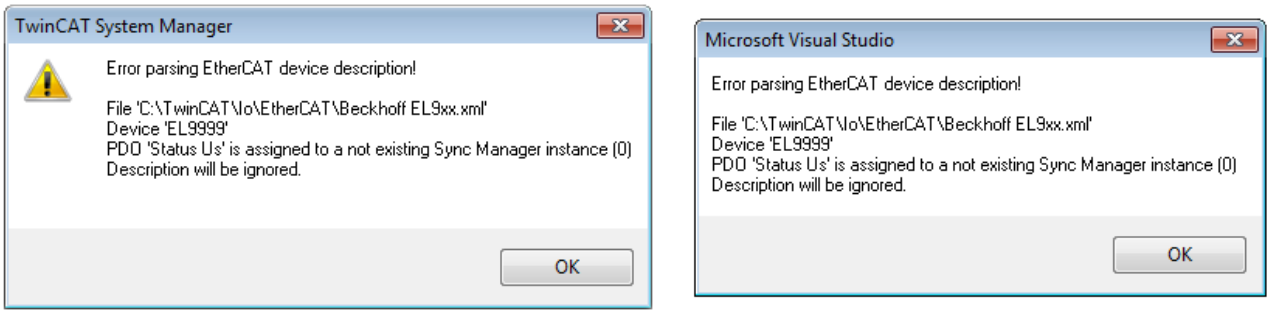


Fig. 92: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.2.3 TwinCAT ESI Updater

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

5.2.3.1 TwinCAT 3

Step 1) Updating the ESI data sets

The Updater can be accessed under:
“TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)...”

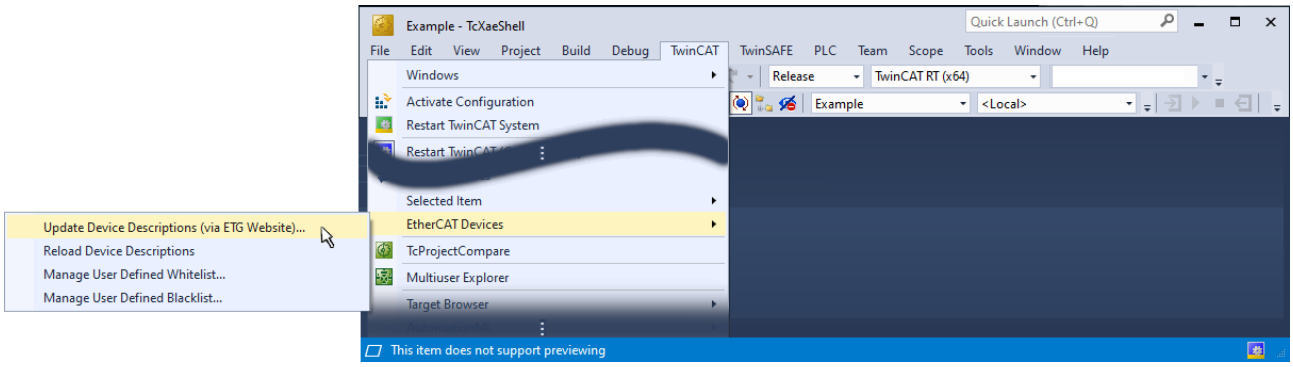
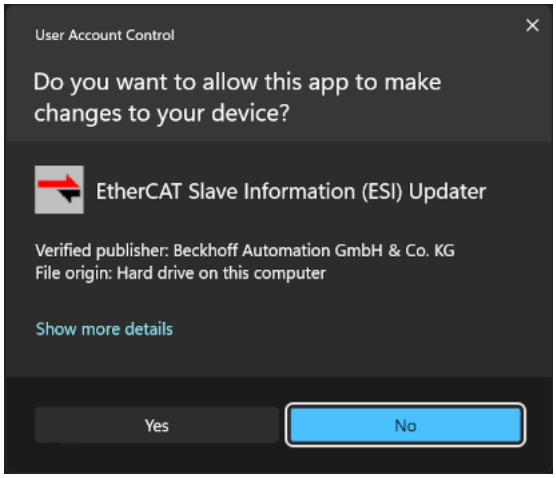


Fig. 93: Calling the ESI Updater (TwinCAT 3)

A prompt for Windows user account control appears after the call, which must be confirmed with [Yes]:



The "Updater" window then appears, showing the configuration data source:

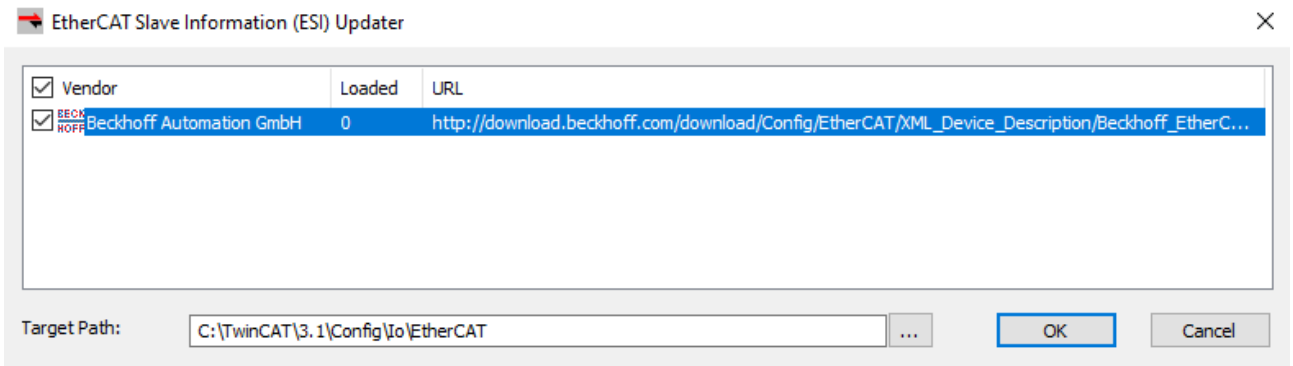
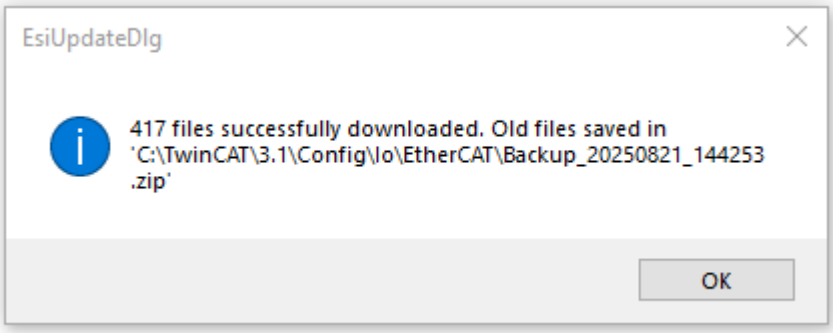


Fig. 94: (ESI) Updater

Select the desired line and click [OK] to start the download or update process. This may take a few minutes.

The completion of the download process is indicated by the message



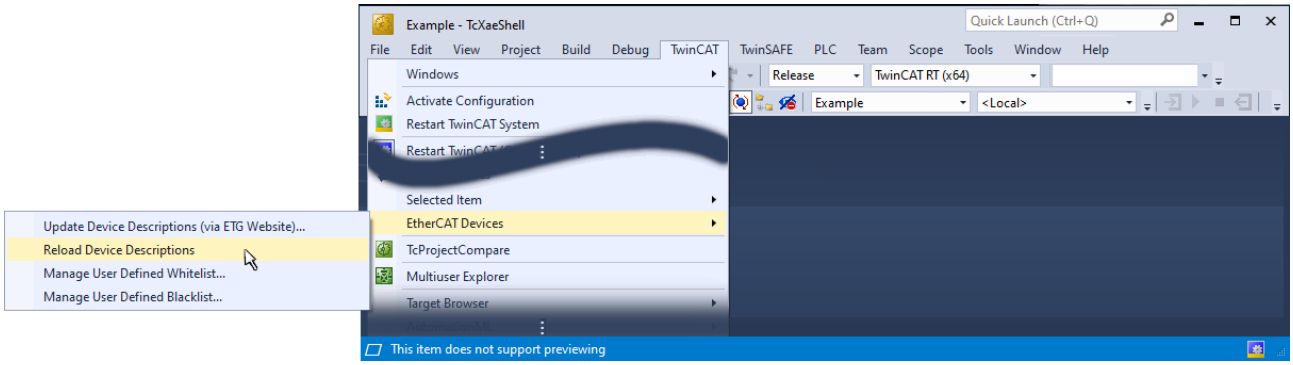
, which shows the number of downloaded files and the storage location or the backup zip folder. Click [OK] to close the window.

Note: Manually added ESI files are not moved to the backup zip folder and remain accessible in TwinCAT. Only the ESI files supplied by Beckhoff are managed automatically.

The (ESI) Updater window that is still open can now be closed with [X] (top right) or [Cancel].

Step 2) Update ESI cache

The ESI files must now be loaded into the application cache in order to be able to use them. This is done either by restarting TwinCAT or by calling up
"TwinCAT" → "EtherCAT Devices" → "Reload Device Descriptions":



The status of the cache rebuild can be seen in the lower status bar in TwinCAT.

5.2.3.2 TwinCAT 2

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

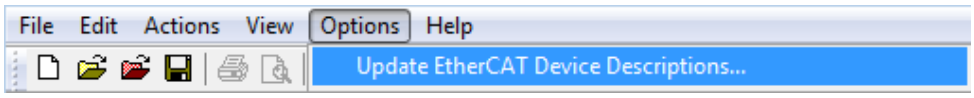


Fig. 95: Using the ESI Updater (>= TwinCAT 2.11)

The call is made via:
“Options” → “Update EtherCAT Device Descriptions”.

5.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 96].

For preparation of a configuration:

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed
- the devices/modules must be connected via EtherCAT cables or in the terminal/ module strand in the same way as they are intended to be used later
- the devices/modules be connected to the power supply and ready for communication
- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 108] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 109]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 112]

The scan with existing configuration [▶ 113] can also be carried out for comparison.

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

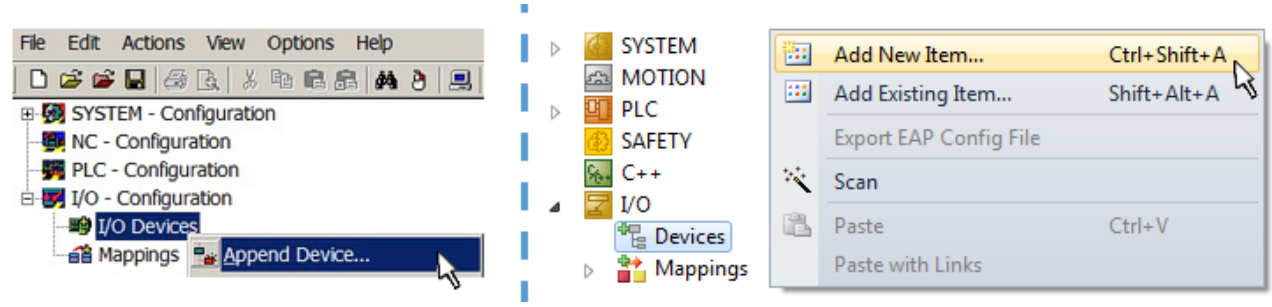


Fig. 96: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

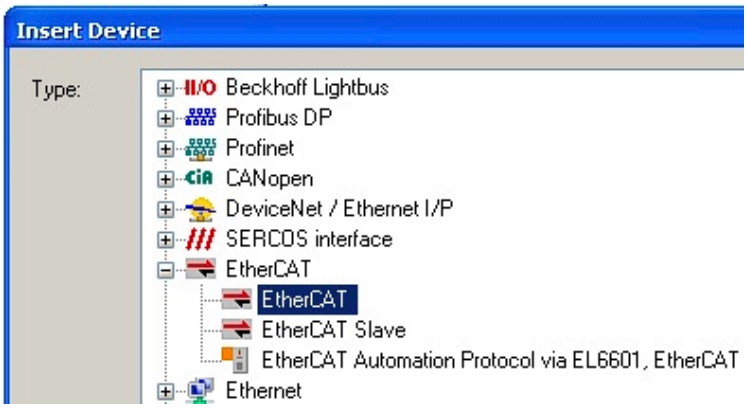


Fig. 97: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

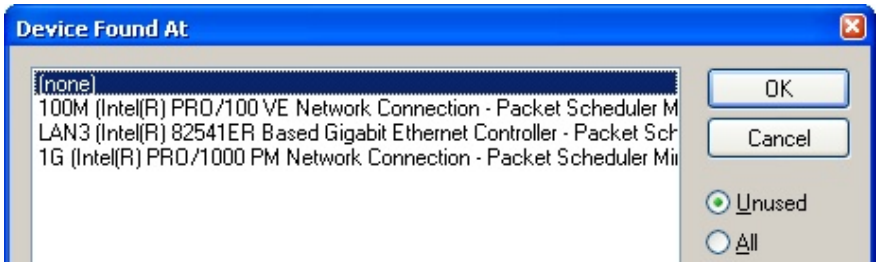


Fig. 98: Selecting the Ethernet port

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)”.

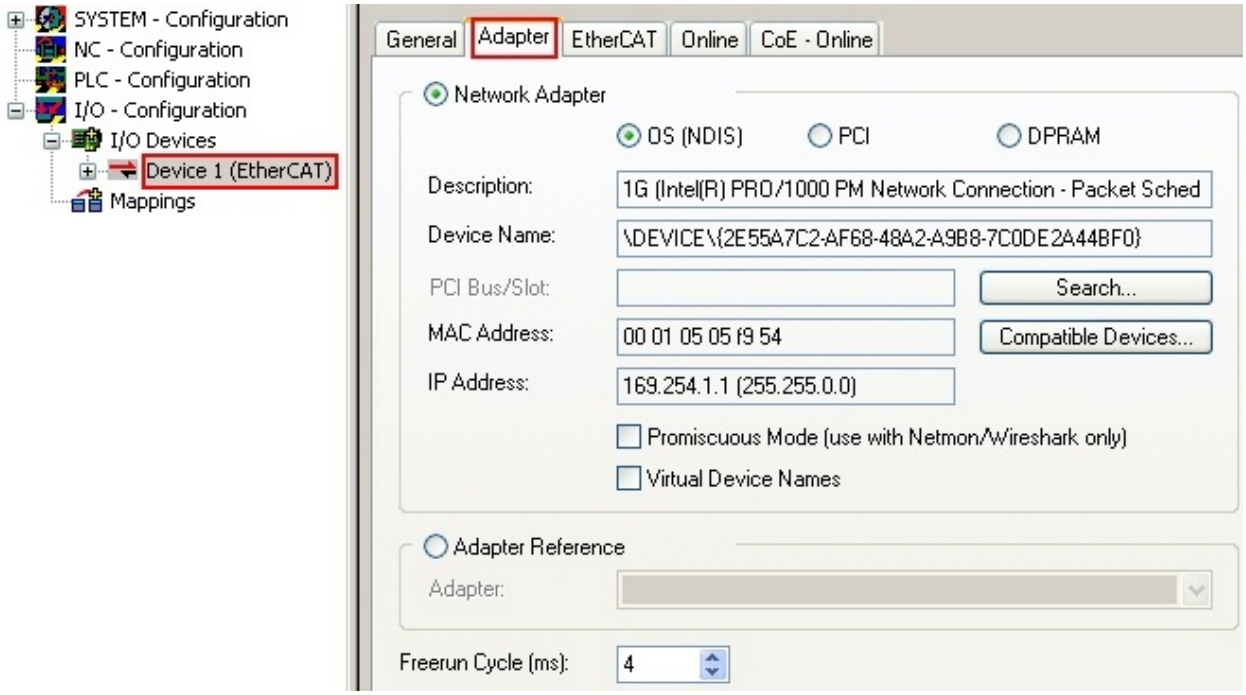
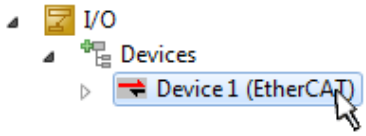


Fig. 99: EtherCAT device properties (TwinCAT 2)

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



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 [▶ 90].

Defining EtherCAT slaves

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

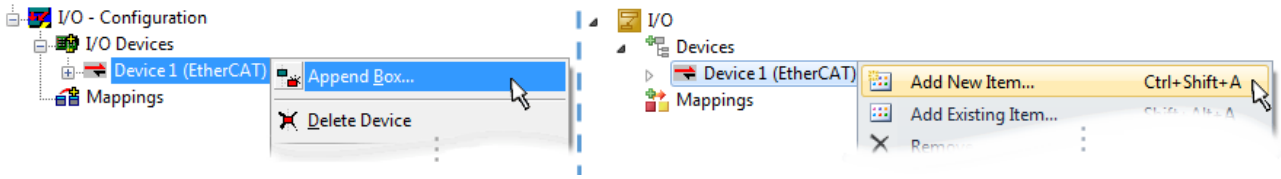


Fig. 100: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector
- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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

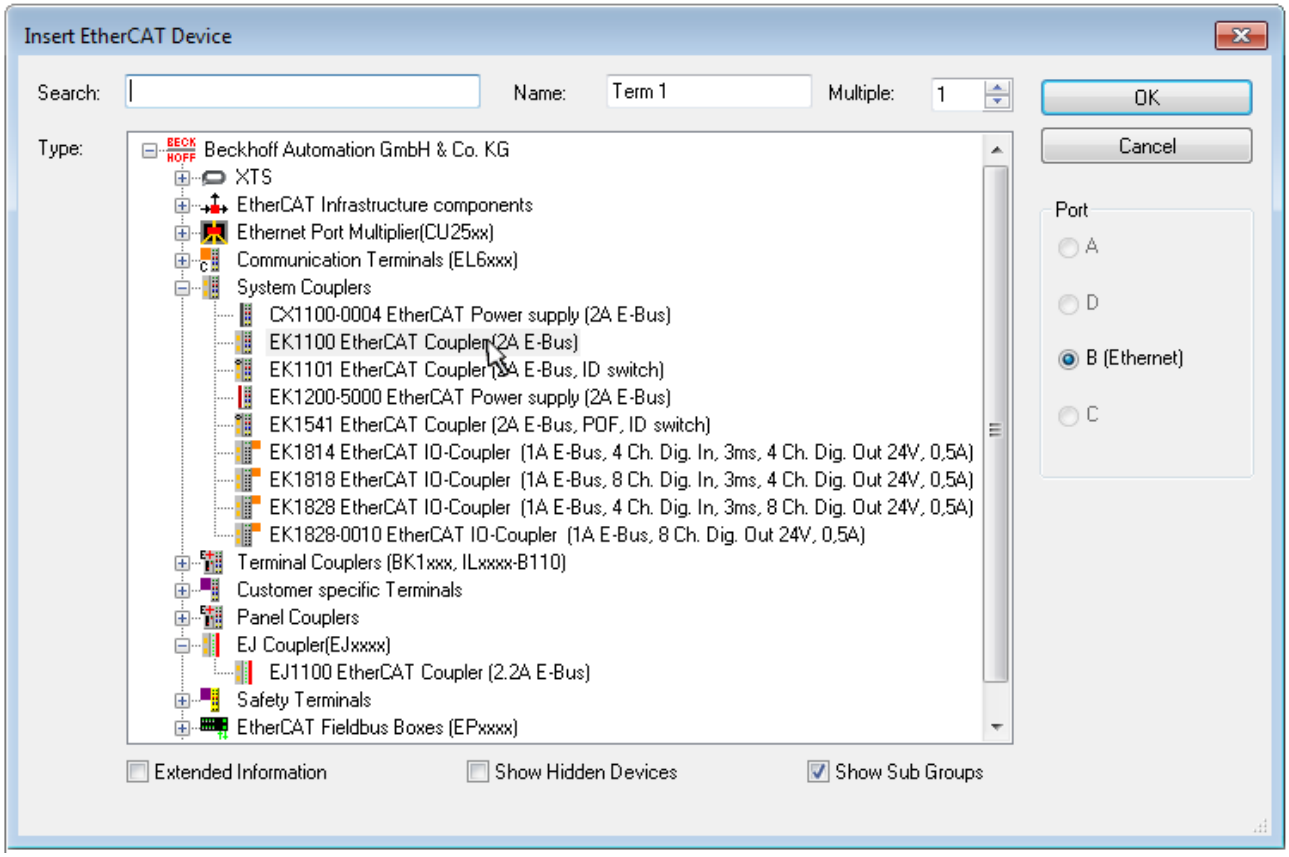


Fig. 101: Selection dialog for new EtherCAT device

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”.

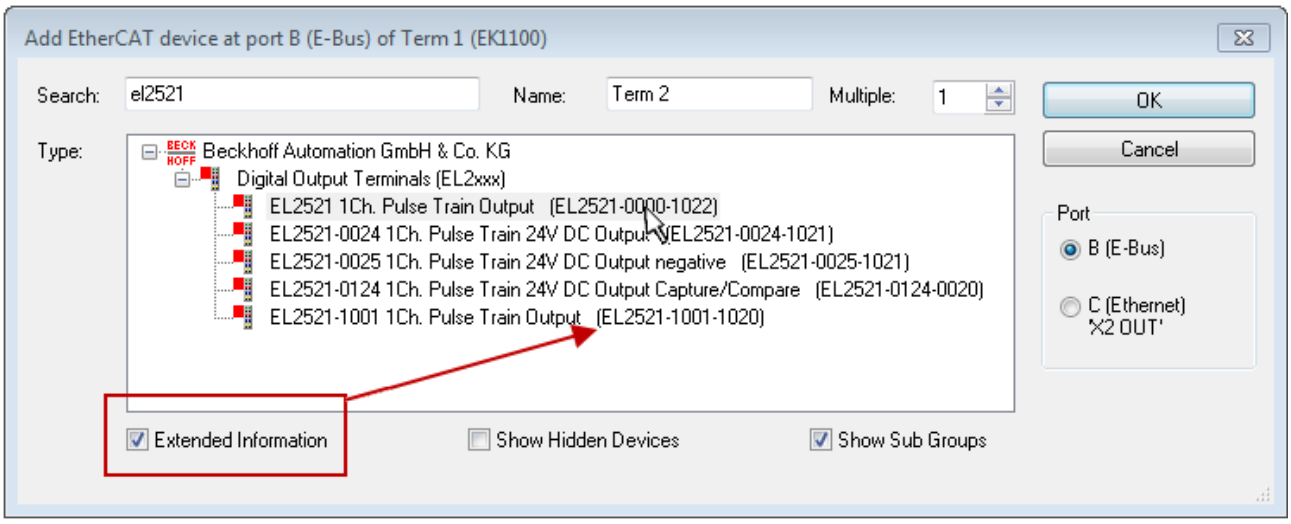


Fig. 102: Display of device revision

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”.

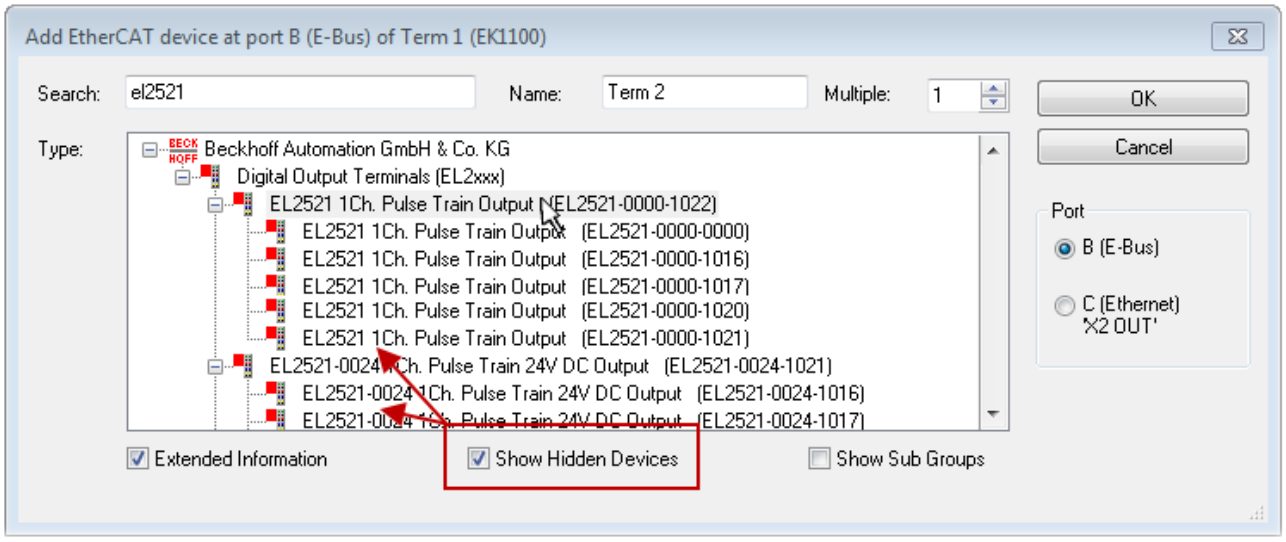


Fig. 103: Display of previous revisions

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.

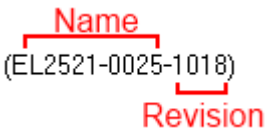


Fig. 104: Name/revision of the terminal

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

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

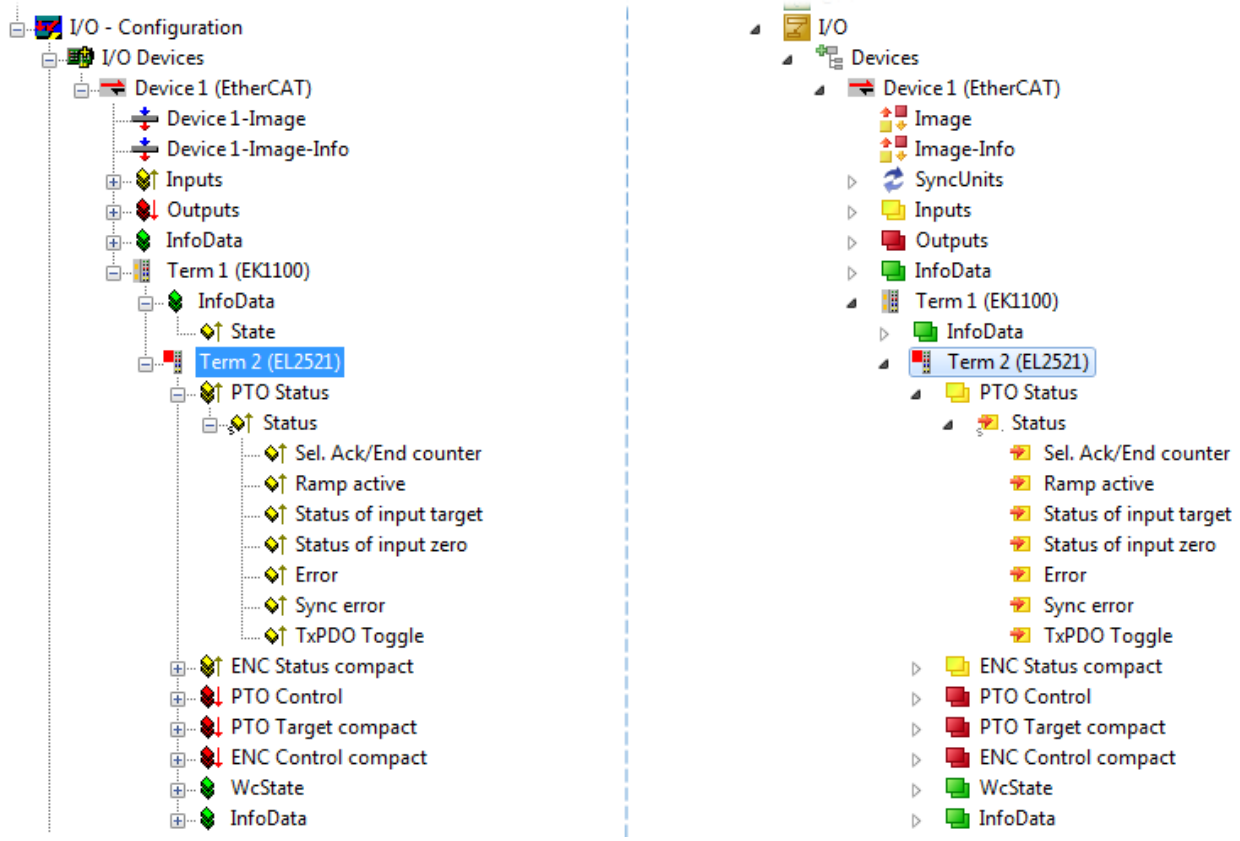


Fig. 105: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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.

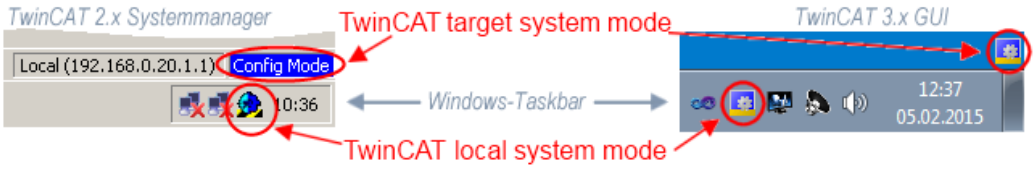


Fig. 106: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

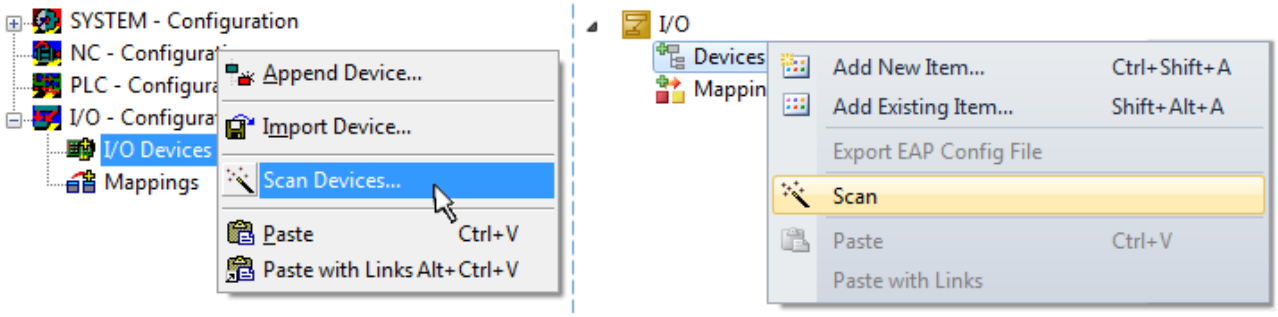


Fig. 107: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

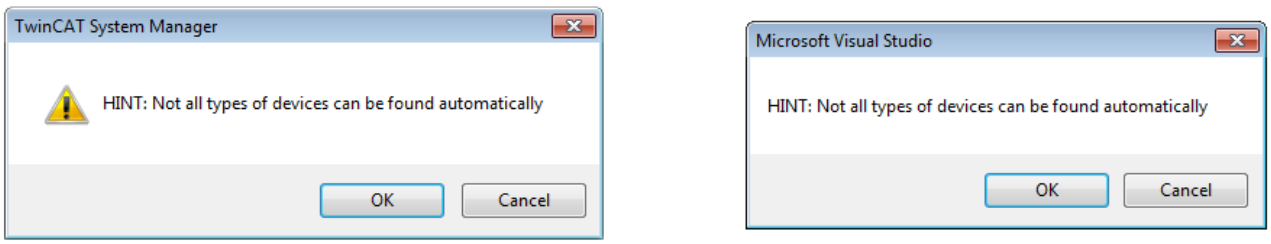


Fig. 108: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

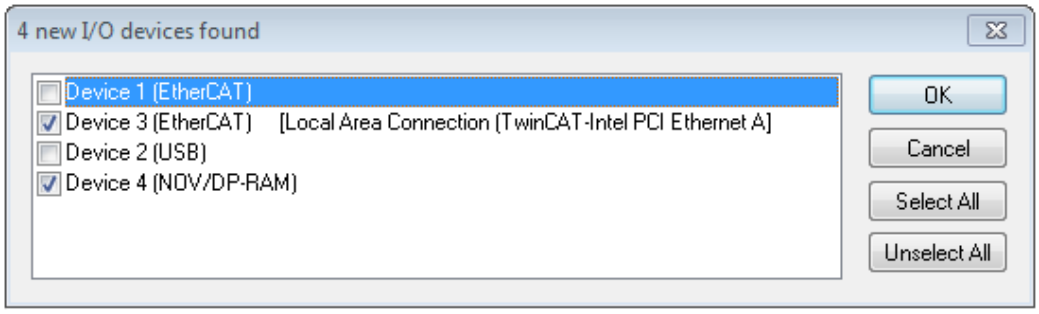


Fig. 109: Detected Ethernet devices

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 [▶ 90].

Detecting/Scanning the EtherCAT devices

Online scan functionality

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

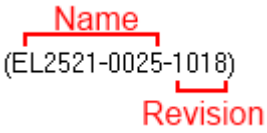


Fig. 110: Example default state

NOTICE

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 [▶ 113] 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:

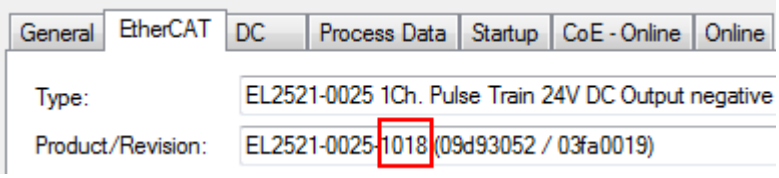


Fig. 111: Installing EthetCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [▶ 113] 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:

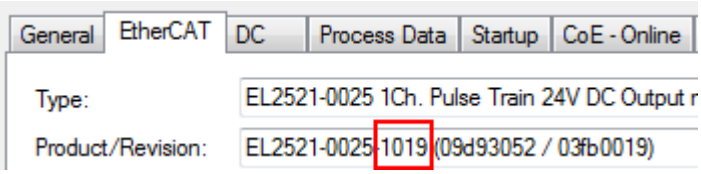


Fig. 112: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since a new configuration is essentially 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. Þ 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. 113: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

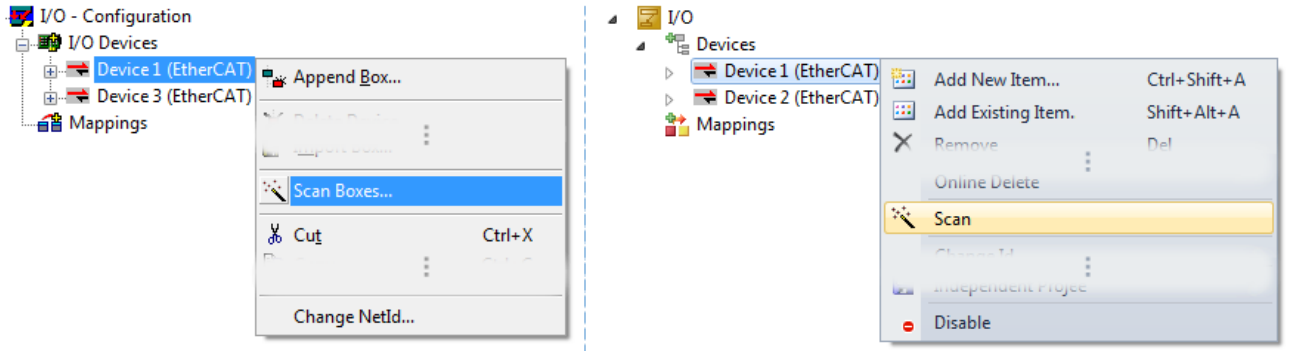


Fig. 114: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

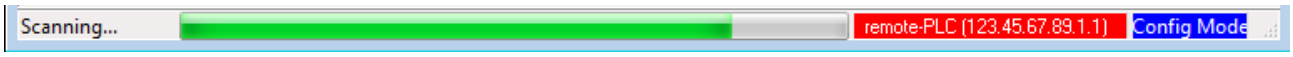


Fig. 115: *Scan progressexemplary by TwinCAT 2*

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



Fig. 116: 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).

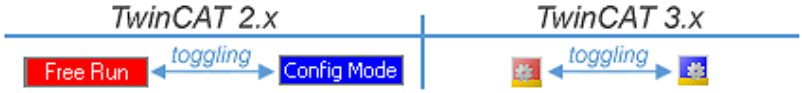


Fig. 117: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 118: 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*.

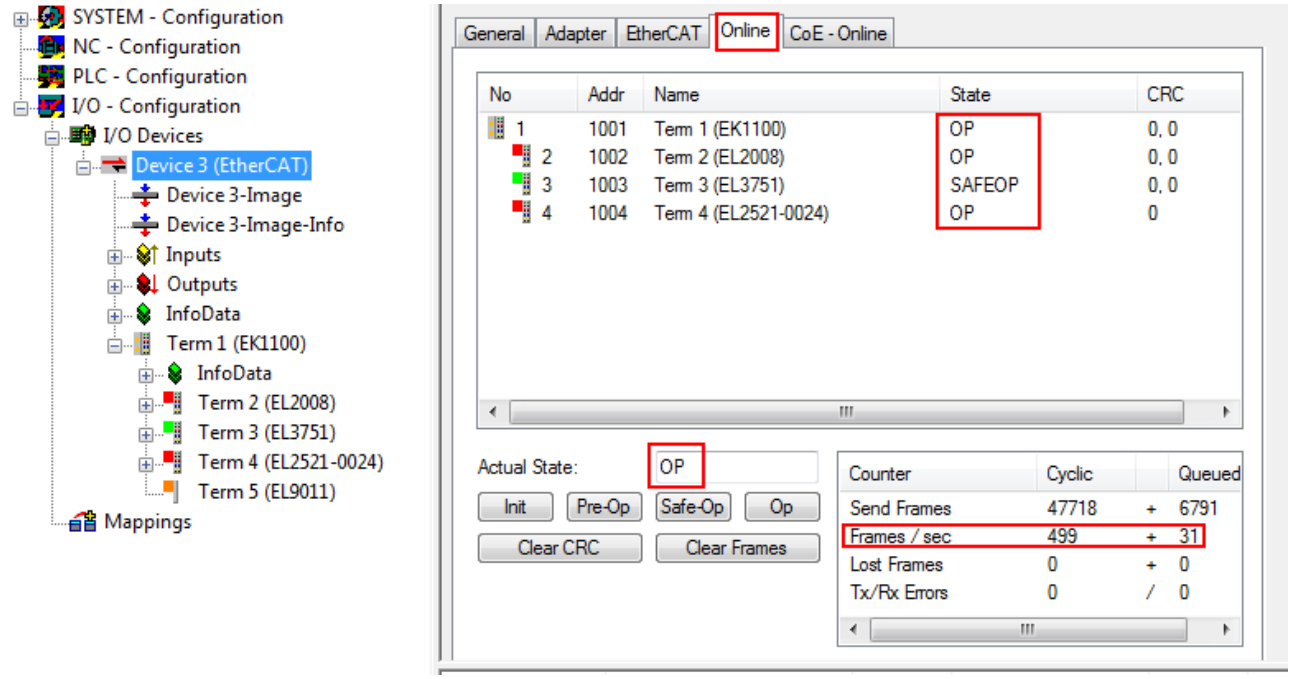


Fig. 119: Online display example

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 [▶ 102].

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.

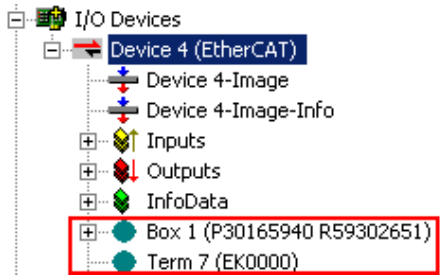


Fig. 120: Faulty identification

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

NOTICE

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. 121: 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.

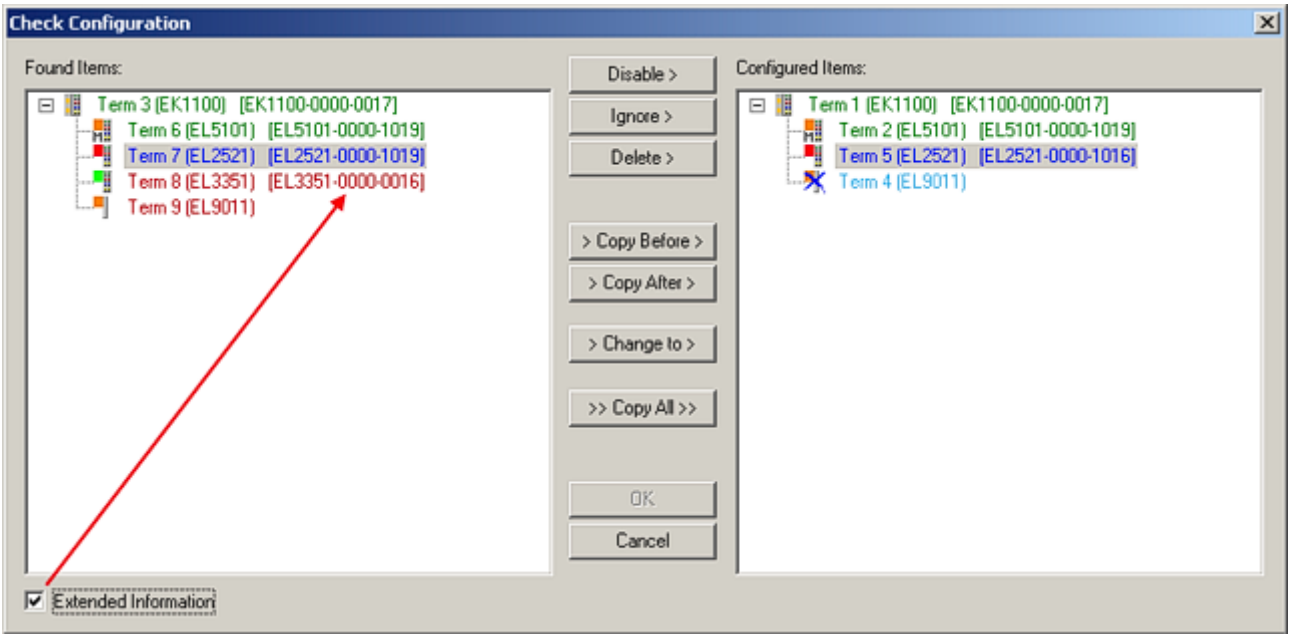


Fig. 122: Correction dialog

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

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored (“Ignore” button)
red	• 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

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.

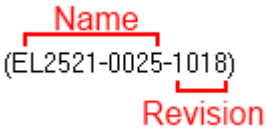


Fig. 123: Name/revision of the terminal

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

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

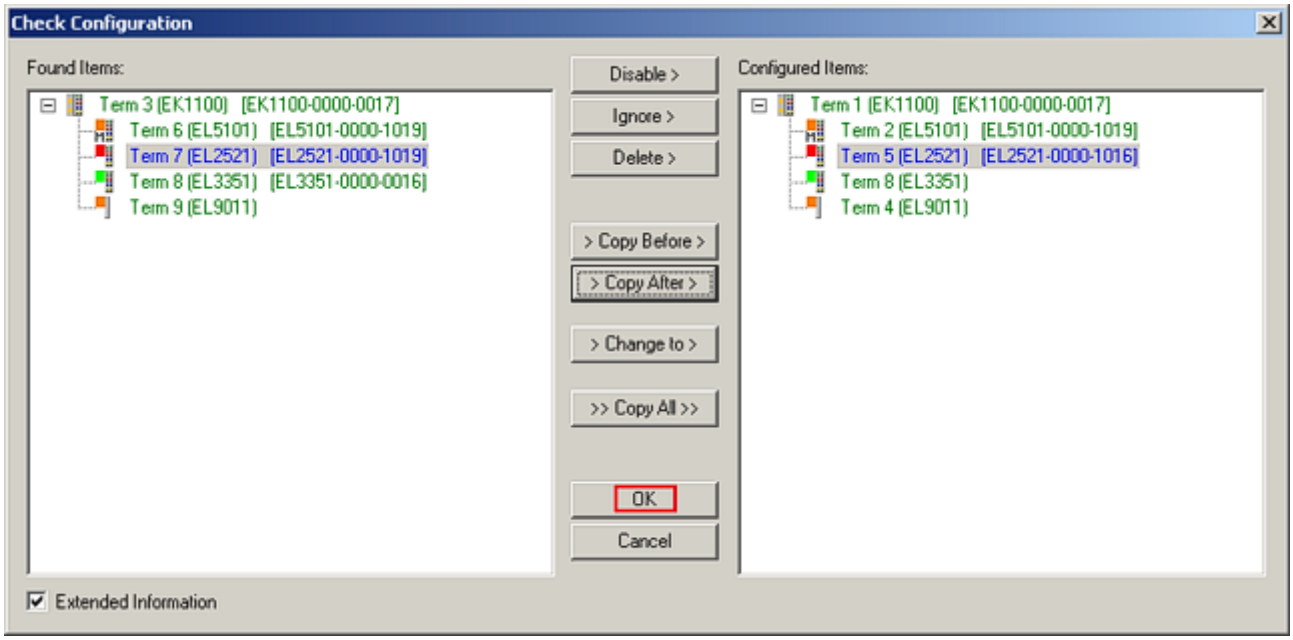


Fig. 124: Correction dialog with modifications

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.

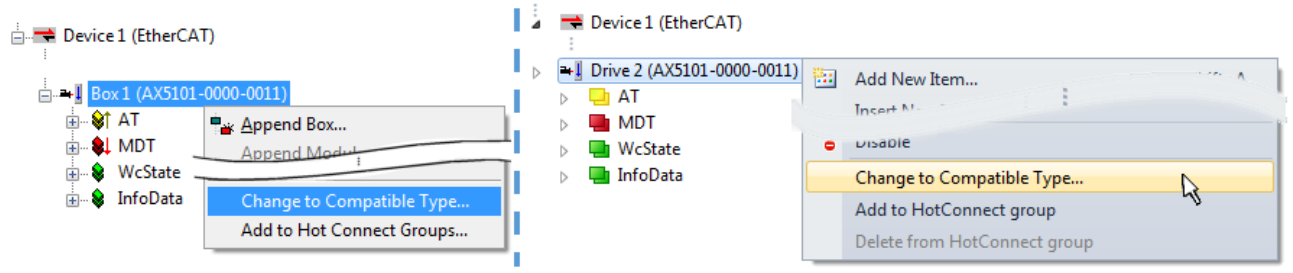


Fig. 125: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

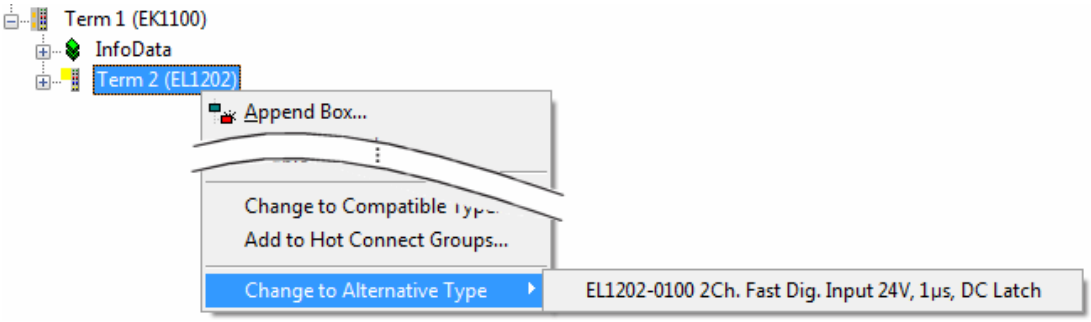


Fig. 126: TwinCAT 2 Dialog Change to Alternative Type

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).

5.2.7 EtherCAT subscriber configuration

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

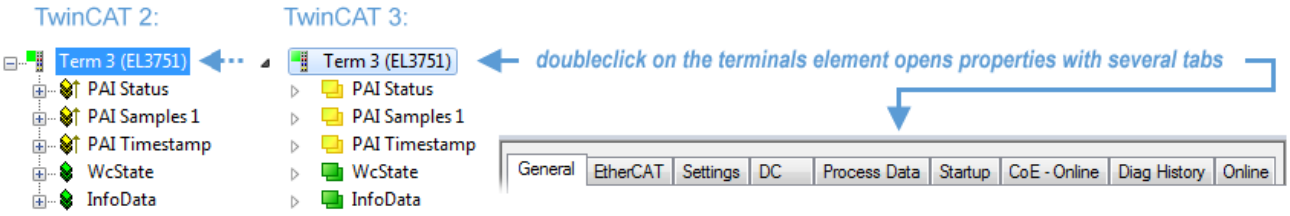


Fig. 127: Branch element as terminal EL3751

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

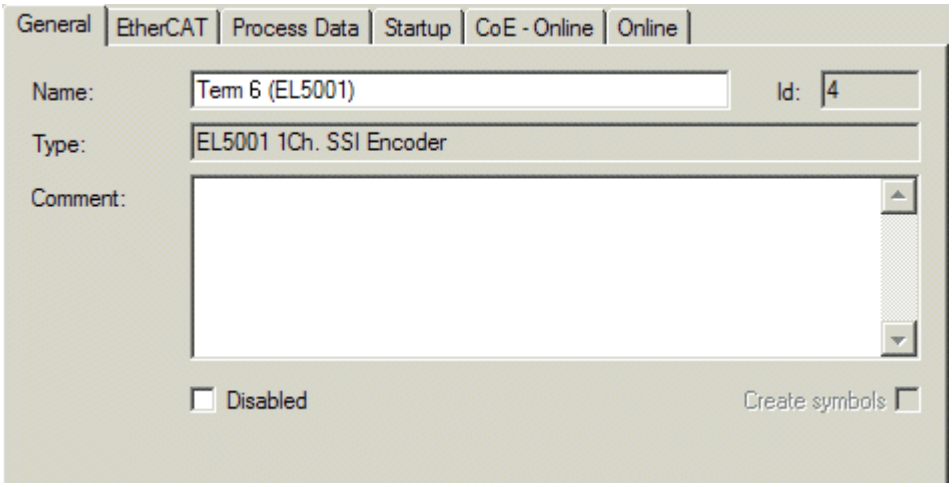


Fig. 128: “General” tab

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

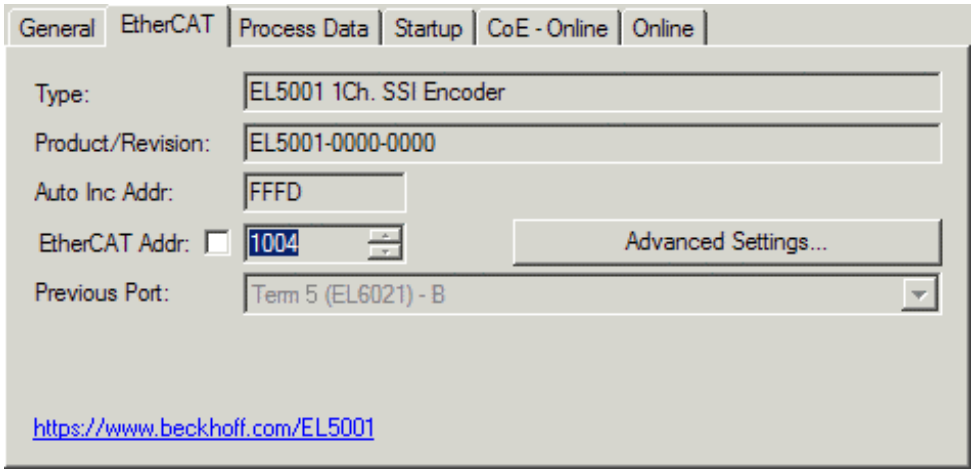


Fig. 129: “EtherCAT” tab

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 (**P**rocess **D**ata **O**bjects, 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.

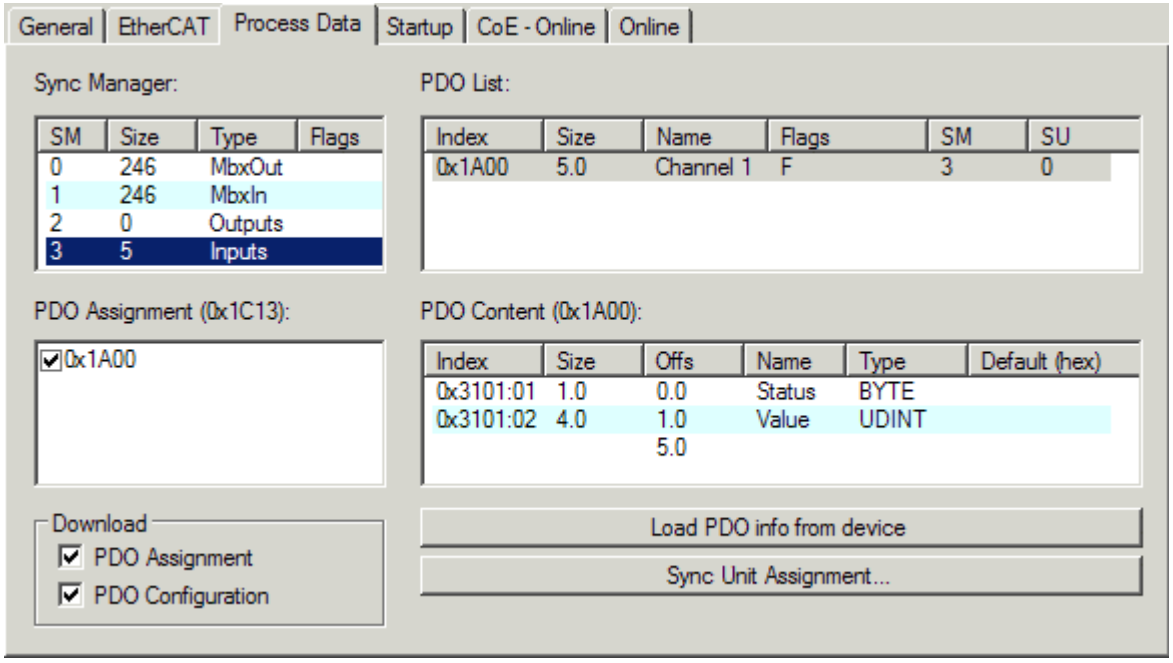


Fig. 130: “Process Data” tab

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 Sync Manager (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”).

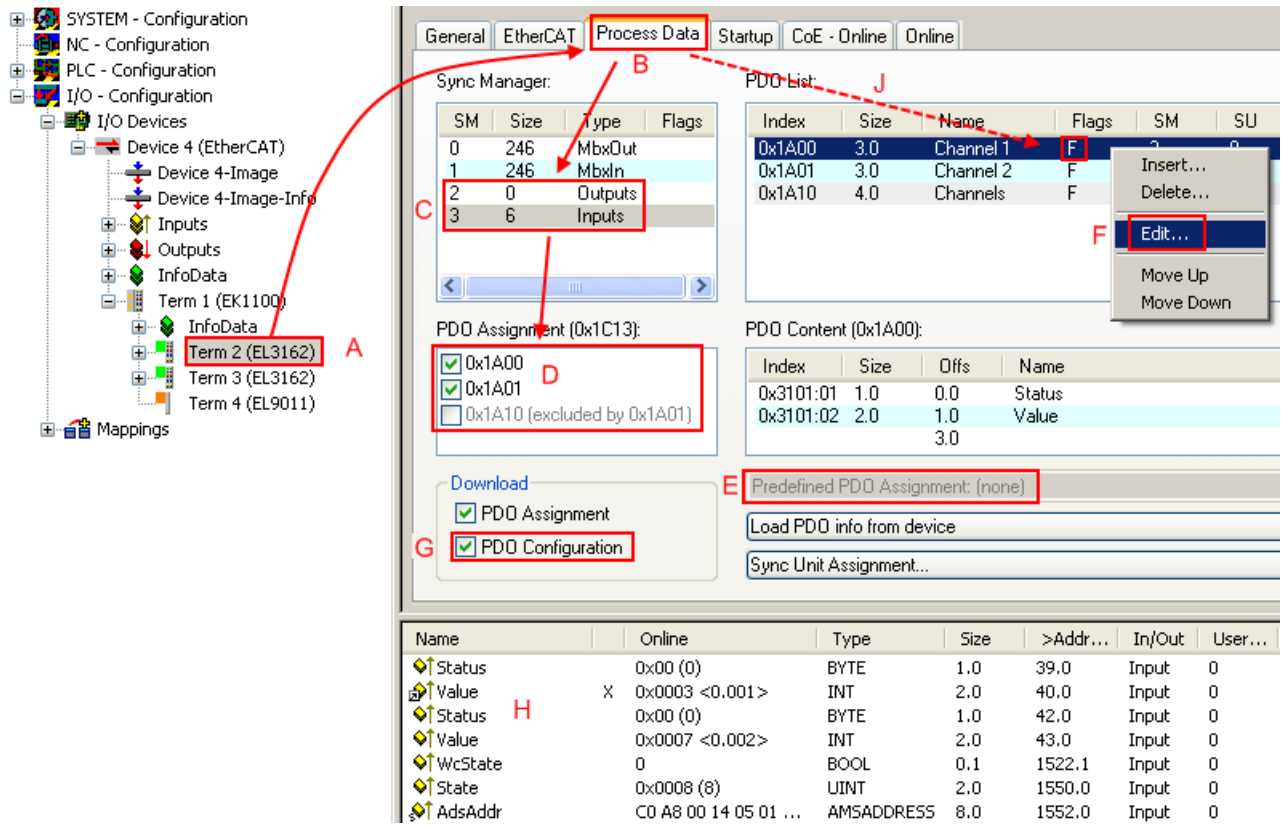


Fig. 131: Configuring the process data

Manual modification of the process data

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

A detailed description [▶ 124] 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.

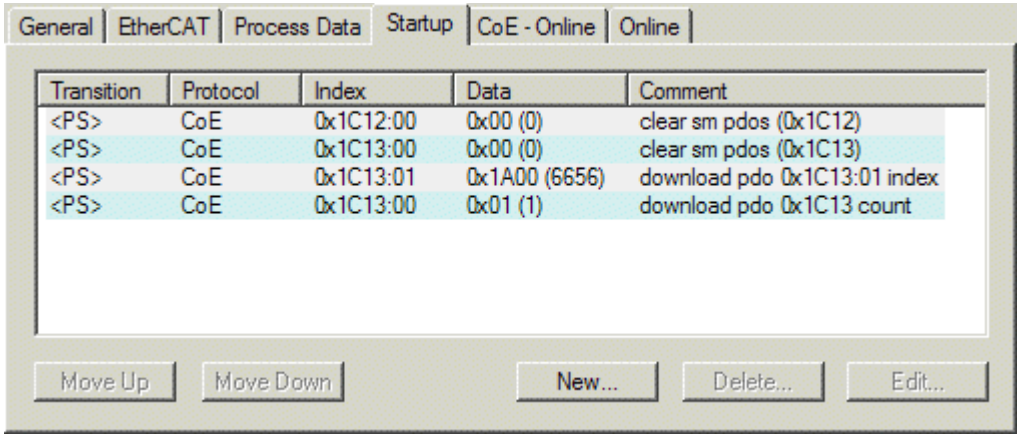


Fig. 132: “Startup” tab

Column	Description
Transition	Transition to which the request is sent. This can either be • 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.

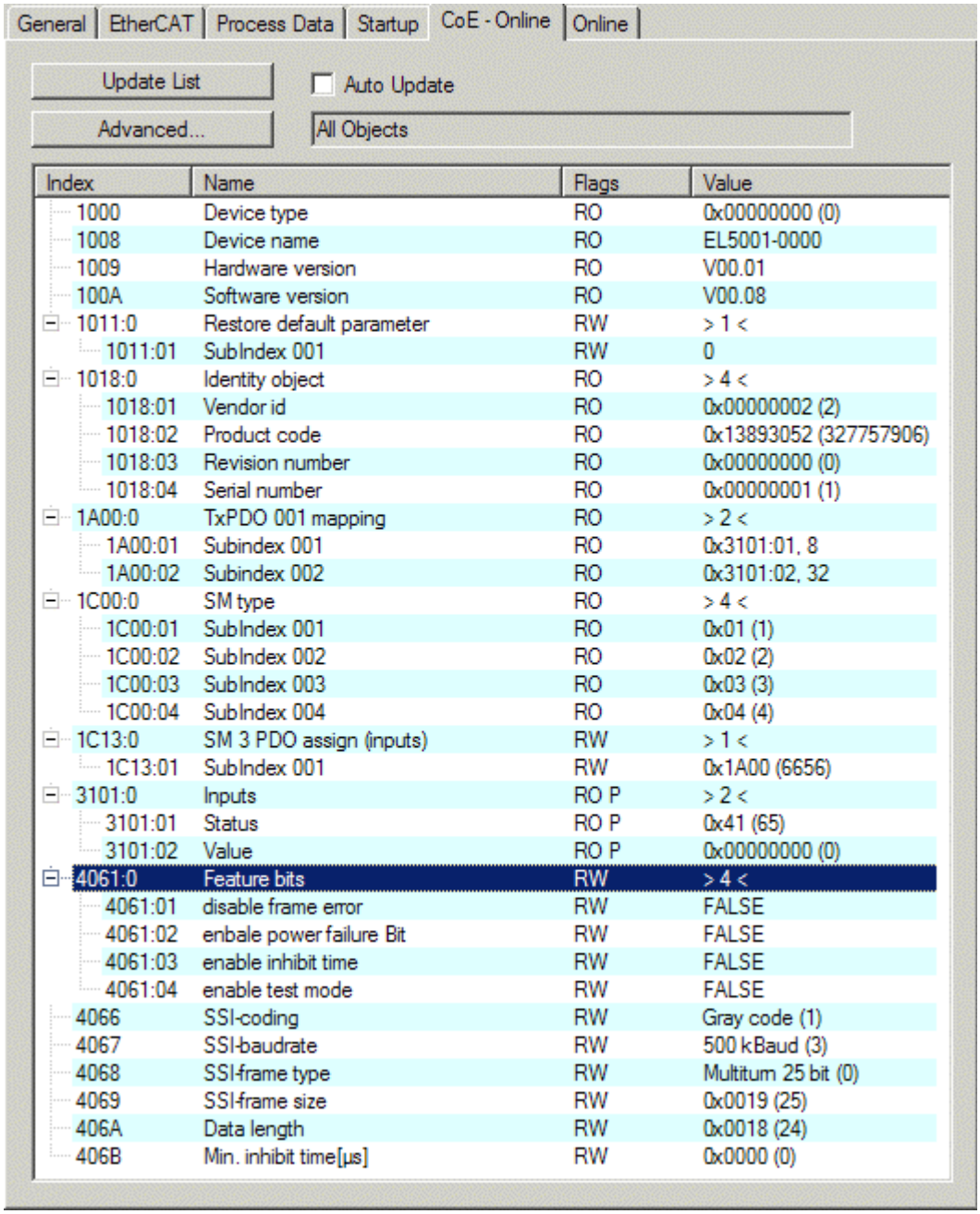


Fig. 133: “CoE - Online” tab

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.

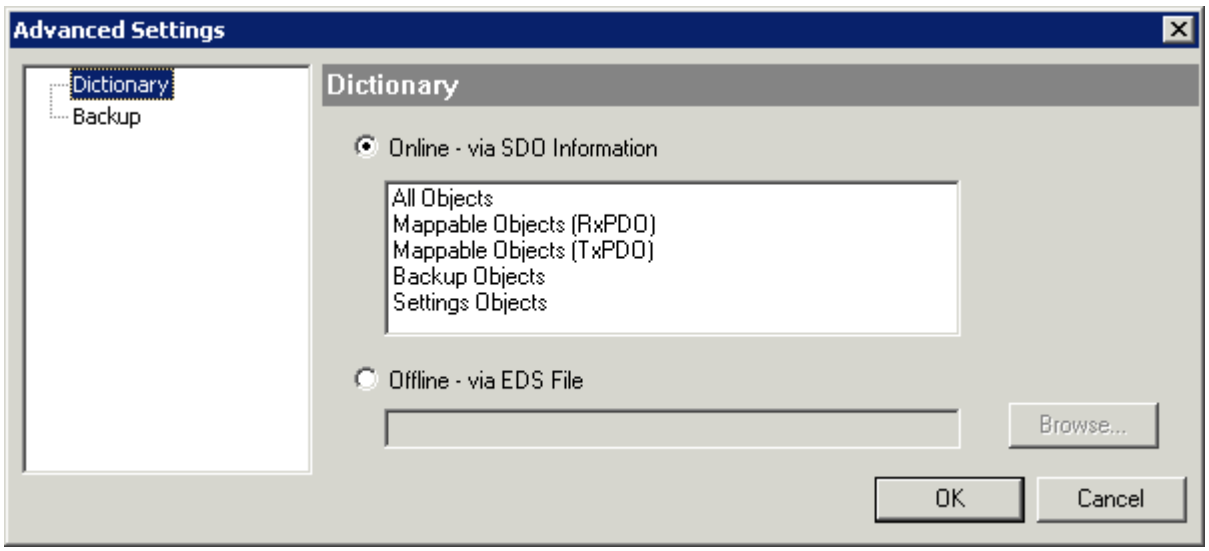


Fig. 134: Dialog “Advanced settings”

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

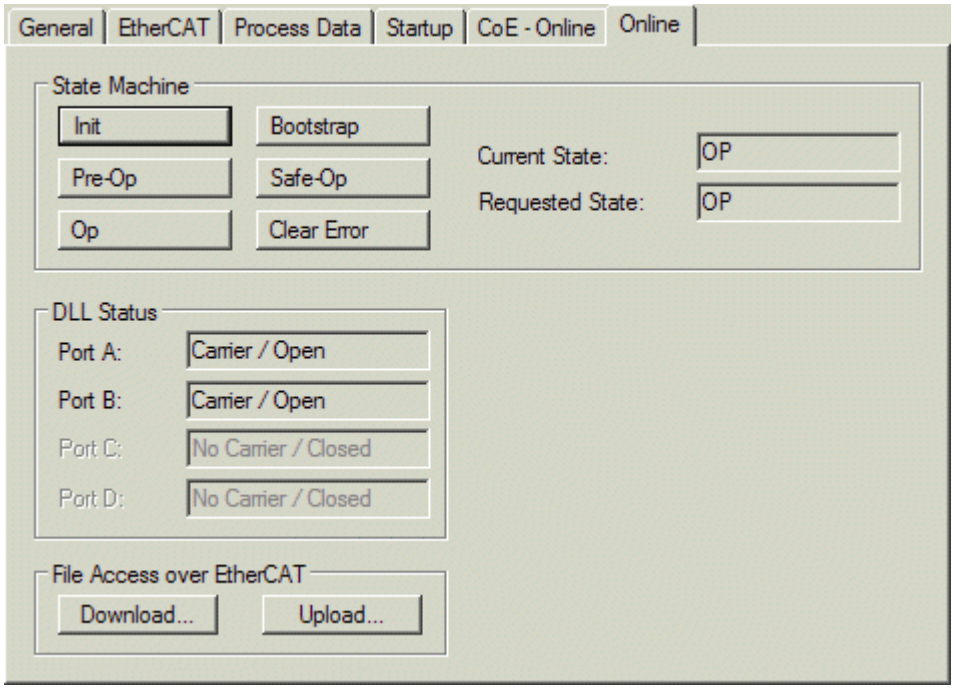


Fig. 135: “Online” tab

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)

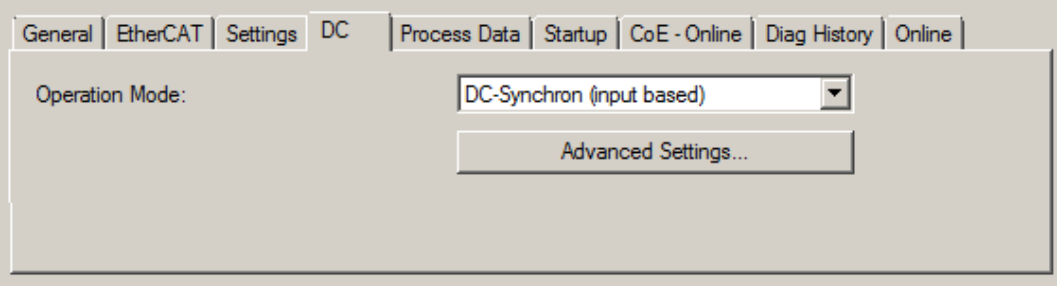


Fig. 136: “DC” tab (Distributed Clocks)

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

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

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

5.2.7.1 Download revision

Download revision in Start-up list

Several terminals / modules generate the entry from object 0xF081:01 in the Start-up list automatically (see fig. “Download revision in Start-up list“).

The object 0xF081:01 (Download revision) describes the revision of the terminal / module, e.g. 0x00**18**00*0A* for EL7201-00*10*-00**24**, and is necessary to ensure compatibility.

Please note, that you must not delete this entry from the Start-up list!

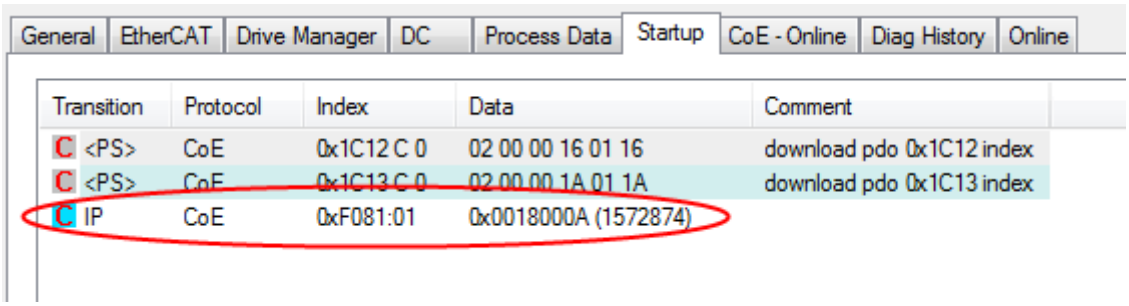


Fig. 137: Download revision in Start-up list

5.2.7.2 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.

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 [▶ 122]),

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 *PDO Assignment* 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 [▶ 119] 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.

5.3 Start-up and parameter configuration

5.3.1 Integration into the NC configuration

(Master: TwinCAT 2.11 R3)

Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be downloaded from the Beckhoff Website and installed according to the installation instructions.

Integration into the NC can be accomplished as follows:

- The terminal must already have been added manually under I/O devices or have been scanned in by the system (see section "Configuration set-up in TwinCAT [▶ 90]").

Adding an axis automatically

- Once the terminals have been scanned successfully, TwinCAT detects the new axes automatically. The user is asked whether the detected axes should be added automatically (see Fig. *Axis detected*). If this is confirmed, all axes are automatically liked to the NC.

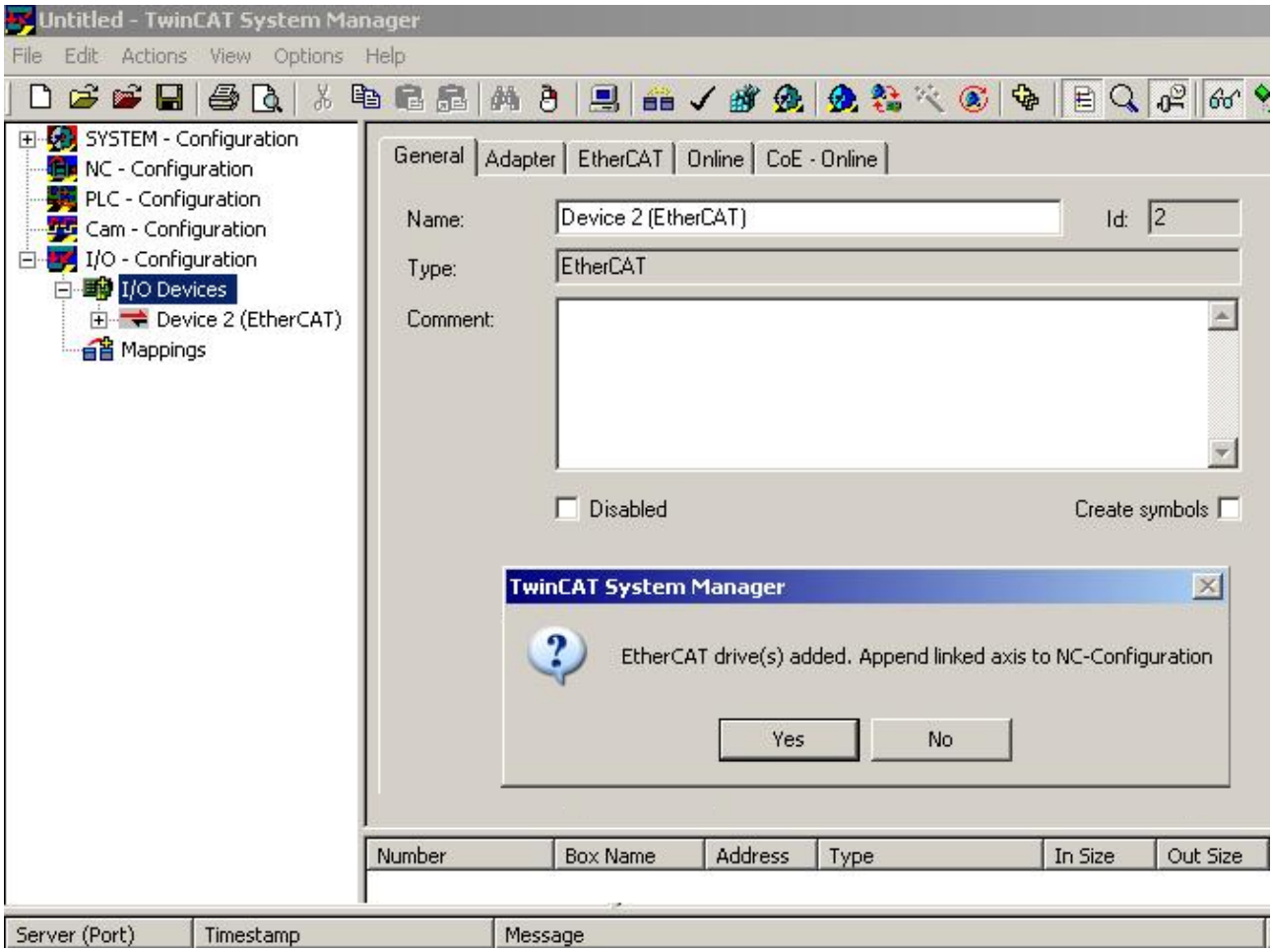


Fig. 138: Axis detected

- Several parameters have to be set before the motor can be started up. The values can be found in section "Configuration of the main parameters [▶ 135]".
  Please set these parameters before continuing with the motor commissioning procedure.

Adding an axis manually

- First add a new task. Right-click on NC configuration and select "Append Task..." (see Fig. *Adding a new task*).
- Rename the task if required and confirm with OK.

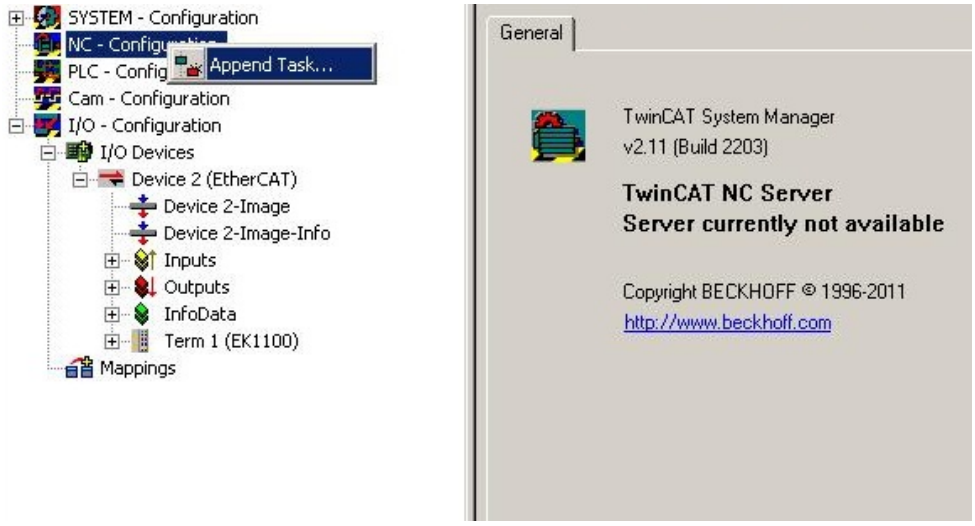


Fig. 139: Adding a new task

- Right-click on *Axes*, then add a new axis (see Fig. *Adding a new axis*).

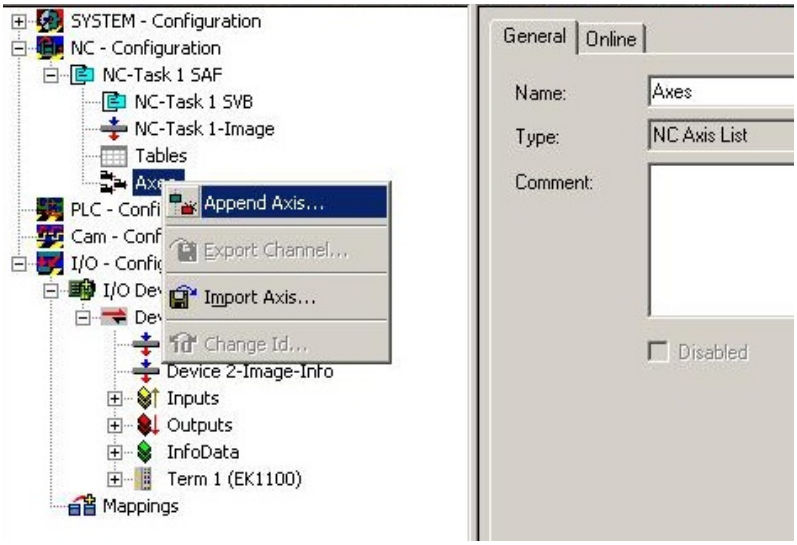


Fig. 140: Adding a new axis

- Select Continuous Axis type and confirm with OK (see Fig. *Selecting and confirming the axis type*).

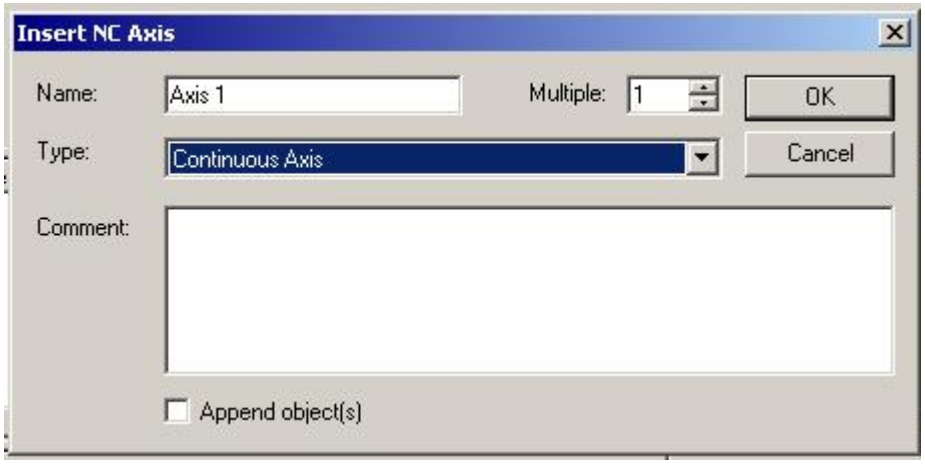


Fig. 141: Selecting and confirming the axis type

- Left-click your axis to select it. Under the S*ettings* tab select "Link To..." (see Fig. *Linking the axis with the terminal*).

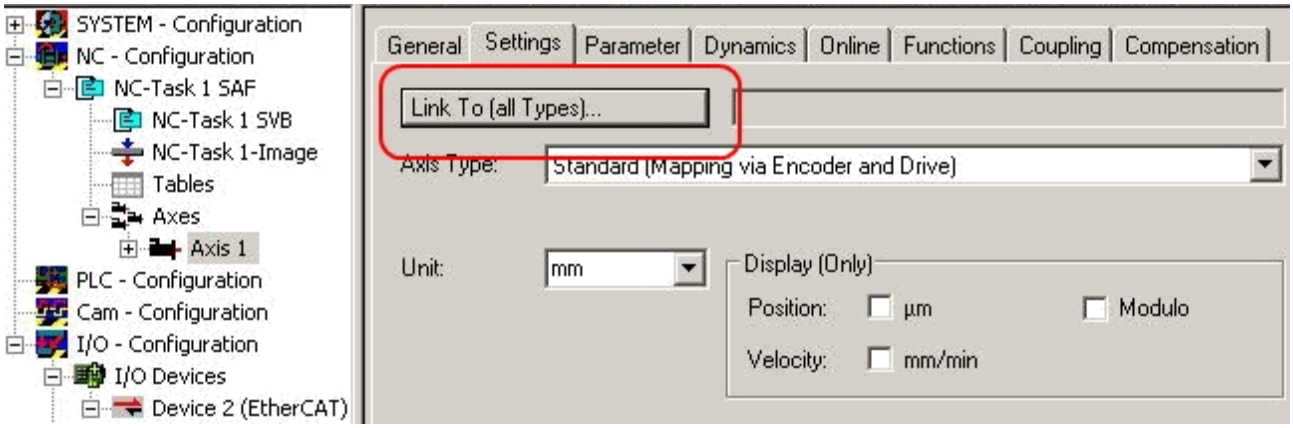


Fig. 142: Linking the axis with the terminal

- Select the required terminal (CANopen DS402, EtherCAT CoE) and confirm with OK.

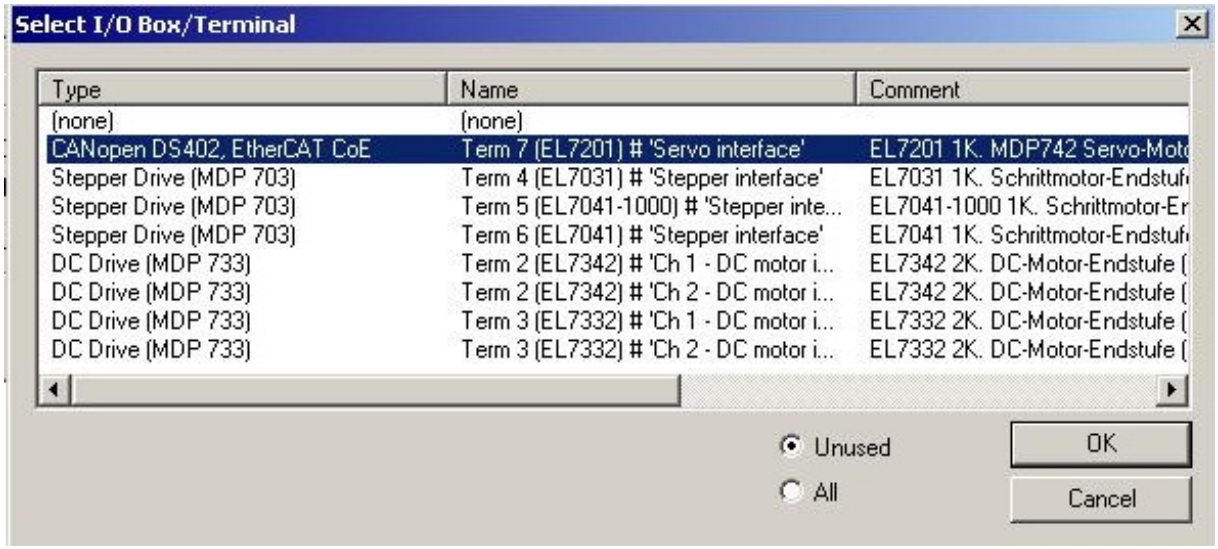


Fig. 143: Selecting the right terminal

- All main links between the NC configuration and the terminal are set automatically (see Fig. *Automatic linking of all main variables*)

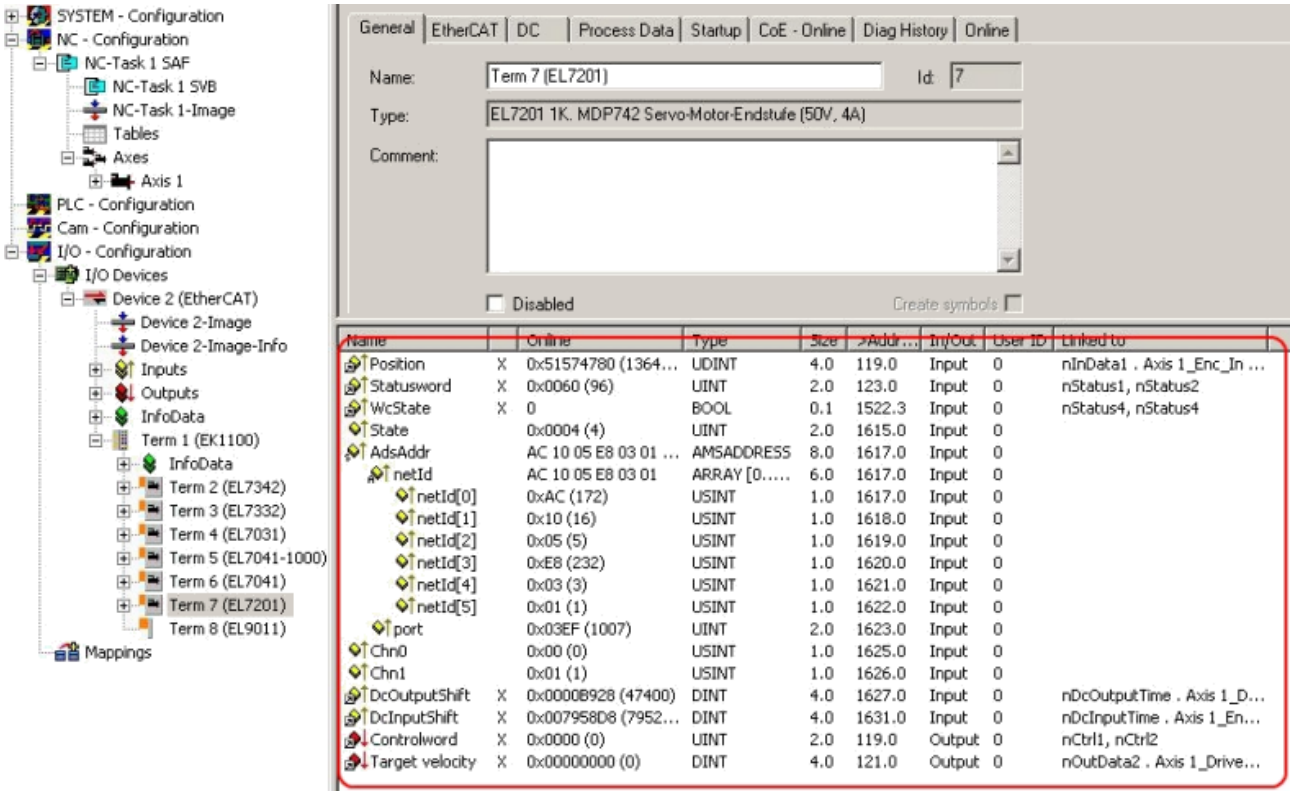


Fig. 144: Automatic linking of all main variables

- Several parameters have to be set before the motor can be started up. The values can be found in sections "CoE settings [▶ 135]" and "NC settings".
  Please set these parameters before continuing with the motor commissioning procedure.

5.3.2 Settings with the Drive Manager

(Master TwinCAT 2.11 R3)

The data given here serve as an example for a servomotor type AM8131-0F20-0000 from Beckhoff Automation. For other motors the values may vary, depending on the application.

Using the Drive Manager from revision -0019

The Drive Manager is only supported from revision -0019 of the EL72x1-xxxx. If you use an older version, the settings have to be made manually. See chapters "CoE settings [▶ 135]" and "NC settings"

Table of contents

- Start-up with the Drive Manager [▶ 130]
- Setting further parameters with the Drive Manager [▶ 134]
  - Integral velocity controller component Tn [▶ 134]
  - Proportional velocity controller component Kp [▶ 134]

The TwinCAT Drive Manager is available for download from the Beckhoff website.

The TwinCAT Drive Manager for parameterizing an EL72x1-xxxx servo terminal is integrated in the System Manager, so that no separate configuration tool is required. Once a servo terminal has been detected or entered, the TwinCAT Drive Manager is available in the Configuration tab.

The following instructions are intended to enable you to start up the servo terminal relatively quickly. More detailed information on the Drive Manager can be found in the corresponding documentation "AX5000 Introduction in the TCDrivemanager"

Start-up with the Drive Manager

- The terminal must already have been added manually under I/O devices or have been scanned in by the system (see section "Configuration set-up in TwinCAT [▶ 102]")
- The terminal must already be integrated in the NC (see section "Integration in the NC configuration [▶ 126]")
- Select the *Configuration* tab for the EL72x1-xxxx.
- Select the connected voltage under *Power Management*.

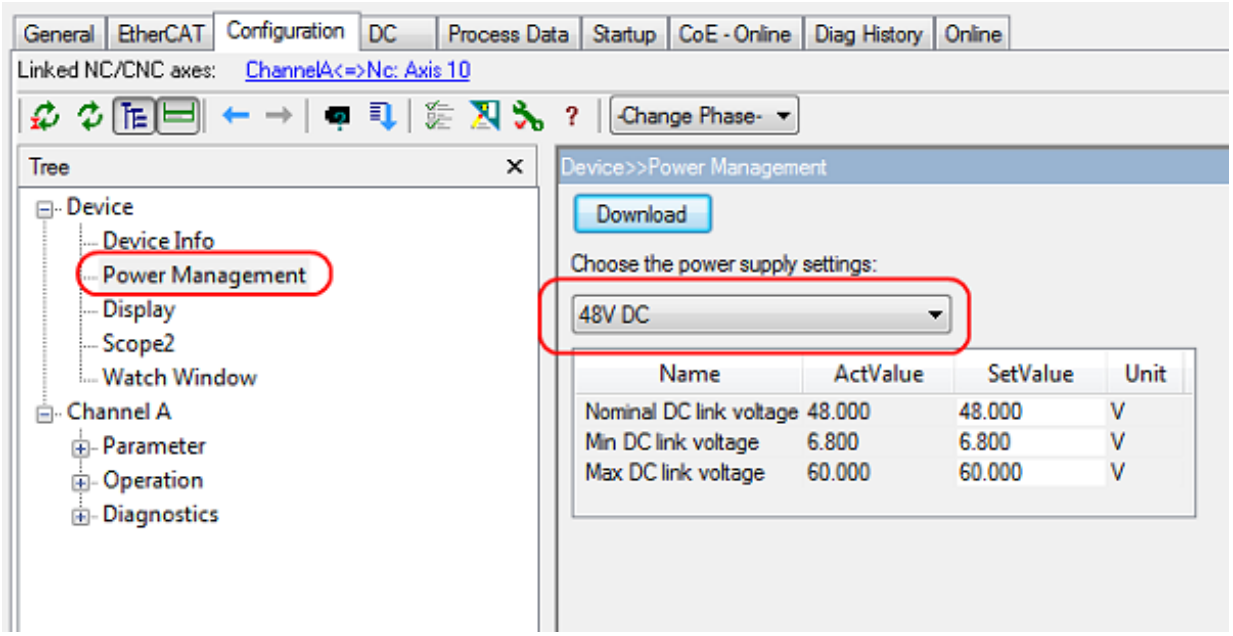


Fig. 145: Selecting the connected voltage

- You can subsequently scan or select the connected motor under *Motor and Feedback*. If you decide to use automatic scanning, click on *Scan motor and feedback*. The electronic type plate of the AM81xx-x2xx motor will then be read automatically. To do this it is necessary for automatic scanning of the motor to be activated in the terminal (Index 0x8008 [▶ 199], MDP or Index 0x2018 [▶ 238], DS402)

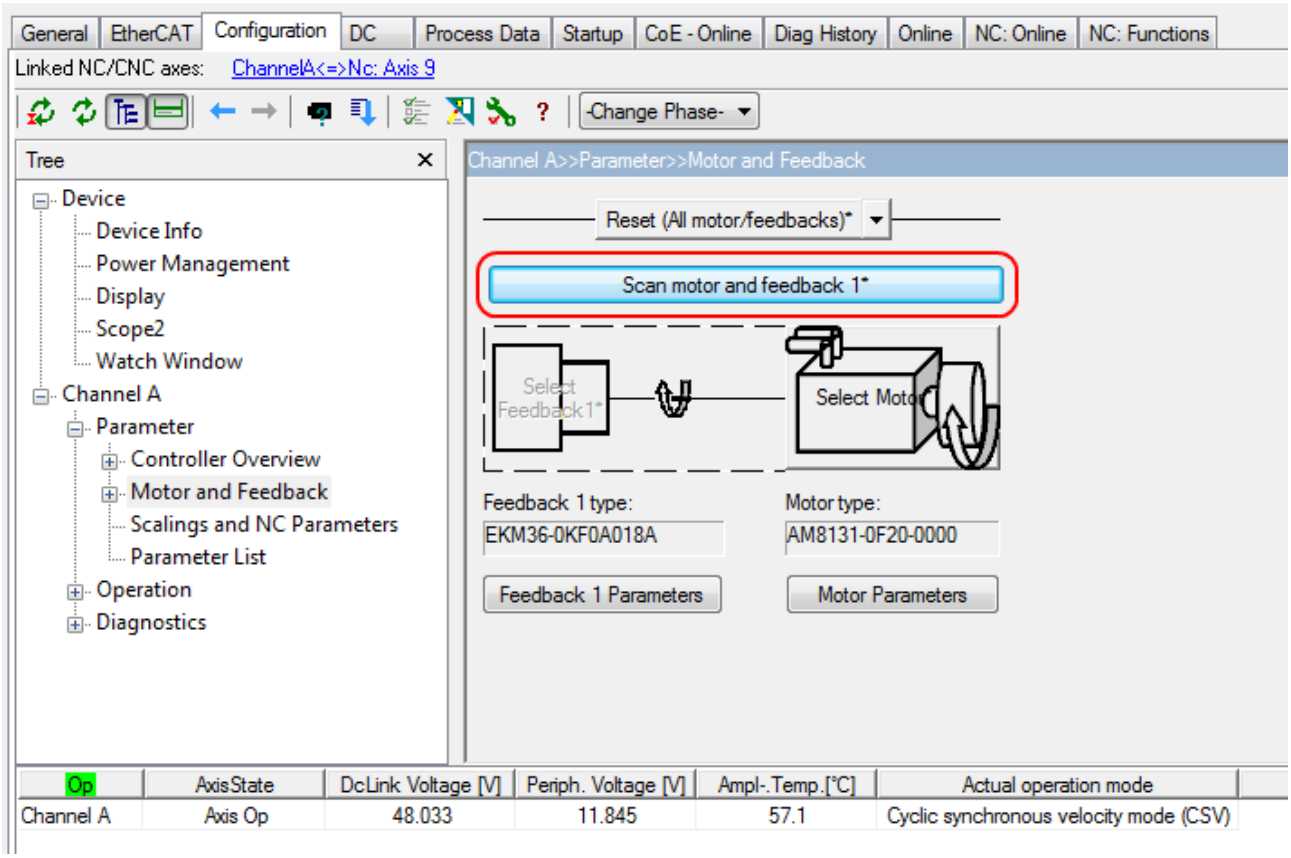


Fig. 146: Automatic scanning of the connected motor

- If you decide to manually input the connected motor, please click on *Select Motor*.

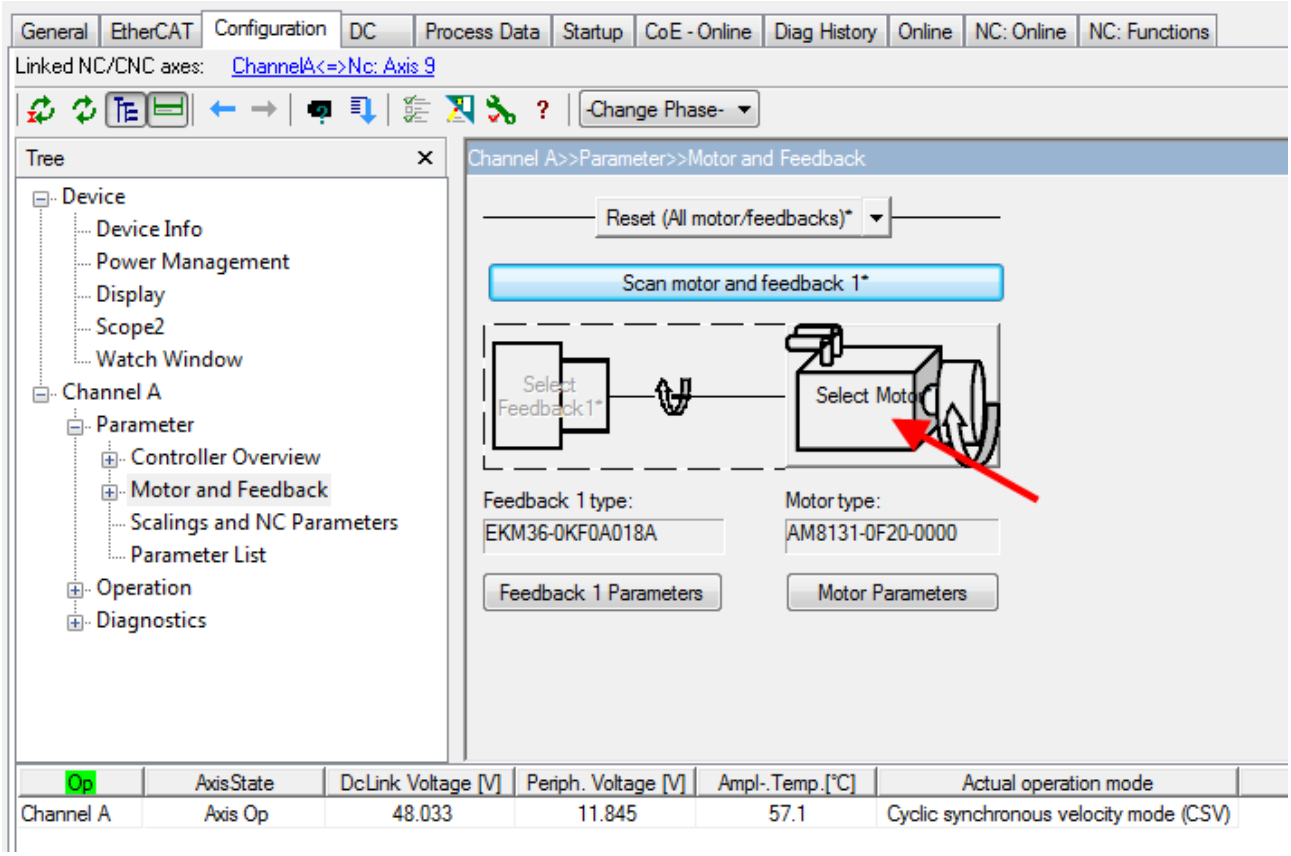


Fig. 147: Selecting the connected motor

- Select the suitable motor in the selection window and confirm with *Ok*.

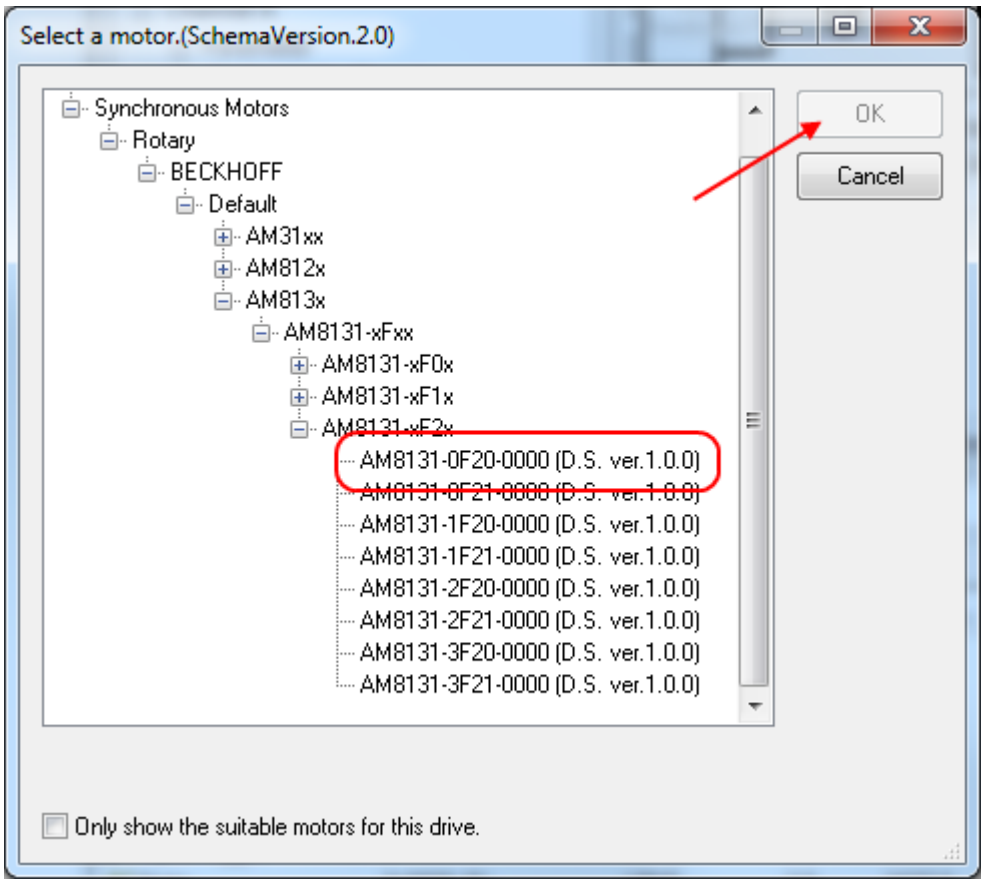


Fig. 148: List of available motors

- Confirm the next dialog box with OK. All required parameters are automatically entered in the NC, and the scaling factor is calculated. If this is not confirmed, these settings have to be entered manually. See section "NC settings".

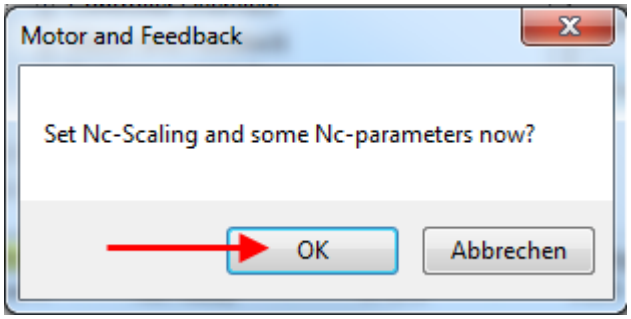


Fig. 149: Confirmation of the automatic NC settings parameters

- The scaling can be determined under *Scalings and NC Parameters*. A motor revolution is defined as 360° as an example. All required parameters are adjusted automatically. The setting only becomes active once the configuration is activated.

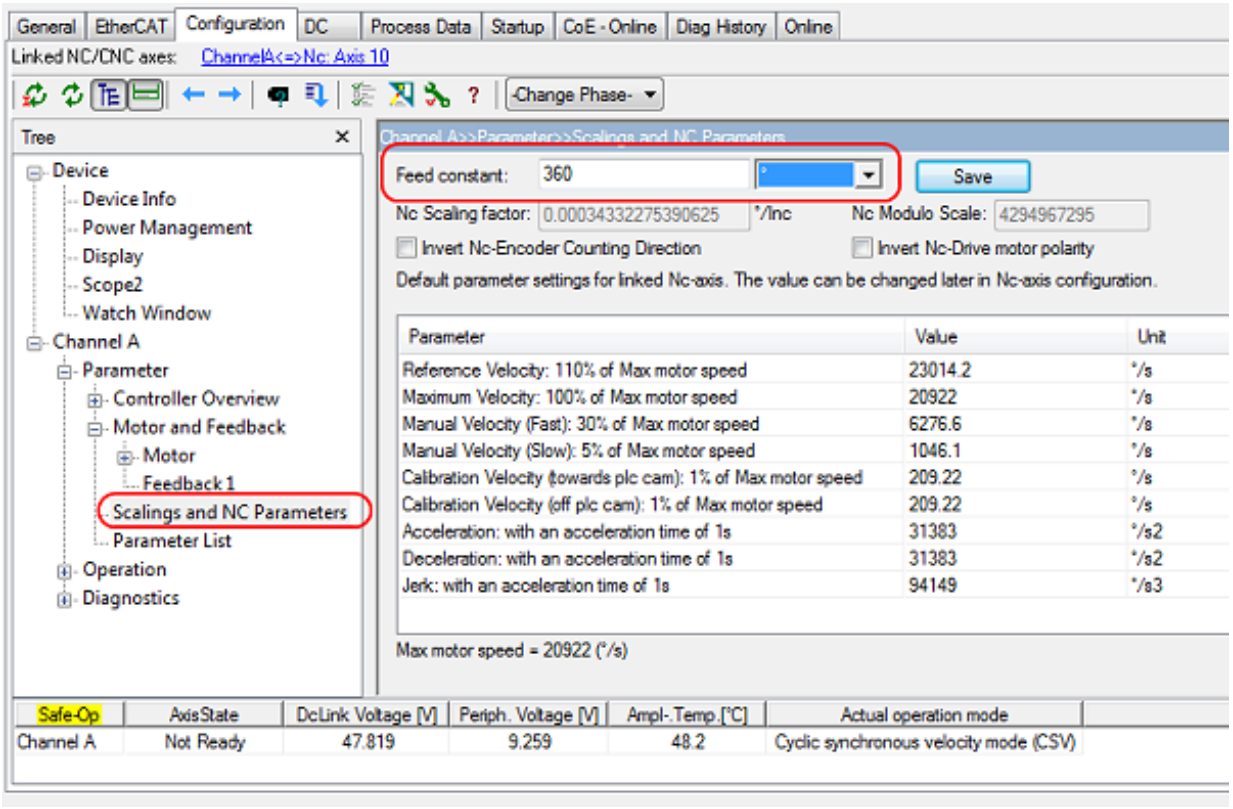


Fig. 150: Adapting the scaling

All main parameters for the commissioning the motor are now set. The motor can now be commissioned with the NC, for example. A brief description can be found in section "Commissioning the motor with the NC [▶ 143]". Or the NC can be addressed from the PLC. A small example program [▶ 144] is included in the documentation.
Some parameters can be adjusted manually for your particular application.

Setting further parameters with the Drive Manager

The values specified here are exemplary, although in most cases they have led to excellent results. Depending on the application, other values may yield better results.
These values can be changed during operation. Click on *Download* to apply the values.

Integral velocity controller component Tn

- Reduce the value, until the motor starts to oscillate slightly. Then increase the value by 10%.

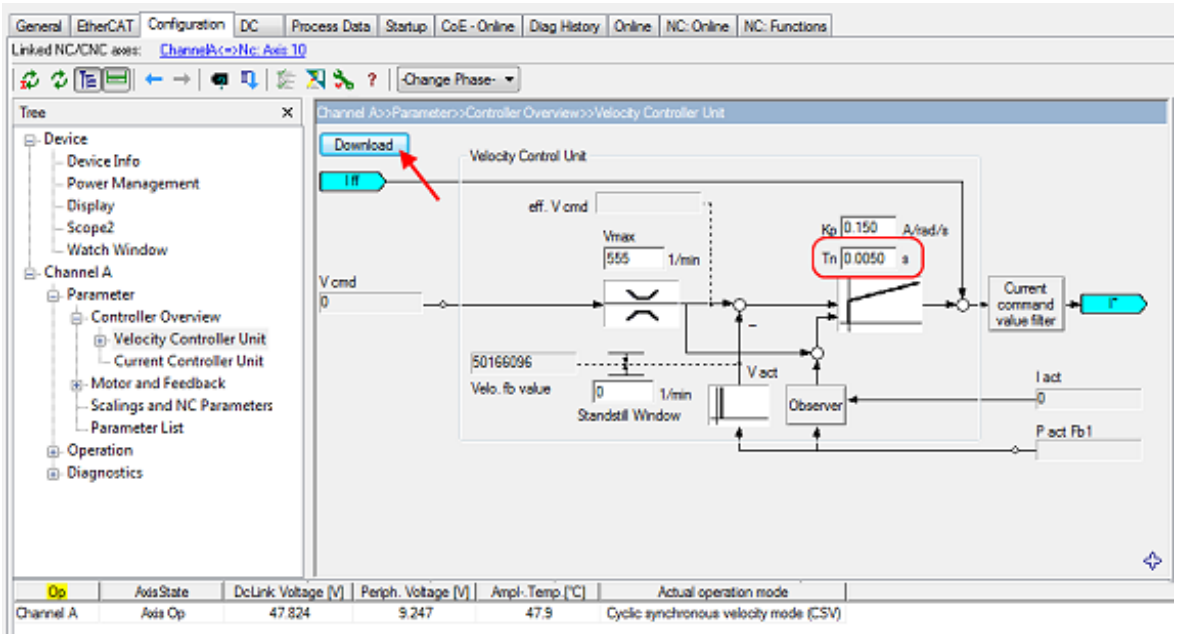


Fig. 151: Adapting Tn

Proportional velocity controller component Kp

- Increase the value, until the motor starts to oscillate slightly. Then reduce the value by 80%.

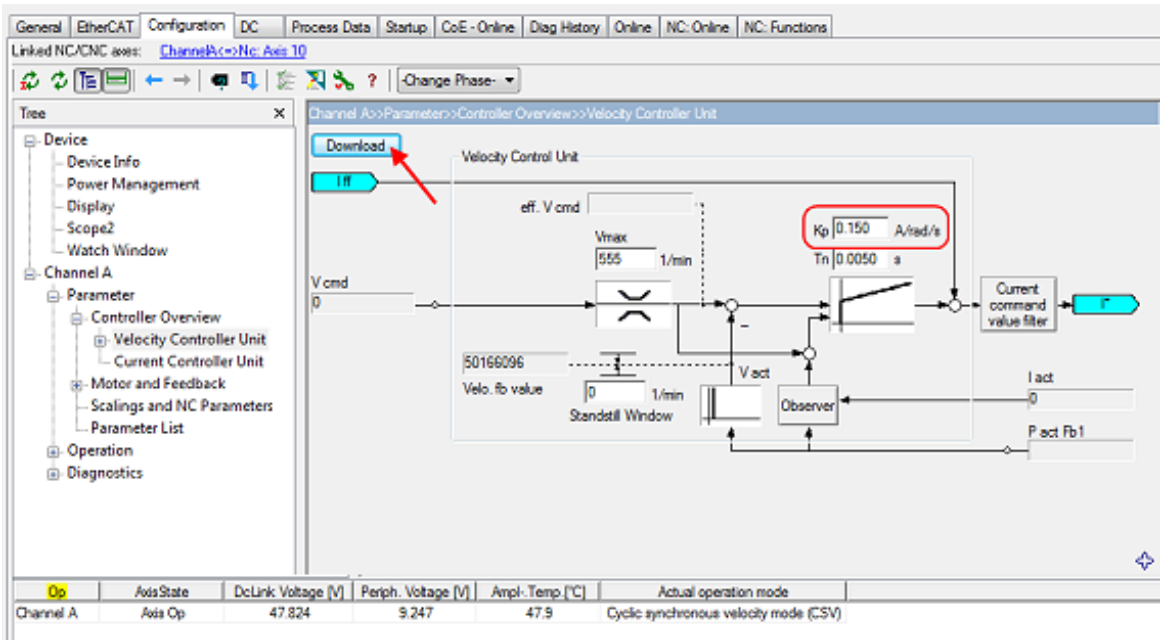


Fig. 152: Adapting Kp

5.3.3 Settings in the CoE register

(Master TwinCAT 2.11 R3)

The data given here serve as an example for a servomotor type AM8131-0F20-0001 from Beckhoff Automation. For other motors the values may vary, depending on the application.

Table of contents

- Inserting the motor XML file [▶ 135]
  - Adaptation of current and voltage [▶ 137]
- Setting further parameters [▶ 137]
  - Single turn bits / Multi turn bits [▶ 137]
  - Torque limitation [▶ 137]
  - Integral velocity controller component Tn [▶ 137]
  - Proportional velocity controller component Kp [▶ 137]

Inserting the motor XML file

Downloading the EL72x1-xxxx motor XML files

The motor XML files are available for download from the Beckhoff website.

To facilitate commissioning of the EL72x1-xxxx servo terminal, motor XML files are provided for the servomotors that are supported by the EL72x1-xxxx. The XML files can be read in the System Manager. All CoE parameters and DS402 parameters are then set as required.

- To read the motor XML file select the EL72x1-xxxx and open the *Startup* tab. Right-click in the empty field and select *Import from XML*...(see Fig. *Importing the motor XML file*).

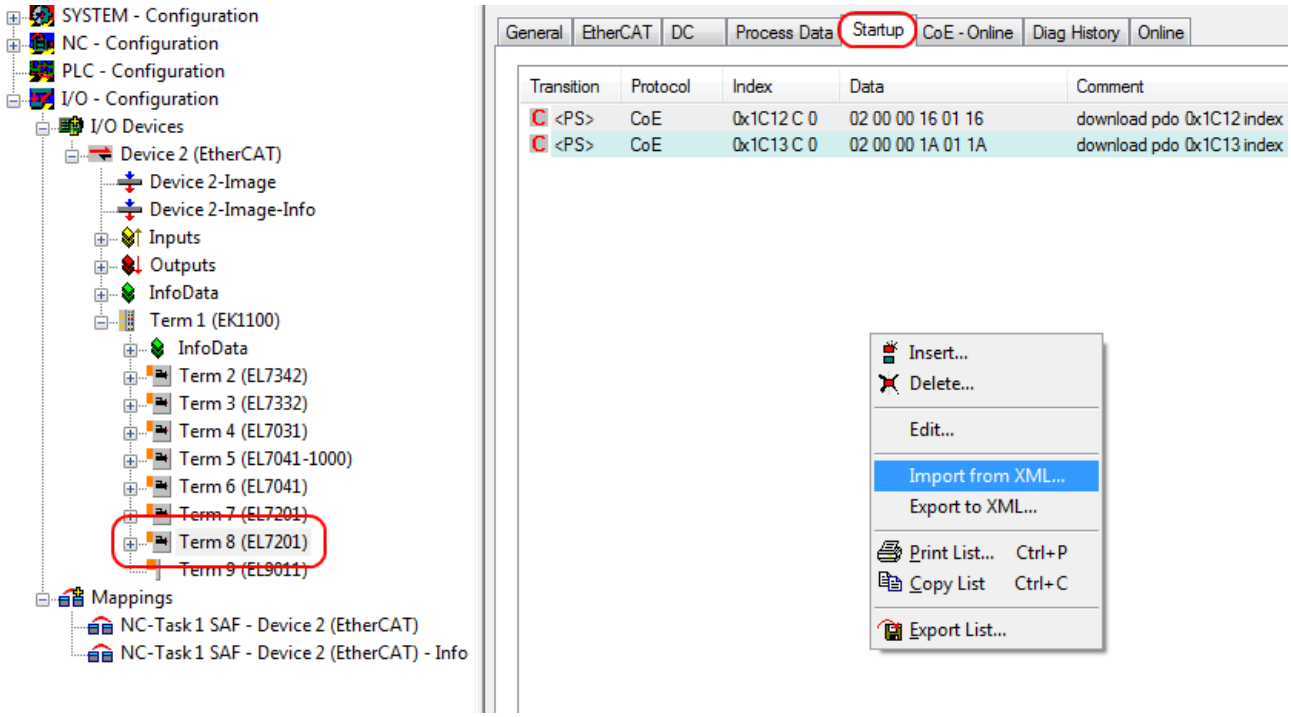


Fig. 153: Importing the motor XML file

- Select the motor XML file that matches the connected motor (see Fig. *Selecting the correct motor XML file*)

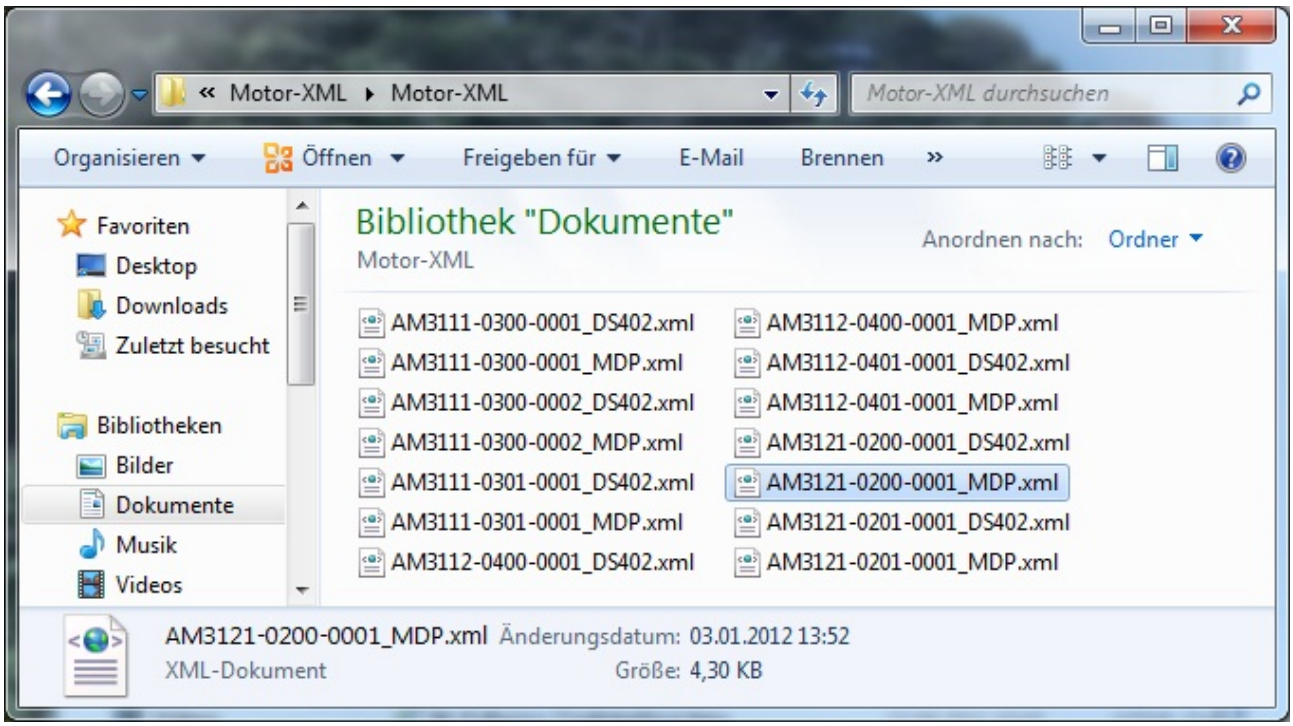


Fig. 154: Selecting the correct motor XML file

- All required parameters are then set, and the motor can be put into operation (see Fig. *CoE parameters of the motor XML file*).

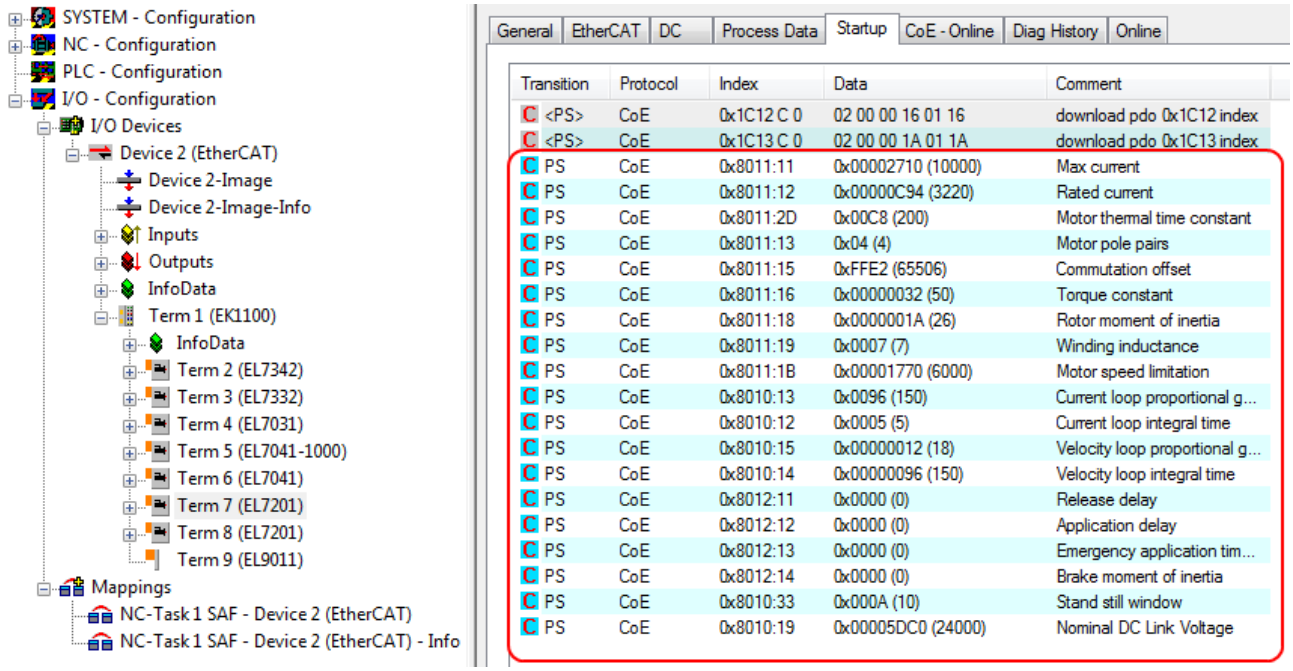


Fig. 155: CoE parameters of the motor XML file

Startup list

Any further application-specific settings should also be implemented in the Startup list. Otherwise the modified settings will be overwritten next time the terminal starts up.

Adaptation of current and voltage

NOTICE

The motor may overheat!

In order to prevent overheating of the connected motor, it is important to adjust the voltage of the servo terminal to the actually connected voltage.

This requires the index 0x8010:19 [▶ 200] (0x2002:19 [▶ 233], DS402 profile) "Nominal DC Link Voltage" of the connected voltage to be set accordingly

Setting further parameters

Single-turn Bits (MDP742: Index 0x8000:12 [▶ 199] / DS402: Index 0x2010:12 [▶ 238]) / Multi-turn Bits (MDP742: Index 0x8000:13 [▶ 199] / DS402: Index 0x2010:13 [▶ 238])

Here the user can specify how many single-turn and multi-turn bits the terminal should display. A total of 32 bits are available. These 32 bits can be subdivided as required.
The standard setting is 20 single-turn bits and 12 multi-turn bits.

Singleturn bits: number of bits relating to the resolution of one rotor rotation.

Multiturn bits: after a rotor rotation the multi-turn bits are incremented by one.

The motor may overheat!

If the number of single-turn bits is changed, the scaling factor in the NC has to be adjusted

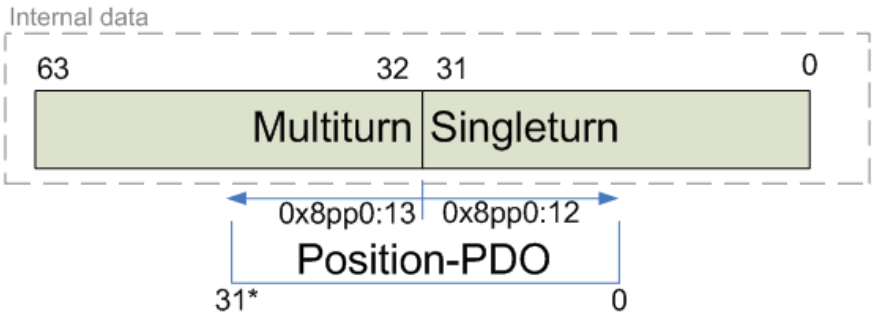


Fig. 156: Multi-turn / single-turn bits:

Torque limitation (MDP742: Index 0x7010:0B [▶ 211] / DS402: Index 0x6072:0 [▶ 241])

Limits the current / torque to this value. The value is specified in 1000th of the rated current.

Integral velocity controller component Tn (MDP742: Index 0x8010:14 [▶ 200] / DS402: Index 0x2002:14 [▶ 233])

The values specified here are exemplary, although in most cases they have led to excellent results. Depending on the application, other values may yield better results.

- Reduce the value, until the motor starts to oscillate slightly. Then increase the value by 10%.

Proportional velocity controller component Kp (MDP742: Index 0x8010:15 [▶ 200] / DS402: Index 0x2002:15 [▶ 233])

The values specified here are exemplary, although in most cases they have led to excellent results. Depending on the application, other values may yield better results.

- Increase the value, until the motor starts to oscillate slightly. Then reduce the value by 80%.

5.3.4 NC settings

(Master TwinCAT 2.11 R3)

The data given here serve as an example for a servomotor type AM8122-0F20-0000 from Beckhoff Automation. For other motors the values may vary, depending on the application.

Table of contents

- Definition of the unit [▶ 138]
- Selecting the maximum velocity [▶ 139]
- Dead time compensation [▶ 139]
- Setting the encoder mask [▶ 140]
- Scaling factor [▶ 141]
  - Calculation of the scaling factor [▶ 141]
  - Scaling output [▶ 141]
- Position lag monitoring [▶ 142]
- Commissioning the motor with the NC [▶ 143]

Several important parameters are required for the commissioning with the NC. These should be set as follows before commissioning. A fundamental factor for setting the following parameters is the unit in which the NC is set to operate. For the following parameters it was assumed that one revolution corresponds to 360°.

Definition of the unit

The unit can be defined in the *Settings* tab for the axis.

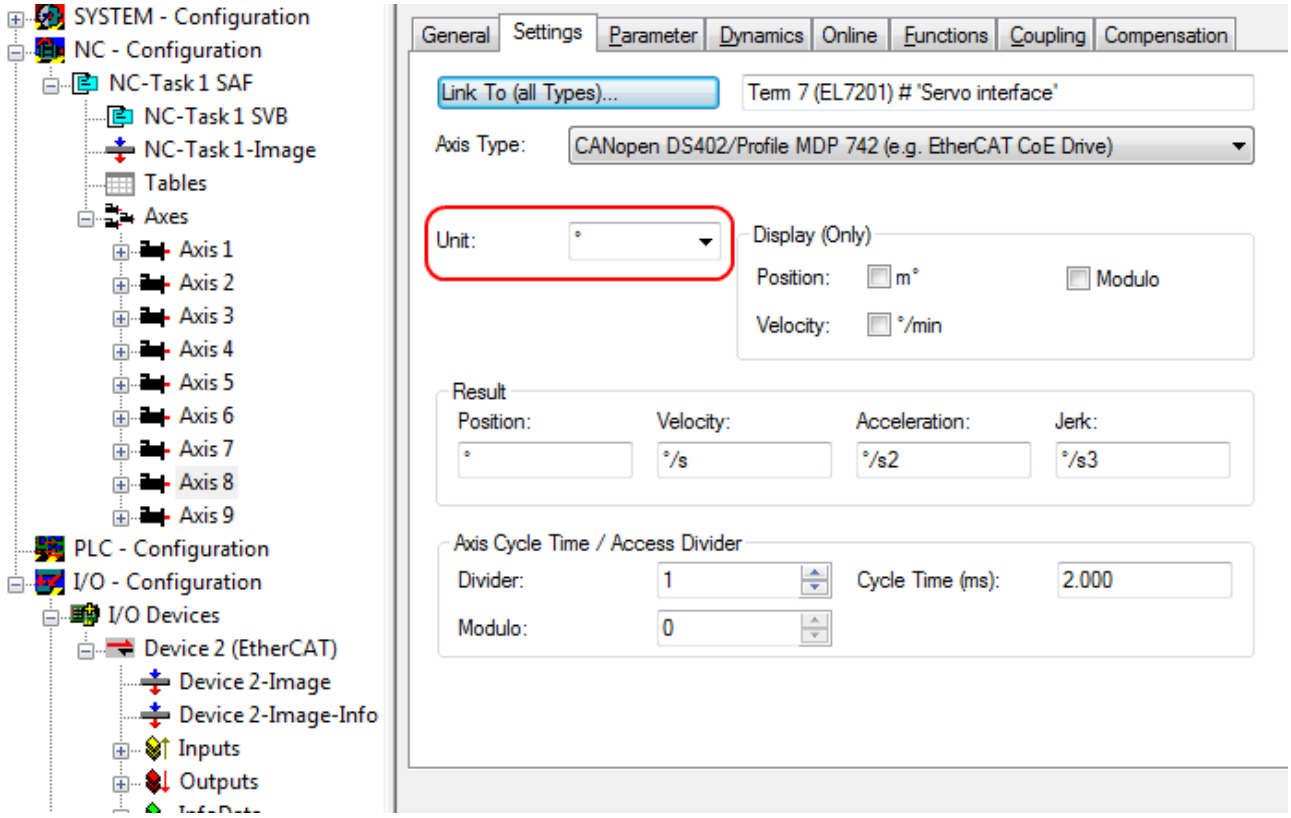


Fig. 157: Definition of the unit

Selecting the maximum velocity

The *maximum permitted velocity* is calculated based on the maximum motor speed (name plate) and the distance, in this case in relation to 360° per second.

$$v_{Bez} = \frac{v_{maxMotor} \times 360°}{60\ s} = \frac{2000\ {}^{1}/_{min} \times 360°}{60\ s} = 12000\ {}^{°}/_{s}$$

$$v_{max} = \frac{v_{maxMotor} \times 360°}{60\ s} = \frac{2000\ {}^{1}/_{min} \times 360°}{60\ s} = 12000\ {}^{°}/_{s}$$

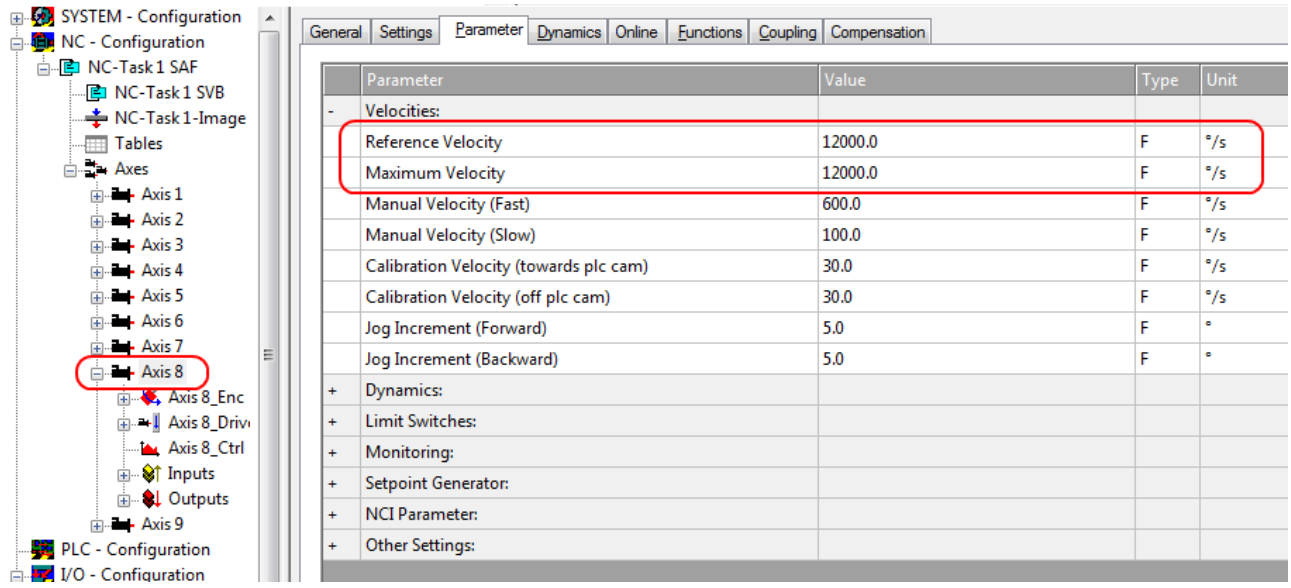


Fig. 158: Adjusting the reference velocity

The *reference velocity* matches the *maximum permitted velocity*.
Below that separate values for the maximum and minimum velocity for manual NC mode can be set.

Dead time compensation

The dead time compensation can be adjusted on the *Time Compensation* tab of *Axis1_ENC*.
It should theoretically be 3 cycles of the NC cycle time, although in practice 4 cycles are preferable.
Therefore, the settings of the parameters *Time Compensation Mode Encoder* should be ‚ON (with velocity)‘ and *Encoder Delay in Cycles* ‘4’.

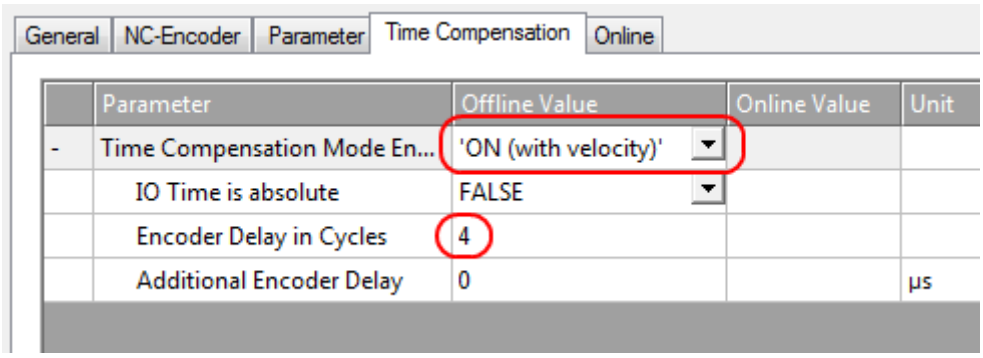


Fig. 159: Dead time compensation parameter

Setting the encoder mask

The maximum values for the encoder mask can be set in the *Parameter* tab for the *Axis1_ENC* encoder settings. The EL72x1-xxxx provides a maximum of 32 bits for the encoder.
The parameter Encoder Mask (maximum encoder value) can be used to set the maximum number of available bits. By default this is set to 0xFFFF FFFF, which corresponds to 32 bits (20 single-turn bits and 12 multi-turn bits). The calculation is based on the following equation.

$$GM_{max} = 2^{Singleturn\,Bits+Multiturn\,Bits} - 1 = 2^{20+12} - 1 = 4\,294\,967\,295 => 0x\ FFFF\ FFFF$$

The parameter Encoder Sub Mask (absolute range maximum value) indicates how many bits of the maximum encoder value are single-turn bits. The default setting is 20 (and therefore 12 multi-turn bits). The calculation is based on the following equation.

$$GM_{ST} = 2^{Singleturn\,Bits} - 1 = 2^{20} - 1 = 1\,048\,575 => 0x\ 000F\ FFFF$$

Further calculation example with 13 single-turn bits and 8 multi-turn bits.

$$GM_{max} = 2^{Singleturn\,Bits+Multiturn\,Bits} - 1 = 2^{13+8} - 1 = 2\,097\,151 => 0x\ 001F\ FFFF$$

$$GM_{ST} = 2^{Singleturn\,Bits} - 1 = 2^{13} - 1 = 8\,191 => 0x\ 0000\ 1FFF$$

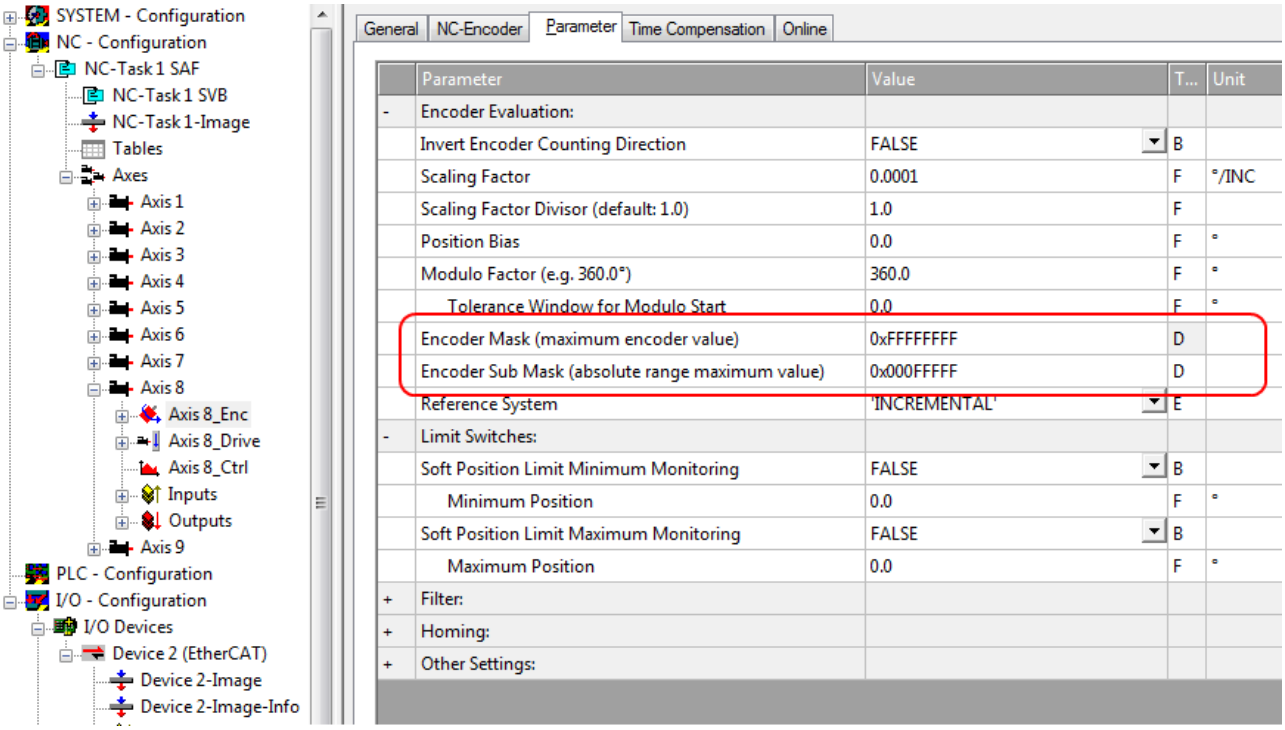


Fig. 160: Setting the encoder mask

Scaling factor

The scaling factor can be changed by selecting "*Axis 1_Enc*" and tab *Parameter* in the NC (see *Setting the Scaling Factor*). The value can be calculated with the formulas specified below. The calculation is based on the assumption that one revolution corresponds to 360°.
The number of single-turn bits is taken into account in the calculation of the scaling factor. As indicated above, the default setting for the EL72x1-xxxx is 20 single-turn bits. This value is also used for calculating the scaling factor. If the single-turn bit value is changed, the scaling factor must be adjusted.

Calculation of the scaling factor

$$SF = \frac{distance\ per\ round}{2^{Singleturn\ Bits}} = \frac{360^{\circ}}{2^{20}} = 0{,}000343322753906\ {}^{\circ}/_{INC}$$

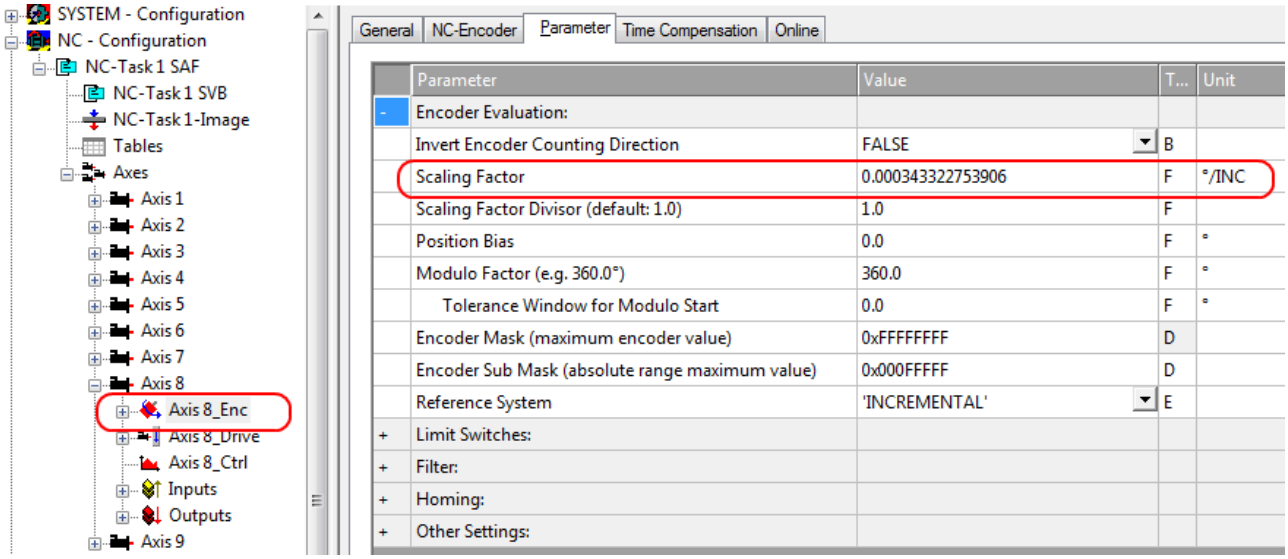


Fig. 161: Setting the Scaling Factor

Scaling output

Enter the value 32 in the *Parameter* tab for the drive settings under *Output Scaling (Velocity)*.

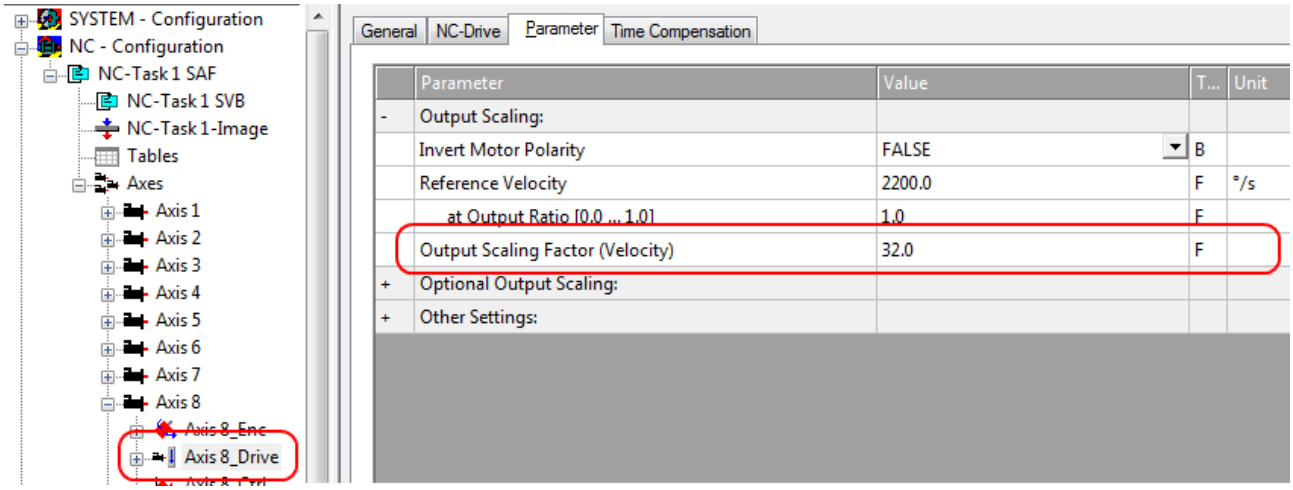


Fig. 162: Output scaling

Position lag monitoring

The position lag monitoring function checks whether the current position lag of an axis has exceeded the limit value. The position lag is the difference between the set value (control value) and the actual value reported back. If the terminal parameters are set inadequately, the position lag monitoring function may report an error when the axis is moved. During commissioning it may therefore be advisable to increase the limits of the *Position lag monitoring* slightly.

NOTICE

Damage to equipment, machines and peripheral components possible!

Setting the position lag monitoring parameters too high may result in damage to equipment, machines and peripheral components.

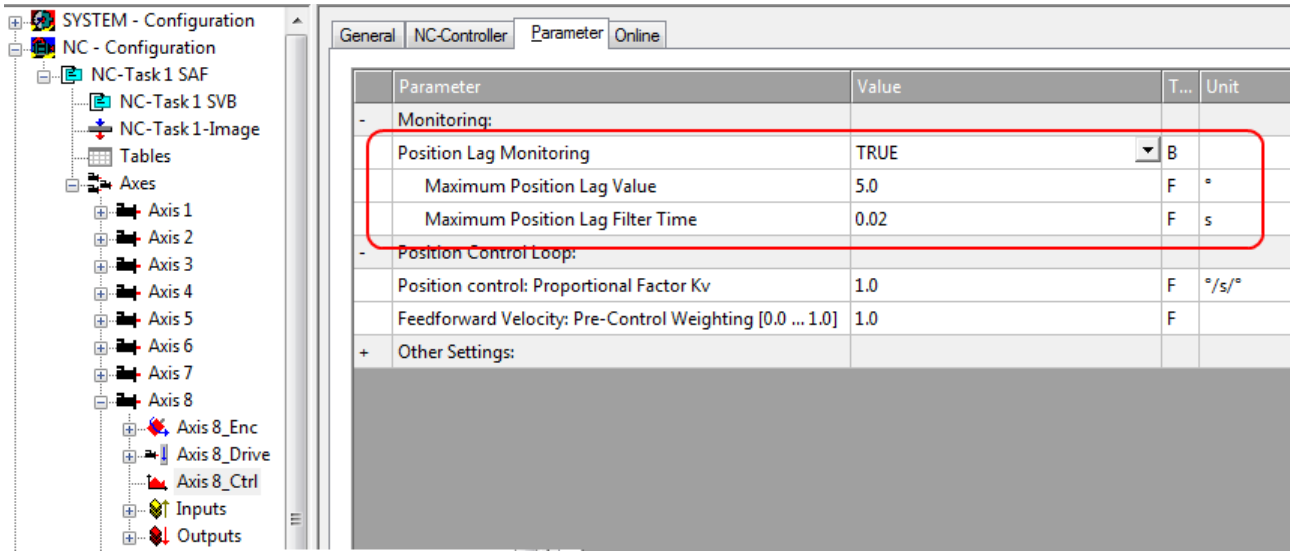


Fig. 163: Lag monitoring

Commissioning the motor with the NC

- Once the parameters are set, the motor is basically ready for operation. Individual further parameters have to be adapted to the respective application.
- To commission the axis, activate the configuration (Ctrl+Shift+F4), select the axis, select tab *Online* and enable the axis under Set.
- Set all tick marks and set *Override* to 100% (see Fig. *Enabling an axis*). The axis can then be moved.

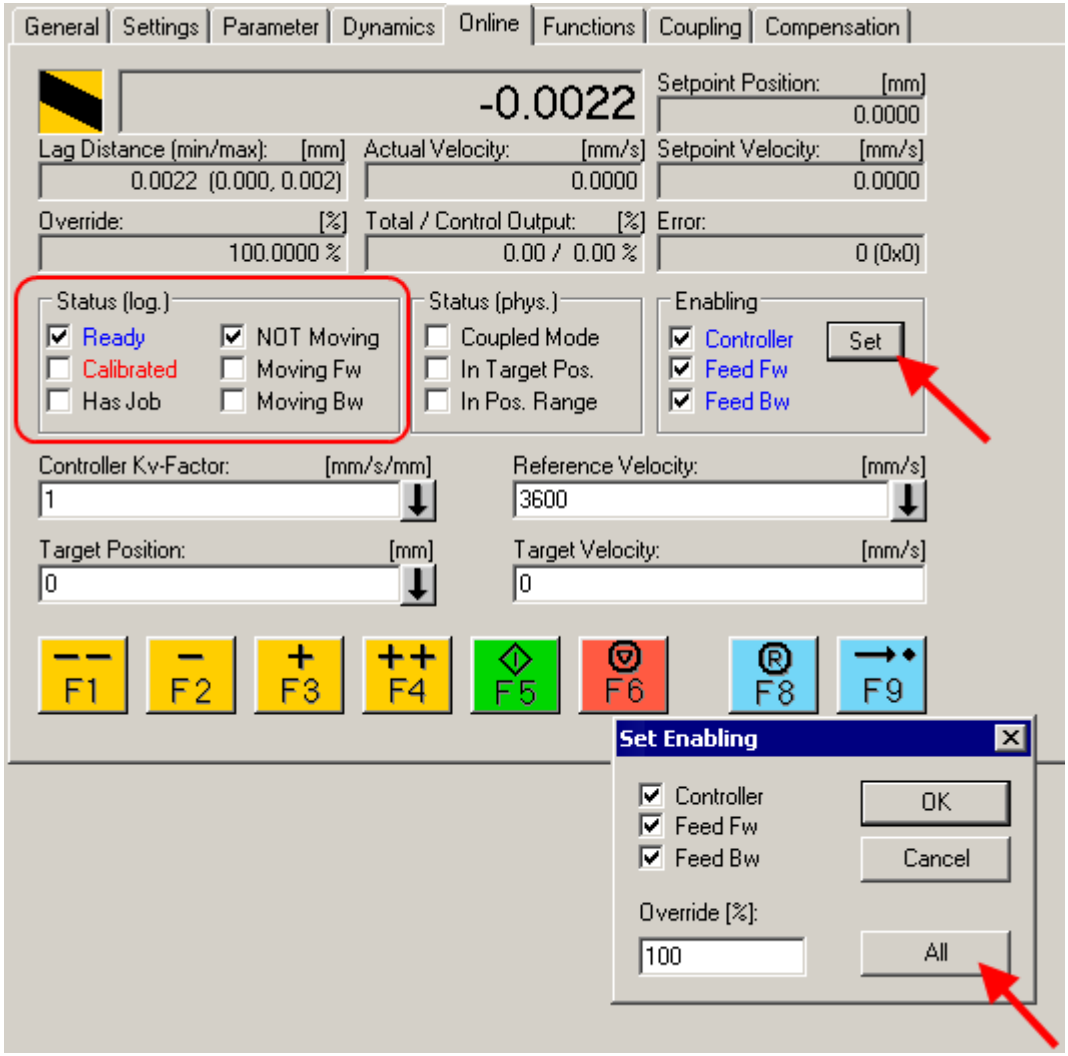


Fig. 164: Enabling an axis

You can now move the axis with the function keys F1, F2 (Backward) or F3, F4 (Forward).
You can adjust the Kv factor in order to approach a suitable factor. Set the value to 0 initially in order to set the correct reference velocity. For calculating the reference velocity please refer to section "Selecting the maximum velocity [▶ 139]". The calculation provides a relatively precise value, although the value may have to be corrected slightly. To this end move the motor with a Kv factor of 0 until the actual velocity matches the setpoint velocity.

Alternatively you can control the axis via the *Functions* tab. An example is provided below.

- Select as *Reversing Sequence* as the start type.
- Enter the required *Target Position2*, e.g. 12000°.
- Enter the required Target Velocity, e.g. 12000°/s.
- Enter the required Target Position1, e.g. 0°.
- Enter the required *Idle Time*, e.g. 2 s.
- Select Start.

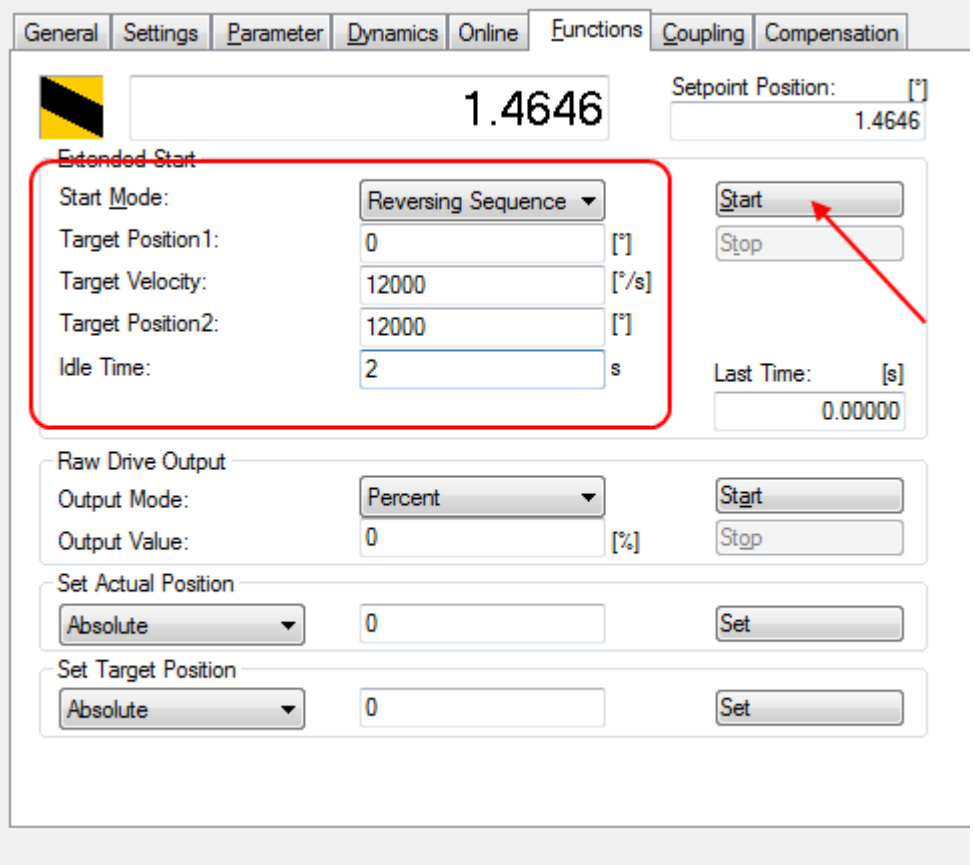


Fig. 165: Reversing Sequence

The motor now turns to position 2, remains there for 2 seconds and returns to position 1. This is repeated until Stop is pressed.

5.3.5 Application example

Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be downloaded from the Beckhoff Website and installed according to the installation instructions.

Motor control with visualization

Download (https://infosys.beckhoff.com/content/1033/el72x1-901x/Resources/1859339787.zip):

Used Master: TwinCAT 2.11 (for older versions the control loop has to be programmed manually; in this case it is already implemented in the NC).
This application example demonstrates movement of a motor to any position or in continuous mode with the aid of visualization. The velocity, the starting acceleration and the deceleration can be specified.

The sample program consists of two files (PLC file and System Manager file).

First open the PLC file and compile it so that you have the *.tpy file available that is required for the System Manager.

Please note that you may have to adjust the target platform in the PLC program (default: PC or CX 8x86). If required, you can select the target platform under *Resources -> Controller configuration*.

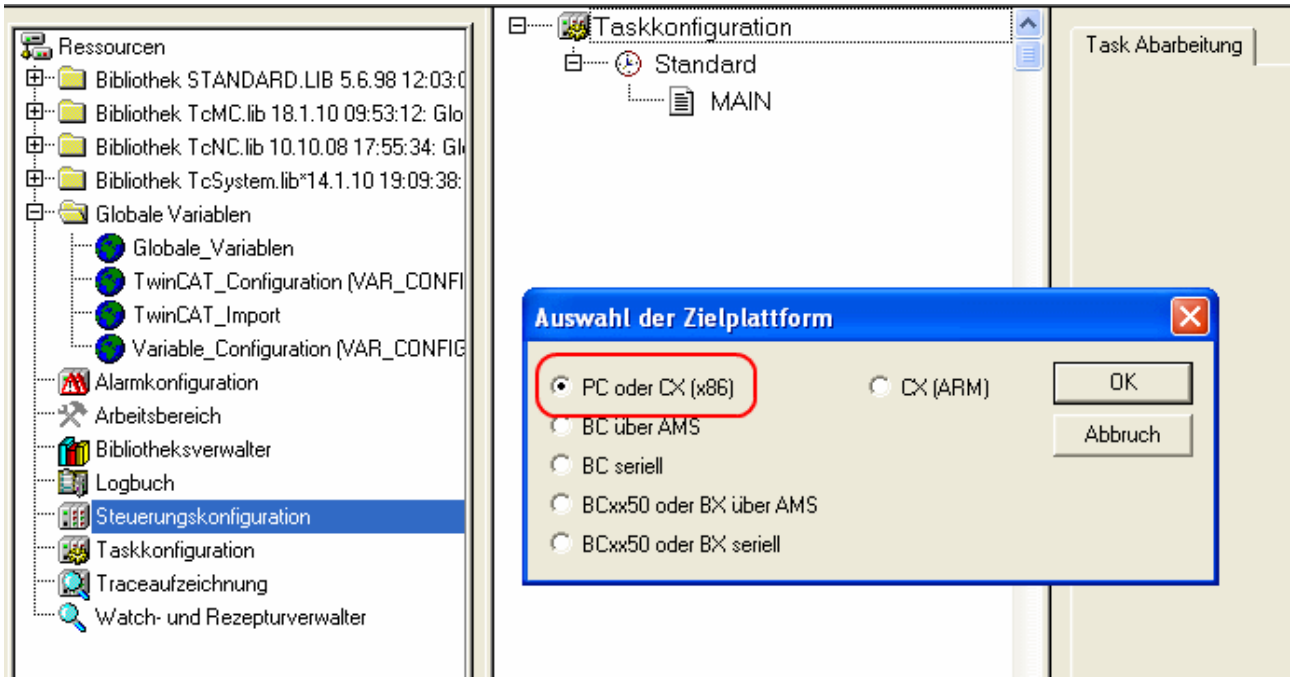


Fig. 166: Selection of the target platform

Please note the following for the System Manager file:

- Start the System Manager in Config mode.
- Please ensure that the I/O configuration matches your actual configuration. In the sample program only one EL7041 is integrated. If further terminals are connected you have to add them or re-scan your configuration.
- You have to adjust the MAC address. To do this, click on your *EtherCAT device*, then select the *Adapter* tab and click on *Search* after the MAC address (see Fig. *Selecting the MAC address*). Select the right adapter.

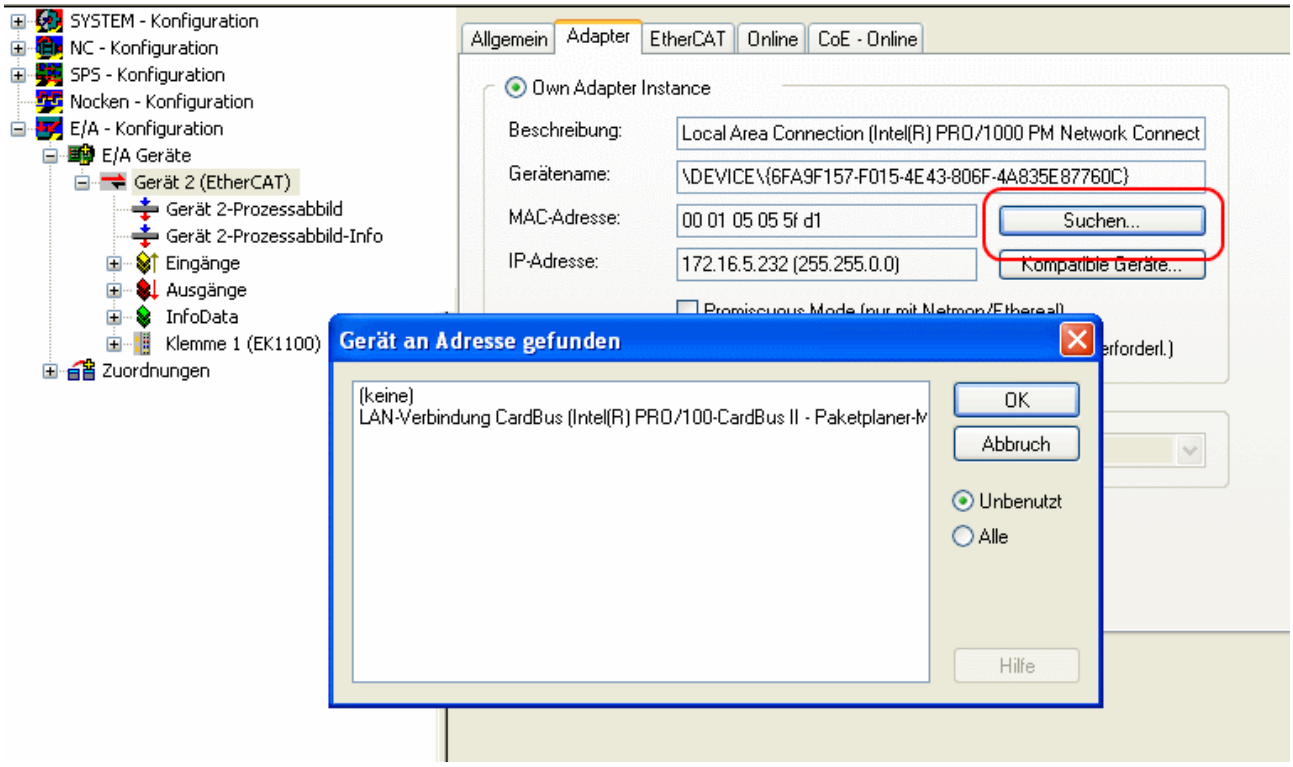


Fig. 167: Selecting the MAC address

- In the PLC configuration you have to adjust the path for the PLC program. Click on the appended PLC program and select the tab *IEC1131* (see Fig. *Changing the PLC path*). Select *Change* and enter the correct path.

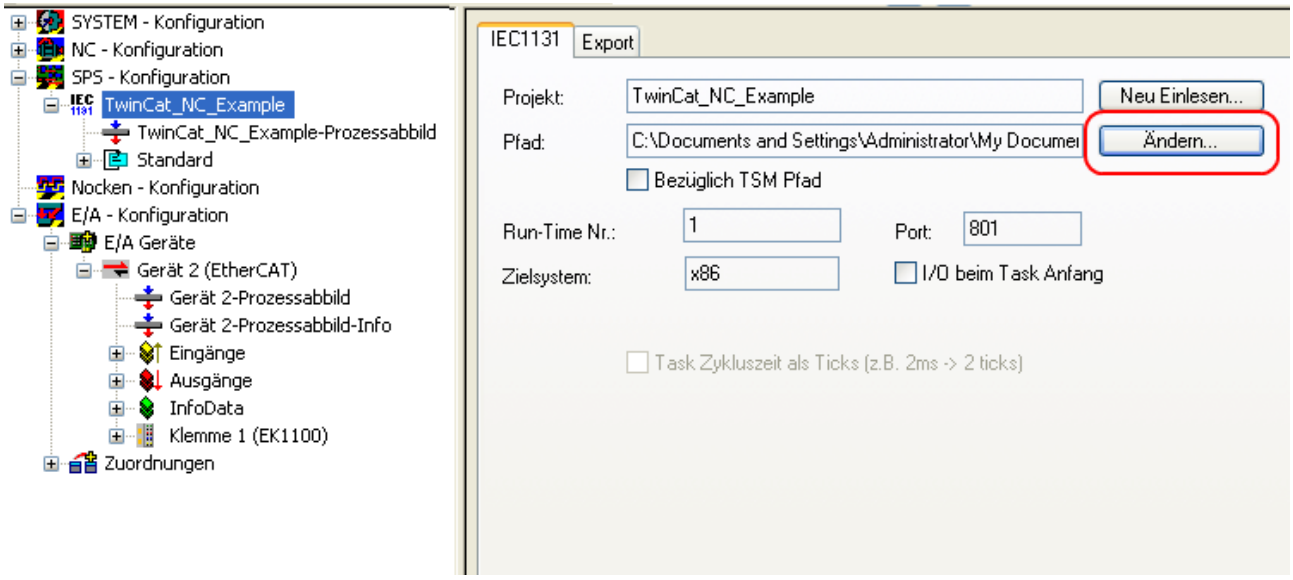


Fig. 168: Changing the PLC path

- Under NC configuration an EL7041 is already linked to the NC. To change the link or add additional devices proceed as described under “Integration into the NC configuration [▶ 126]”.

The PLC program is configured as follows. The libraries *TcMC.lib* and *TcNC.lib* must be integrated (see Fig. *Required libraries*).

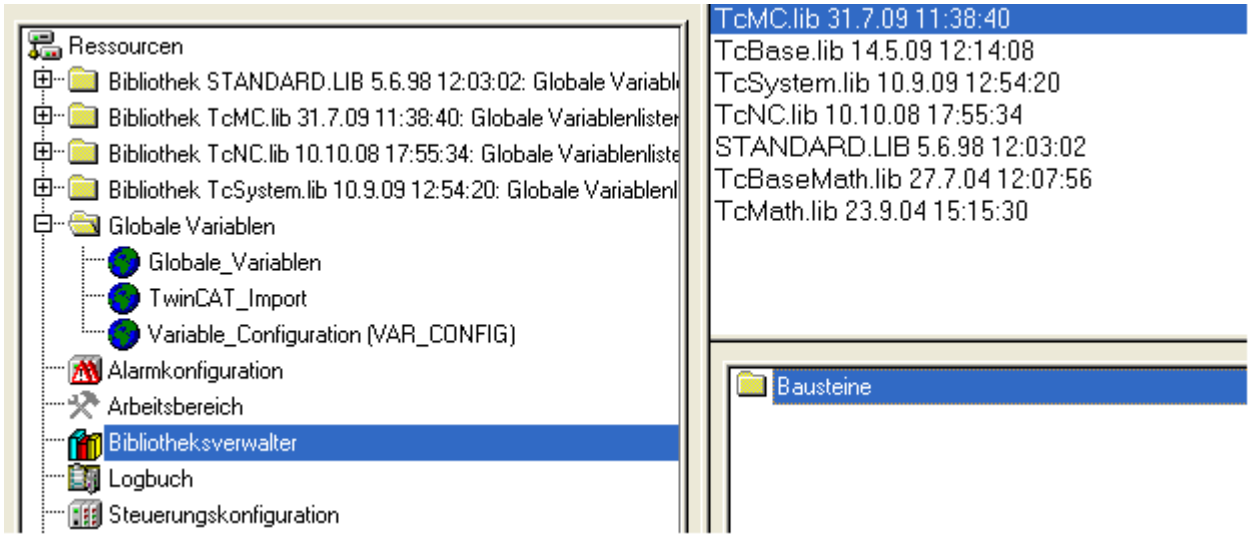


Fig. 169: Required libraries

Once this is done, certain global variables are declared (see Fig. *Global variables*). The data types *PLCTONC_AXLESTRUCT* and *NCTOPLC_AXLESTRUCT*deal with the communication between the PLC and the NC.

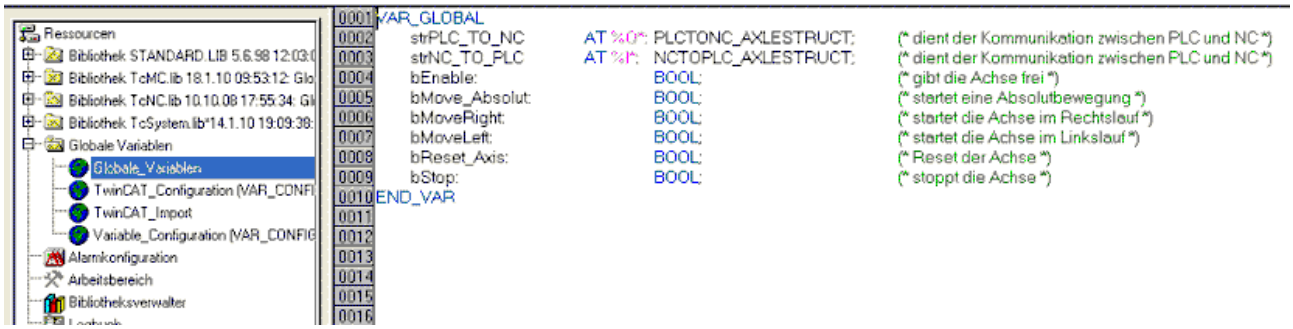


Fig. 170: Global variables

Once the global variables have been declared, programming can commence. Start with declaring local variables (see Fig. *Local variables*).
MC_Direction is an enumeration type that specifies the direction of travel for the block *MC_MoveVelocity*, which in turn initiates continuous travel of the motor.
An axis reset is carried out with the function block *MC_Reset*. Absolute positioning is carried out with the function block *MC_MoveAbsolute*. The current axis position can be read with the function block *MC_ActualPosition*.
MC_Power enables the axis; *MC_Stop* is required for stopping the axis.

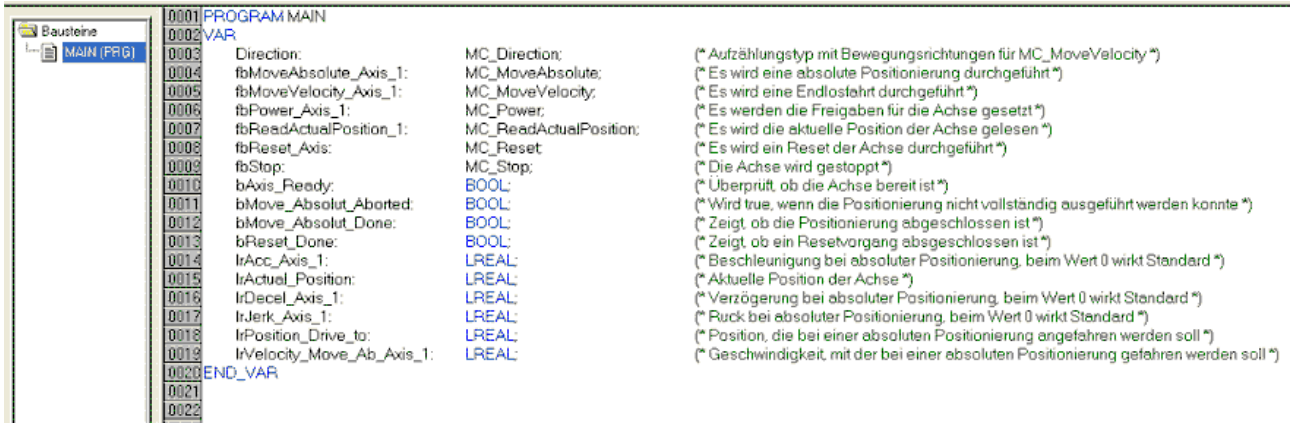
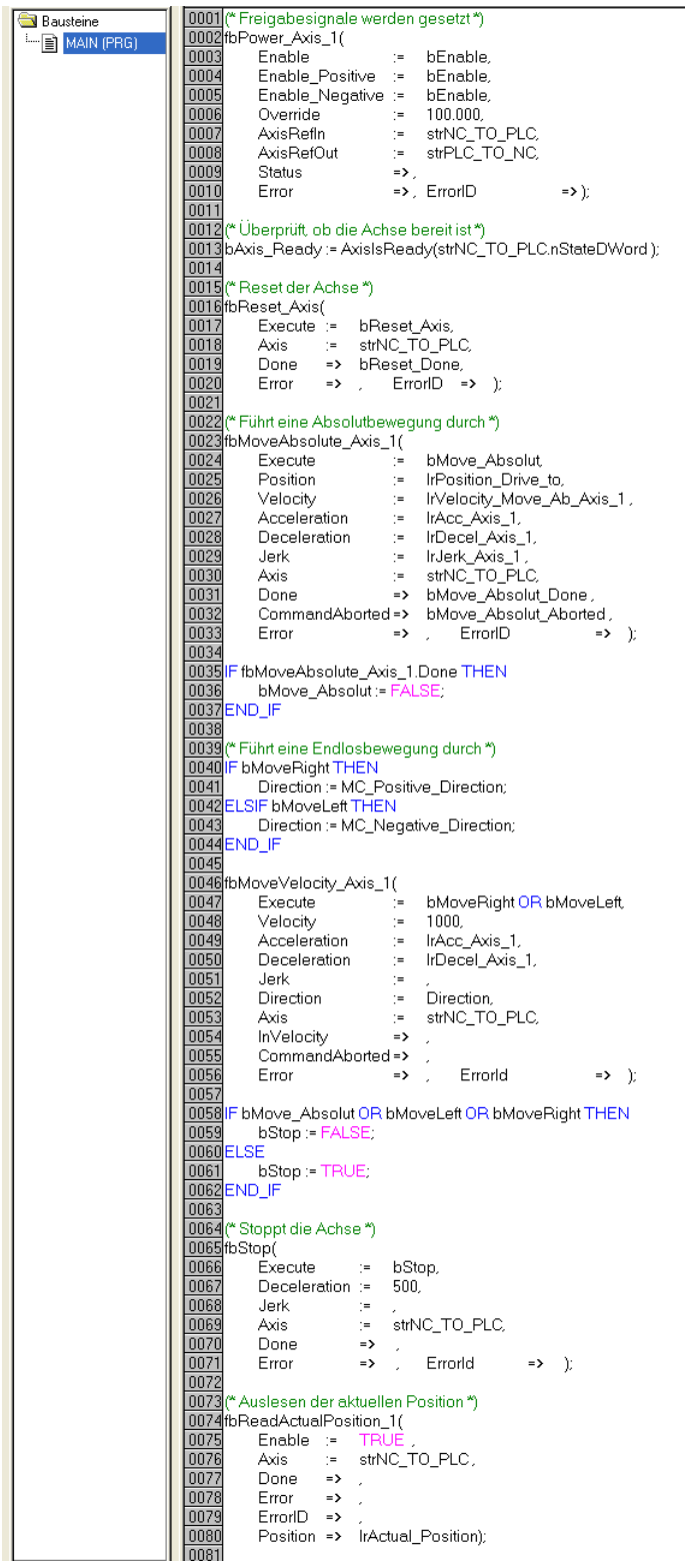


Fig. 171: Local variables

The program code is as follows (see Fig. *Program code*):




```
(* Freigabesignale werden gesetzt *)
fbPower_Axis_1(
    Enable          :=  bEnable,
    Enable_Positive :=  bEnable,
    Enable_Negative :=  bEnable,
    Override        :=  100.000,
    AxisRefIn       :=  strNC_TO_PLC,
    AxisRefOut      :=  strPLC_TO_NC,
    Status          =>  ,
    Error           =>  , ErrorID       => );

(* Überprüft, ob die Achse bereit ist *)
bAxis_Ready := AxisIsReady(strNC_TO_PLC.nStateDWord );

(* Reset der Achse *)
fbReset_Axis(
    Execute :=  bReset_Axis,
    Axis    :=  strNC_TO_PLC,
    Done    =>  bReset_Done,
    Error   =>  ,   ErrorID =>  );

(* Führt eine Absolutbewegung durch *)
fbMoveAbsolute_Axis_1(
    Execute         :=  bMove_Absolut,
    Position        :=  lrPosition_Drive_to,
    Velocity        :=  lrVelocity_Move_Ab_Axis_1 ,
    Acceleration    :=  lrAcc_Axis_1,
    Deceleration    :=  lrDecel_Axis_1,
    Jerk            :=  lrJerk_Axis_1 ,
    Axis            :=  strNC_TO_PLC,
    Done            =>  bMove_Absolut_Done ,
    CommandAborted  =>  bMove_Absolut_Aborted ,
    Error           =>  ,   ErrorID     =>  );

IF fbMoveAbsolute_Axis_1.Done THEN
    bMove_Absolut := FALSE;
END_IF

(* Führt eine Endlosbewegung durch *)
IF bMoveRight THEN
    Direction := MC_Positive_Direction;
ELSIF bMoveLeft THEN
    Direction := MC_Negative_Direction;
END_IF

fbMoveVelocity_Axis_1(
    Execute         :=  bMoveRight OR bMoveLeft,
    Velocity        :=  1000,
    Acceleration    :=  lrAcc_Axis_1,
    Deceleration    :=  lrDecel_Axis_1,
    Jerk            :=  ,
    Direction       :=  Direction,
    Axis            :=  strNC_TO_PLC,
    InVelocity      =>  ,
    CommandAborted  =>  ,
    Error           =>  ,   ErrorId     =>  );

IF bMove_Absolut OR bMoveLeft OR bMoveRight THEN
    bStop := FALSE;
ELSE
    bStop := TRUE;
END_IF

(* Stoppt die Achse *)
fbStop(
    Execute     :=  bStop,
    Deceleration :=  500,
    Jerk        :=  ,
    Axis        :=  strNC_TO_PLC,
    Done        =>  ,
    Error       =>  ,   ErrorId     =>  );

(* Auslesen der aktuellen Position *)
fbReadActualPosition_1(
    Enable  :=  TRUE ,
    Axis    :=  strNC_TO_PLC ,
    Done    =>  ,
    Error   =>  ,
    ErrorID =>  ,
    Position =>  lrActual_Position);
```

Fig. 172: Program code

The motor can then be operated with the aid of the following visualization (see Fig. *Visualization*). Press *Enable* to enable the axis. In "Free run mode" you can now use the *Left* or *Right* buttons, and the motor will run with a speed defined under *fbMoveVelocity_Axis_1* in the selected direction. In "Absolute mode" you can specify a *Velocity*, *Acceleration*, *Deceleration* and the Setpoint *Position* and initiate the motion with *Start Job*. If no values are entered for *acceleration* and *deceleration* the default value of the NC is used.

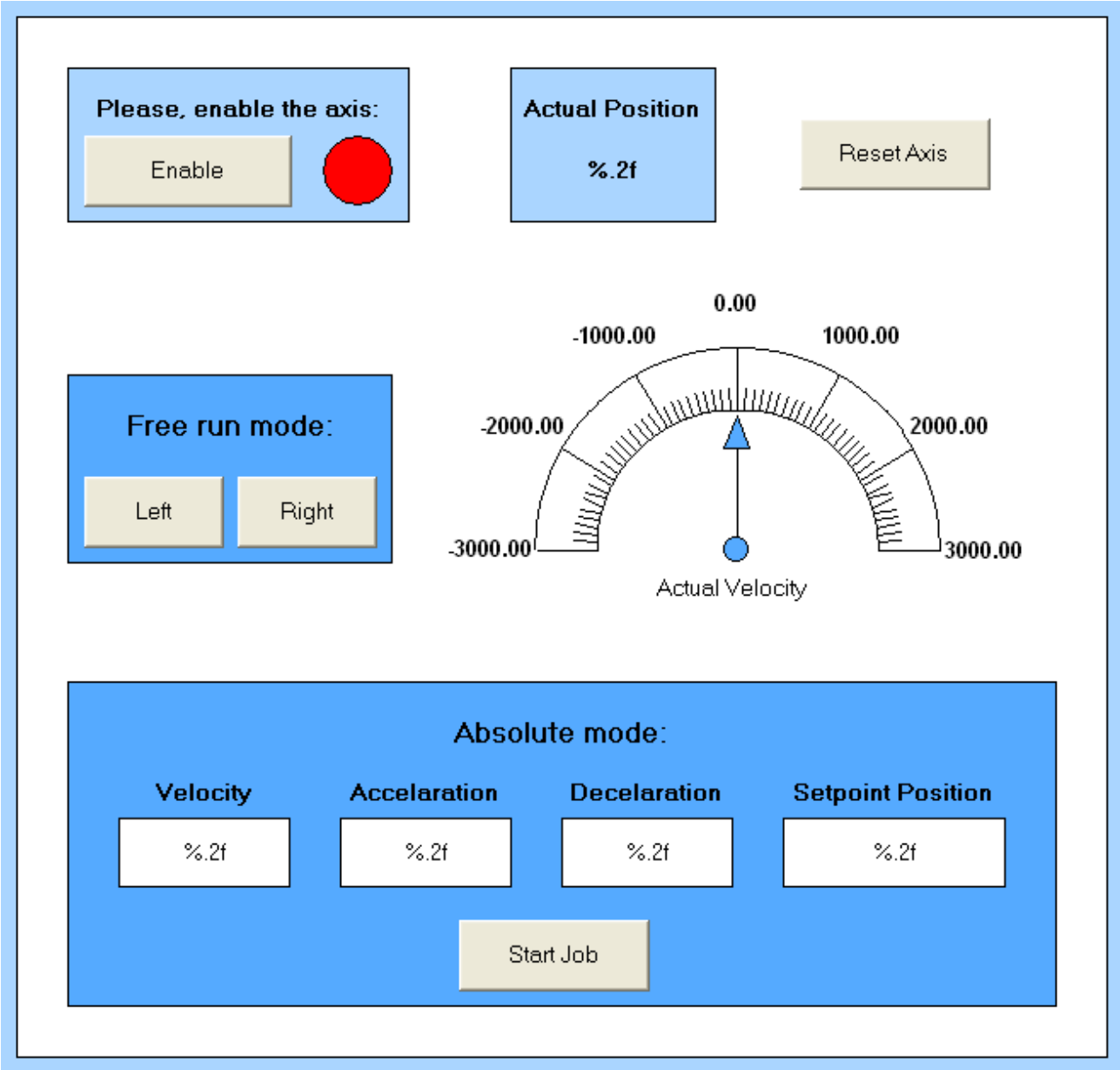


Fig. 173: Visualization

Information on function blocks and data types

Further information on the function blocks and data types used can be found in the Beckhoff Information System.

5.3.6 Commissioning without NC, status word/control word

(Master: TwinCAT 2.11 R3)

In principle, the operating modes CST, CSTCA, CSV and CSP can be used without TwinCAT NC.

Output stage enabled via control word

The output stage has to be enabled for each operating mode. To this end enter the following values in the specified order via the PLC control word (MDP742 [▶ 211] / DS402 [▶ 240]) (see Fig. *DS402 State Machine*). The respective status messages are output in the status word (MDP742 [▶ 208] / DS402 [▶ 240]).

0_{hex}
80_{hex} (Fault reset)
6_{hex} (Shutdown)
7_{hex} (Switch on)
F_{hex} (Enable operation)

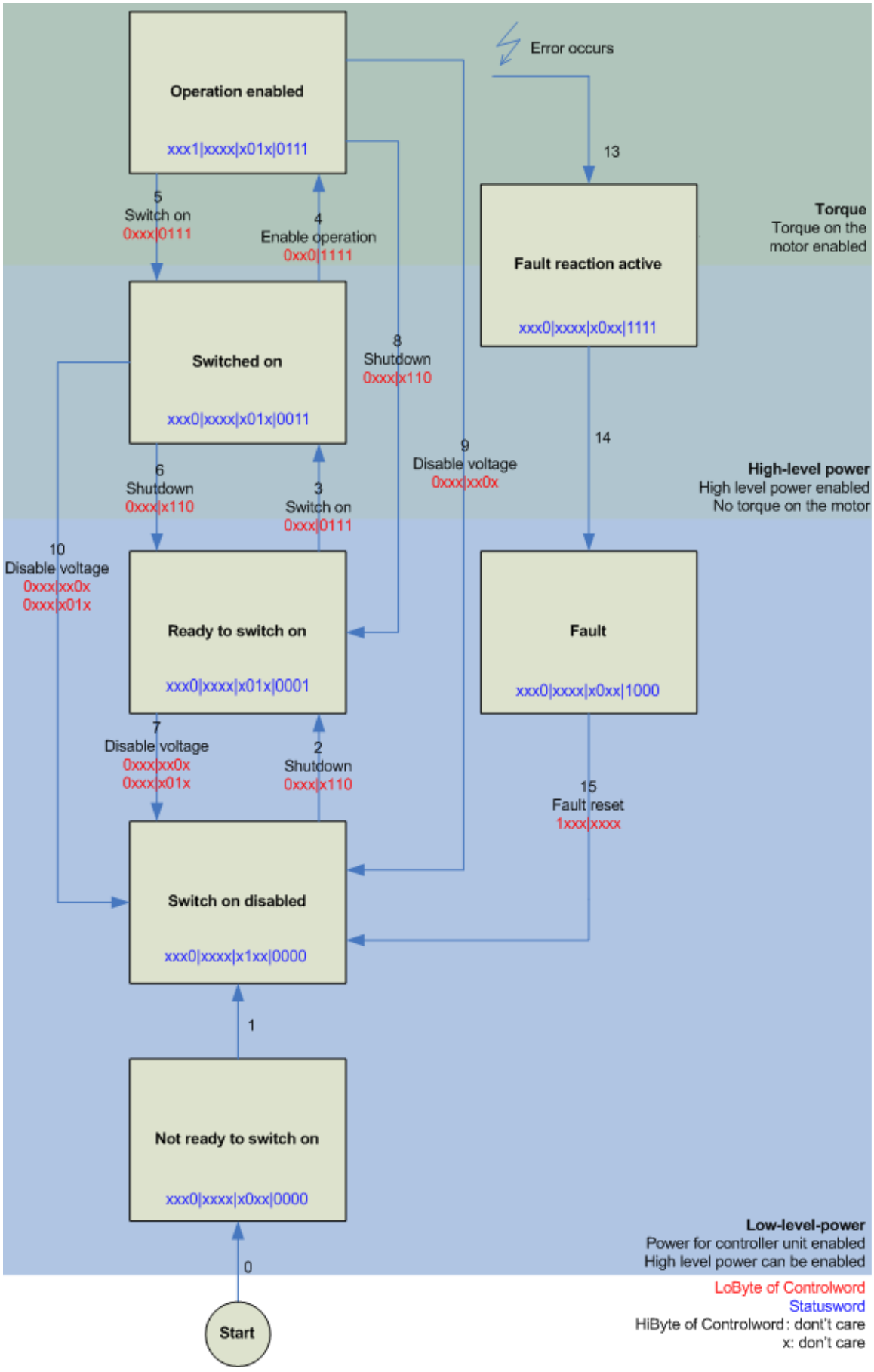


Fig. 174: DS402 State Machine

CST - cyclic synchronous torque

Select *Cyclic synchronous torque mode* in index 0x7010:03 [▶ 211] *Modes of operation* (MDP) or 0x6060:0 [▶ 240] *Modes of operation* (DS402). The *Predefined PDO Assignment: 'Cyclic synchronous torque mode (CST)'* should also be selected in the respective process data (see CoE process data [▶ 177] or DS402 process data [▶ 183]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 208] *Modes of operation display* (MDP) or 0x6061:0 [▶ 240] *Modes of operation display* (DS402) can be used to check the actual mode of the servo terminal.

Via the PLC a defined torque can be set in the variable *Target torque* as a target value for the servo terminal. The torque is specified in 1000th of the rated current. A value of 1000_{dec}, for example, corresponds to the set index 0x8011:12 [▶ 203] *Rated current* (MDP) or index 0x6075:0 [▶ 241] *Motor rated current* (DS402). The value 1_{dec} corresponds to one 1000th of the rated current.

CSTCA - cyclic synchronous torque with commutation angle

Select *Cyclic synchronous torque mode with commutation angle* in index 0x7010:03 [▶ 211] *Modes of operation* (MDP) or 0x6060:0 [▶ 240]*Modes of operation* (DS402). The *Predefined PDO Assignment: 'Cyclic synchronous torque mode with commutation angle mode (CSTCA)'* should also be selected in the respective process data (see CoE process data [▶ 177] or DS402 process data [▶ 183]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 208] *Modes of operation display* (MDP) or 0x6061:0 [▶ 240] *Modes of operation display* (DS402) can be used to check the actual mode of the servo terminal.

Via the PLC a defined torque can be set in the *Target torque* variable as a basis for the servo terminal control. In the *Commutation angle* variable the angle to be maintained with the set torque can be specified. The torque is specified in 1000th of the rated current. A value of 1000_{dec}, for example, corresponds to the set index 0x8011:12 [▶ 203] *Rated current* (MDP) or index 0x6075:0 [▶ 241] *Motor rated current* (DS402). The value 1_{dec} corresponds to one 1000th of the rated current.
The angle value must be converted, 65536_{dec} corresponds to 360°.

CSV - cyclic synchronous velocity

Select *Cyclic synchronous velocity* in index 0x7010:03 [▶ 211] *Modes of operation* (MDP) or index 0x6060:0 [▶ 240] *Modes of operation* (DS402). The *Predefined PDO Assignment: 'Cyclic synchronous velocity mode (CSV)'* should also be selected in the respective process data (see CoE process data [▶ 177] or DS402 process data [▶ 183]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 208] *Modes of operation display* (MDP) or 0x6061:0 [▶ 240] *Modes of operation display* (DS402) can be used to check the actual mode of the servo terminal.

Via the PLC a defined speed can be set as control parameter for the servo terminal in the variable *Target velocity* 0x7010:06 [▶ 211] (MDP) or 0x60FF:0 [▶ 244] (DS402). The constant value *Velocity encoder resolution* in CoE object 0x9010:14 [▶ 215] (MDP) or 0x6090:0 [▶ 242] (DS402) corresponds to 1 revolution per second. If this value is entered under *Target velocity*, the motor speed is 1 rev. per sec. The velocity can be increased by entering a suitable multiple of the *Velocity encoder resolution* value under *Target velocity*.

CSP - cyclic synchronous position

Select *Cyclic synchronous position* in index 0x7010:03 [▶ 211] *Modes of operation* (MDP) or index 0x6060:0 [▶ 240] *Modes of operation* (DS402).
Similarly, the *Predefined PDO Assignment: 'Cyclic synchronous position mode (CSP)'* should be selected in the respective process data (see CoE process data [▶ 177] or DS402 process data [▶ 183]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 208] *Modes of operation display* (MDP) or 0x6061:0 [▶ 240] *Modes of operation display* (DS402) can be used to check the actual mode of the servo terminal.

Via the PLC a defined position can be set in the variable *Target position* 0x7010:05 [▶ 211] (MDP) or 0x607A:0 [▶ 242] (DS402) to which the motor is to drive. The calculated scaling factor [▶ 141] is taken as the basis for the calculation of the position. The value entered in the *Target position* variable must be multiplied by the calculated scaling factor.

5.3.7 Settings for the automatic configuration

(Master TwinCAT 2.11 R3)

The EL72x1-xx1x offers the user the possibility to automatically configure the connected motor from the AM81xx series. The electronic type plate integrated in the motor is read and the necessary parameters of the terminal are adapted accordingly.

The automatic configuration is switched off on delivery. The user has the possibility to adapt the automatic configuration according to the flow chart shown below (see fig. *Flow chart for the automatic configuration*).

Overwriting of the parameters during automatic configuration

The parameters manually changed by the user in the parameter list of the automatic configuration are automatically overwritten at the next start-up if automatic configuration is switched on.

- The automatic configuration can be switched on in the index 0x8008:01 [▶ 199] (0x2018:01 [▶ 238], DS402 Profile) *Enable autoconfig*.
- In the index 0x8008:02 [▶ 199] (0x2018:02 [▶ 238], DS402 Profile) *Reconfig identical motor*, the user can decide in the case of replacing an identical motor whether the terminal should automatically re-configure the motor (setting = TRUE) or whether the motor should be operated with the stored settings (setting = FALSE). The deactivation of this function can be advantageous, for example, if the user has specially adjusted the motor to his application and does not want to lose these settings after replacing the motor.
- In the index 0x8008:03 [▶ 199] (0x2018:03 [▶ 238], DS402 Profile) *Reconfig non-identical motor*, the user can decide in the case of replacing a non-identical motor whether the terminal should automatically re-configure the motor (setting = TRUE) or whether the motor should be operated with the stored settings (setting = FALSE).

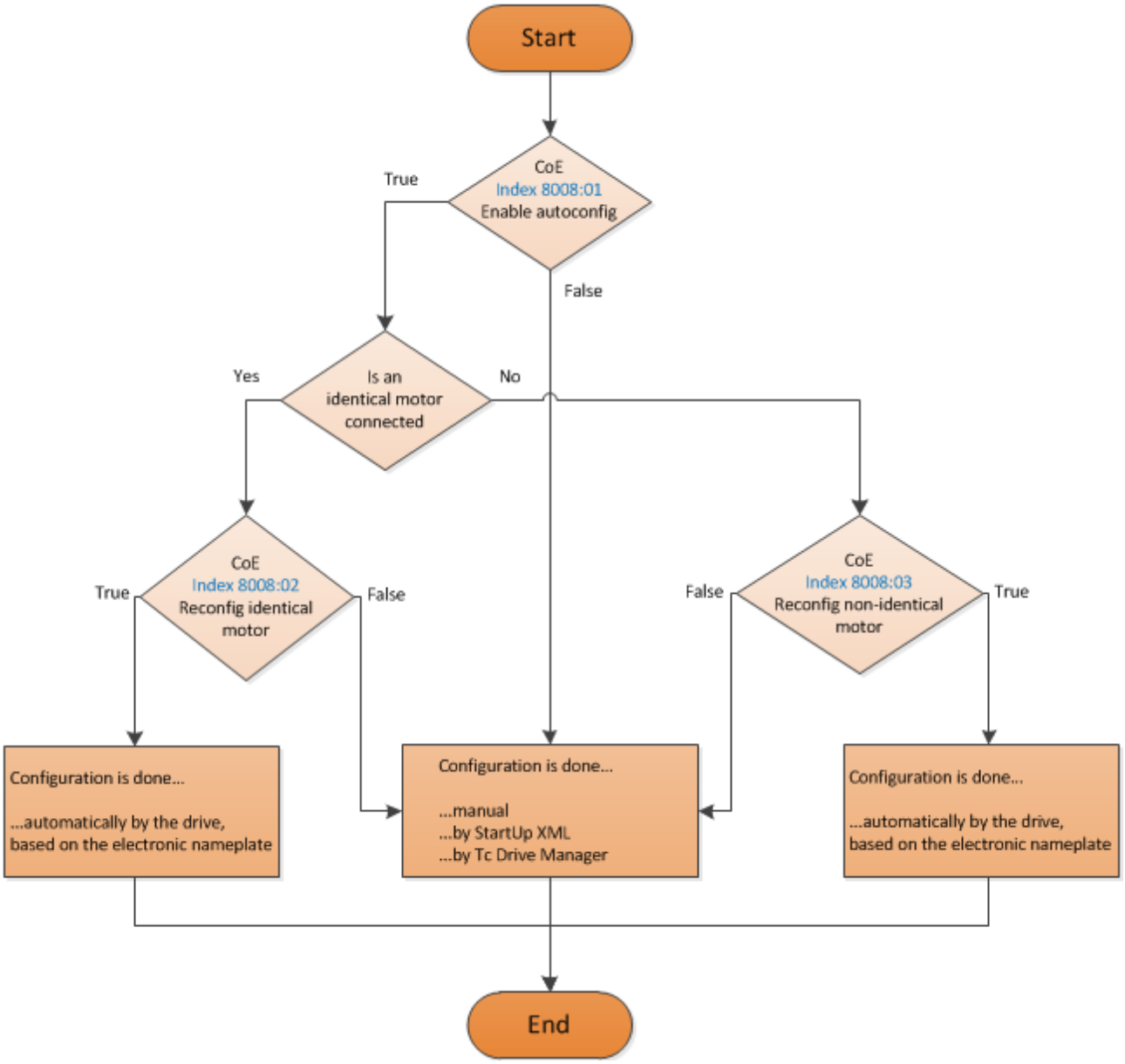


Fig. 175: Flow chart for the automatic configuration

Parameter list of the automatic configuration

The following parameters are affected by the automatic configuration.

Index (hex)		Name	Meaning
MDP 407 Profile	DS402 Profile		
8010:12 [▶ 200]	2002:12 [▶ 233]	Current loop integral time	is calculated according to the symmetrical optimum
8010:13 [▶ 200]	2002:13 [▶ 233]	Current loop proportional gain	is calculated according to the symmetrical optimum
8011:11 [▶ 203]	2003:11 [▶ 236]	Max. current	is adopted directly from the electronic type plate of the connected motor
8011:12 [▶ 203]	2003:12 [▶ 236]	Rated current	is adopted directly from the electronic type plate of the connected motor
8011:13 [▶ 203]	2003:13 [▶ 236]	Motor pole pairs	is adopted directly from the electronic type plate of the connected motor
8011:15 [▶ 203]	2003:15 [▶ 236]	Commutation offset	is always set to -90°
8011:16 [▶ 203]	2003:16 [▶ 236]	Torque constant	is adopted directly from the electronic type plate of the connected motor
8011:18 [▶ 203]	2003:18 [▶ 236]	Rotor moment of inertia	is adopted directly from the electronic type plate of the connected motor
8011:19 [▶ 203]	2003:19 [▶ 236]	Winding inductance	is adopted directly from the electronic type plate of the connected motor
8011:1B [▶ 203]	2003:1B [▶ 236]	Motor speed limitation	Calculation of the max. speed of the connected motor
8011:2B [▶ 203]	2003:2B [▶ 236]	Motor temperature warn level	is adopted directly from the electronic type plate of the connected motor
8011:2C [▶ 203]	2003:2C [▶ 236]	Motor temperature error level	is adopted directly from the electronic type plate of the connected motor
8011:2D [▶ 203]	2003:2D [▶ 236]	Motor thermal time constant	is adopted directly from the electronic type plate of the connected motor
8012:11 [▶ 204]	2004:11 [▶ 237]	Release delay	is adopted directly from the electronic type plate of the connected motor
8012:12 [▶ 204]	2004:12 [▶ 237]	Application delay	is adopted directly from the electronic type plate of the connected motor
8012:14 [▶ 204]	2004:14 [▶ 237]	Brake moment of inertia	is adopted directly from the electronic type plate of the connected motor

5.3.8 Configuring the limit switch

Software end position monitoring

The TwinCAT NC can be used to set software end position monitoring for the EL72x1-xxxx to ensure the safety of the system. The axis does not move beyond the set position (maximum/minimum end position). End position monitoring can be activated in the Parameter tab for the corresponding axis.

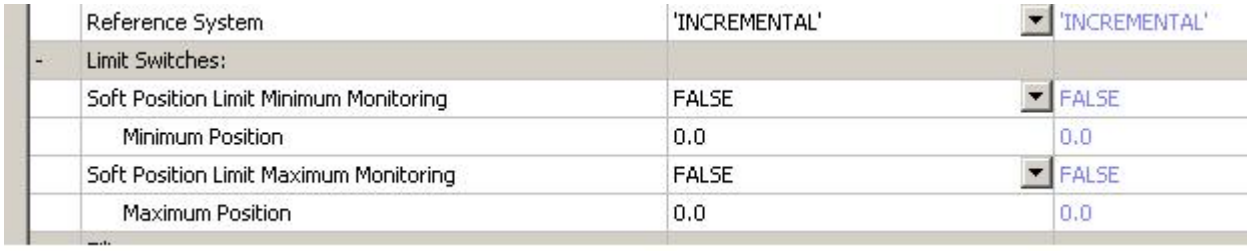


Fig. 176: Pull-down menu for activating end position monitoring

Limit switch

It is not possible to connect a limit switch directly to the terminal for direct evaluation. Alternatively, the limit switch can be read via a digital input terminal, or the software end position monitoring can be used.

5.3.9 Homing

(Master TwinCAT 2.11 R3)

The data given here serve as an example for a servomotor type AM8131-0F20-0000 from Beckhoff Automation. For other motors the values may vary, depending on the application.

Table of contents

- Referencing [▶ 156]
  - Function block "MC_Home" [▶ 157]
  - Reference modes [▶ 158]

Referencing

Referencing does not work via the online commissioning tab of the axis (see Fig. *Online homing in the NC*).

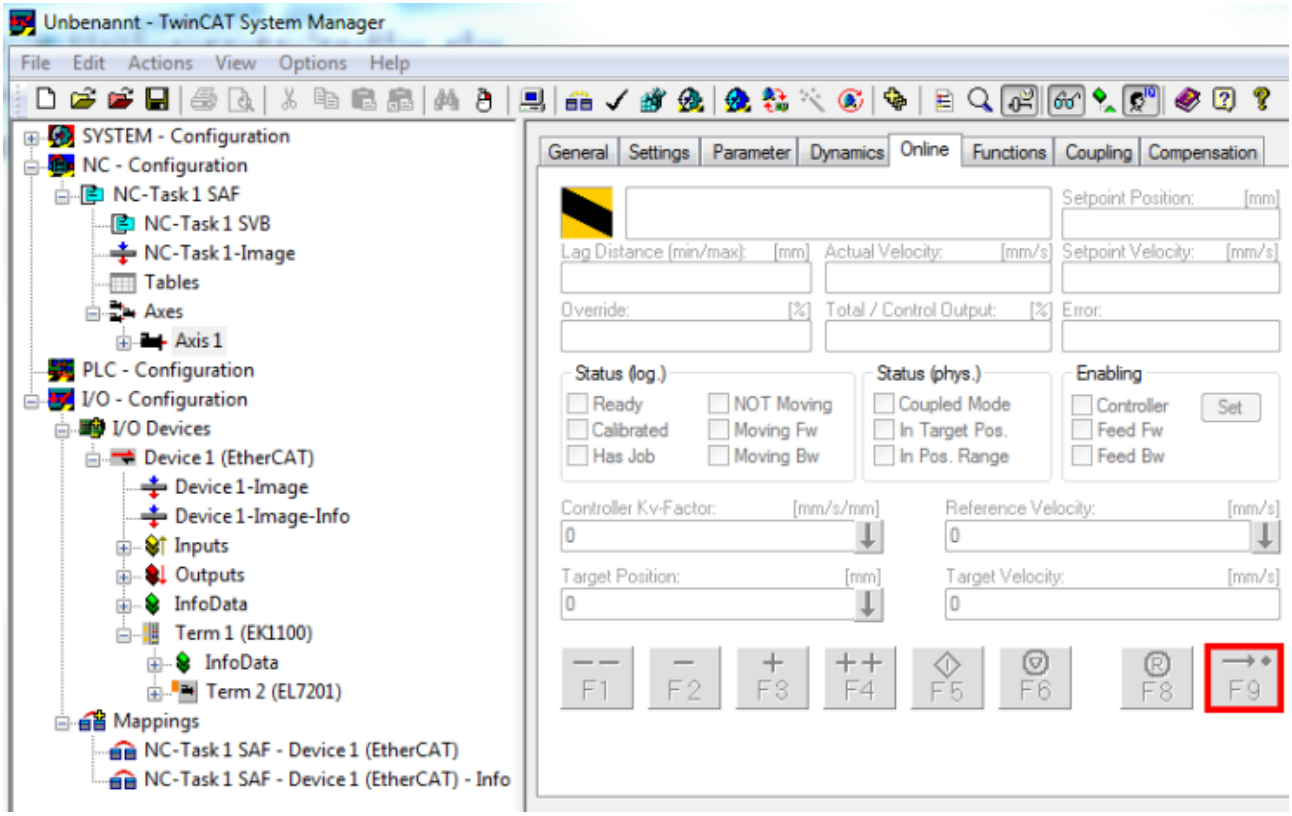


Fig. 177: Online homing in the NC

Function block "MC_Home"

- Referencing must be done from the PLC. The function block *MC_Home* from the TC MC2 Lib is used for this purpose.
- The following minimum configuration is required in *MC_Home*.
  - *HomingMode* enables selection of mode to be used for referencing.
  - *Execute* is used to initiate homing.
  - *bCalibrationCam*, which has to be linked with your reference cam, is used to stop homing.

```
0046 (*Homing*)
0047
0048 MC_Home(
0049     Execute:= bStartHoming,          (*start homing*)
0050     Position:= ,
0051     HomingMode:= MC_DefaultHoming,   (*execute a standard homing*)
0052     BufferMode:= ,
0053     Options:= ,
0054     bCalibrationCam:= bReferenceStop,  (*reference cam*)
0055     Axis:= axis1,
0056     Done=> ,
0057     Busy=> ,
0058     Active=> ,
0059     CommandAborted=> ,
0060     Error=> ,
0061     ErrorID=> );
0062
```

Fig. 178: Configuration of the MC_Home block

- The following figure *Extract from the functional description for MC_Home* shows an extract from the functional description of *MC_Home*. Full information can be found in the corresponding functional description.

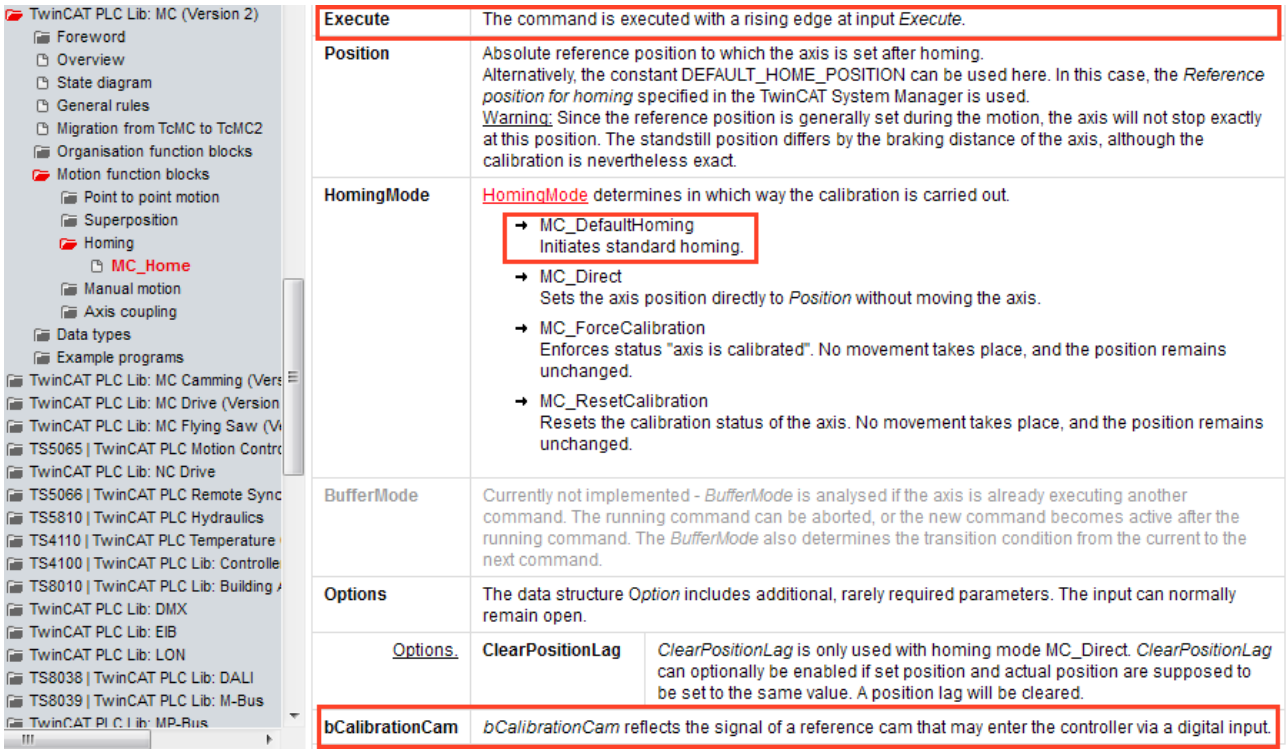


Fig. 179: Extraction from the functional description for MC_Home

Reference modes

- The EL72x1-xxxx can be operated with the following NC reference modes (see Fig. *Selection of the reference modes in the NC*).
- **Default:** Is suitable as a general setting and for most applications.
  Once the motor reaches the reference cam, the direction is reversed. The declining cam signal causes the motor to stop. The reference position is then set.
- **Software Sync:** The C track is modelled virtually.

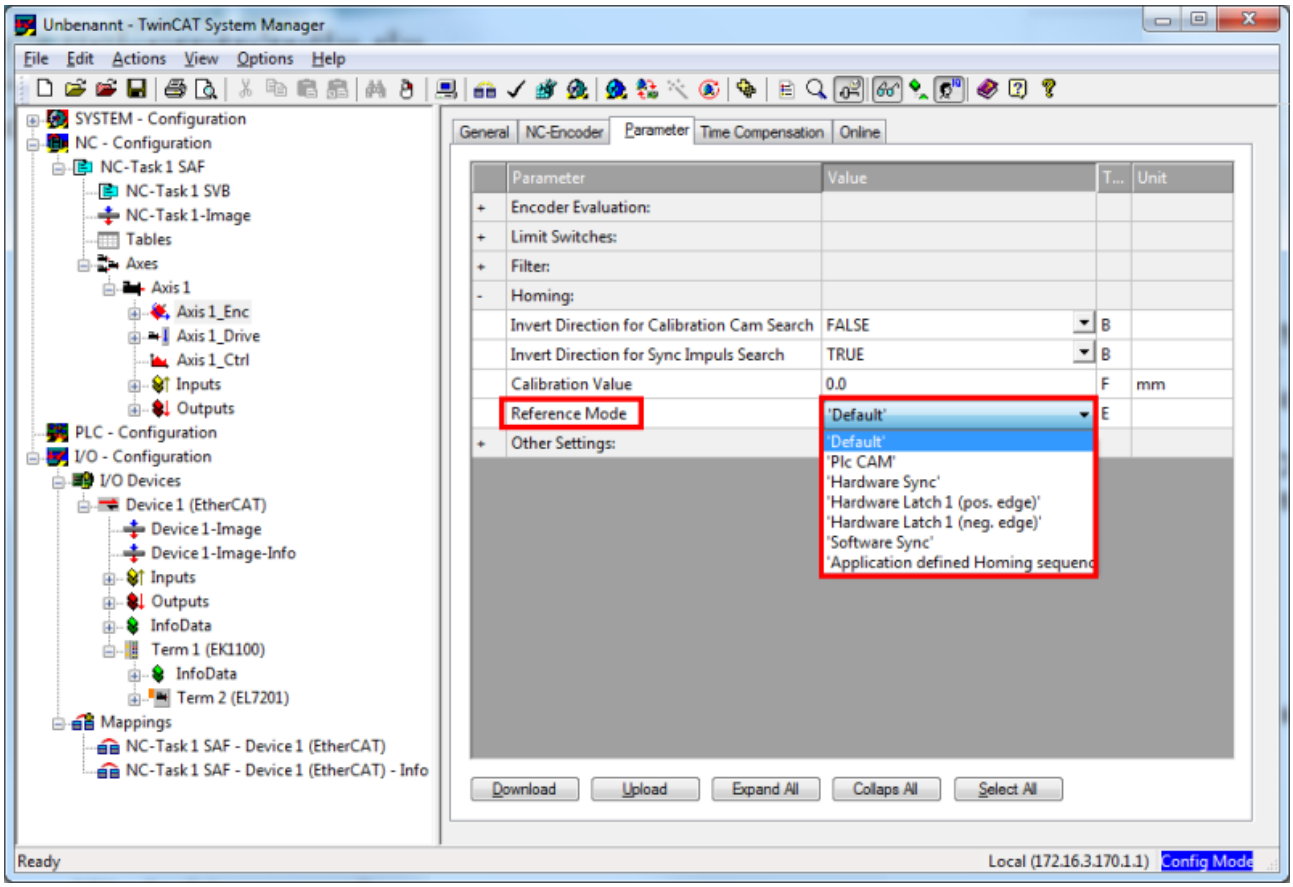


Fig. 180: Selection of the reference modes in the NC

The velocity to be used for homing can also be set in the NC (Fig. *Setting the reference velocity*).

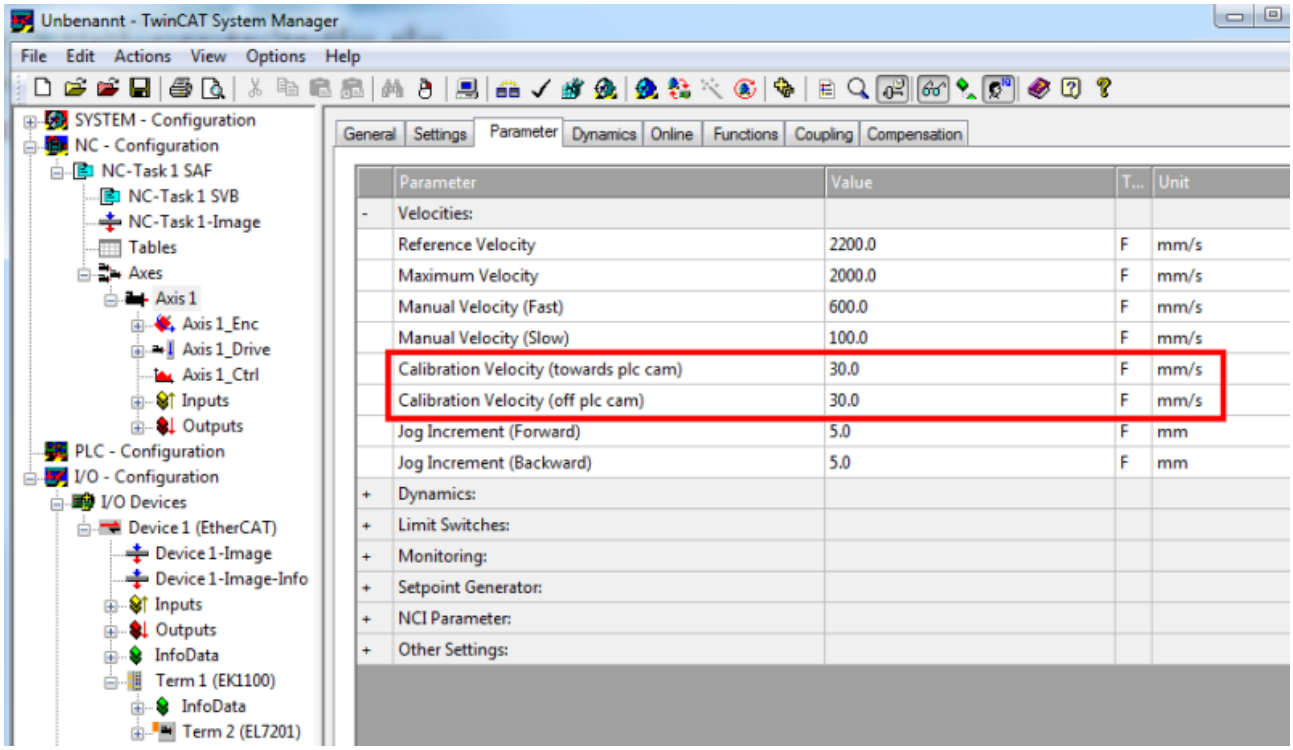


Fig. 181: Setting the reference velocity

5.3.10 Touch Probe

(Master TwinCAT 2.11 R3)

The data given here serve as an example for a servomotor type AM8131-0F20-0000 from Beckhoff Automation. For other motors the values may vary, depending on the application.

Functional description

The *Touch Probe* function implemented in the EL72x1-xxxx provides the user with the possibility to save the current position of the connected motor at a defined point in time.

The required inputs and outputs can be added in the Process data [▶ 177] tab (see Fig. *Touch Probe inputs* and Fig. *Touch Probe outputs*).

The EL72x1-xxxx has 2 digital inputs that can be used for the Touch Probe function. The abbreviation TP1 stands for Touch Probe 1 and is linked to input 1 of the terminal (connection pin 3), while the abbreviation TP2 stands for Touch Probe 2 and is linked to input 2 of the terminal (connection pin 11). TP1 is used here as an example for the description of the function.

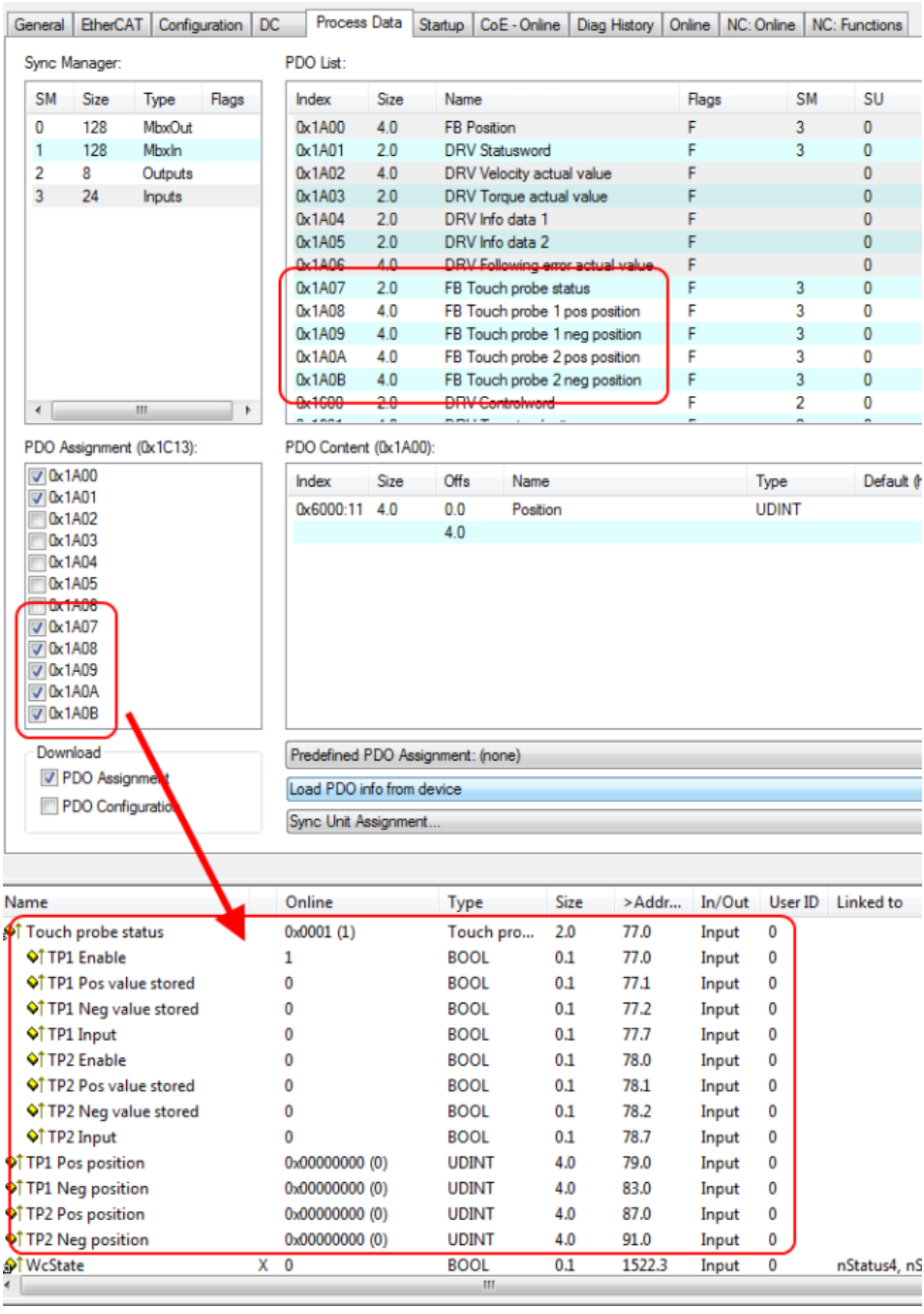


Fig. 182: Touch Probe inputs

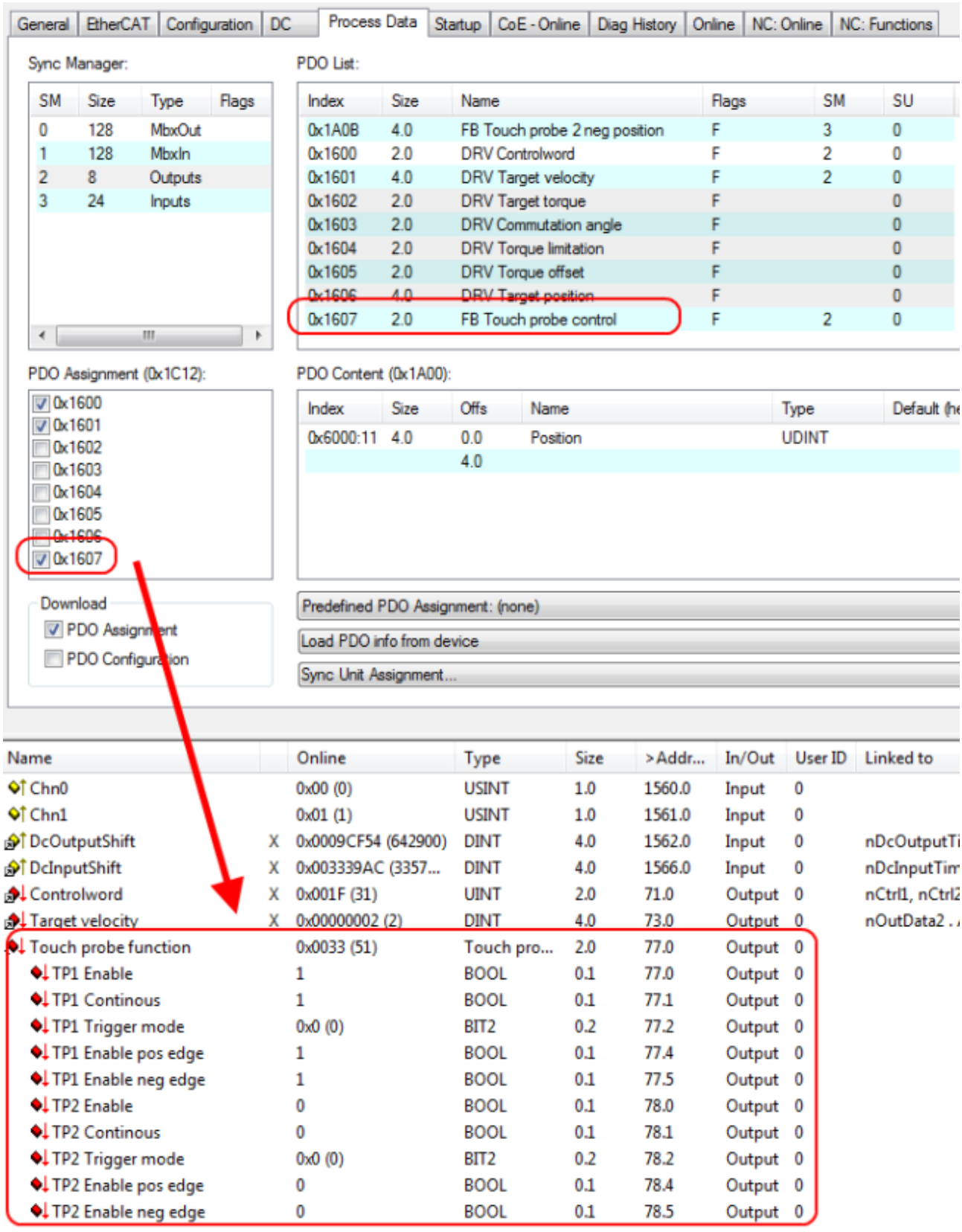


Fig. 183: Touch Probe outputs

Step-by-step

- *TP1 Enable* must be set to TRUE in order to generally activate the Touch Probe function.
- Subsequently, you must decide whether the position is to be saved on a positive edge at input 1 (*TP1 Enable pos edge* = TRUE), on a negative edge (*TP1 Enable neg edge* = TRUE), or in both cases (both set to TRUE).
- With *TP1 Continuous* you can decide whether the position is to be saved only at the first event (*TP1 Continuous* = FALSE) or whether this should take place at every event (*TP1 Continuous* = TRUE).
  For example, if *TP1 Continous* and *TP1 Enablepos edge* are set, the position is saved on each positive edge at input 1 of the terminal.
  If *TP1 Enable neg edge* is set and *TP1 Continuous* is not set, the position will only be saved on the first negative edge at input 1 of the terminal. If you wish to repeat this procedure, you must first deactivate *TP1 Enable* and then activate it again. Then the position is saved again on the first negative edge.
- *TP1 Trigger mode* has no function in the case of the EL72x1-xxxx.
- The saved position of the positive edge can be read in the inputs of the process data under *TP1 Pos position*, that of the negative edge under *TP1 Neg position*.
- The variables under *Touch probe status* are for the diagnosis.
- The Touch Probe inputs must be addressed with a 1-wire +24 V signal.

5.4 Operation modes

5.4.1 Overview

Operating modes CST, CSTCA, CSV and CSP are supported. The operating mode is set in the CoE list in index 0x7010:03 [▶ 211] Modes of operation (MDP) or index 0x6060:0 [▶ 240] Modes of operation (DS402). In the respective process data the user can additionally select the respective *Predefined PDO Assignment*. All required variables are then in the process data.

CSV [▶ 163] - cyclic synchronous velocity (velocity control)

In CSV mode the EL72x1-xxxx operates with the cyclic velocity interface. A defined velocity can be set via the *Target velocity* variable.

CST [▶ 166] - cyclic synchronous torque (torque control)

In CST mode the EL72x1-xxxx operates in the cyclic torque interface. A defined torque can be set via the *Target torque* variable.

CSTCA [▶ 169] - cyclic synchronous torque with commutation angle (torque control with commutation angle)

This operating mode is also intended for use with the cyclic torque interface. In addition the user can specify the commutation angle. The variable *Commutation angle* can be used to set an angle which is to be maintained with a defined torque set in variable *Target torque*.

CSP [▶ 172] - cyclic synchronous position (position control)

In the CSP operating mode the EL72x1-xxxx operates in the cyclic position interface. A defined position can be set via the *Target position* variable.

For further information on the three operating modes described above please refer to section “Commissioning without NC”.

5.4.2 CSV

CSV - cyclic synchronous velocity (velocity control)

In CSV mode the EL72x1-xxxx operates with the cyclic velocity interface. A defined velocity can be set via the *Target velocity* variable.

Step-by-Step

- Add the terminal to the configuration as described in the chapter TwinCAT configuration settings [▶ 102] - manual or - Online scan [▶ 108].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 126].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 135].
- Set the mode of operation in the CoE directory to *Cyclic synchronous velocity mode (CSV), Fig. Selection of the mode of operation*.

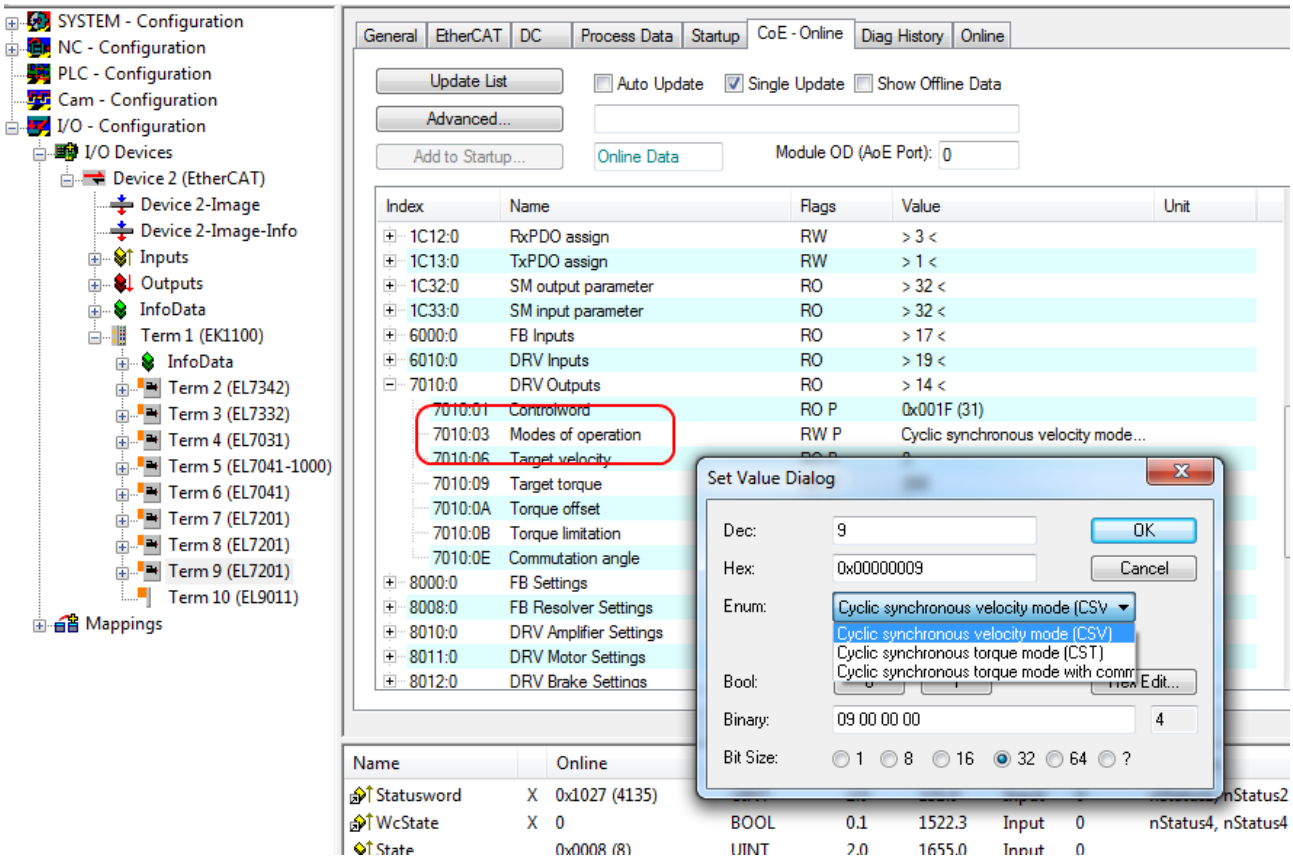


Fig. 184: Selection of the mode of operation

- Under Predefined PDO assignment, also select *Cyclic synchronous velocity mode (CSV)*, Fig. *Selecting a predefined PDO assignment*.

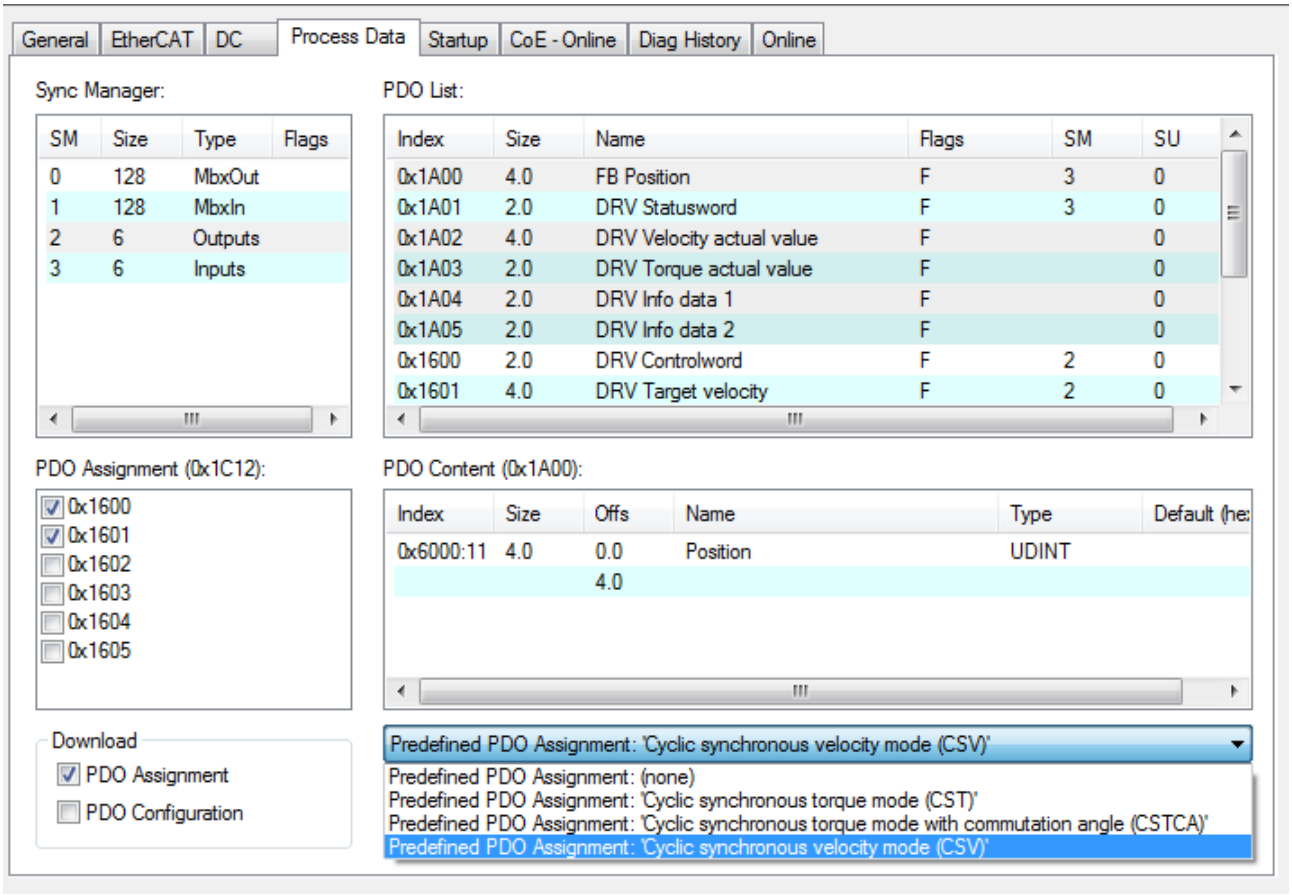


Fig. 185: Selecting a predefined PDO assignment

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. There are two ways to do this:
  - If you use the TwinCAT NC.
    The State Machine is run through automatically by the NC. You can enable the axis in the *Online* tab of the axis.
    Set all tick marks and set *Override* to 100% (see Fig. *Set enables*). The axis can then be moved.

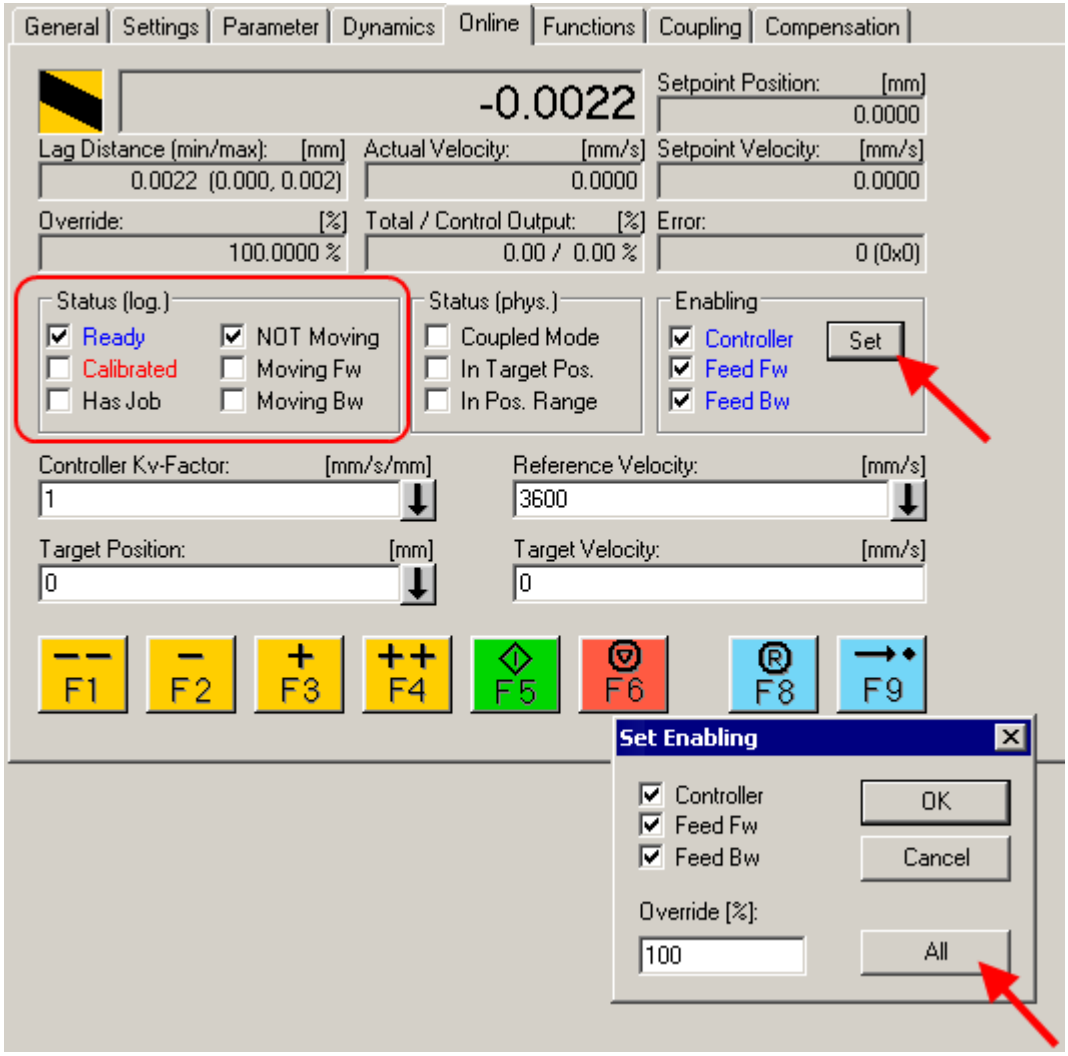


Fig. 186: Set enables

◦ If you don’t use the TwinCAT NC.
In this case you must run through the State Machine manually. To do this, follow the instructions in the chapter Commissioning without the NC [▶ 150].

- The cyclic variable *Target velocity* (Fig. *Torque specification*) can be used to specify a defined velocity. The value in the index 0x9010:14 (0x6090:0, DS402)*Velocity encoder resolution* corresponds to 1 rpm.

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Position	X	0x00000000 (0)	UDINT	4.0	132.0	Input	0	nInData1 . Axis 10_Enc_I...
Statusword	X	0x0000 (0)	UINT	2.0	136.0	Input	0	nStatus1, nStatus2
WcState	X	1	BOOL	0.1	1522.3	Input	0	nStatus4, nStatus4
State		0x0042 (66)	UINT	2.0	1655.0	Input	0	
AdsAddr		AC 11 28 29 03 01 ...	AMSADDR...	8.0	1657.0	Input	0	
Chn0		0x00 (0)	USINT	1.0	1665.0	Input	0	
Chn1		0x01 (1)	USINT	1.0	1666.0	Input	0	
DcOutputShift	X	0x0009E854 (649300)	DINT	4.0	1667.0	Input	0	nDcOutputTime . Axis 1...
DcInputShift	X	0x003320AC (3350...	DINT	4.0	1671.0	Input	0	nDcInputTime . Axis 10_...
Controlword	X	0x0006 (6)	UINT	2.0	132.0	Output	0	nCtrl1, nCtrl2
Target velocity		0x00000000 (0)	DINT	4.0	134.0	Output	0	

Fig. 187: Torque specification

5.4.3 CST

CST - cyclic synchronous torque (torque control)

In CST mode the EL72x1-xxxx operates in the cyclic torque interface. A defined torque can be set via the *Target torque* variable.

Step-by-Step

- Add the terminal to the configuration as described in the chapter TwinCAT configuration settings [▶ 102] - manual or - Online scan [▶ 108].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 126].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 135].
- Set the mode of operation in the CoE directory to *Cyclic synchronous torque mode (CST)*, Fig. *Selection of the mode of operation*.

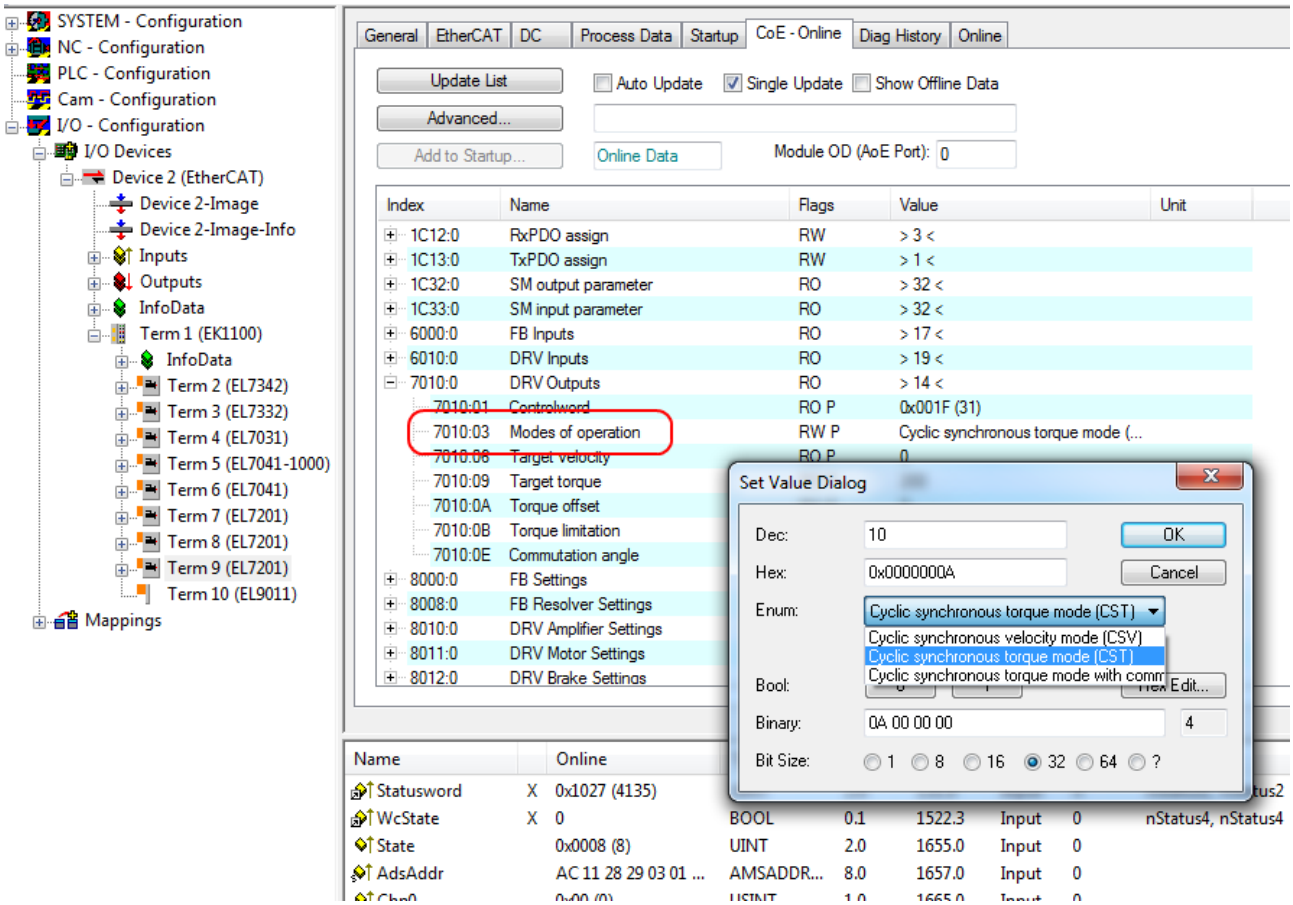


Fig. 188: Selection of the mode of operation

- Under Predefined PDO assignment, also select *Cyclic synchronous torque mode (CST)*, Fig. *Selecting a predefined PDO assignment*.

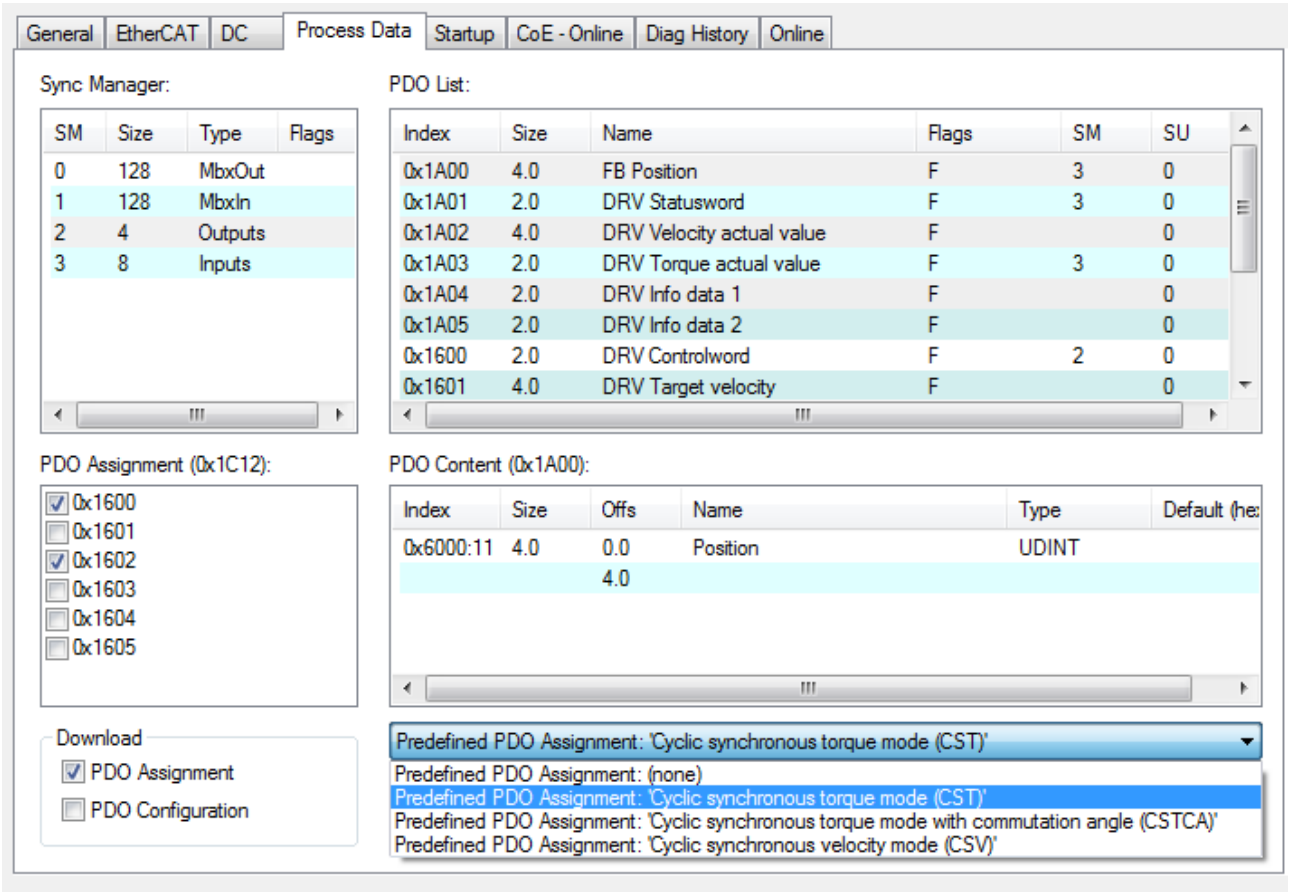


Fig. 189: Selecting a predefined PDO assignment

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. There are two ways to do this:
  - If you use the TwinCAT NC.
    The State Machine is run through automatically by the NC. You can enable the axis in the *Online* tab of the axis.
    Set all tick marks and set *Override* to 100% (see Fig. *Set enables*). The axis can then be moved.

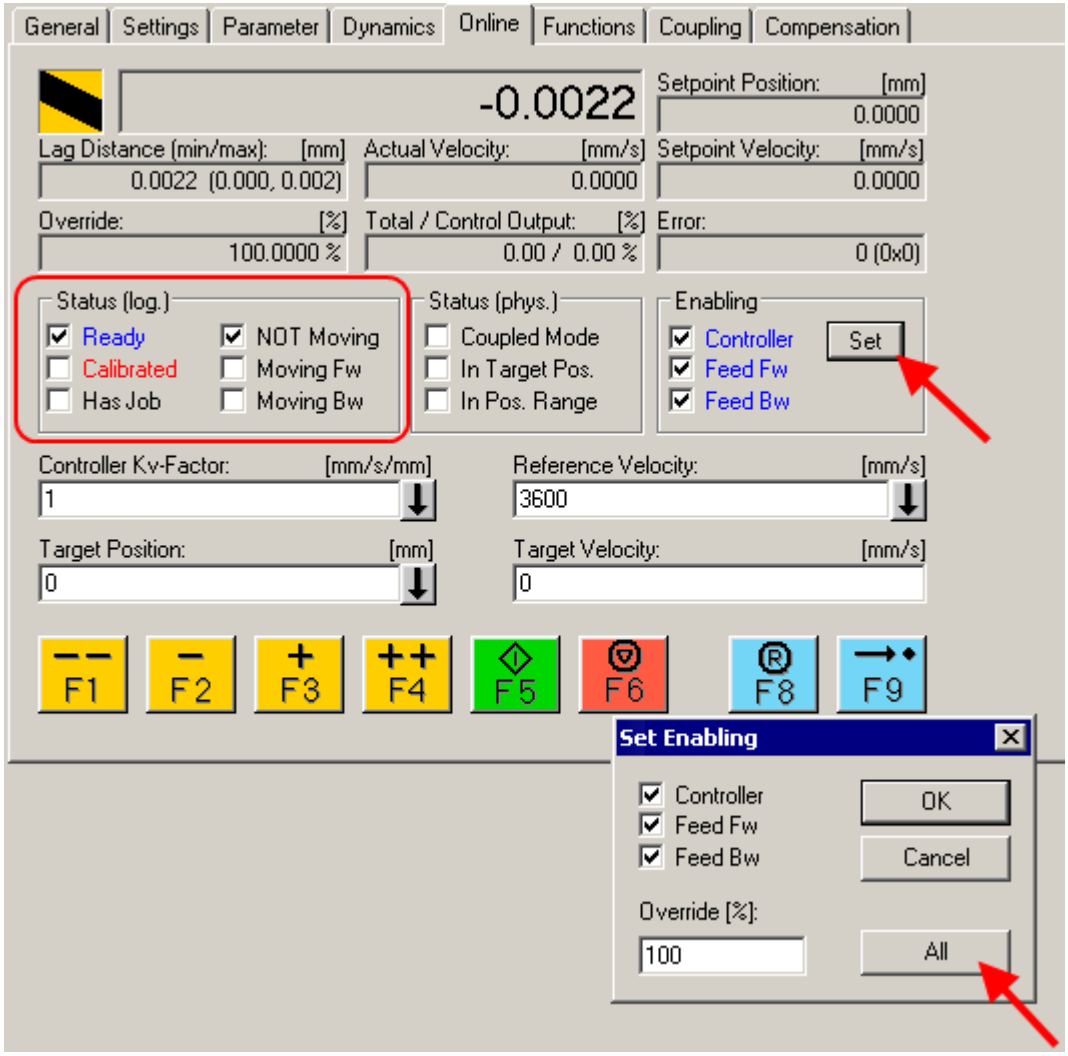


Fig. 190: Set enables

◦ If you don't use the TwinCAT NC.
In this case you must run through the State Machine manually. To do this, follow the instructions in the chapter Commissioning without the NC [▶ 150].

- The cyclic variable *Target torque* (Fig. *Torque specification*) can be used to specify a defined torque. The value is specified in 1000ths of the *rated current* and the torque is calculated according to the following equation, where the *rated current* refers to the value in the index 0x8011:12 (*rated current*).

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Position	X	0x00000000 (0)	UDINT	4.0	132.0	Input	0	nInData1 . Axis 10_Enc_I...
Statusword	X	0x0000 (0)	UINT	2.0	136.0	Input	0	nStatus1, nStatus2
Torque actual v...		0x0000 (0)	INT	2.0	138.0	Input	0	
WcState	X	1	BOOL	0.1	1522.3	Input	0	nStatus4, nStatus4
State		0x0042 (66)	UINT	2.0	1655.0	Input	0	
AdsAddr		AC 11 28 29 03 01 ...	AMSADDR...	8.0	1657.0	Input	0	
Chn0		0x00 (0)	USINT	1.0	1665.0	Input	0	
Chn1		0x01 (1)	USINT	1.0	1666.0	Input	0	
DcOutputShift	X	0x0009E854 (649300)	DINT	4.0	1667.0	Input	0	nDcOutputTime . Axis 1...
DcInputShift	X	0x003320AC (3350...	DINT	4.0	1671.0	Input	0	nDcInputTime . Axis 10_...
Controlword	X	0x0006 (6)	UINT	2.0	132.0	Output	0	nCtrl1, nCtrl2
Target torque		0x0000 (0)	INT	2.0	134.0	Output	0	

Fig. 191: Torque specification

5.4.4 CSTCA

CSTCA - cyclic synchronous torque with commutation angle (torque control with commutation angle)

This mode of operation is also intended for use with the cyclic torque interface. In addition the user can specify the commutation angle. The variable *Commutation angle* can be used to set an angle which is to be maintained with a defined torque set in variable *Target torque*.

Step-by-Step

- Add the terminal to the configuration as described in the chapter TwinCAT configuration settings [▶ 102] - manual or - Online scan [▶ 108].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 126].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 135].
- Set the mode of operation in the CoE directory to *Cyclic synchronous torque mode with commutation angle (CSTCA)*, Fig. *Selection of the mode of operation*

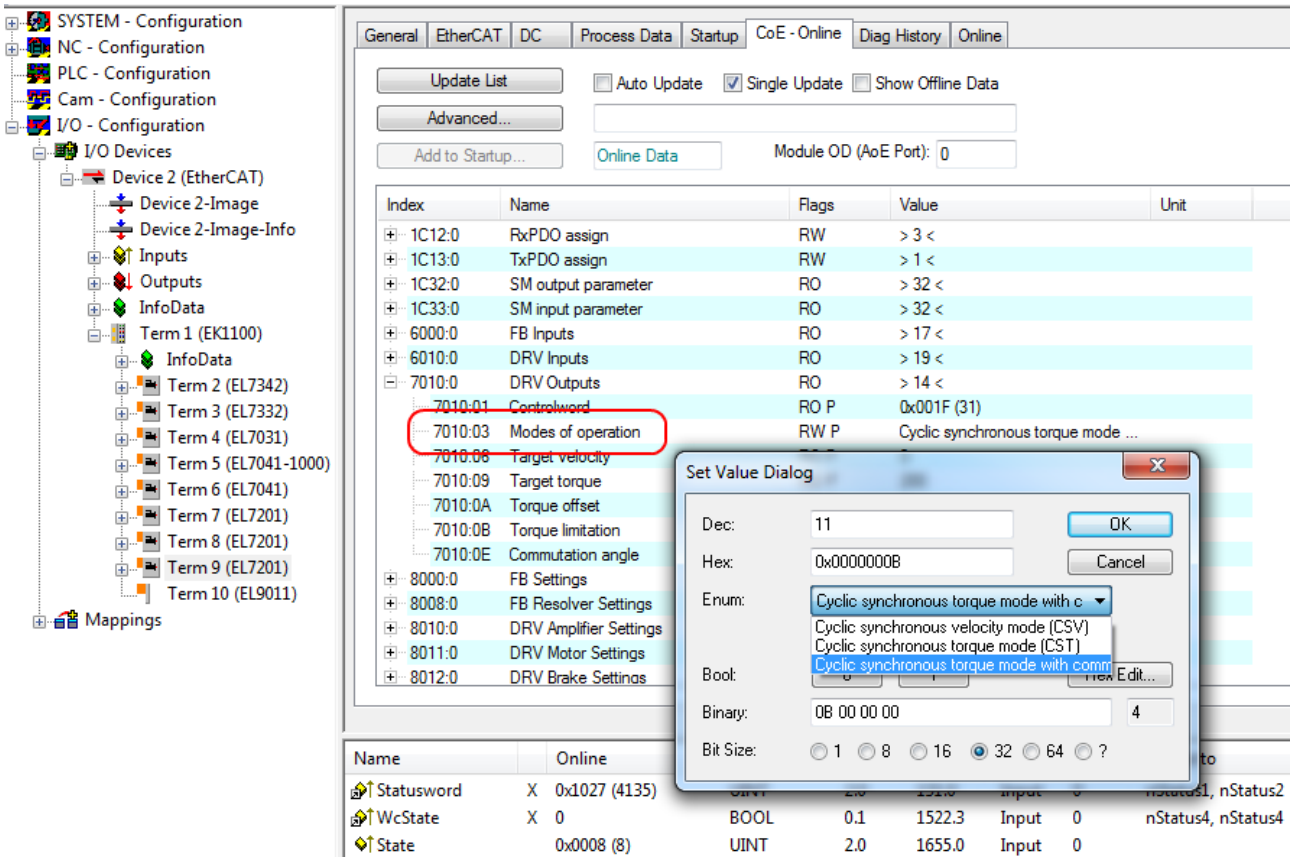


Fig. 192: Selection of the mode of operation

- Under Predefined PDO assignment, also select *Cyclic synchronous torque mode with commutation angle (CSTCA)*, Fig. *Selecting a predefined PDO assignment*.

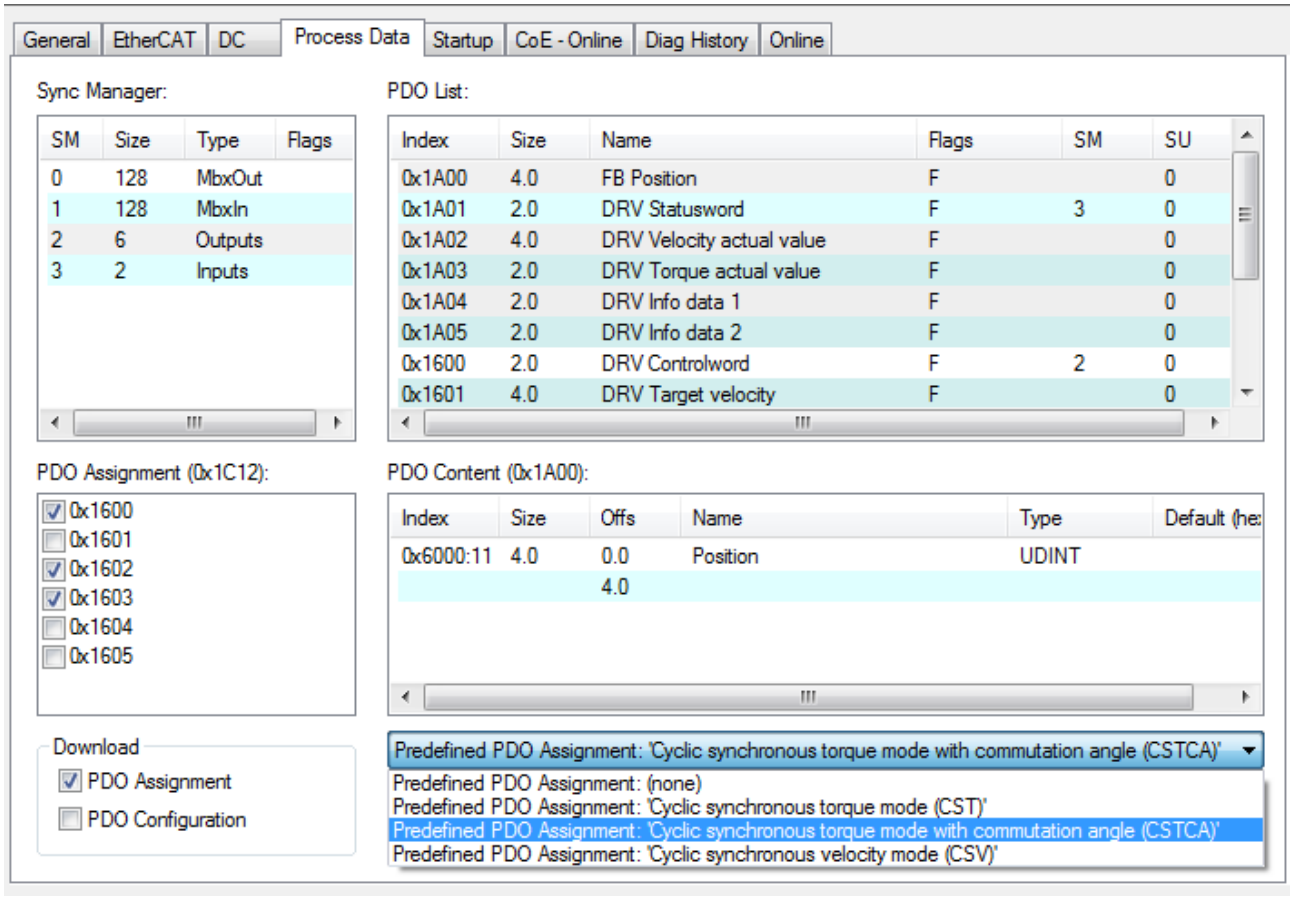


Fig. 193: Selecting a predefined PDO assignment

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. There are two ways to do this:
  - If you use the TwinCAT NC.
    The State Machine is run through automatically by the NC. You can enable the axis in the *Online* tab of the axis.
    Set all tick marks and set *Override* to 100% (see Fig. *Set enables*). The axis can then be moved.

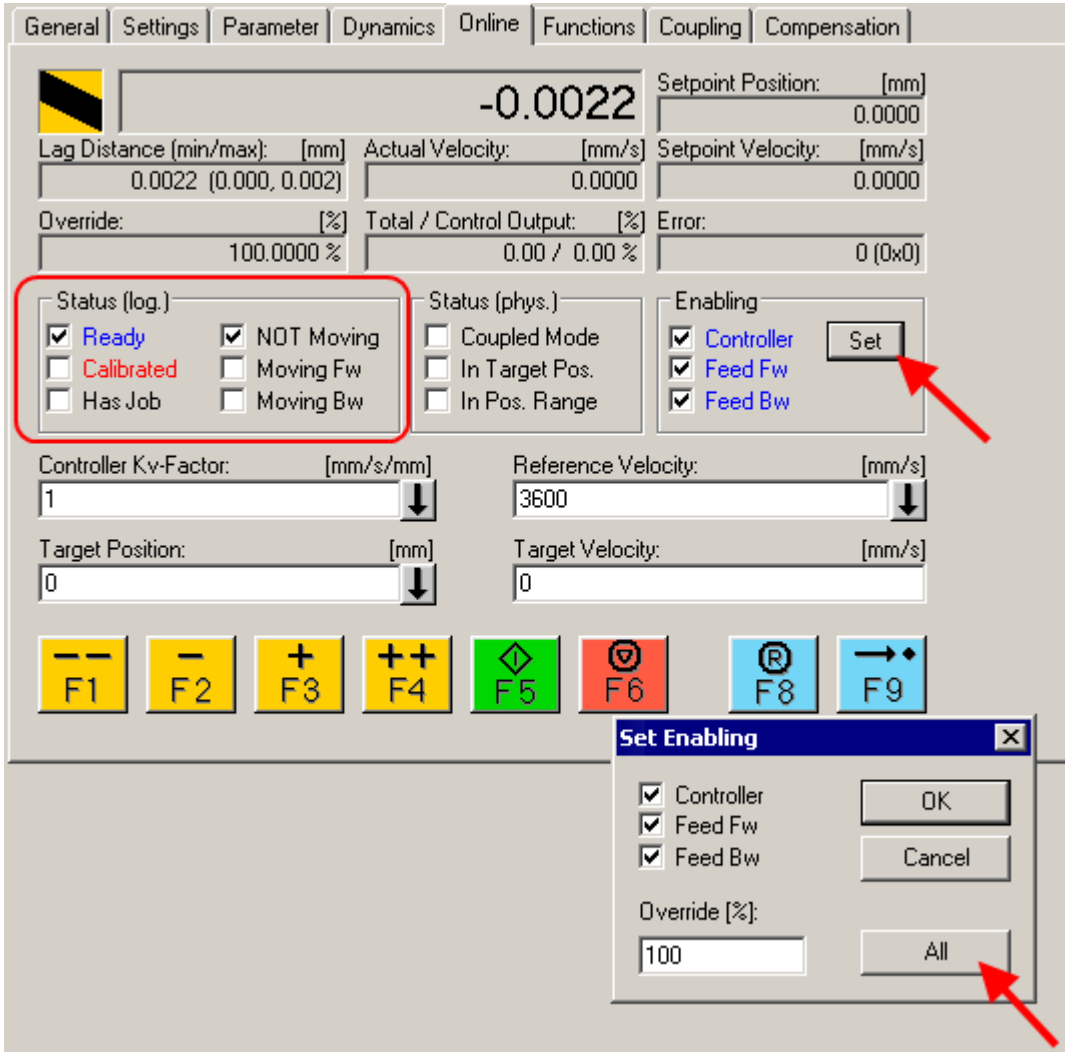


Fig. 194: Set enables

- If you don't use the TwinCAT NC.
  In this case you must run through the State Machine manually. To do this, follow the instructions in the chapter Commissioning without the NC [▶ 150].

- You can specify a defined torque via the cyclic variable *Target torque*. The value is specified in 1000ths of the *rated current* and the torque is calculated according to the following equation, where the rated current refers to the value in the index 0x8011:12 (*rated current*).

You can specify a defined angle via the cyclic variable *Commutation angle*. The value is specified in $360°/2^{16}$.

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Statusword	X	0x0000 (0)	UINT	2.0	132.0	Input	0	nStatus1, nStatus2
WcState	X	1	BOOL	0.1	1522.3	Input	0	nStatus4, nStatus4
State		0x0042 (66)	UINT	2.0	1655.0	Input	0	
AdsAddr		AC 11 28 29 03 01 ...	AMSADDR...	8.0	1657.0	Input	0	
Chn0		0x00 (0)	USINT	1.0	1665.0	Input	0	
Chn1		0x01 (1)	USINT	1.0	1666.0	Input	0	
DcOutputShift	X	0x0009E854 (649300)	DINT	4.0	1667.0	Input	0	nDcOutputTime . Axis 1...
DcInputShift	X	0x003320AC (3350...	DINT	4.0	1671.0	Input	0	nDcInputTime . Axis 10_...
Controlword	X	0x0006 (6)	UINT	2.0	132.0	Output	0	nCtrl1, nCtrl2
Target torque		0x0000 (0)	INT	2.0	134.0	Output	0	
Commutation angle		0x0000 (0)	UINT	2.0	136.0	Output	0	

Fig. 195: Specification of torque and commutation angle

5.4.5 CSP

CSP - cyclic synchronous position (position control)

In the CSP operating mode the EL72x1-xxxx operates in the cyclic position interface. A defined position can be set via the *Target position* variable.

> **Minimum cycle time**
>
> The cycle times in CSP mode with 2^n * 125 µs (for n =1 to 8) are:
> 250 µs, 500 µs, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms or 32 ms.

Step-by-Step

- Add the terminal to the configuration as described in the chapter TwinCAT configuration settings [▶ 102] - manual or - Online scan [▶ 108].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 126].
- Configure the motor with the help of the automatic configuration [▶ 153] (only OCT types), using the Drive Manager [▶ 129] or import the motor XML file into the Start-up directory as described in the chapter Settings in the CoE [▶ 135].
- Set the mode of operation in the CoE directory to *Cyclic synchronous position mode (CSP), Fig. Selection of the mode of operation*.

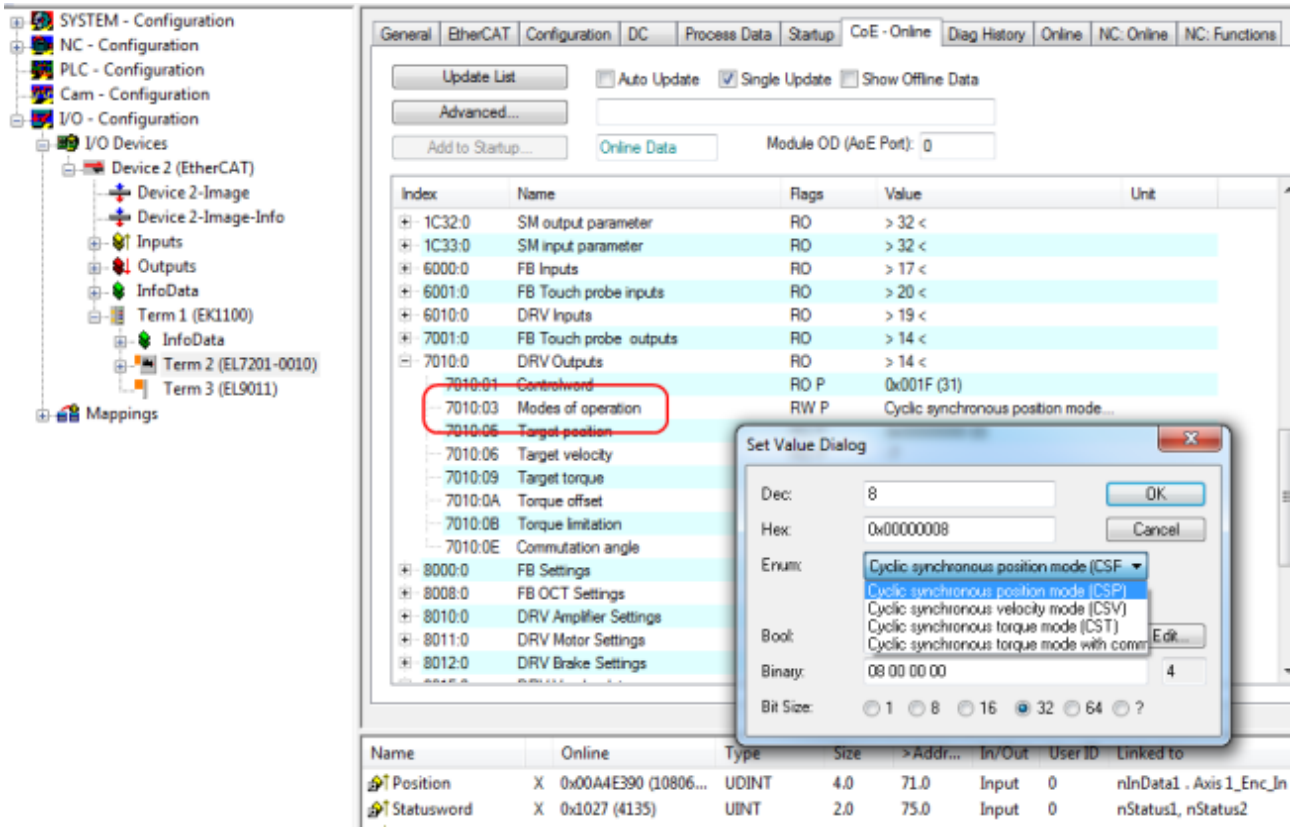


Fig. 196: Selection of the mode of operation

- Under Predefined PDO assignment, also select *Cyclic synchronous position mode (CSP)*, Fig. *Selecting a predefined PDO assignment*.

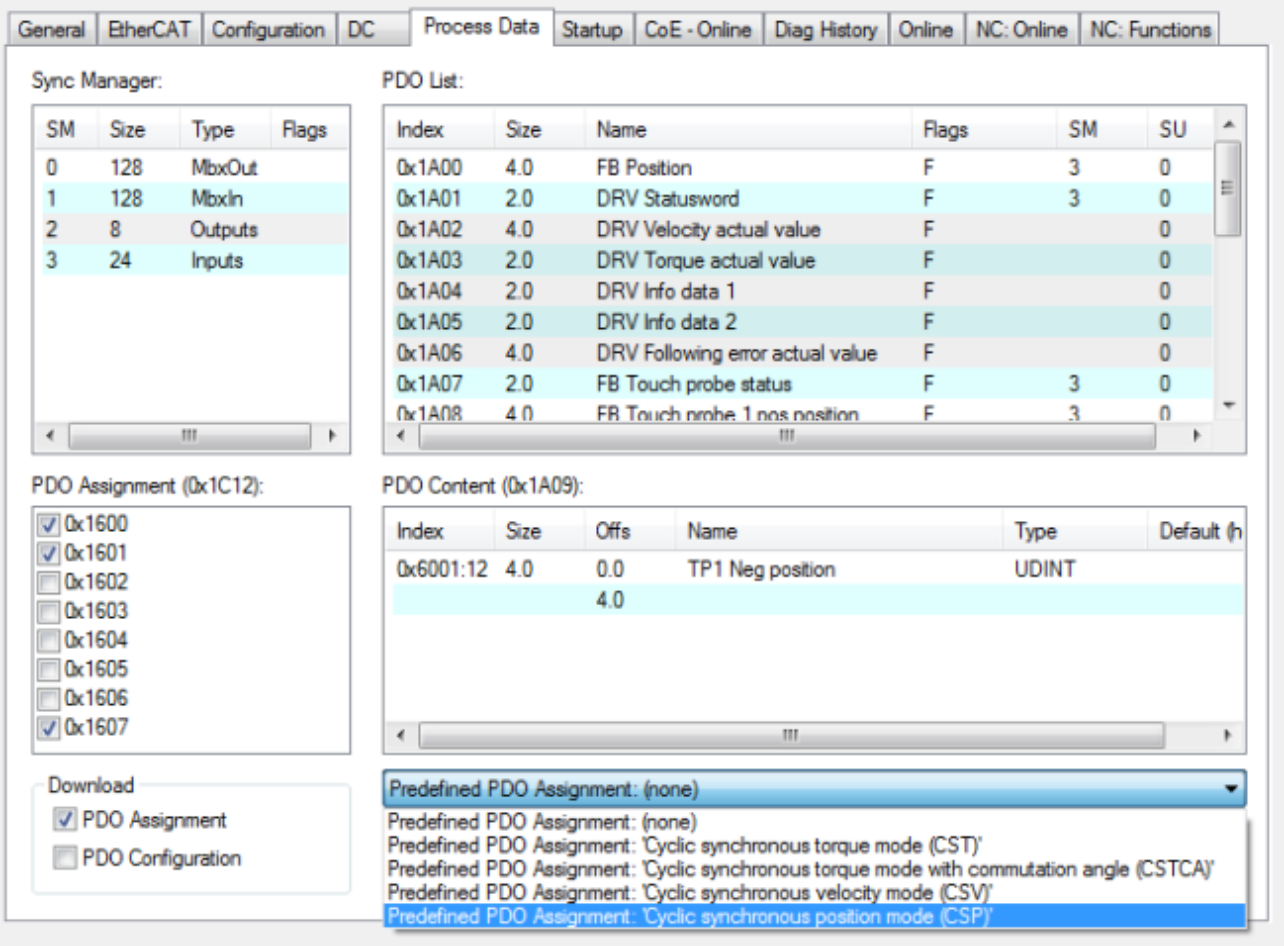


Fig. 197: Selecting a predefined PDO assignment

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. There are two ways to do this:
  - If you use the TwinCAT NC.
    The State Machine is run through automatically by the NC. You can enable the axis in the “Online” tab of the axis.
    Set all tick marks and set Override to 100% (see Fig. *Set enables*). The axis can then be moved.

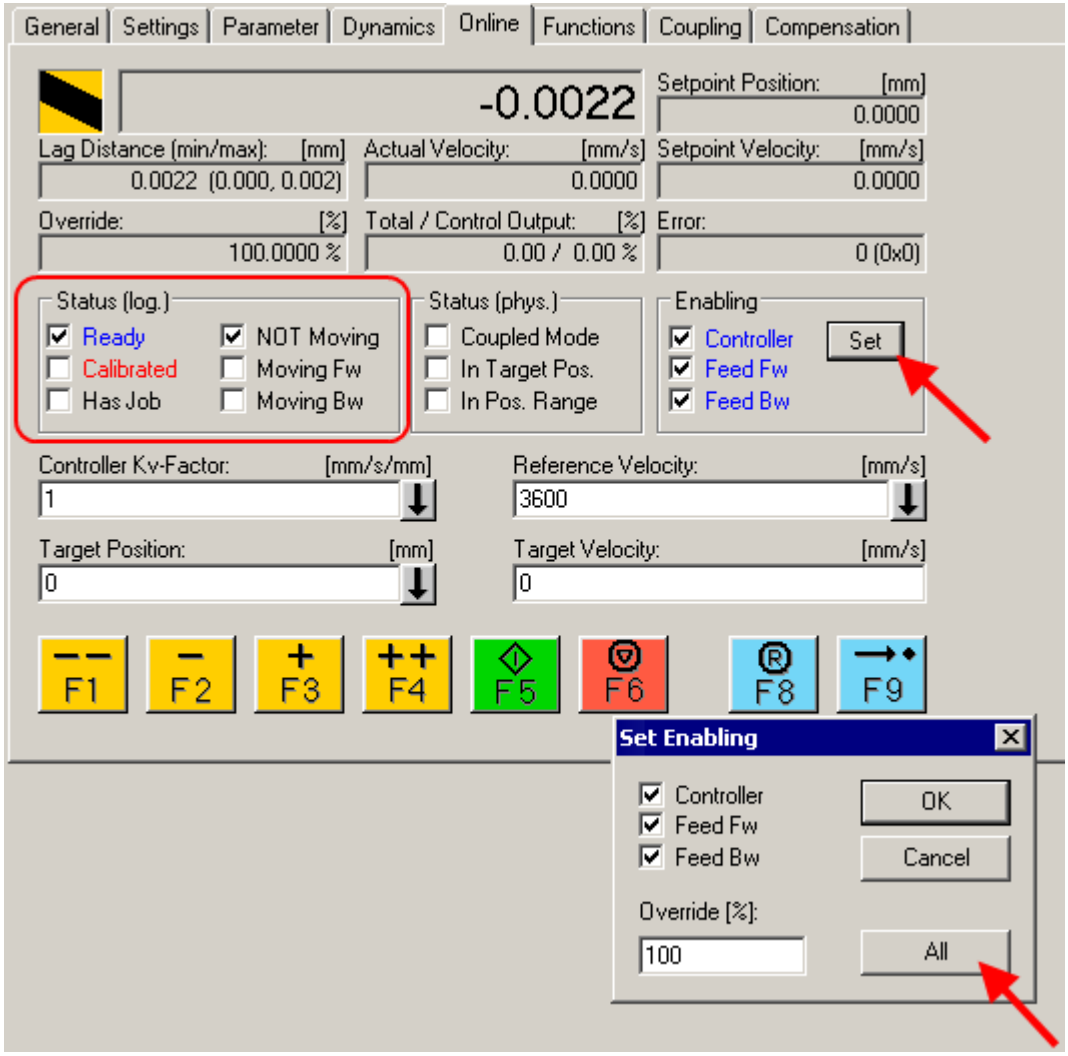


Fig. 198: Set enables

- If you don't use the TwinCAT NC.
  In this case you must run through the State Machine manually. To do this, follow the instructions in the chapter Commissioning without the NC [▶ 150].

- You can specify a defined position via the cyclic variable *Target position* (fig. *Position specification*).
  The value must be multiplied by the calculated scaling factor in order to obtain the correct position.

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Position	X	0x00A4BB64 (10795876)	UDINT	4.0	71.0	Input	0	nInData1 . Axis 1
Statusword	X	0x0021 (33)	UINT	2.0	75.0	Input	0	nStatus1, nStatu
WcState	X	0	BOOL	0.1	1522.3	Input	0	nStatus4, nStatu
InputToggle	X	1	BOOL	0.1	1524.3	Input	0	nStatus4, nStatu
State		0x0008 (8)	UINT	2.0	1550.0	Input	0	
AdsAddr		AC 11 28 29 03 01 EA 03	AMSADDR...	8.0	1552.0	Input	0	
Chn0		0x00 (0)	USINT	1.0	1560.0	Input	0	
Chn1		0x01 (1)	USINT	1.0	1561.0	Input	0	
DcOutputShift	X	0x0009CB6C (641900)	DINT	4.0	1562.0	Input	0	nDcOutputTime
DcInputShift	X	0x00333D94 (3358100)	DINT	4.0	1566.0	Input	0	nDcInputTime .
Controlword	X	0x0006 (6)	UINT	2.0	71.0	Output	0	nCtrl1, nCtrl2
Target position		0x00000000 (0)	UDINT	4.0	73.0	Output	0	

Fig. 199: Position specification

Following error monitor

Furthermore, there is an option in *CSP* mode to activate a following error monitor. The following error monitor is switched off on delivery. In all other modes this is not used and is ignored.

- The window of the following error monitor can be adjusted with the *Following error window* (Index 0x8010:50 MDP742 / Index 0x6065 DS402). The value set here – multiplied by the scaling factor – specifies by what position the actual position may differ from the set position, positively and negatively. The total accepted tolerance is thus twice as large as the position entered in the *Following error window* (see fig. *Following error window*).

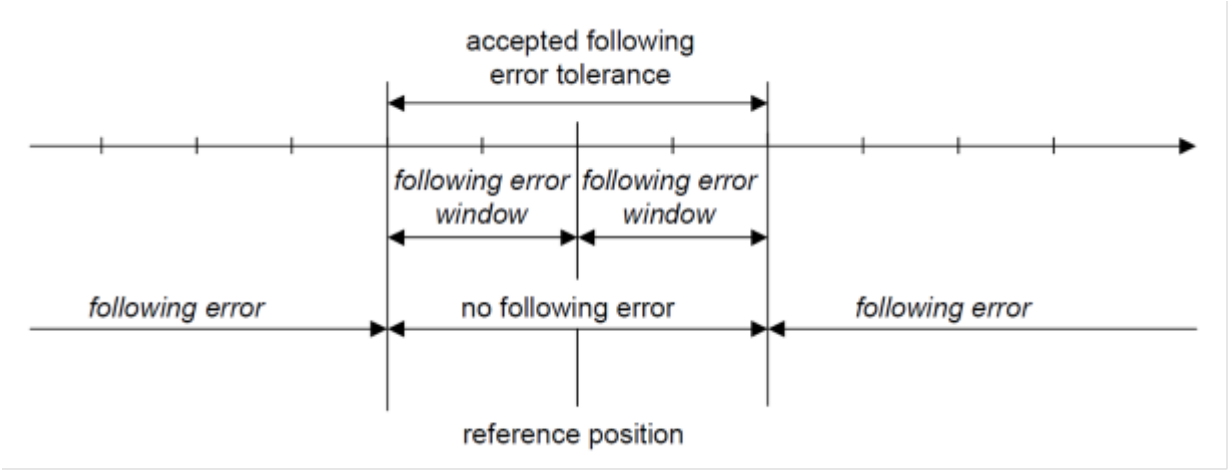


Fig. 200: Following error window

- The time (in ms) allowed for a following error exceedance can be set with the *Following error time out* (Index 0x8010:51 MDP742 / Index 0x6066 DS402). As soon as the target position is exceeded by more than the position entered in the *Following error window* for the time entered in the *Following error time out*, the terminal outputs an error and stops immediately.
- The current following error can be read in the *Following error actual value* (Index 0x6010:09 MDP742 / Index 0x60F4 DS402).

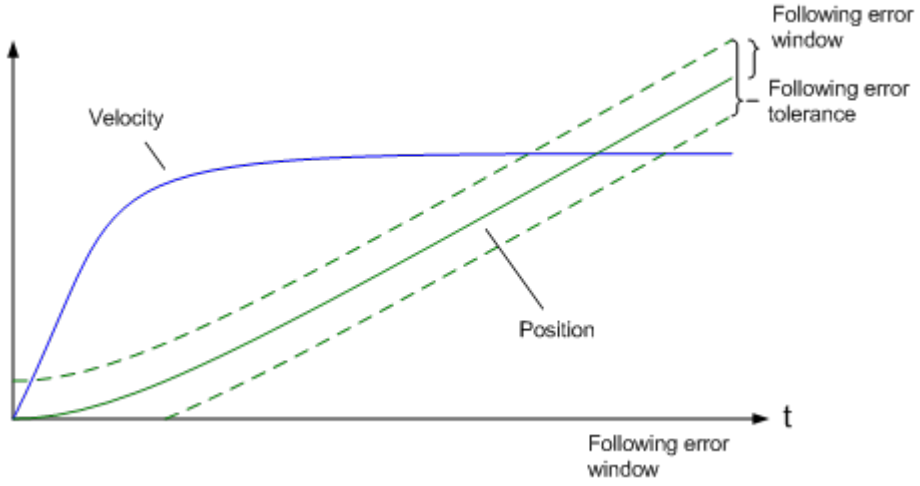


Fig. 201: Following error over time

The value 0xFFFFFF (- 1) in the *Following error window* means that the following error monitor is switched off and corresponds to the delivery status.
The *Following error time out* is 0x0000 (0) on delivery.

5.5 Profile MDP742 or DS402

The EL72x1-x01x supports the MDP742 and DS402 drive profiles. The profiles define the presentation of parameters for the EtherCAT Terminal and the index, under which the respective parameters are arranged in the object directory.

Both profiles contain the same parameters. They only differ in terms of the specified designations and the parameter index. The CoE objects in the MDP742 profile (Modular Device Profile) are allocated in the way that is common for the Beckhoff EtherCAT Terminals. The DS402 drive profile is specified in IEC61800-7-200 (CiA402). It uses a different object directory structure.

In both profiles, the drive state machine of the EL72x1-x01x is based on the CiA402 State Machine [▶ 150], which means the functional behavior is identical.
The terminal is supplied from the factory with the MDP742 profile.

Changing the profile

After a profile change an EEPROM update [▶ 268] is required. The corresponding ESI description [▶ 268] can then be loaded into the terminal.

Please note that the CoE object description and the process data are different for both profiles. The motor XML files that match the set profile must be used.

Terminal designation MDP742 and DS402 profile

Note that the servomotor terminal is given a different type designation in TwinCAT as a result of the profile change process described above.
The terminal then represents itself in the TwinCAT system manager as one of the following devices:
- Servomotor terminal with MDP742 profile: EL72x1-**0010** or EL72x1-**9014**
- Servomotor terminal with DS402 profile: EL72x1-**0011** or EL72x1-**9015**

5.6 MDP742 process data

Table of contents

- Sync Manager [▶ 177]
- PDO Assignment [▶ 178]
- Predefined PDO Assignment [▶ 182]

Sync Manager (SM)

Sync Manager (SM) The extent of the process data that is made available can be changed through the "Process data" tab (see following Fig.).

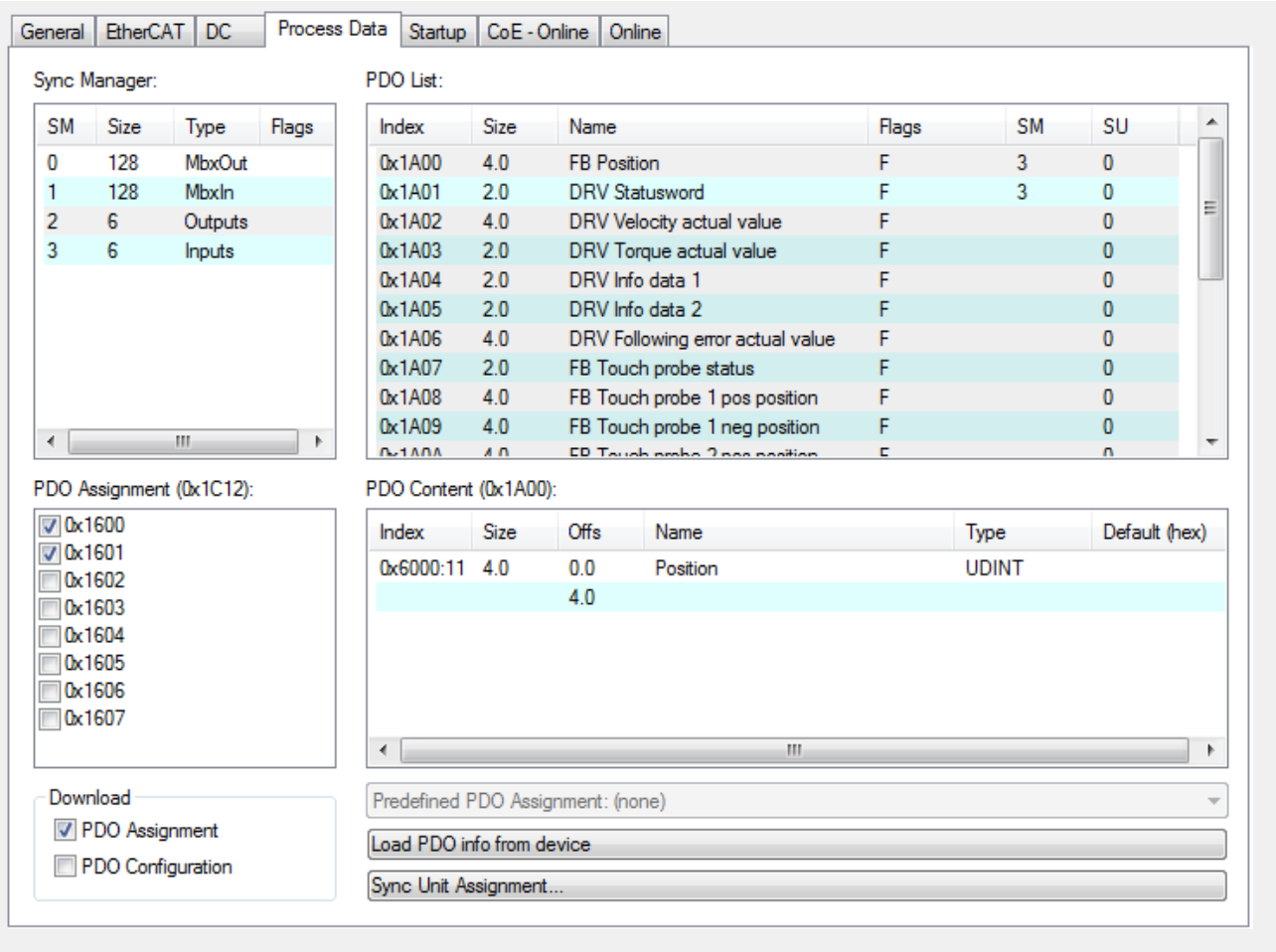


Fig. 202: Process Data tab SM2, EL72x1-0010 (default)

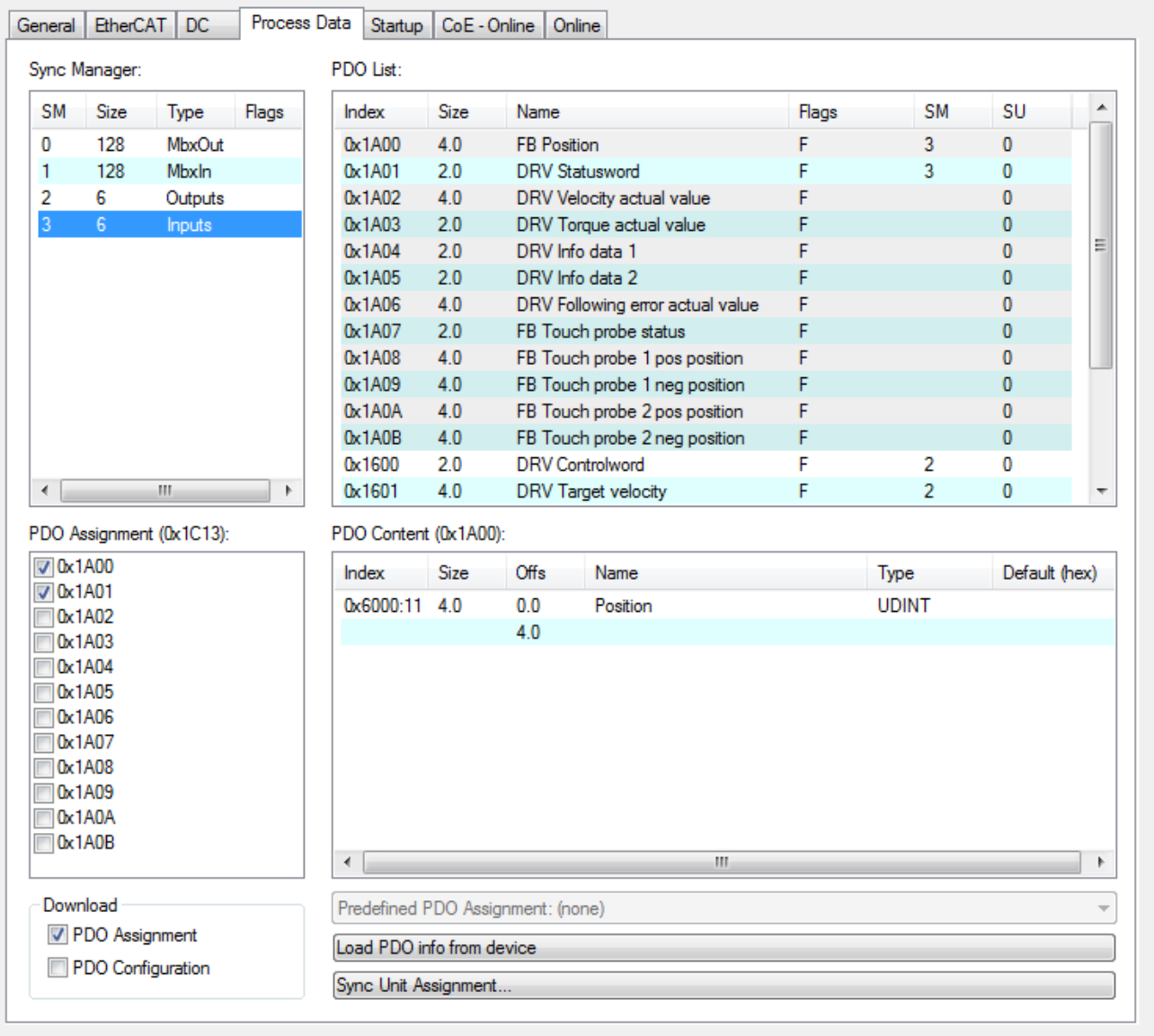


Fig. 203: Process Data tab SM3, EL72x1-0010 (default)

PDO Assignment

- To configure the process data, select the required Sync Manager (SM2 or SM3) in the "Sync Manager" field at the top left (see Fig. *Process Data tab SM3, EL72x1-0010*).
- The process data assigned to this Sync Manager can then be switched on or off in the “PDO Assignment” box underneath.
- Restarting the EtherCAT system, or reloading the configuration in configuration mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM2, PDO assignment 0x1C12

Index	Size (byte.bit)	Name	PDO content Index - name, size (byte.bit)
0x1600 (default)	2.0	DRV Controlword	0x7010:01 [▶ 211] - Controlword (2.0)
0x1601 (default)	4.0	DRV Target velocity	0x7010:06 [▶ 211] - Target velocity (4.0)
0x1602	2.0	DRV Target torque	0x7010:09 [▶ 211] - Target torque (2.0)
0x1603	2.0	DRV Commutation angle	0x7010:0E [▶ 211] - Commutation angle (2.0)
0x1604	2.0	DRV Torque limitation	0x7010:0B [▶ 211] - Torque limitation (2.0)
0x1605	2.0	DRV Torque offset	0x7010:0A [▶ 211] - Torque offset (2.0)
0x1606	4.0	DRV Target position	0x7010:05 [▶ 211] - Target position (4.0)
0x1607	2.0	FB Touch probe control	0x7001:01 [▶ 210] - Touch probe function_TP1 Enable (0.1) 0x7001:02 [▶ 210] - Touch probe function_TP1 Continous (0.1) 0x7001:03 [▶ 210] - Touch probe function_TP1 Trigger mode (0.2) 0x7001:05 [▶ 210] - Touch probe function_TP1 Enable pos. edge (0.1) 0x7001:06 [▶ 210] - Touch probe function_TP1 Enable neg. edge (0.1) 0x7001:09 [▶ 210] - Touch probe function_TP2 Enable (0.1) 0x7001:0A [▶ 210] - Touch probe function_TP2 Continous (0.1) 0x7001:0B [▶ 210] - Touch probe function_TP2 Trigger mode (0.2) 0x7001:0D [▶ 210] - Touch probe function_TP2 Enable pos. edge (0.1) 0x7001:0E [▶ 210] - Touch probe function_TP2 Enable neg. edge (0.1)
0x1608	1.0	DRV Modes of operation	0x7010:03 [▶ 211] - Modes of operation (1.0)
0x1630*	40.0	DMC Outputs	0x7030:02 [▶ 212] - DMC_FeedbackControl_Enable latch extern on positive edge (0.1) 0x7030:03 [▶ 212] - DMC_FeedbackControl_Set counter (0.1) 0x7030:04 [▶ 212] - DMC_FeedbackControl_Enable latch extern on negative edge (0.1) 0x7030:11 [▶ 212] - DMC_DriveControl_Enable (0.1) 0x7030:12 [▶ 212] - DMC_DriveControl_Reset (0.1) 0x7030:21 [▶ 212] - DMC_PositioningControl_Execute (0.1) 0x7030:22 [▶ 212] - DMC_PositioningControl_Emergency stop (0.1) 0x7030:31 [▶ 212] - DMC_Set counter value (8.0) 0x7030:32 [▶ 212] - DMC_Target position (8.0) 0x7030:3 [▶ 212]3 - DMC_Target velocity (2.0) 0x7030:34 [▶ 212] - DMC_Start typev (2.0) 0x7030:35 [▶ 212] - DMC_Target acceleration (2.0) 0x7030:3 [▶ 212]6 - DMC_Target deceleration (2.0)
0x1630*	40.0	DMC Outputs 32 Bit	0x7030:02 [▶ 212] - DMC_FeedbackControl_Enable latch extern on positive edge (0.1) 0x7030:03 [▶ 212] - DMC_FeedbackControl_Set counter (0.1) 0x7030:04 [▶ 212] - DMC_FeedbackControl_Enable latch extern on negative edge (0.1) 0x7030:11 [▶ 212] - DMC_DriveControl_Enable (0.1) 0x7030:12 [▶ 212] - DMC_DriveControl_Reset (0.1) 0x7030:21 [▶ 212] - DMC_PositioningControl_Execute (0.1) 0x7030:22 [▶ 212] - DMC_PositioningControl_Emergency stop (0.1) 0x7030:31 [▶ 212] - DMC_Set counter value (4.0) 0x7030:32 [▶ 212] - DMC_Target position (4.0) 0x7030:3 [▶ 212]3 - DMC_Target velocity (2.0) 0x7030:34 [▶ 212] - DMC_Start type (2.0) 0x7030:35 [▶ 212] - DMC_Target acceleration (2.0) 0x7030:3 [▶ 212]6 - DMC_Target deceleration (2.0)

*) from firmware 19

SM3, PDO assignment 0x1C13

Index	Size (byte.bit)	Name	PDO content Index - name, size (byte.bit)
0x1A00 (default)	4.0	FB Position	0x6000:11 [▶ 206] - Position (4.0)
0x1A01 (default)	2.0	DRV Statusword	0x6010:01 [▶ 208] - Statusword (2.0)
0x1A02	4.0	DRV Velocity actual value	0x6010:07 [▶ 208] - Velocity actual value (4.0)
0x1A03	2.0	DRV Torque actual value	0x6010:08 [▶ 208] - Torque actual value (2.0)
0x1A04	2.0	DRV Info data 1	0x6010:12 [▶ 208] - Info data 1 (2.0)
0x1A05	2.0	DRV Info data 2	0x6010:13 [▶ 208] - Info data 2 (2.0)
0x1A06	4.0	DRV Following error actual value	0x6010:09 [▶ 208] - Following error actual value (4.0)
0x1A07	2.0	FB Touch probe status	0x6001:01 [▶ 207] - Touch probe status_ TP1 Enable (0.1) 0x6001:02 [▶ 207] - Touch probe status_ TP1 Pos. value stored (0.1) 0x6001:03 [▶ 207] - Touch probe status_ TP1 Neg. value stored (0.1) 0x6001:08 [▶ 207] - Touch probe status_ TP1 Input (0.1) 0x6001:09 [▶ 207] - Touch probe status_ TP2 Enable (0.1) 0x6001:0A [▶ 207] - Touch probe status_ TP2 Pos. value stored (0.1) 0x6001:0B [▶ 207] - Touch probe status_ TP2 Neg. value stored (0.1) 0x6001:10 [▶ 207] - Touch probe status_ TP2 Input (0.1)
0x1A08	4.0	FB Touch probe 1 pos. position	0x6001:11 [▶ 207] - TP1 Pos position (4.0)
0x1A09	4.0	FB Touch probe 1 neg. position	0x6001:12 [▶ 207] - TP1 Neg position (4.0)
0x1A0A	4.0	FB Touch probe 2 pos. position	0x6001:13 [▶ 207] - TP2 Pos position (4.0)
0x1A0B	4.0	FB Touch probe 2 neg. position	0x6001:14 [▶ 207] - TP2 Neg position (4.0)
0x1A0C	2.0	FB Status	0x6000:0E [▶ 206] - Status_TxPDO State (0.1) 0x6000:0F [▶ 206] - Status_Input Cycle Counter (0.2)
0x1A0E	1.0	DRV Modes of operation display	0x6010:03 [▶ 208] - Modes of operation display (1.0)
0x1A30*	64.0	DMC Inputs	0x6030:02 [▶ 209] - DMC_FeedbackStatus_Latch extern valid (0.1) 0x6030:03 [▶ 209] - DMC_FeedbackStatus_Set counter done (0.1) 0x6030:0D [▶ 209] - DMC_FeedbackStatus_Status of extern Latch (0.1) 0x6030:11 [▶ 209] - DMC_DriveStatus_Ready to enable (0.1) 0x6030:12 [▶ 209] - DMC_DriveStatus_Ready (0.1) 0x6030:13 [▶ 209] - DMC_DriveStatus_Warning (0.1) 0x6030:14 [▶ 209] - DMC_DriveStatus_Error (0.1) 0x6030:15 [▶ 209] - DMC_DriveStatus_Movin positive (0.1) 0x6030:16 [▶ 209] - DMC_DriveStatus_Moving negative (0.1) 0x6030:1C [▶ 209] - DMC_DriveStatus_Digital input 1 (0.1) 0x6030:1D [▶ 209] - DMC_DriveStatus_Digital input 2 (0.1) 0x6030:21 [▶ 209] - DMC_PositioningStatus_Busy (0.1) 0x6030:22 [▶ 209] - DMC_PositioningStatus_In-Target (0.1) 0x6030:23 [▶ 209] - DMC_PositioningStatus_Warning (0.1) 0x6030:24 [▶ 209] - DMC_PositioningStatus_Error (0.1) 0x6030:25 [▶ 209] - DMC_PositioningStatus_Calibrated (0.1) 0x6030:26 [▶ 209] - DMC_PositioningStatus_Accelerate (0.1) 0x6030:27 [▶ 209] - DMC_PositioningStatus_Decelerate (0.1) 0x6030:28 [▶ 209] - DMC_PositioningStatus_Ready to execute (0.1) 0x6030:31 [▶ 209] - DMC_Set position (8.0) 0x6030:32 [▶ 209] - DMC_Set velocity (2.0) 0x6030:33 [▶ 209] - DMC_Actual drive time (4.0) 0x6030:34 [▶ 209] - DMC_Actual position lag (8.0) 0x6030:35 [▶ 209] - DMC_Actual velocity (2.0) 0x6030:36 [▶ 209] - DMC_Actual position (8.0) 0x6030:37 [▶ 209] - DMC_Error id (4.0) 0x6030:38 [▶ 209] - DMC_Input cycle counter (1.0) 0x6030:3A [▶ 209] - DMC_Latch value (8.0) 0x6030:3B [▶ 209] - DMC_Cyclic info data 1 (2.0) 0x6030:3C [▶ 209] - DMC_Cyclic info data 2 (2.0)

Index	Size (byte.bit)	Name	PDO content Index - name, size (byte.bit)
0x1A31*	64.0	DMC Inputs 32 Bit	0x6030:02 [▶ 209] - DMC_FeedbackStatus_Latch extern valid (0.1) 0x6030:03 [▶ 209] - DMC_FeedbackStatus_Set counter done (0.1) 0x6030:0D [▶ 209] - DMC_FeedbackStatus_Status of extern Latch (0.1) 0x6030:11 [▶ 209] - DMC_DriveStatus_Ready to enable (0.1) 0x6030:12 [▶ 209] - DMC_DriveStatus_Ready (0.1) 0x6030:13 [▶ 209] - DMC_DriveStatus_Warning (0.1) 0x6030:14 [▶ 209] - DMC_DriveStatus_Error (0.1) 0x6030:15 [▶ 209] - DMC_DriveStatus_Movin positive (0.1) 0x6030:16 [▶ 209] - DMC_DriveStatus_Moving negative (0.1) 0x6030:1C [▶ 209] - DMC_DriveStatus_Digital input 1 (0.1) 0x6030:1D [▶ 209] - DMC_DriveStatus_Digital input 2 (0.1) 0x6030:21 [▶ 209] - DMC_PositioningStatus_Busy (0.1) 0x6030:22 [▶ 209] - DMC_PositioningStatus_In-Target (0.1) 0x6030:23 [▶ 209] - DMC_PositioningStatus_Warning (0.1) 0x6030:24 [▶ 209] - DMC_PositioningStatus_Error (0.1) 0x6030:25 [▶ 209] - DMC_PositioningStatus_Calibrated (0.1) 0x6030:26 [▶ 209] - DMC_PositioningStatus_Accelerate (0.1) 0x6030:27 [▶ 209] - DMC_PositioningStatus_Decelerate (0.1) 0x6030:28 [▶ 209] - DMC_PositioningStatus_Ready to execute (0.1) 0x6030:31 [▶ 209] - DMC_Set position (4.0) 0x6030:32 [▶ 209] - DMC_Set velocity (2.0) 0x6030:33 [▶ 209] - DMC_Actual drive time (4.0) 0x6030:34 [▶ 209] - DMC_Actual position lag (4.0) 0x6030:35 [▶ 209] - DMC_Actual velocity (2.0) 0x6030:36 [▶ 209] - DMC_Actual position (4.0) 0x6030:37 [▶ 209] - DMC_Error id (4.0) 0x6030:38 [▶ 209] - DMC_Input cycle counter (1.0) 0x6030:3A [▶ 209] - DMC_Latch value (4.0) 0x6030:3B [▶ 209] - DMC_Cyclic info data 1 (2.0) 0x6030:3C [▶ 209] - DMC_Cyclic info data 2 (2.0)

*) from firmware 19

Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the "Process Data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.

Six PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Cyclic synchronous velocity mode (CSV)	0x1600 [▶ 217] (DRV Controlword) 0x1601 [▶ 218] (DRV Target velocity)	0x1A00 [▶ 222] (FB Position) 0x1A01 [▶ 222] (DRV Statusword)
Cyclic synchronous torque mode (CST)	0x1600 [▶ 217] (DRV Controlword) 0x1602 [▶ 218] (DRV Target torque)	0x1A00 [▶ 222] (FB Position) 0x1A01 [▶ 222] (DRV Statusword) 0x1A03 [▶ 222] (DRV Torque actual value)
Cyclic synchronous torque mode with commutation angle (CSTCA)	0x1600 [▶ 217] (DRV Controlword) 0x1602 [▶ 218] (DRV Target torque) 0x1603 [▶ 218] (DRV Commutation angle)	0x1A01 [▶ 222] (DRV Statusword)
Cyclic synchronous position mode (CSP)	0x1600 [▶ 217] (DRV Controlword) 0x1606 [▶ 218] (DRV Target position)	0x1A00 [▶ 222] (FB Position) 0x1A01 [▶ 222] (DRV Statusword) 0x1A06 [▶ 223] (DRV Following error actual value)
Drive motion control (For TC3 DriveMotionControl Lib)*	0x1630 [▶ 220] (DMC Outputs)	0x1A30 [▶ 224] (DMC Inputs)
Drive motion control (32 Bit)*	0x1631 [▶ 221] (DMC Outputs 32 Bit)	0x1A31 [▶ 226] (DMC Inputs 32 Bit)

*) from firmware 19

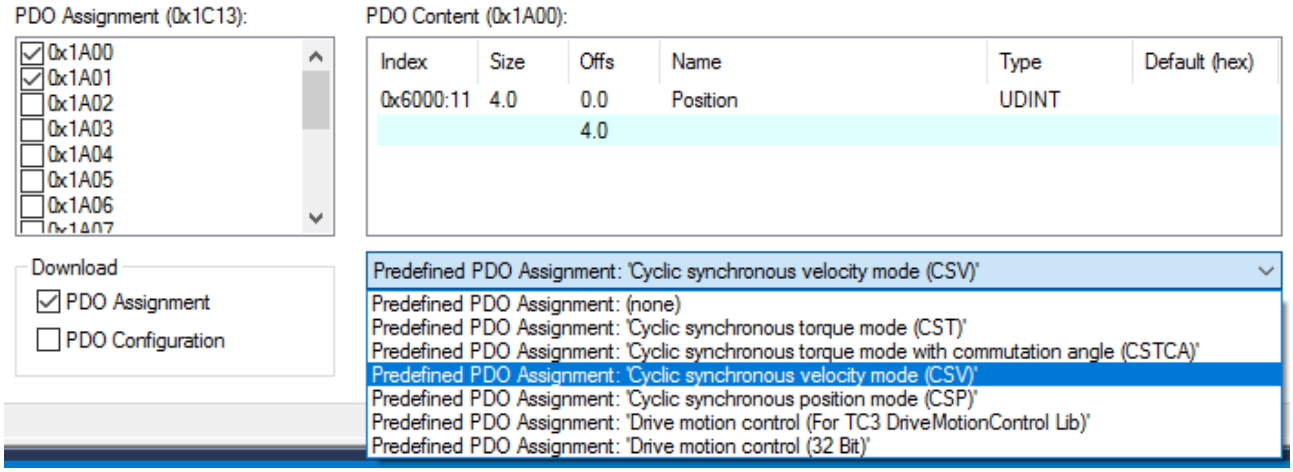


Fig. 204: Process data tab - Predefined PDO Assignment, EL72x1-0010

5.7 DS402 Process data

Table of contents

- Sync Manager [▶ 183]
- PDO Assignment [▶ 184]
- Predefined PDO Assignment [▶ 186]

Sync Manager (SM)

Sync Manager (SM) The extent of the process data that is made available can be changed through the "Process data" tab (see Fig. Process Data tab SM2, EL72x1-0010 (default)).

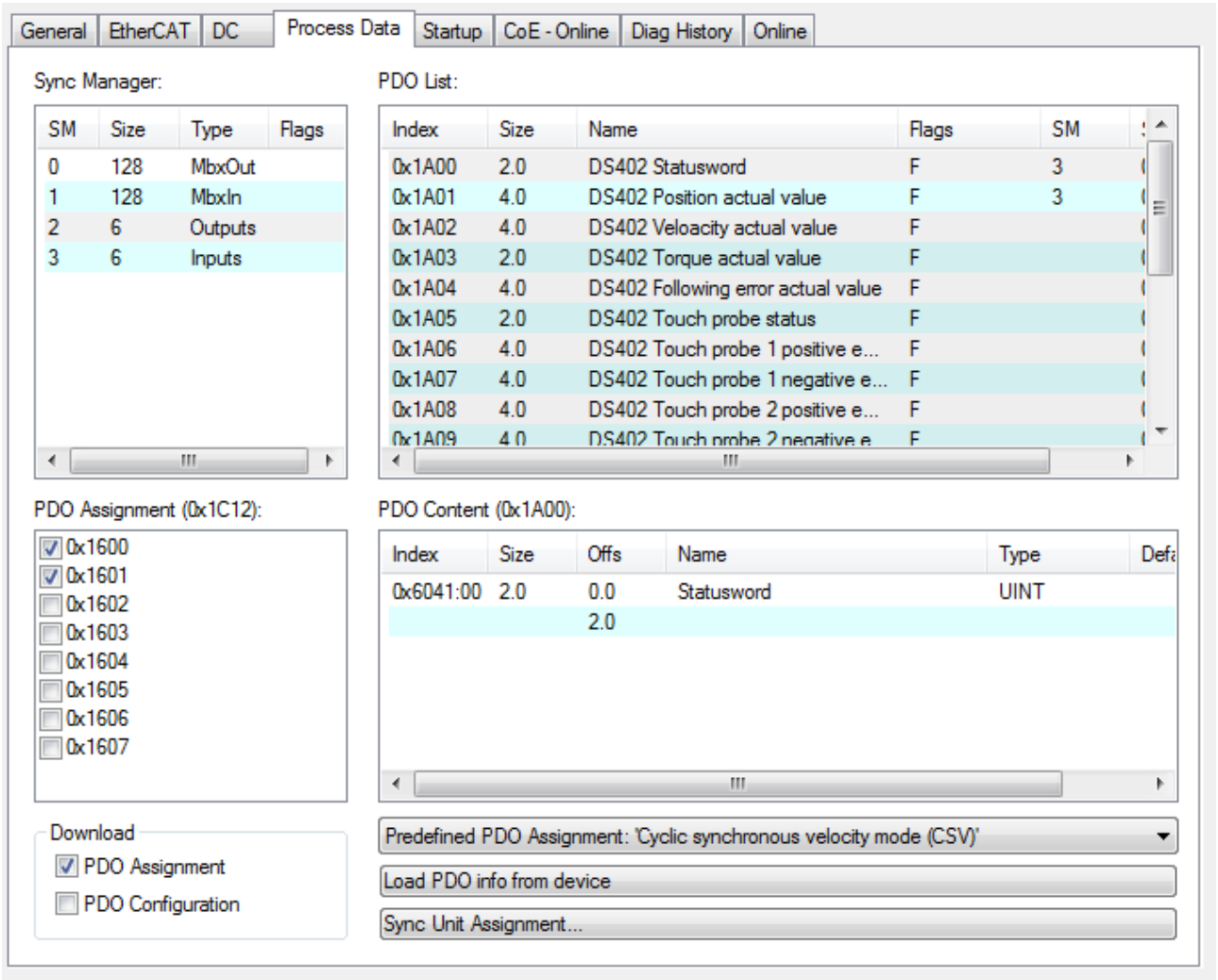


Fig. 205: Process Data tab SM2, EL72x1-0010 (default)

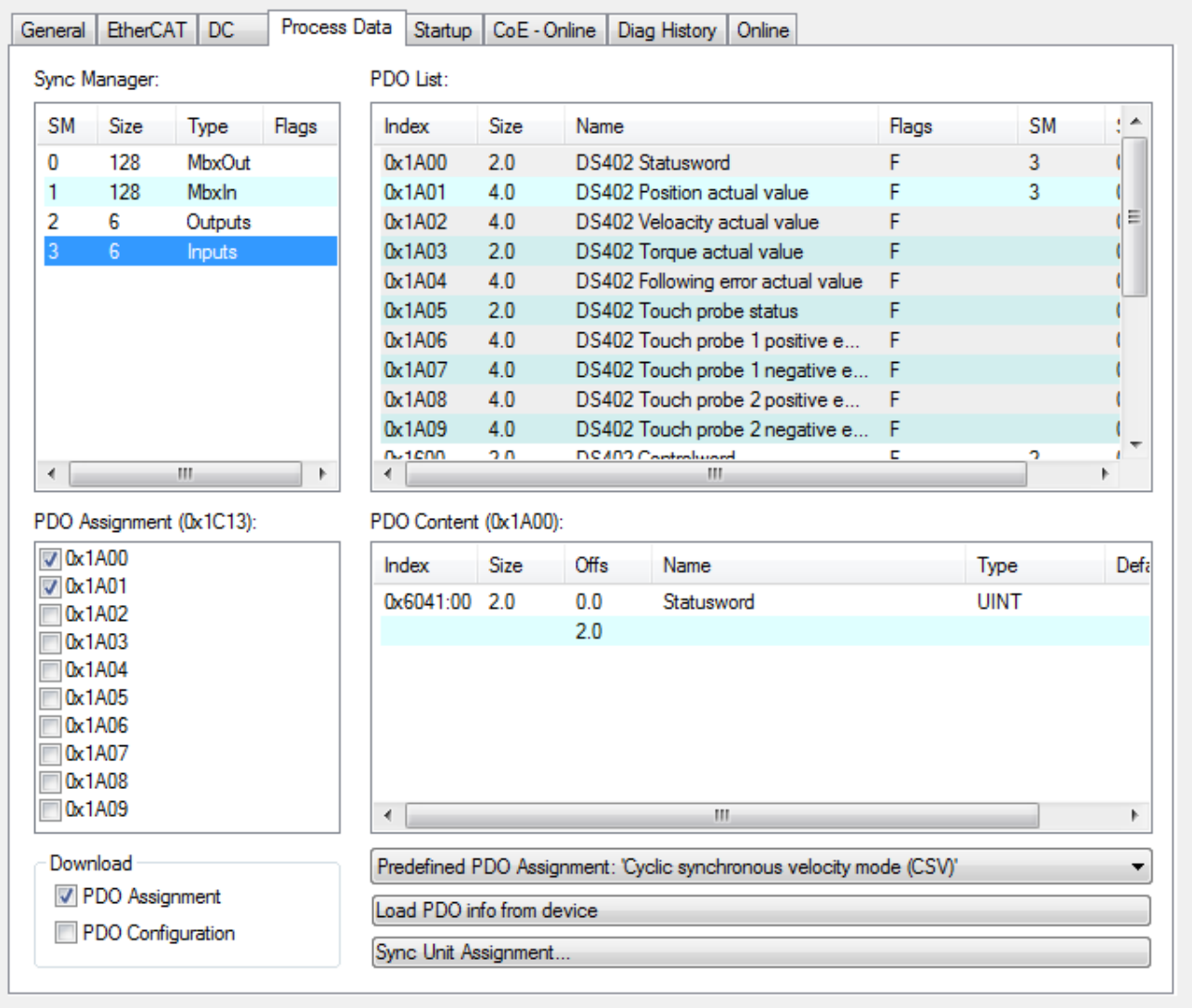


Fig. 206: Process Data tab SM3, EL72x1-0010 (default)

PDO Assignment

- In order to configure the process data, select the desired Sync Manager (SM2 and SM3 can be edited) in the upper left-hand "Sync Manager" box (see fig.).
- The process data assigned to this Sync Manager can then be switched on or off in the “PDO Assignment” box underneath.
- Restarting the EtherCAT system, or reloading the configuration in configuration mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM2, PDO assignment 0x1C12

Index	Size (byte.bit)	Name	PDO Content Index - name, size (byte.bit)
0x1600 (default)	2.0	DS402 Controlword	0x6040 [▶ 240] - Controlword (2.0)
0x1601 (default)	4.0	DS402 Target velocity	0x60FF [▶ 244] -Target velocity (4.0)
0x1602	2.0	DS402 Target torque	0x6071 [▶ 241] - Target torque (2.0)
0x1603	2.0	DS402 Commutation angle	0x60EA [▶ 244] - Commutation angle (2.0)
0x1604	2.0	DS402 Torque limitation	0x6072 [▶ 241] - Max. torque (2.0)
0x1605	2.0	DS402 Torque offset	0x2001:11 [▶ 239] - Torque offset (2.0)
0x1606	4.0	DS402 Target position	0x607A [▶ 242] - Target position (4.0)
0x1607	2.0	DS402 FB Touch probe function	0x60B8 [▶ 242] - Touch probe function (2.0)
			Bit 0 TP1 Enable Bit 1 TP1 Continous Bit 2 TP1 Trigger mode Bit 4 TP1 Enable pos. edge Bit 5 TP1 Enable neg. edge Bit 8 TP2 Enable Bit 9 TP2 Continous Bit 10 TP2 Trigger mode Bit 12 TP2 Enable pos. edge Bit 13 TP2 Enable neg. edge
0x1608	1.0	DS402 Modes of operation	0x6060 [▶ 240] - Modes of operation (1.0)

SM3, PDO Assignment 0x1C13

Index	Size (byte.bit)	Name	PDO Content Index - name, size (byte.bit)
0x1A00 (default)	2.0	DS402 Statusword	0x6041 [▶ 240] - Statusword (2.0)
0x1A01 (default)	4.0	DS402 Position actual value	0x6064 [▶ 240] - Position actual value (4.0)
0x1A02	4.0	DS402 Velocity actual value	0x606C [▶ 241] - Velocity actual value (4.0)
0x1A03	2.0	DS402 Torque actual value	0x6077 [▶ 242] - Torque actual value (2.0)
0x1A04	4.0	DS402 Following error actual value	0x60F4 [▶ 244] - Following error actual value (4.0)
0x1A05	2.0	DS402 Touch probe status	0x60B9 [▶ 243] - Touch probe status (2.0)
			Bit 0 TP1 Enable Bit 1 TP1 Pos. value stored Bit 2 TP1 Neg. value stored Bit 7 TP1 Input Bit 8 TP2 Enable Bit 9 TP2 Pos. value stored Bit 10 TP2 Neg. value stored Index 6001:10 TP2 Input
0x1A06	4.0	DS402 Touch probe 1 positive edge	0x60BA [▶ 243] - Touch probe 1 positive edge (4.0)
0x1A07	4.0	DS402 Touch probe 1 negative edge	0x60BB [▶ 243] - Touch probe 1 negative edge (4.0)
0x1A08	4.0	DS402 Touch probe 2 positive edge	0x60BC [▶ 243] - Touch probe 2 positive edge (4.0)
0x1A09	4.0	DS402 Touch probe 2 negative edge	0x60BD [▶ 243] - Touch probe 2 negative edge (4.0)
0x1A0A	2.0	DS402 TxPDO Data Invalid	0x603E:02 [▶ 239] - TxPDO Data invalid_Position actual value (0.1)
0x1A0B	2.0	DS402 Info data 1	0x2008:01 [▶ 239] - Info data 1 (2.0)
0x1A0C	2.0	DS402 Info data 2	0x2008:02 [▶ 239] - Info data 2 (2.0)
0x1A0E	1.0	DS402 Modes of operation display	0x6061:00 [▶ 240] - Modes of operation display (1.0)

Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the "Process Data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.

Four PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Cyclic synchronous velocity mode (CSV)	0x1600 [▶ 248] (DS402 Controlword) 0x1601 [▶ 249] (DS402 Target velocity)	0x1A00 [▶ 249] (DS402 Statusword) 0x1A01 [▶ 250] (DS402 Position actual value)
Cyclic synchronous torque mode (CST)	0x1600 [▶ 248] (DS402 Controlword) 0x1602 [▶ 249] (DS402 Target torque)	0x1A00 [▶ 249] (DS402 Statusword) 0x1A01 [▶ 250] (DS402 Position actual value) 0x1A03 [▶ 250] (DS402 Torque actual value)
Cyclic synchronous torque mode with commutation angle (CSTCA)	0x1600 [▶ 248] (DS402 Controlword) 0x1602 [▶ 249] (DS402 Target torque) 0x1603 [▶ 249] (DS402 Commutation angle)	0x1A00 [▶ 249] (DS402 Statusword)
Cyclic synchronous position mode (CSP)	0x1600 [▶ 248] (DS402 Controlword) 0x1606 [▶ 249] (DS402 Target position)	0x1A00 [▶ 249] (DS402 Statusword) 0x1A01 [▶ 250] (DS402 Position actual value) 0x1A04 [▶ 250] (DS402 Following error actual value)

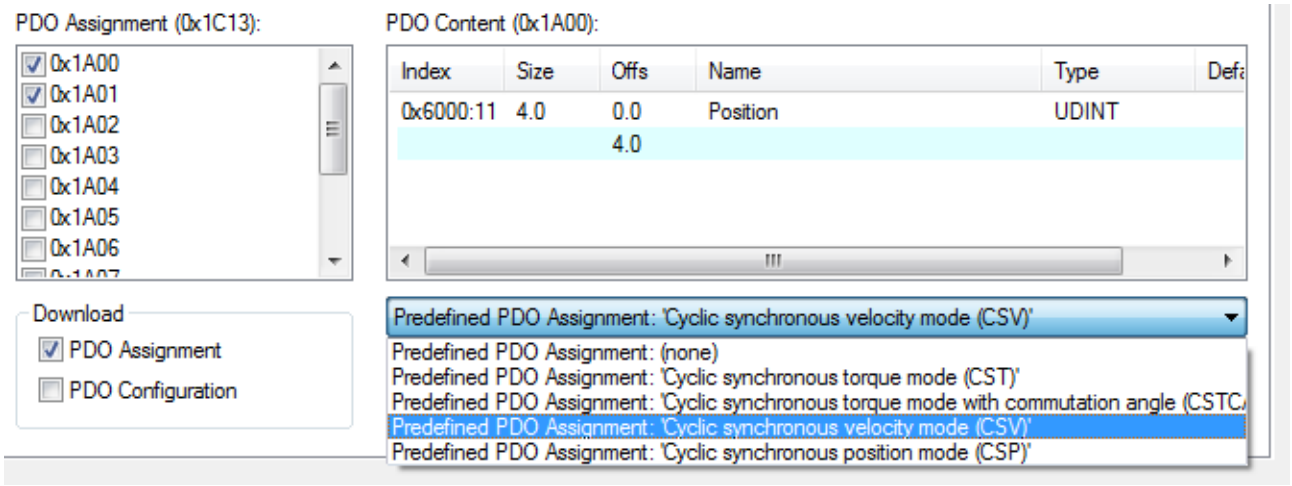


Fig. 207: Process data tab - Predefined PDO Assignment, EL72x1-0011

6 Integrated safety

6.1 Safety regulations

⚠ WARNING

Caution - Risk of injury!

Electronic equipment is not fail-safe. The machine manufacturer is responsible for ensuring that the connected motors and the machine are brought into a safe state in the event of a fault in the drive system.

NOTICE

Caution – electrostatic charging may lead to destruction of the Safety Card!

The Safety Card is an ESD-sensitive component. Follow the usual ESD safety procedures when handling the card (anti-static wrist straps, earthing of the relevant components etc.).

6.2 Description of product and function

6.2.1 Intended use

⚠ WARNING

Caution - Risk of injury!

TwinSAFE terminals may only be used for the purposes described below!

The TwinSAFE terminals expand the application range of Beckhoff Bus Terminal system with functions that enable them to be used for machine safety applications. The TwinSAFE terminals are designed for machine safety functions and directly associated industrial automation tasks. They are therefore only approved for applications with a defined fail-safe state. This safe state is the wattless state. Fail-safety according to the relevant standards is required.

The TwinSAFE Terminals enable connection of:

- 24 V_{DC} sensors (EL1904) such as
  emergency off pushbutton switches, pull cord switches, position switches, two-hand switches, safety mats, light curtains, light barriers, laser scanner, etc.
- 24 V_{DC} actuators (EL2904) such as
  contactors, protection door switches with tumbler, signal lamps, servo drives, etc.

Test pulses

When selecting actuators please ensure that the EL2904 test pulses do not lead to actuator switching or diagnostic message from the EL2904.

The following modules were developed for these tasks:

- The EL1904 terminal is an input module with digital inputs.
- The EL2904 terminal is an output module with digital outputs.
- The EL6900 terminal is a logic module.

These modules are suitable for operation with

- Beckhoff EKxxxx series Bus Couplers
- Beckhoff CXxxxx series Embedded PCs with E-bus connection

⚠ CAUTION

Follow the machinery directive

The TwinSAFE terminals may only be used in machines according to the machinery directive.

⚠ CAUTION

Ensure traceability

The buyer has to ensure the traceability of the device via the serial number.

6.2.2 Dimensions

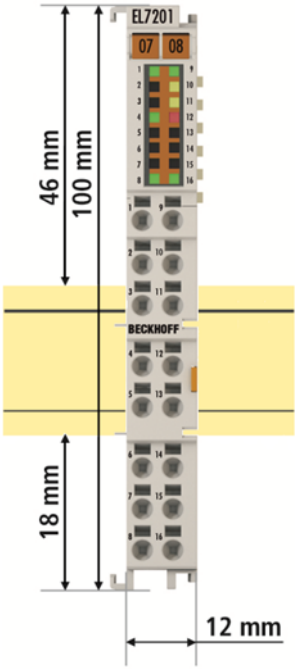


Fig. 208: Dimensions of the EL7201-xxxx

Width: 12 mm (side-by-side installation)
Height: 100 mm
Depth: 68 mm

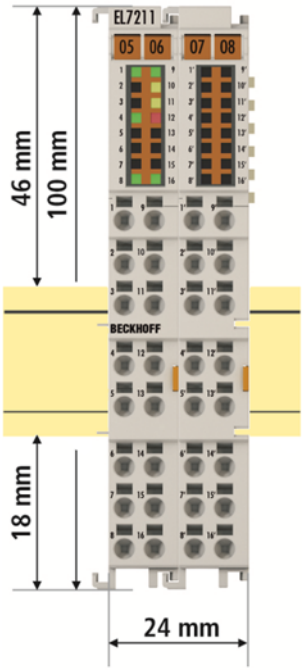


Fig. 209: Dimensions of the EL7211-xxxx

Width: 24 mm (side-by-side installation)
Height: 100 mm
Depth: 68 mm

6.2.3 TwinSAFE reaction times

The TwinSAFE terminals form a modular safety system that exchanges safety-oriented data via the Safety-over-EtherCAT protocol. This chapter is intended to help you determine the system's reaction time from the change of signal at the sensor to the reaction at the actuator.

Typical reaction time

The typical reaction time is the time that is required to transmit information from the sensor to the actuator, if the overall system is working without error in normal operation.

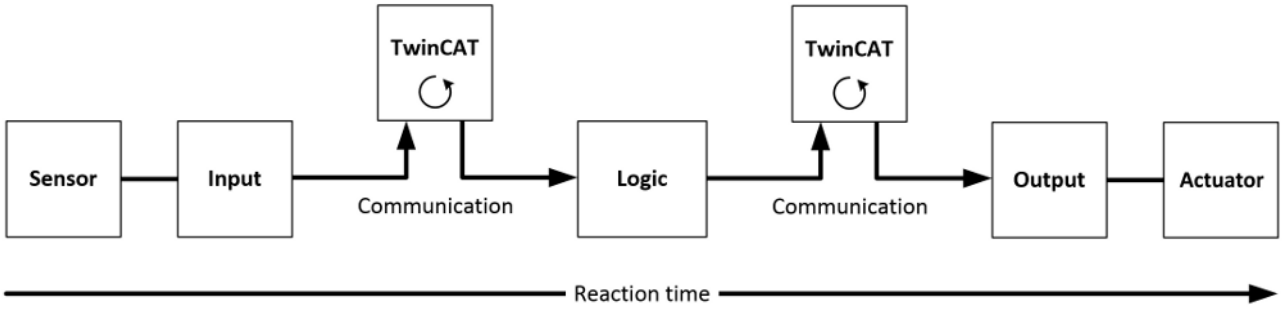


Fig. 210: Typical reaction time

Definition	Description
RTSensor	Reaction time of the sensor until the signal is provided at the interface. Typically supplied by the sensor manufacturer.
RTInput	Reaction time of the safe input, such as EL1904 or EP1908. This time can be found in the technical data. In the case of the EL1904 it is 4 ms.
RTComm	Reaction time of the communication This is typically 3x the EtherCAT cycle time, because new data can only be sent in a new Safety-over-EtherCAT telegram. These times depend directly on the higher-level standard controller (cycle time of the PLC/NC).
RTLogic	Reaction time of the logic terminal. This is the cycle time of the logic terminal and typically ranges from 500 µs to 10 ms for the EL6900, depending on the size of the safety project. The actual cycle time can be read from the terminal.
RTOutput	Reaction time of the output terminal. This typically lies within the range of 2 to 3 ms.
RTActor	Reaction time of the servo terminal from the cut-off of the 24 V at the terminal point until the safe cut-off of the gate driver (Under Voltage Lockout). This time is typically about 20 ms.
WDComm	Watchdog time of the communication

This results in the following equation for the typical-case reaction time:

$$ReactionTime_{typ} = RT_{Sensor} + RT_{Input} + 3 * RT_{Comm} + RT_{Logic} + 3 * RT_{Comm} + RT_{Output} + RT_{Actuator}$$

with, for example

$$ReactionTime_{typ} = 5\,ms + 4\,ms + 3 * 1\,ms + 10\,ms + 3 * 1\,ms + 3\,ms + 20\,ms = 48\,ms$$

Worst-case reaction time

The worst case reaction time is the maximum time required to switch off the actuator in the case of an error.

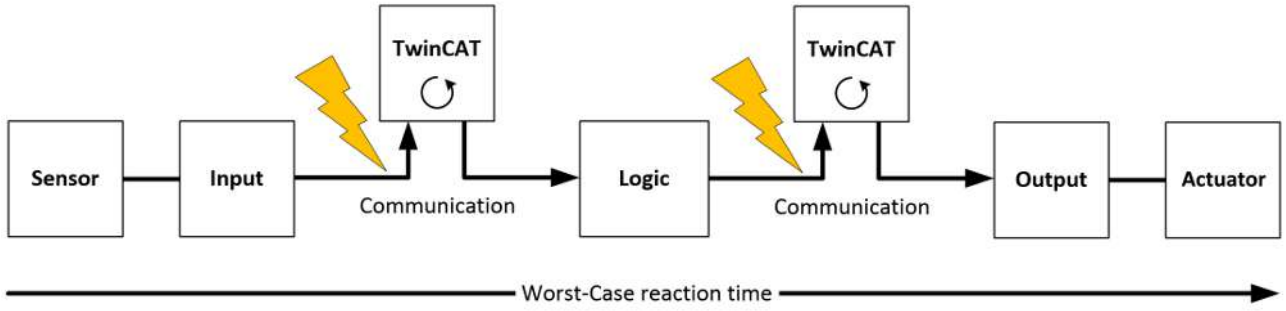


Fig. 211: Worst-case reaction time

This assumes that a signal change occurs at the sensor and is transmitted to the input. A communication error occurs at precisely the moment when the signal is to be transferred to the communication interface. This is detected by the logic following the watchdog time of the communication link. This information should then be transferred to the output, but a further communication error occurs here. This error is detected at the output following the expiry of the watchdog time and leads to the switch-off.

This results in the following equation for the worst-case reaction:

$$ReactionTime_{max} = WD_{Comm} + WD_{Comm} + RT_{Actuator}$$

with, for example

$$ReactionTime_{max} = 15\,ms + 15\,ms + 20\,ms = 50\,ms$$

6.2.4 Application example for STO function (Cat. 3, PL d)

Application example (STO – Safe Torque Off)

The following application example shows how the EL72x1-9014 can be wired together with an EL2904 in order to implement an STO function according to EN 61800-5-2.

The user must realize an appropriate evaluation for the wiring between the safety output terminal (EL2904) and the servo terminal (EL72x1-9014), so that a fault exclusion is permissible for external supply and cross-circuit in this wiring.

Components involved

- Emergency stop device (control switch S1) according to ISO 13850 and control switch S2
- 1 safety input terminal (EL1904) and 1 input terminal (EL 1004)
- 1 safety output terminal (EL2904)
- 1 safety logic terminal (EL6900) with function block "ESTOP"
- 1 servo terminal (EL72x1-9014) with STO input
- Programmable logic controller (PLC) and EtherCAT fieldbus

A safety door (S1 and S2) and a restart signal (S3) are logically linked on an ESTOP function block. The EStopOut signal is transferred to the NC controller, with which, for example, the Enable signal of the EL72x1-9014 can be switched. The STO input of the EL72x1-9014 is operated via the delayed output EStopDelOut. The EL72x1-9014 supplies the information that the STO function is active via the standard controller. This information is transferred to the EDM input of the ESTOP function block and additionally to the EDM function block in order to generate an expectation for this signal.

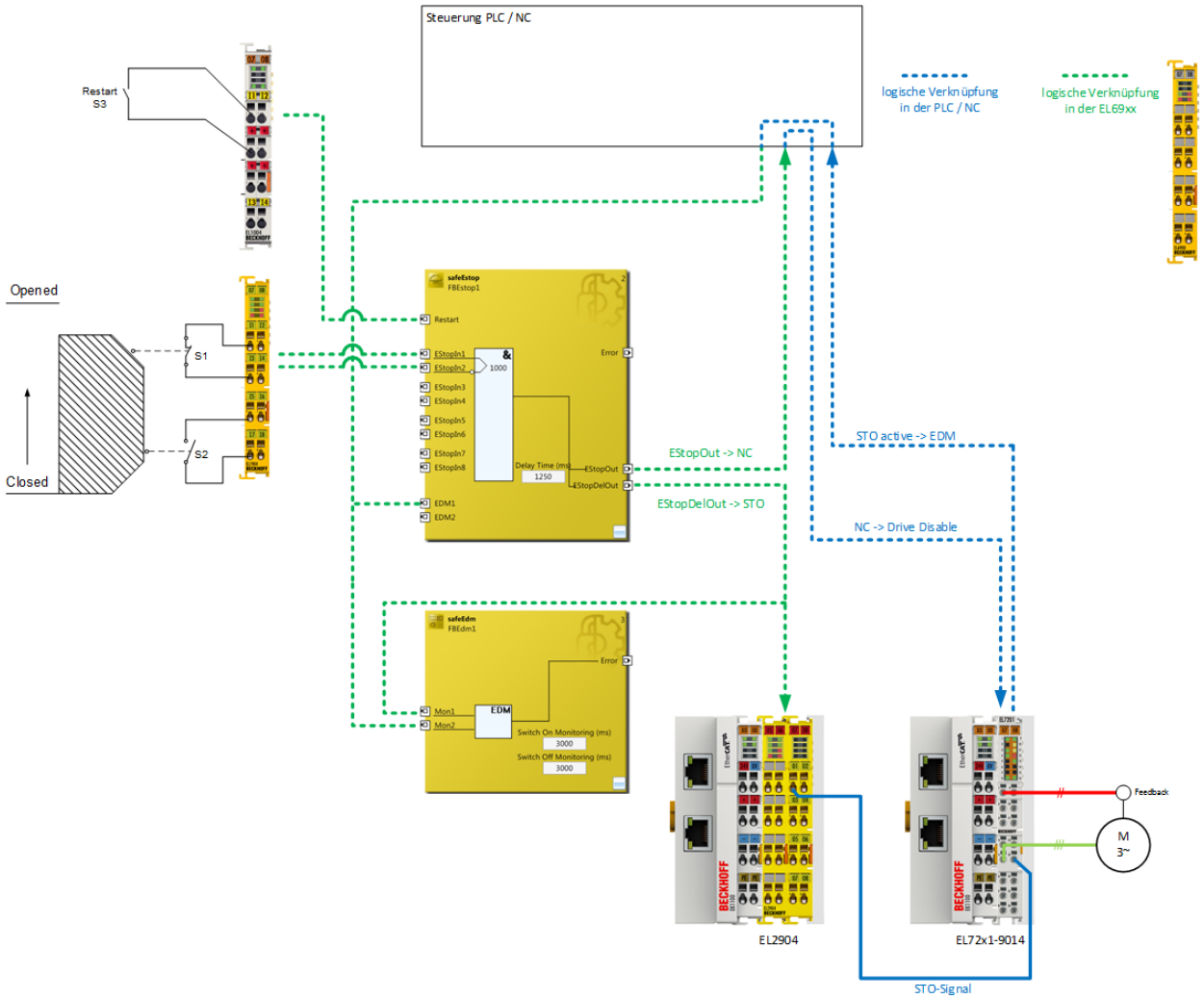


Fig. 212: Connection example for EL72x1_9014 with STO

⚠ CAUTION

Implement a restart lock in the machine!

The restart lock is NOT part of the safety chain and must be implemented in the machine!
If the risk analysis returns the result that a restart is to be realized in the safety controller, then the restart **must** also be placed on a safe input.

⚠ WARNING

Wiring only inside the control cabinet

The wiring between the EL2904 and the STO input of the EL72x1-9014 must be located in the same control cabinet in order to be able to assume a fault exclusion for the cross-circuit or external power supply of the wiring between EL2904 and EL72x1-9014.
The evaluation of this wiring and the evaluation of whether the fault exclusion is permissible must be done by the machine manufacturer or user.

Calculation EL72x1-9014

The EL72x1-9014 is not taken into account in the calculation of the Performance Level DIN EN ISO 13849-1 since it behaves non-reactively to the safety function. The PFH value goes into the calculation according to EN 62061 with a value of 0.

Operation of several EL72x1-9014 terminals at the same time

A maximum of 10 STO inputs of the corresponding EL72x1-9014 can be operated at the same time with a cut-off channel of the EL2904.

Parameters of the safe input and output terminals

EL1904

Parameter	Value
Sensor test channel 1 active	Yes
Sensor test channel 2 active	Yes
Sensor test channel 3 active	Yes
Sensor test channel 4 active	Yes
Logic channel 1 and 2	Single Logic
Logic channel 3 and 4	Single Logic

EL2904

Parameter	Value
Current measurement active	No
Output test pulses active	Yes

Block formation and safety loops
Safety function 1

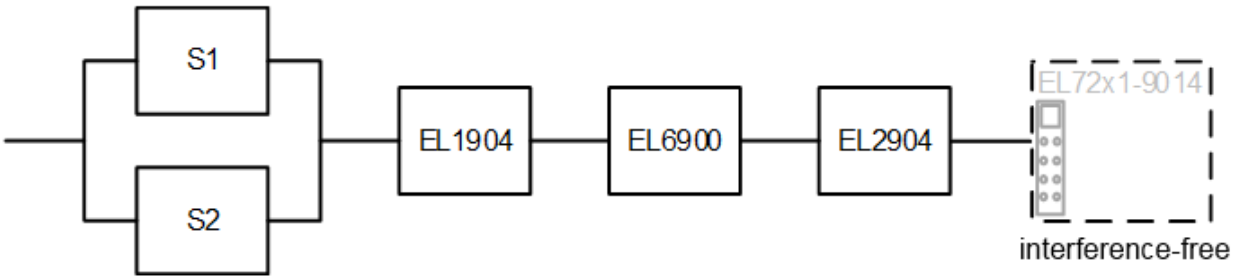


Fig. 213: EL72x1_9014_block_diagram

Calculation
PFH / MTTFd /B10d – values

Component	Value
EL1904 – PFH	1.11E-09
EL2904 – PFH	1.25E-09
EL6900 – PFH	1.03E-09
EL72x1-9014 - PFH	0.00
S1 – B10d	1,000,000
S2 – B10d	2,000,000
Days of operation (dop)	230
Hours of operation / day (hop)	16
Cycle time (minutes) (Tcycle)	15 (4x per hour)
Lifetime (T1)	20 years = 175200 hours

Diagnostic Coverage DC

Component	Value
S1/S2 with testing/plausibility	DCavg=99%
EL2904 with testing	DCavg=99%

Calculation for block 1

Calculation of the PFH and $MTTF_d$ values from the $B10_d$ values:

From:

$$n_{op} = \frac{d_{op} * h_{op} * 60}{T_{Zyklus}}$$

and:

$$MTTF_d = \frac{B10_d}{0{,}1 * n_{op}}$$

Inserting the values, this produces:

S1:

$$n_{op} = \frac{230 * 16 * 60}{15} = 14720$$

$$MTTF_d = \frac{1000000}{0{,}1 * 14720} = 679{,}3\text{y} = 5951087\text{h}$$

S2:

$$n_{op} = \frac{230 * 16 * 60}{15} = 14720$$

$$MTTF_d = \frac{2000000}{0{,}1 * 14720} = 1358{,}7\text{y} = 11902174\text{h}$$

and the assumption that S1 and S2 are each single-channel:

$$MTTF_d = \frac{1}{\lambda_d}$$

produces for

$$PFH = \frac{0{,}1 * n_{op} * (1 - DC)}{B10_d} = \frac{1 - \mathrm{DC}}{\mathrm{MTTF_d}}$$

S1:

$$PFH = \frac{1 - 0{,}99}{679{,}3 * 8760} = 1{,}68\mathrm{E} - 9$$

S2:

$$PFH = \frac{1 - 0{,}99}{1358{,}7 * 8760} = 8{,}4\mathrm{E} - 10$$

The following assumptions must now be made:

The door switches S1/S2 are always actuated in opposite directions. Since the switches have different values, but the complete protective door switch consists of a combination of normally closed and normally open contacts and both switches must function, the poorer of the two values (S1) can be taken for the combination!

There is a coupling coefficient between the components that are connected via two channels. Examples are temperature, EMC, voltage peaks or signals between these components. This is assumed to be the worst-case estimation, where ß =10%. EN 62061 contains a table with which this ß-factor can be precisely determined. Further, it is assumed that all usual measures have been taken to prevent both channels failing unsafely at the same time due to an error (e.g. overcurrent through relay contacts, overtemperature in the control cabinet).

This produces for the calculation of the PFH value for block 1:

PFHtot = β (PFH(S1)+ PFH(S2))/2 + PFH(EL1904) + PFH(EL6900) + PFH(EL2904) + PFH(EL7201-9014)*

to:

PFHtot = 10%* (1.68E-09+1.68E-09)/2 +1.11E-09 + 1.03E-09 + 1.25E-09 + 0.00 = **3.558E-09**

The $MTTF_d$ value for block 1 (based on the same assumption) is calculated with:

$$\frac{1}{MTTF_{d\ ges}} = \sum_{i=1}^{n} \frac{1}{MTTF_{d\ n}}$$

as:

$$\frac{1}{MTTF_{d\ ges}} = \frac{1}{MTTF_d\ (S1)} + \frac{1}{MTTF_{\ d}(EL1904)} + \frac{1}{MTTF_d\ (EL6900)} + \frac{1}{MTTF_{\ d}(2904)}$$

with:

$$MTTF_d(\mathrm{S1}) = \frac{B10_d(S1)}{0{,}1 * n_{op}}$$

$$MTTF_d(\mathrm{S2}) = \frac{B10_d(S2)}{0{,}1 * n_{op}}$$

If only PFH values are available for EL1904, EL2904 and EL6900, the following estimation applies:

$$MTTF_d(\text{ELxxxx}) = \frac{(1 - DC(ELxxx))}{PFH(ELxxx)}$$

Hence:

$$MTTF_d(\text{EL1904}) = \frac{(1 - DC(EL1904))}{PFH(EL1904)} = \frac{(1 - 0{,}99)}{1{,}11\text{E} - 09\frac{1}{\text{h}} * 8760\frac{\text{h}}{\text{y}}} = \frac{0{,}01}{9{,}72\text{E} - 06\frac{1}{\text{y}}} = 1028{,}8\text{y}$$

$$MTTF_d(\text{EL6900}) = \frac{(1 - DC(EL6900))}{PFH(EL6900)} = \frac{(1 - 0{,}99)}{1{,}03\text{E} - 09\frac{1}{\text{h}} * 8760\frac{\text{h}}{\text{y}}} = \frac{0{,}01}{9{,}02\text{E} - 06\frac{1}{\text{y}}} = 1108{,}6\text{y}$$

$$MTTF_d(\text{EL2904}) = \frac{(1 - DC(EL2904))}{PFH(EL2904)} = \frac{(1 - 0{,}99)}{1{,}25\text{E} - 09\frac{1}{\text{h}} * 8760\frac{\text{h}}{\text{y}}} = \frac{0{,}01}{1{,}1\text{E} - 05\frac{1}{\text{y}}} = 913{,}2\text{y}$$

$$MTTF_{dges} = \frac{1}{\frac{1}{679{,}3y} + \frac{1}{1028{,}8y} + \frac{1}{1108{,}6y} + \frac{1}{913{,}2y}} = 225{,}2y$$

$$DC_{avg} = \frac{\frac{99\%}{679{,}3} + \frac{99\%}{1358{,}7} + \frac{99\%}{1028{,}8} + \frac{99\%}{1108{,}6} + \frac{99\%}{913{,}2}}{\frac{1}{679{,}3} + \frac{1}{1358{,}7} + \frac{1}{1028{,}8} + \frac{1}{1108{,}6} + \frac{1}{913{,}2}} = 99{,}00\%$$

⚠ CAUTION

Category

This structure is possible up to category 3 at the most.

$MTTF_d$	
Designation for each channel	**Range for each channel**
low	3 years ≤ $MTTF_d$ < 10 years
medium	10 years ≤ $MTTF_d$ < 30 years
high	30 years ≤ $MTTF_d$ ≤ 100 years

DC_{avg}	
Designation	**Range**
none	DC < 60 %
low	60 % ≤ DC < 90 %
medium	90 % ≤ DC < 99 %
high	**99 % ≤ DC**

Category	B	1	2	2	3	3	4
DC / $MTTF_d$	**none**	**none**	**low**	**medium**	**low**	**medium**	**high**
low	a	-	a	b	b	c	-
medium	b	-	b	c	c	d	-
high	-	c	c	d	d	**d**	e

Manufacturer data for interface type C – Senke

Parameter	min.	typically	max.
Class		2.3	
Test pulse duration $\mathbf{t_i}$	-	-	500 µs
Test pulse interval **T**	10 ms	-	-
Input resistance **R**	4.7 kΩ	-	-
Input capacitance $\mathbf{C_L}$	-	-	1.21 µF

The *Testing* parameter can be switched on in conjunction with the EL2904.

6.3 Maintenance

Maintenance

The TwinSAFE components are maintenance-free!

Environmental conditions

⚠ WARNING

Observe the specified environmental conditions!

Please ensure that the TwinSAFE components are only stored and operated under the specified conditions (see technical data).

If the TwinSAFE component is operated outside the permitted temperature range it will switch to *Global Shutdown* state.

Cleaning

Protect the TwinSAFE component from unacceptable soling during operation and storage!

If the TwinSAFE component was subjected to unacceptable soiling it may no longer be operated!

⚠ WARNING

Have soiled terminals checked!

Cleaning of the TwinSAFE component by the user is not permitted!
Please send soiled terminals to the manufacturer for inspection and cleaning!

7 Object description and parameterization

7.1 EL72x1-9014 (MDP742)

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs).

Please note the following general CoE information [▶ 32] 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

NOTICE

Risk of damage to the device!

We strongly advise not to change settings in the CoE objects while the axis is active, since this could impair the control.

7.1.1 Restore object

Index 1011 Restore default parameters

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

7.1.2 Configuration data

Index 8000 FB Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	FB Settings	Maximum subindex	UINT8	RO	0x17 (23_{dec})
8000:01	Invert feedback direction	Inverting the count direction	BOOLEAN	RW	0x00 (0_{dec})
8000:02	Referenced	Can be set to TRUE by the user. Is automatically set to FALSE as soon as a motor with a deviating serial number is recognized or if the feedback is single-turn.	BOOLEAN	RW	0x00 (0_{dec})
8000:0D	Offset position actual value source	Permitted values 0: Offset disabled No position offset is used. 1: Encoder memory The position offset from the feedback memory is used (0x9008:20 [▶ 213]) 2: Drive memory (default) The position offset from the servo drive is used (0x8000:17)			
8000:11	Device type	3: OCT (not changeable)	UINT32	RW	0x00000003 (3_{dec})
8000:12	Singleturn bits	Number of single-turn-bits [▶ 137]	UINT8	RW	0x14 (20_{dec})
8000:13	Multiturn bits	Number of multi-turn-bits [▶ 137]	UINT8	RW	0x0C (12_{dec})
8000:14	Observer bandwidth	Bandwidth of observer [Hz]	UINT16	RW	0x01F4 (500_{dec})
8000:15	Observer feed-forward	Load ratio[%] 100 % = free of load 50 % = moments of inertia of input and output are equal	UINT8	RW	0x01 (0_{dec})
8000:17	Positon offset	The position offset is subtracted from the raw position of the encoder. It can only be written with the axis stopped.	UINT32	RW	0x00000000 (0_{dec})

Index 8001 FB Touch probe Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	FB Touch probe Settings	Maximum subindex	UINT8	RO	0x12 (18_{dec})
8001:11	Touch probe 1 source	Permitted values: 1: Touch probe input 1	UINT16	RW	0x0001 (1_{dec})
8001:12	Touch probe 2 source	Permitted values: 2: Touch probe input 2	UINT16	RW	0x0002 (2_{dec})

Index 8008 FB OCT Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8008:0	FB OCT Settings	Maximum subindex	UINT8	RO	0x03 (3_{dec})
8008:01	Enable autoconfig	Configuration takes place automatically after the reading of the electronic type plate (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})
8008:02	Reconfig identical motor	When replacing identical motors, reconfiguration takes place automatically after reading the electronic type plate. *Enable autoconfig* must be activated. (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})
8008:03	Reconfig non-identical motor	When replacing non-identical motors, reconfiguration takes place automatically after reading the electronic type plate. *Enable autoconfig* must be activated. (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})

Index 8010 DRV Amplifier Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0	DRV Amplifier Settings	Maximum subindex	UINT8	RO	0x102 (66_{dec})
8010:01	Enable TxPDOToggle	Show TxPDO toggle in status word (bit 10)	BOOLEAN	RW	0x00 (0_{dec})
8010:02	Enable input cycle counter	1: enabled Two-bit counter that is incremented with each process data cycle up to a maximum value of 3, after which it starts again at 0 The low bit is represented in bit 10 and the high bit in bit 14 of the status word.	BOOLEAN	RW	0x00 (0_{dec})
8010:11	Device type	1: Servo drive (cannot be changed)	UINT32	RW	0x00000001 (1_{dec})
8010:12*	Current loop integral time	Integral component of current controller **Unit**: 0.1 ms This value is affected by automatic scanning. (see Automatic scanning of the electronic nameplates [▶ 153])	UINT16	RW	0x000A (10_{dec})
8010:13*	Current loop proportional gain	Proportional component of current controller **Unit**: 0.1 V/A This value is affected by automatic scanning. (see Automatic scanning of the electronic nameplates [▶ 153])	UINT16	RW	0x0064 (100_{dec})
8010:14	Velocity loop integral time	Integral component of velocity controller **Unit**: 0.1 ms	UINT32	RW	0x00000032 (50_{dec})
8010:15	Velocity loop proportional gain	Proportional component of velocity controller **Unit**: mA / (rad/s)	UINT32	RW	0x00000096 (150_{dec})
8010:17	Position loop proportional gain	Proportional component of position controller **Unit**: (rad/s) / rad	UINT32	RW	0x0000000A (10_{dec})
8010:19	Nominal DC link voltage	Rated DC link voltage **Unit**: mV	UINT32	RW	0x0000BB80 (48000_{dec})
8010:1A	Min DC link voltage	Minimum DC link voltage **Unit**: mV	UINT32	RW	0x00001A90 (6800_{dec})
8010:1B	Max DC link voltage	Maximum DC link voltage **Unit**: mV	UINT32	RW	0x0000EA60 (60000_{dec})
8010:29	Amplifier I2T warn level	I²T model warning threshold **Unit**: %	UINT8	RW	0x50 (80_{dec})
8010:2A	Amplifier I2T error level	I²T model error threshold **Unit**: %	UINT8	RW	0x69 (105_{dec})
8010:2B	Amplifier Temperature warn level	Overtemperature warning threshold **Unit**: 0.1°C	UINT16	RW	0x0320 (800_{dec})
8010:2C	Amplifier Temperature error level	Overtemperature error threshold **Unit**: 0.1°C	UINT16	RW	0x03E8 (1000_{dec})
8010:31	Velocity limitation	Speed limitation **Unit**: 1/min	UINT32	RW	0x00040000 (262144_{dec})
8010:32	Short-circuit brake duration max	Max. duration of armature short-circuit brake **Unit**: ms	UINT16	RW	0x03E8 (1000_{dec})
8010:33	Stand still window	Standstill window **Unit**: 1/min	UINT16	RW	0x0000 (0_{dec})

*) see index 0x9009 FB OCT Nameplate

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:39	Select info data 1	Selection "Info data 1" Optional display of additional information in the cyclic process data. The following information is available. 1_{dec}: Torque current (filtered 1 ms) [1000th of rated current] 2_{dec}: DC link voltage [mV] 4_{dec}: PCB temperature [0.1°C] 5_{dec}: Errors: Bit0: ADC Error Bit1: Overcurrent Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder Bit8: Watchdog 6_{dec}: Warnings: Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder 7_{dec}: I2T Motor [%] 8_{dec}: I2T Amplifier [%] 10_{dec}: Input Level: Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit8: STO Input Level (STO variant only)	UINT8	RW	0x01 (1_{dec})
8010:3A	Select info data 2	Selection "Info data 2" Optional display of additional information in the cyclic process data. The following information is available. 1_{dec}: Torque current (filtered 1 ms) [1000th of rated current] 2_{dec}: DC link voltage [mV] 4_{dec}: PCB temperature [0.1°C] 5_{dec}: Errors: Bit0: ADC Error Bit1: Overcurrent Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder Bit8: Watchdog 6_{dec}: Warnings: Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder 7_{dec}: I2T Motor [%] 8_{dec}: I2T Amplifier [%] 10_{dec}: Input Level: Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit8: STO Input Level (STO variant only)	UINT8	RW	0x01 (1_{dec})
8010:41	Low-pass filter frequency	Load filter frequency **Unit**: Hz The following values can be set: 0 Hz = Off 160 Hz 320 Hz	UINT16	RW	0x0140 (320_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:49	Halt ramp deceleration	Halt ramp deceleration **Unit**: 0.1 rad / s²	UINT32	RW	0x0000F570 (62832_{dec})
8010:50	Following error window	Following error monitoring: following error window **Unit**: The specified value must be multiplied by the corresponding scaling factor [▶ 141] 0xFFFFFFFF (-1_{dec}) = following error monitoring off Any other value = following error monitoring on	UINT32	RW	0xFFFFFFFF (-1_{dec})
8010:51	Following error time out	Following error monitoring: timeout **Unit:** ms If the following error is larger than the following error window for a time that exceeds the timeout, this leads to an error reaction.	UINT16	RW	0x0000 (0_{dec})
8010:52	Fault reaction option code	Allowed values 0: Disable drive function; motor is free to rotate 1: Slow down on slow down ramp	UINT16	RW	0x0000 (0_{dec})
8010:54	Feature bits	The adjustable motor current values can be interpreted as peak values or RMS values. The feature bit enables the conversion. Peak value → Bit 0 = 0 (default for EL7201-001x) RMS value → Bit 0 = 1 (default for EL7211-001x) In the case of the EL7201-001x the output current can be increased with the help of the ZB8610 fan cartridge. Normal output current → Bit 1 = 0 (default) Increased output current → Bit 1 = 1 This allows you to set the following combinations: 0_{dec} → normal output current interpreted as peak value 1_{dec} → normal output current interpreted as RMS value 2_{dec} → increased output current interpreted as peak value 3_{dec} → increased output current interpreted as RMS value	UINT32	RW	0x00000000 (0_{dec})
8010:57	Velocity feed forward gain	Velocity feedforward **Unit:** %	UINT8	RW	0x64 (100_{dec})
8010:65	Invert direction of rotation	Reversal of the direction of rotation	BOOLEAN	RW	0x00 (0_{dec})
8010:66	Enable cogging torque compensation	Activate cogging compensation	BOOLEAN	RW	0x00 (0_{dec})

Index 8011 DRV Motor Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8011:0	DRV Motor Settings	Maximum subindex	UINT8	RO	0x2D (45_{dec})
8011:11*	Max current	Peak current **Unit**: mA The adjustable motor current values can be interpreted as peak values or rms values. The feature bit (0x8010:54 [▶ 200]) enables the conversion. Peak value → Bit 0 = 0 (default with EL7201-001x) Rms value → Bit 0 = 1 (default with EL7211-001x) This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00001770 (6000_{dec})
8011:12*	Rated current	Rated current **Unit**: mA The adjustable motor current values can be interpreted as peak values or rms values. The feature bit (0x8010:54 [▶ 200]) enables the conversion. Peak value → Bit 0 = 0 (default with EL7201-001x) Rms value → Bit 0 = 1 (default with EL7211-001x) This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x000003E8 (1000_{dec})
8011:13*	Motor pole pairs	Number of pole pairs This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT8	RW	0x03 (3_{dec})
8011:15*	Commutation offset	Commutation offset (between electrical zero position and mechanical single-turn zero position) **Unit**:° This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	INT16	RW	0x0000 (0_{dec})
8011:16*	Torque constant	Torque constant **Unit**: mNm / A This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00000000 (0_{dec})
8011:18*	Rotor moment of inertia	Mass moment of inertia of the motor **Unit**: g cm^2 This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00000000 (0_{dec})
8011:19*	Winding inductance	Inductance **Unit**: 0.1 mH This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x000E (14_{dec})
8011:1B*	Motor speed limitation	Velocity limitation **Unit**: rpm This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00040000 (262144_{dec})

*) see index 0x9009 FB OCT Nameplate

Index (hex)	Name	Meaning	Data type	Flags	Default
8011:29	I2T warn level	I2T model warning threshold **Unit**: %	UINT8	RW	0x50 (80_{dec})
8011:2A	I2T error level	I2T model error threshold **Unit**: %	UINT8	RW	0x69 (105_{dec})
8011:2B*	Motor Temperature warn level	Overtemperature warning threshold **Unit**: 0.1 °C This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x03E8 (1000_{dec})
8011:2C*	Motor Temperature error level	Overtemperature error threshold **Unit**: 0.1 °C This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x05DC (1500_{dec})
8011:2D*	Motor thermal time constant	Thermal time constant **Unit**: 0.1 s This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0028 (40_{dec})

*) see index 0x9009 FB OCT Nameplate

Index 8012 DRV Brake Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8012:0	DRV Brake Settings	Maximum subindex	UINT8	RO	0x14 (20_{dec})
8012:01	Enable manual override	Manual release of the motor holding brake	BOOLEAN	RW	0x00 (0_{dec})
8012:02	Manual brake state	0: Release Release the brake 1: Apply Apply the brake	BOOLEAN	RW	0x00 (0_{dec})
8012:11*	Release delay	Time the holding brake requires for opening (releasing) after the current was applied This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0000 (0_{dec})
8012:12*	Application delay	Time the holding brake requires for closing (holding) after the current was switched off This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0000 (0_{dec})
8012:13	Emergency application timeout	Time the amplifier waits, until the rotation speed reaches the stand still window after stop request (set rotation speed 0 or Torque off). If the set waiting time is exceeded, the holding brake is triggered independently of the rotation speed. **Note:** For rotatory axes and the setting „torque off“ in error case, this parameter has to be set at least to the „coast to the stop“ time of the axis. For suspended (hanging) axes and the setting „torque off“ in error case, this parameter has to be set to an appropriate short time, to prevent the axis/load from drop/fall.	UINT16	RW	0x0000 (0_{dec})
8012:14*	Brake moment of inertia	Mass moment of inertia of the brake **Unit**: g cm^2 This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0000 (0_{dec})

*) see index 0x9009 FB OCT Nameplate

The following note relates to the DMC objects 0x6030, 0x7030, 0x8030 and 0x8031.

Data type INT64 for all positions in the travel distance control

The data type INT64 is used for all positions in the travel distance control.

- The single-turn position is located in the lower 32 bits.
- The multi-turn position is located in the upper 32 bits.

Index 8030 DMC settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8030:0	DMC Settings	Maximum subindex	UINT8	RO	0x14 (20_{dec})
8030:07	Emergency deceleration	Deceleration for the emergency stop ramp. (In ms from rated motor speed to standstill) **Unit:** 1 ms	UINT16	RW	0x64 (100_{dec})
8030:08	Calibration position	If homing is successful, the "Actual position" is set to this value.	INT64	RW	0x0000000000000000 (0_{dec})
8030:09	Calibration velocity (towards plc cam)	Velocity on contact with the cam in 10000ths of the rated motor speed	INT16	RW	0x0064 (100_{dec})
8030:0A	Calibration velocity (off plc cam)	Velocity on separation from the cam in 10000ths of the rated motor speed	INT16	RW	0x000A (10_{dec})
8030:0E	Modulo factor	Feedback increments for one mechanical revolution	INT64	RW	0x0000000100000000 (4294967296_{dec})
8030:12	Block calibration torque limit	Torque limitation for approaching the end stop. In 1000ths of the rated motor current.	UINT16	RW	0x64 (100_{dec})
8030:13	Block calibration stop distance	After reaching the calibration position, the axis moves out of the end position by this distance.	INT64	RW	0x0000000100000000 (4294967296_{dec})
8030:14	Block calibration lag threshold	If this lag distance is exceeded, the axis is in the end position	INT64	RW	0x0000000100000000 (4294967296_{dec})
8030:15	Target position window	Target position window: The In-Target bit is set when the axis is within this window for at least the time set under 0x8030:16.	INT64	RW	0x16c16c1
8030:16	Target position monitor time	s. 0x8030:15 time in **Unit:** ms	UINT16	RW	0x0014 (20_{dec})
8030:17	Target position timeout	When the set value generator has reached its end position and the axis is not in the target window after this time has elapsed, the task is terminated and the in-target bit is not set.	UINT16	RW	0x1770 (6000_{dec})

Index 8031 DMC features

Index (hex)	Name	Meaning	Data type	Flags	Default
8031:0	DMC Features	Maximum subindex	UINT8	RO	0x1B (27_{dec})
8031:13	Invert calibration cam search direction	Invert travel direction to search for limit switch (Default: FALSE = search with positive direction of rotation)	BOOLEAN	RW	0x00 (0_{dec})
8031:14	Invert sync impulse search direction	Direction of rotation to leave the limit switch (Default: TRUE = leave in the negative direction of rotation)	BOOLEAN	RW	0x01 (1_{dec})
8031:19	Calibration cam source	Source for the reference switch 0: Input 1 1: Input 2	ENUM8	RW	0x00 (0_{dec})
8031:1A	Calibration cam active level	State of the reference switch in the actuated state 0: Hi 1: Low	ENUM8	RW	0x00 (0_{dec})
8031:B	Latch source	Source for the latch unit 0: Input 1 1: Input 2	ENUM8	RW	0x00 (0_{dec})

7.1.3 Configuration data (vendor-specific)

Index 801F DRV Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
801F:0	DRV Vendor data	Maximum subindex	UINT8	RO	0x15 (21_{dec})
801F:11	Amplifier peak current	Peak current of the amplifier (peak value) **Unit**: mA	UINT32	RW	0x00001F40 (8000_{dec})
801F:12	Amplifier rated current	Rated current of the amplifier (peak value) **Unit**: mA	UINT32	RW	0x00000FA0 (4000_{dec})
801F:13	Amplifier thermal time constant	Thermal time constant of the amplifier **Unit**: 0.1 ms	UINT16	RW	0x0023 (35_{dec})
801F:14	Amplifier overcurrent threshold	Threshold value for short-circuit detection **Unit**: mA	UINT32	RW	0x00002EE0 (12000_{dec})
801F:15	Max rotary field frequency	Max. rotary filed frequency **Unit**: Hz	UINT16	RW	0x0257 (599_{dec})

7.1.4 Command object

Index FB00 command

Index (hex)	Name	Meaning			Data type	Flags	Default
FB00:0	DCM Command	Max. subindex			UINT8	RO	0x03 (3_{dec})
FB00:01	Request	0x1000	Clear the Diag History	Clear the Diag History	OCTET-STRING[2]	RW	{0}
		0x1100	Get build number	Read out the build number			
		0x1101	Get build date	Read out the build date			
		0x1102	Get build time	Read out the build time			
		0x8000	Software reset	Perform a software reset (hardware is re-initialized with the current CoE configuration; this otherwise happens only during the transition to INIT)			
FB00:02	Status	0	Finished, no error, no response	Command terminated without error and without response	UINT8	RO	0x00 (0_{dec})
		1	Finished, no error, response	Command terminated without error and with response			
		2	Finished, error, no response	Command terminated with error and without response			
		3	Finished, error, response	Command terminated with error and with response			
		255	Executing	Command is being executed			
FB00:03	Response	dependent on the request			OCTET-STRING[4]	RO	{0}

7.1.5 Input data

Index 6000 FB Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	FB Inputs	Maximum subindex	UINT8	RO	0x11 (17_{dec})
6000:0E	TxPDO State	TRUE: The position data are invalid FALSE: The positon data are valid	BOOLEAN	RO	0x00 (0_{dec})
6000:0F	Input Cycle Counter	Incremented with each process data cycle, switches to 0 after reaching the maximum value of 3.	BIT2	RO	0x00 (0_{dec})
6000:11	Position	Position	UINT32	RO	0x00000000 (0_{dec})

Index 6001 FB Touch probe inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	FB Touch probe inputs	Maximum subindex	UINT8	RO	0x14 (20_{dec})
6001:01	TP1 Enable	Touchprobe 1 switched on	BOOLEAN	RO	0x00 (0_{dec})
6001:02	TP1 pos value stored	Positive value of Touchprobe 1 saved	BOOLEAN	RO	0x00 (0_{dec})
6001:03	TP1 Neg value stored	Negative value of Touchprobe 1 saved	BOOLEAN	RO	0x00 (0_{dec})
6001:08	TP1 Input	Digital input Touch probe 1 The input must be addressed with a single conductor + 24 V signal.	BOOLEAN	RO	0x00 (0_{dec})
6001:09	TP2 Enable	Touchprobe 2 switched on	BOOLEAN	RO	0x00 (0_{dec})
6001:0A	TP2 pos value stored	Positive value of Touchprobe 2 saved	BOOLEAN	RO	0x00 (0_{dec})
6001:0B	TP2 neg value stored	Negative value of Touchprobe 2 saved	BOOLEAN	RO	0x00 (0_{dec})
6001:10	TP1 Input	Digital input Touch probe 2 The input must be addressed with a single conductor + 24 V signal.	BOOLEAN	RO	0x00 (0_{dec})
6001:11	TP1 pos position	Positive value of Touchprobe 1 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})
6001:12	TP1 neg position	Negative value of Touchprobe 1 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})
6001:13	TP2 pos position	Positive value of Touchprobe 2 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})
6001:14	TP2 neg position	Negative value of Touchprobe 2 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})

Index 6010 DRV Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	DRV Inputs	Maximum subindex	UINT8	RO	0x13 (19_{dec})
6010:01	Statusword	Statusword Bit 0: Ready to switch on Bit 1: Switched on Bit 2: Operation enabled Bit 3: Fault Bit 4: reserved Bit 5: Quick stop (inverse) Bit 6: Switch on disabled Bit 7: Warning Bit 8 + 9: reserved Bit 10: TxPDOToggle (selection/deselection via 0x8010:01) Bit 11: Internal limit active Bit 12: (Target value ignored) Bit 13 - 15: reserved	UINT16	RO	0x0000 (0_{dec})
6010:03	Modes of operation display	Modes of operation display. Permitted values: 9: Cyclic synchronous velocity mode (CSV) 10: Cyclic synchronous torque mode (CST) 11: Cyclic synchronous torque mode with commutation angle (CSTCA)	UINT8	RO	0x00 (0_{dec})
6010:06	Following error actual value	Following error **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})
6010:07	Velocity actual value	Display of the current velocity value **Unit:** see Index 0x9010:14 [▶ 215]	INT32	RO	0x00000000 (0_{dec})
6010:08	Torque actual value	Display of current torque value The value is specified in 1000th of the *rated current* (0x8011:12). Equation for index 0x8010:54 [▶ 200] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x8011:16 [▶ 203]) Equation for index 0x8010:54 [▶ 200] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x8011:16 [▶ 203])	INT16	RO	0x0000 (0_{dec})
6010:12	Info data 1	Synchronous information (selection via subindex 0x8010:39 [▶ 200])	UINT16	RO	0x0000 (0_{dec})
6010:13	Info data 2	Synchronous information (selection via subindex 0x8010:3A [▶ 200])	UINT16	RO	0x0000 (0_{dec})

The following note relates to the DMC objects 0x6030, 0x7030, 0x8030 and 0x8031.

Data type INT64 for all positions in the travel distance control

The data type INT64 is used for all positions in the travel distance control.

- The single-turn position is located in the lower 32 bits.
- The multi-turn position is located in the upper 32 bits.

Index 6030 DMC inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6030:0	DMC Inputs	Maximum subindex	UINT8	RO	0x3C (60_{dec})
6030:02	DMC_FeedbackStatus__Latch extern valid	An edge was detected on the external input and latched	BOOLEAN	RO	0x00 (0_{dec})
6030:03	DMC_FeedbackStatus__Set counter done	The setting of the feedback position was successful. This bit remains present until "Set counter" is released again.	BOOLEAN	RO	0x00 (0_{dec})
6030:0D	DMC_FeedbackStatus__Status of extern Latch	Status of the external latch input	BOOLEAN	RO	0x00 (0_{dec})
6030:11	DMC_DriveStatus__Ready to enable	The drive hardware is ready for activation.	BOOLEAN	RO	0x00 (0_{dec})
6030:12	DMC_DriveStatus__Ready	The drive hardware is activated.	BOOLEAN	RO	0x00 (0_{dec})
6030:13	DMC_DriveStatus__Warning	A warning is pending in the drive.	BOOLEAN	RO	0x00 (0_{dec})
6030:14	DMC_DriveStatus__Error	An error is pending in the drive. The "Ready to enable" bit and the "Ready" bit are set to FALSE.	BOOLEAN	RO	0x00 (0_{dec})
6030:15	DMC_DriveStatus__Moving positive	The axis moves in positive direction.	BOOLEAN	RO	0x00 (0_{dec})
6030:16	DMC_DriveStatus__Moving negative	The axis moves in negative direction.	BOOLEAN	RO	0x00 (0_{dec})
6030:1C	DMC_DriveStatus__Digital Input 1	Status of the first digital input	BOOLEAN	RO	0x00 (0_{dec})
6030:1D	DMC_DriveStatus__Digital Input 2	Status of the second digital input	BOOLEAN	RO	0x00 (0_{dec})
6030:21	DMC_PositioningStatus__Busy	The positioning task is running.	BOOLEAN	RO	0x00 (0_{dec})
6030:22	DMC_PositioningStatus__In-Target	The axis is at the target position.	BOOLEAN	RO	0x00 (0_{dec})
6030:23	DMC_PositoningStatus__Warning	Warning	BOOLEAN	RO	0x00 (0_{dec})
6030:24	DMC_PositoningStatus__Error	error	BOOLEAN	RO	0x00 (0_{dec})
6030:25	DMC_PositoningStatus__Calibrated	The axis is calibrated.	BOOLEAN	RO	0x00 (0_{dec})
6030:26	DMC_PositioningStatus__Accelerate	The axis accelerates.	BOOLEAN	RO	0x00 (0_{dec})
6030:27	DMC_PositioningStatus__Decelerate	The axis is decelerating.	BOOLEAN	RO	0x00 (0_{dec})
6030:28	DMC_PositoningStatus__Ready to execute	The path control is ready to accept a command. This bit is FALSE if: • The drive has a fault • The drive is not activated • As long as "PositioningControl__Execute" is pending.	BOOLEAN	RO	0x00 (0_{dec})
6030:31	DMC__Set position	Current target position specified by the ramp generator in feedback increments	INT64	RO	0x0000000000000000 (0_{dec})
6030:32	DMC__Set velocity	Current velocity specified by the ramp generator in 10000ths of the rated motor speed.	INT16	RO	0x0000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
6030:33	DMC__Actual drive time	Time since the start of the motion command in ms. Stops when the target position is reached.	UINT32	RO	0x00000000 (0_{dec})
6030:34	DMC__Actual position lag	Following error	INT64	RO	0x0000000000000000 (0_{dec})
6030:35	DMC__Actual velocity	Current velocity in 10000ths of the rated motor speed	INT16	RO	0x0000 (0_{dec})
6030:36	DMC__Actual position	Current position from the feedback (incl. possible offsets due to homing, ...)	INT64	RO	0x0000000000000000 (0_{dec})
6030:37	DMC__Error id	Error Id (identical to Diag History)	UINT32	RO	0x00000000 (0_{dec})
6030:38	DMC__Input cycle counter	Incremented with each process data cycle	UINT8	RO	0x00 (0_{dec})
6030:3A	DMC__Latch value	Feedback position at latch time	INT64	RO	0x0000000000000000 (0_{dec})
6030:3B	DMC__Cyclic info data 1	Synchronous info data 1	INT16	RO	0x0000 (0_{dec})
6030:3C	DMC__ Cyclic info data 2	Synchronous info data 2	INT16	RO	0x0000 (0_{dec})

7.1.6 Output data

Index 7001 FB Touch probe outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	FB Touch probe outputs	Maximum subindex	UINT8	RO	0x0E (14_{dec})
7001:01	TP1 Enable	Switch on Touchprobe 1	BOOLEAN	RO	0x00 (0_{dec})
7001:02	TP1 Continous	0: triggered only on the first event 1. triggered on every event	BOOLEAN	RO	0x00 (0_{dec})
7001:03	TP1 Trigger mode	Input 1 is triggered (not changeable).	BIT2	RO	0x00 (0_{dec})
7001:05	TP1 Enable pos edge	Trigger on positive edge	BOOLEAN	RO	0x00 (0_{dec})
7001:06	TP1 Enable neg edge	Trigger on negative edge	BOOLEAN	RO	0x00 (0_{dec})
7001:09	TP2 Enable	Switch on Touchprobe 2	BOOLEAN	RO	0x00 (0_{dec})
7001:0A	TP2 Continous	0: triggered only on the first event 1. triggered on every event	BOOLEAN	RO	0x00 (0_{dec})
7001:0B	TP2 Trigger mode	Input 2 is triggered (not changeable).	BIT2	RO	0x00 (0_{dec})
7001:0D	TP2 Enable pos edge	Trigger on positive edge	BOOLEAN	RO	0x00 (0_{dec})
7001:0E	TP2 Enable neg edge	Trigger on negative edge	BOOLEAN	RO	0x00 (0_{dec})

Index 7010 DRV Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	DRV Outputs	Maximum subindex	UINT8	RO	0x0E (14_{dec})
7010:01	Controlword	Controlword Bit 0: Switch on Bit 1: Enable voltage Bit 2: Quick stop (inverse) Bit 3: Enable operation Bit 4 - 6: reserved Bit 7: Fault reset Bit 8 - 15: reserved	UINT16	RO	0x0000 (0_{dec})
7010:03	Modes of operation	Permitted values: 0x08: Cyclic synchronous position mode (CSP) 0x09: Cyclic synchronous velocity mode (CSV) 0x0A: Cyclic synchronous torque mode (CST) 0x0B: Cyclic synchronous torque mode with commutation angle (CSTCA)	UINT8	RW	0x00 (0_{dec})
7010:05	Target position	Configured target position **Unit**: the value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RW	0x00000000 (0_{dec})
7010:06	Target velocity	Configured target velocity The velocity scaling can be found in index 0x9010:14 [▶ 215] (Velocity encoder resolution).	INT32	RO	0x00000000 (0_{dec})
7010:09	Target torque	Configured input value for torque monitoring The value is specified in 1000th of the *rated current* (0x8011:12 [▶ 204]). Equation for index 0x8010:54 [▶ 200] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x8011:16 [▶ 203]) Equation for index 0x8010:54 [▶ 200] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x8011:16 [▶ 203])	INT16	RO	0x0000 (0_{dec})
7010:0A	Torque offset	Torque value offset The value is specified in 1000th of the *rated current* (0x8011:12 [▶ 204]). Equation for index 0x8010:54 [▶ 200] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x8011:16 [▶ 203]) Equation for index 0x8010:54 [▶ 200] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x8011:16 [▶ 203])	INT16	RO	0x0000 (0_{dec})
7010:0B	Torque limitation	Torque threshold value for torque monitoring (bipolar limit) The value is specified in 1000th of the *rated current* (0x8011:12 [▶ 204]). Equation for index 0x8010:54 [▶ 200] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x8011:16 [▶ 203]) Equation for index 0x8010:54 [▶ 200] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x8011:16 [▶ 203])	UINT16	RW	0x7FFF (32767_{dec})
7010:0E	Commutation angle	Commutation angle (for CSTCA mode) **Unit:** 360° / 2^{16}	UINT16	RO	0x0000 (0_{dec})

The following note relates to the DMC objects 0x6030, 0x7030, 0x8030 and 0x8031.

Data type INT64 for all positions in the travel distance control

The data type INT64 is used for all positions in the travel distance control.

- The single-turn position is located in the lower 32 bits.
- The multi-turn position is located in the upper 32 bits.

Index 7030 DMC outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7030:0	DMC Outputs	Maximum subindex	UINT8	RO	0x36 (54_{dec})
7030:02	DMC_FeedbackControl__Enable latch extern on positive edge	Latches to the positive edge of the external input	BOOLEAN	RO	0x00 (0_{dec})
7030:03	DMC_FeedbackControl__Set counter	With a rising edge, "Actual position" is set to the value of "Set counter value".	BOOLEAN	RO	0x00 (0_{dec})
7030:04	DMC_FeedbackControl__Enable latch extern on negative edge	Latches to the negative edge of the external input	BOOLEAN	RO	0x00 (0_{dec})
7030:11	DMC_DriveControl__Enable	Activate drive	BOOLEAN	RO	0x00 (0_{dec})
7030:12	DMC_DriveControl__Reset	Perform a reset of the drive hardware	BOOLEAN	RO	0x00 (0_{dec})
7030:21	DMC_PositionControl__Execute	Start motion command with rising edge The task runs as long as this bit is set or until the command is completed. If the level drops during travel, the axis is brought to a standstill with the deceleration specified for the task.	BOOLEAN	RO	0x00 (0_{dec})
7030:22	DMC_PositionControl__Emergency stop	In the event of a rising edge, decelerate to a standstill with the emergency stop ramp	BOOLEAN	RO	0x00 (0_{dec})
7030:31	DMC__Set counter value	s. 0x7030:03	INT64	RO	0x0000000000000000 (0_{dec})
7030:32	DMC__Target position	Position specification in feedback increments	INT64	RO	0x0000000000000000 (0_{dec})
7030:33	DMC__Target velocity	Maximum velocity during the motion command in 10000ths of the rated motor speed	INT16	RO	0x0000 (0_{dec})
7030:34	DMC__Start type	Type of positioning task: 0x0001: Absolute 0x0002: Relative 0x0003: Endless + 0x0004: Endless - 0x0105: Modulo short 0x0205: Modulo + 0x0305: Modulo - 0x6000: Cali PLC cam 0x6200: Cali Block 0x6E00: Cali set 0x6F00: Cali clear	UINT16	RO	0x0000 (0_{dec})
7030:35	DMC__Target acceleration	Acceleration: Time in ms from standstill to reaching the rated motor speed	UINT16	RO	0x0000 (0_{dec})
7030:36	DMC__Target deceleration	Delay: Time in ms for deceleration from rated motor speed to standstill	UINT16	RO	0x0000 (0_{dec})

7.1.7 Information / diagnosis data

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x37 (55_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages. A maximum of 50 messages can be stored	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[28]	RO	{0}
...	...	...	...	...	...
10F3:37	Diagnosis Message 050	Message 50	OCTET-STRING[28]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Time stamp	UINT64	RO	

Index 9008 FB OCT Info data

(these data are always read in automatically from the electronic type plate of the motor and serve purely informative purposes)

Index (hex)	Name	Meaning	Data type	Flags	Default
9008:0	FB OCT Info data	Maximum subindex	UINT8	RO	0x20 (32_{dec})
9008:11	Encoder Type	Feedback type 2: rotary encoder, unipolar counting	UINT16	RO	0x0000 (0_{dec})
9008:12	Resolution	Resolution of the feedback **Unit**: Steps per revolution	UINT32	RO	0x00000000 (0_{dec})
9008:13	Range	Working range of the feedback. On leaving this range there is an overflow of the position. **Unit**: Revolutions	UINT32	RO	0x00000000 (0_{dec})
9008:14	Type Code Name	Name of the feedback	STRING	RO	
9008:15	Serial No	Serial number of the feedback	STRING	RO	
9008:16	Firmware Revision No	Revision of the firmware	STRING	RO	
9008:17	Firmware Date	Date of the firmware	STRING	RO	
9008:18	EEPROM Size	EEPROM size	UINT16	RO	0x0000 (0_{dec})
9008:19	Temperature	Temperature **Unit**: 0,1°	INT16	RO	0x0000 (0_{dec})
9008:1A	LED Current	Current of the feedback LED **Unit**: 0.1 mA	UINT16	RO	0x0000 (0_{dec})
9008:1B	Supply voltage	Supply voltage of the feedback **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9008:1C	Life- time	Operating hour counter **Unit**: Minutes	UINT32	RO	0x00000000 (0_{dec})
9008:1D	Received Signal Strength Indicator	Received signal strength at the terminal **Unit**: %	UINT16	RO	0x0000 (0_{dec})
9008:1E	Slave Received Signal Strength Indicator	Received signal strength at the encoder **Unit**: %	UINT16	RO	0x0000 (0_{dec})
9008:1F	Line delay	Running time of the signal in the cable **Unit**: ns	UINT16	RO	0x0000 (0_{dec})
9008:20	Encoder position offset	Positon offset stored in the motor feedback	UINT32	RO	0x00000000 (0_{dec})

Index 9009 FB OCT Nameplate

The parameters described in this index are always read from the electronic type plate of the connected motor. These parameters automatically lead to the parameters marked with an asterisk (*) in this chapter, if automatic scanning of the electronic type plate is switched on (index 0x8008 [▶ 199]).

Index (hex)	Name	Meaning	Data type	Flags	Default
9009:0	FB OCT Nameplate	Maximum subindex	UINT8	RO	0x24 (36_{dec})
9009:01	Motor vendor	Motor vendor	STRING	RO	
9009:02	Electric motor type	Motor type	STRING	RO	
9009:03	Serial No	Serial number	STRING	RO	
9009:04	Order code	Order number (In case of Autoconfig a check is made on the basis of this index as to whether the motor is identical to the predecessor)	STRING	RO	
9009:05	Motor construction	Type of motor	STRING	RO	
9009:06	Pole pairs	Number of pole pairs	UINT32	RO	0x00000000 (0_{dec})
9009:07	Standstill current (rms)	Effective holding current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
9009:08	Rated current (rms)	Effective rated current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
9009:09	Peak current (rms)	Effective peak current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
9009:0A	Nominal voltage (rms)	Effective nominal voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9009:0B	Max voltage (rms)	Maximum voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9009:0C	Max winding du/dt	Maximum permissible voltage rise on the winding **Unit**: kV/s	UINT32	RO	0x00000000 (0_{dec})
9009:0D	Max torque	Maximum torque **Unit**: mNm	UINT16	RO	0x0000 (0_{dec})
9009:0E	Torque constant	Torque constant **Unit**: mNm / A	UINT16	RO	0x0000 (0_{dec})
9009:0F	EMK (rms)	Reverse voltage **Unit**: mV / (rpm)	UINT32	RO	0x00000000 (0_{dec})
9009:10	Winding resistance Ph-Ph 20°C	Coil resistance **Unit**: mOhm	UINT16	RO	0x0000 (0_{dec})
9009:11	Ld Ph-Ph	Inductance in the direction of flow **Unit**: 0.1 mH	UINT16	RO	0x0000 (0_{dec})
9009:12	Lq Ph-Ph	Inductance in the torque-forming direction **Unit**: 0.1 mH	UINT16	RO	0x0000 (0_{dec})
9009:13	Max speed	Maximum speed **Unit**: rpm	UINT32	RO	0x00000000 (0_{dec})
9009:14	Moment of inertia	Mass moment of inertia **Unit**: g cm^2	UINT16	RO	0x0000 (0_{dec})
9009:15	T motor warn limit	Motor temperature warning threshold **Unit**: 0.1°C	UINT16	RO	0x0000 (0_{dec})
9009:16	T motor shut down	Motor temperature error threshold **Unit**: 0.1°C	UINT16	RO	0x0000 (0_{dec})
9009:17	Time constant i2t	Time constant I2T model **Unit**: s	UINT16	RO	0x0000 (0_{dec})
9009:18	Motor thermal constant	Thermal time constant of the motor **Unit**: s	UINT16	RO	0x0000 (0_{dec})
9009:1B	Brake type	Brake type • no Brake • holding Brake	STRING	RO	
9009:1C	Min brake voltage	Minimum brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9009:1D	Max brake voltage	Maximum brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9009:1E	Min brake monitor current	Minimum current for the monitoring of the brake **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
9009:1F	Brake holding torque	Brake holding torque **Unit**: mNm	UINT32	RO	0x00000000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
9009:20	Brake T on	Time until the brake is applied **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
9009:21	Brake T off	Time until the brake is released **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
9009:22	Brake reduced holding voltage	Reduced brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9009:23	Brake time to red. holding volt.	Time from which the brake holds with reduced voltage **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
9009:24	Motor temp sensor connection	Temperature sensor connection Feedback port (not changeable)	STRING	RO	

Index 9010 DRV Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
9010:0	DRV Info data	Maximum subindex	UINT8	RO	0x17 (23_{dec})
9010:11	Amplifier temperature	Internal terminal temperature **Unit**: 0.1 °C	UINT16	RO	0x0000 (0_{dec})
9010:12	DC link voltage	DC link voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
9010:13	Supported drive modes	Information about supported drive modes. (DS402: object 0x6502) Only modes *CSV*, *CST*, *CSTCA and CSP* are supported Bit 0: PP Bit 1: VL Bit 2: PV Bit 3: TQ Bit 4: R Bit 5: HM Bit 6: IP Bit 7: CSP Bit 8: CSV Bit 9: CST Bit 10: CSTCA Bit 11 - 15: reserved Bit 16-31: Manufacturer-specific	UINT32	RO	0x00000000 (0_{dec})
9010:14	Velocity encoder resolution	Display of configured encoder increments/s and motor revolutions/s. The velocity encoder resolution is calculated according to the following formula: Velocity Encoder Resolution = (encoder_increments / s) / (motor_revolutions / s)	UINT32	RO	0x00041893 (268435_{dec})
9010:15	Position encoder resolution increments	Feedback increments per motor revolution	UINT32	RO	0x00000000 (0_{dec})
9010:16	Position encoder resolution revolutions	Motor revolutions Position encoder resolution = encoder increments (Index 0x9010:15) / motor revolutions (Index 0x9010:16)	UINT32	RO	0x00000000 (0_{dec})
9010:17	Cogging compensation supported	The cogging compensation data are available in the motor's electronic identification plate.	BOOLEAN	RO	0x00 (0_{dec})

Index 9018 DRV Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
9018:0	DRV Info data	Maximum subindex	UINT8	RO	0x11 (17_{dec})
9018:11	Auxiliary voltage (10 V)	Auxiliary voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})

Index A010 DRV Amplifier Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	DRV Amplifier Diag data	Maximum subindex	UINT8	RO	0x11 (17_{dec})
A010:11	Amplifier I2T temperature	I2T model load **Unit**: %	UINT8	RO	0x00 (0_{dec})

Index A011 DRV Motor Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A011:0	DRV Motor Diag data	Maximum subindex	UINT8	RO	0x13 (19_{dec})
A011:11	Motor I2T temperature	I2T model load **Einheit**: %	UINT8	RO	0x00 (0_{dec})
A011:13	Motor temperature	Temperature utilization ratio **Unit**: °	INT16	RO	0x0000 (0_{dec})

Index B001 FB OCT Memory interface

Index (hex)	Name	Meaning	Data type	Flags	Default
B001:0	FB OCT Memory interface	Maximum subindex	UINT8	RO	0x06 (6_{dec})
B001:01	cmd	3_{dec}: Enc Write 6_{dec}: Enc Read Direct 7_{dec}: Enc Read Indirect (Default) 9_{dec}: Enc Reset 10_{dec}: IP Write 15_{dec}: IP Read 16_{dec}: Write encoder position offset	INT16	RW	0x0007 (7_{dec})
B001:02	Len		INT16	RW	0x0000 (0_{dec})
B001:03	Adr		UINT32	RW	0x00000000 (0_{dec})
B001:04	Offset		UINT32	RW	0x00000000 (0_{dec})
B001:05	Ctrl/Status	0_{dec}: Init (Default) 1_{dec}: Execute 2_{dec}: Busy 3_{dec}: Done 4_{dec}: Error	INT16	RW	0x0000 (0_{dec})
B001:06	Data		OCTET-STRING[32]	RW	0x00000000 (0_{dec})

7.1.8 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 CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL72x1-x01x

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x1C213052 (471937106_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index 1430 DMC RxPDO-Par Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1430:0	DMC RxPDO-Par Outputs	PDO Parameter RxPDO 10	UINT8	RO	0x06 (6_{dec})
1430:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 10	OCTET-STRING[6]	RO	00 16 01 16 02 16 03 16 04 16 05 16 06 16 07 16 08 16 31 16

Index 1431 DMC RxPDO-Par Outputs 32-bit

Index (hex)	Name	Meaning	Data type	Flags	Default
1431:0	DMC RxPDO-Par Outputs 32 Bit	PDO Parameter RxPDO 11	UINT8	RO	0x06 (6_{dec})
1431:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 11	OCTET-STRING[6]	RO	00 16 01 16 02 16 03 16 04 16 05 16 06 16 07 16 08 16 30 16

Index 1600 DRV RxPDO-Map Controlword

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DRV RxPDO-Map Controlword	PDO Mapping RxPDO 1	UINT8	RO	0x01 (1_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x01 (Controlword))	UINT32	RO	0x7010:01, 16

Index 1601 DRV RxPDO-Map Target velocity

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DRV RxPDO-Map Target velocity	PDO Mapping RxPDO 2	UINT8	RO	0x01 (1_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x06 (Target velocity))	UINT32	RO	0x7010:06, 32

Index 1602 DRV RxPDO-Map Target torque

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	DRV RxPDO-Map Target torque	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x09 (Target torque))	UINT32	RO	0x7010:09, 16

Index 1603 DRV RxPDO-Map Commutation angle

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	DRV RxPDO-Map Commutation angle	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x0E (Commutation angle))	UINT32	RO	0x7010:0E, 16

Index 1604 DRV RxPDO-Map Torque limitation

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	DRV RxPDO-Map Torque limitation	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x0B (Torque limitation))	UINT32	RO	0x7010:0B, 16

Index 1605 DRV RxPDO-Map Torque offset

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	DRV RxPDO-Map Torque offset	PDO Mapping RxPDO 6	UINT8	RO	0x01 (1_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x0A (Torque offset))	UINT32	RO	0x7010:0A, 16

Index 1606 DRV RxPDO-Map Target position

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	DRV RxPDO-Map Target position	PDO Mapping RxPDO 7	UINT8	RO	0x01 (1_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x05 (Target postion))	UINT32	RO	0x7010:05, 32

Index 1607 FB RxPDO-Map Touch probe control

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	FB RxPDO-Map Touch probe control	PDO Mapping RxPDO 8	UINT8	RO	0x0C (12_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x01 (TP1 Enable))	UINT32	RO	0x7001:01, 1
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x02 (TP1 Continous))	UINT32	RO	0x7001:02, 1
1607:03	SubIndex 003	3. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x03 (TP1 Trigger mode))	UINT32	RO	0x7001:03, 2
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x05 (TP1 Enable pos edge))	UINT32	RO	0x7001:05, 1
1607:05	SubIndex 005	5. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x06 (TP1 Enable neg edge))	UINT32	RO	0x7001:06, 1
1607:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1607:07	SubIndex 007	7. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x09 (TP2 Enable))	UINT32	RO	0x7001:09, 1
1607:08	SubIndex 008	8. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0A (TP2 Continous))	UINT32	RO	0x7001:0A, 1
1607:09	SubIndex 009	9. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0B (TP2 Trigger mode))	UINT32	RO	0x7001:0B, 2
1607:0A	SubIndex 010	10. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0D (TP2 Enable pos edge))	UINT32	RO	0x7001:0D, 1
1607:0B	SubIndex 011	11. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0E (TP2 Enable neg edge))	UINT32	RO	0x7001:0E, 1
1607:0C	SubIndex 012	12. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2

Index 1608 DRV RxPDO-Map Modes of operation

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	DRV RxPDO-Map Modes of operation	PDO Mapping RxPDO 9	UINT8	RO	0x01 (1_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x03 (Modes of operation))	UINT32	RO	0x7010:03, 8

Index 1630 DMC RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1630:0	DMC RxPDO-Map Outputs	PDO Mapping RxPDO 10	UINT8	RO	0x12 (18_{dec})
1630:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1630:02	SubIndex 002	2. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x02 (DMC_FeedbackControl __Enable latch extern on positive edge))	UINT32	RO	0x7030:02, 1
1630:03	SubIndex 003	3. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x03 (DMC_FeedbackControl __Set counter))	UINT32	RO	0x7030:03, 1
1630:04	SubIndex 004	4. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x04 (DMC_FeedbackControl __Enable latch extern on negative edge))	UINT32	RO	0x7030:04, 1
1630:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1630:06	SubIndex 006	6. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x11 (DMC_DriveControl __Enable))	UINT32	RO	0x7030:11, 1
1630:07	SubIndex 007	7. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x12 (DMC_DriveControl __Reset))	UINT32	RO	0x7030:12, 1
1630:08	SubIndex 008	8. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1630:09	SubIndex 009	9. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x21 (DMC_PositionControl __Execute))	UINT32	RO	0x7030:21, 1
1630:0A	SubIndex 010	10. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x22 (DMC_PositionControl __Emergency stop))	UINT32	RO	0x7030:22, 1
1630:0B	SubIndex 011	11. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1630:0C	SubIndex 012	12. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x31 (DMC__Set counter value))	UINT32	RO	0x7030:31, 64
1630:0D	SubIndex 013	13. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x32 (DMC__Target position))	UINT32	RO	0x7030:32, 64
1630:0E	SubIndex 014	14. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x33 (DMC__Target velocity))	UINT32	RO	0x7030:33, 16
1630:0F	SubIndex 015	15. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x34 (DMC__Start types))	UINT32	RO	0x7030:34, 16
1630:10	SubIndex 016	16. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x35 (DMC__Target acceleration))	UINT32	RO	0x7030:35, 16
1630:11	SubIndex 017	17. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x36 (DMC__Target deceleration))	UINT32	RO	0x7030:36, 16
1630:12	SubIndex 018	18. PDO Mapping entry (80 bits align)	UINT32	RO	0x0000:00, 80

Index 1631 DMC RxPDO-Map Outputs 32-bit

Index (hex)	Name	Meaning	Data type	Flags	Default
1631:0	DMC RxPDO-Map Outputs 32	PDO Mapping RxPDO 11	UINT8	RO	0x14 (20_{dec})
1631:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1631:02	SubIndex 002	2. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x02 (DMC_FeedbackControl __Enable latch extern on positive edge))	UINT32	RO	0x7030:02, 1
1631:03	SubIndex 003	3. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x03 (DMC_FeedbackControl __Set counter))	UINT32	RO	0x7030:03, 1
1631:04	SubIndex 004	4. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x04 (DMC_FeedbackControl __Enable latch extern on negative edge))	UINT32	RO	0x7030:04, 1
1631:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1631:06	SubIndex 006	6. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x11 (DMC_DriveControl __Enable))	UINT32	RO	0x7030:11, 1
1631:07	SubIndex 007	7. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x12 (DMC_DriveControl __Reset))	UINT32	RO	0x7030:12, 1
1631:08	SubIndex 008	8. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1631:09	SubIndex 009	9. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x21 (DMC_PositionControl __Execute))	UINT32	RO	0x7030:21, 1
1631:0A	SubIndex 010	10. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x22 (DMC_PositionControl __Emergency stop))	UINT32	RO	0x7030:22, 1
1631:0B	SubIndex 011	11. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1631:0C	SubIndex 012	12. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x31 (DMC__Set counter value))	UINT32	RO	0x7030:31, 32
1631:0D	SubIndex 013	13. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1631:0E	SubIndex 014	14. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x32 (DMC__Target position))	UINT32	RO	0x7030:32, 16
1631:0F	SubIndex 015	15. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1631:10	SubIndex 016	16. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x33 (DMC__Target velocity))	UINT32	RO	0x7030:33, 16
1631:11	SubIndex 017	17. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x34 (DMC__Start types))	UINT32	RO	0x7030:34, 16
1631:12	SubIndex 018	18. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x35 (DMC__Target acceleration))	UINT32	RO	0x7030:35, 16
1631:13	SubIndex 019	19. PDO Mapping entry (object 0x7030 (DMC outputs), entry 0x36 (DMC__Target deceleration))	UINT32	RO	0x7030:36, 16
1631:14	SubIndex 020	20. PDO Mapping entry (80 bits align)	UINT32	RO	0x0000:00, 80

Index 1830 DMC TxPDO-Par Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1830:0	DMC TxPDO-Par Inputs	PDO Parameter TxPDO 15	UINT8	RO	0x06 (6_{dec})
1830:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 15	OCTET-STRING[32]	RO	00 1A 01 1A 02 1A 03 1A 04 1A 05 1A 06 1A 07 1A 08 1A 09 1A 0A 1A 0B 1A 0C 1A 0E 1A 31 1A

Index 1831 DMC TxPDO-Par Inputs 32-bit

Index (hex)	Name	Meaning	Data type	Flags	Default
1831:0	DMC TxPDO-Par Inputs 32 Bit	PDO Parameter TxPDO 16	UINT8	RO	0x06 (6_{dec})
1831:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 16	OCTET-STRING[32]	RO	00 1A 01 1A 02 1A 03 1A 04 1A 05 1A 06 1A 07 1A 08 1A 09 1A 0A 1A 0B 1A 0C 1A 0E 1A 30 1A

Index 1A00 FB TxPDO-Map Position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	FB TxPDO-Map Position	PDO Mapping TxPDO 1	UINT8	RO	0x01 (1_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (FB Inputs), entry 0x11 (Position))	UINT32	RO	0x6000:11, 32

Index 1A01 DRV TxPDO-Map Statusword

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	DRV TxPDO-Map Statusword	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x01 (Statusword))	UINT32	RO	0x6010:01, 16

Index 1A02 DRV TxPDO-Map Velocity actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	DRV TxPDO-Map Velocity actual value	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x07 (Velocity actual value))	UINT32	RO	0x6010:07, 32

Index 1A03 DRV TxPDO-Map Torque actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	DRV TxPDO-Map Torque actual value	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x08 (Torque actual value))	UINT32	RO	0x6010:08, 16

Index 1A04 DRV TxPDO-Map Info data 1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	DRV TxPDO-Map Info data 1	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x12 (Info data 1))	UINT32	RO	0x6010:12, 16

Index 1A05 DRV TxPDO-Map Info data 2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	DRV TxPDO-Map Info data 2	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x13 (Info data 2))	UINT32	RO	0x6010:13, 16

Index 1A06 DRV TxPDO-Map Following error actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	DRV TxPDO-Map Following error actual value	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x06 (Following error actual value))	UINT32	RO	0x6010:06, 32

Index 1A07 FB TxPDO-Map Touch probe status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	FB TxPDO-Map Touch probe status	PDO Mapping TxPDO 8	UINT8	RO	0x0A (10_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x01 (TP1 Enable))	UINT32	RO	0x6001:01, 1
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x02 (TP1 pos value stored))	UINT32	RO	0x6001:02, 1
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x03 (TP1 Neg value stored))	UINT32	RO	0x6001:03, 1
1A07:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x08 (TP1 Input))	UINT32	RO	0x6001:08, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x09 (TP2 Enable))	UINT32	RO	0x6001:09, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0A (TP2 pos value stored))	UINT32	RO	0x6001:0A, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0B (TP2 neg value stored))	UINT32	RO	0x6001:0B, 1
1A07:09	SubIndex 009	9. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A07:0A	SubIndex 010	10. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x10 (TP2 Input))	UINT32	RO	0x6001:10, 1

Index 1A08 FB TxPDO-Map Touch probe 1 pos position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	FB TxPDO-Map Touch probe 1 pos position	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FP Touch probe inputs), entry 0x11 (TP1 Pos position))	UINT32	RO	0x6001:11, 32

Index 1A09 FB TxPDO-Map Touch probe 1 neg position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	FB TxPDO-Map Touch probe 1 neg position	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x12 (TP1 Neg position))	UINT32	RO	0x6001:12, 32

Index 1A0A FB TxPDO-Map Touch probe 2 pos position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	FB TxPDO-Map Touch probe 2 pos position	PDO Mapping TxPDO 11	UINT8	RO	0x01 (1_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x13 (TP2 Pos position))	UINT32	RO	0x6001:13, 32

Index 1A0B FB TxPDO-Map Touch probe 2 neg position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0B:0	FB TxPDO-Map Touch probe 2 neg position	PDO Mapping TxPDO 12	UINT8	RO	0x01 (1_{dec})
1A0B:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x14 (TP2 neg position))	UINT32	RO	0x6001:14, 32

Index 1A0C FB TxPDO-Map Touch probe status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0C:0	FB TxPDO-Map Touch probe status	PDO Mapping TxPDO 13	UINT8	RO	0x03 (3_{dec})
1A0C:01	SubIndex 001	1. PDO Mapping entry (13 bits align)	UINT32	RO	0x0000:00, 13
1A0C:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (FB Inputs), entry 0x0E (TxPDO State))	UINT32	RO	0x6000:0E, 1
1A0C:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (FB Inputs), entry 0x0F (Input Cycle Counter))	UINT32	RO	0x6000:0F, 2

Index 1A0E DRV TxPDO-Map Modes of operation display

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0E:0	DRV TxPDO-Map Modes of operation display	PDO Mapping TxPDO 14	UINT8	RO	0x01 (1_{dec})
1A0E:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x03 (Modes of operation display))	UINT32	RO	0x6010:03, 8

Index 1A30 DMC TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A30:0	DMC RxPDO-Map Inputs	PDO Mapping TxPDO 15	UINT8	RO	0x38 (26_{dec})
1A30:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A30:02	SubIndex 002	2. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x02 (DMC_FeedbackStatus __Latch extern valid))	UINT32	RO	0x6030:02, 1
1A30:03	SubIndex 003	3. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x03 (DMC_FeedbackStatus __Set counter done))	UINT32	RO	0x6030:03, 1
1A30:04	SubIndex 004	4. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A30:05	SubIndex 005	5. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x0D (DMC_FeedbackStatus __Status of extern latch))	UINT32	RO	0x6030:0D, 1
1A30:06	SubIndex 006	6. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A30:07	SubIndex 007	7. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x11 (DMC_DriveStatus __Ready to enable))	UINT32	RO	0x6030:11, 1
1A30:08	SubIndex 008	8. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x12 (DMC_DriveStatus __Ready))	UINT32	RO	0x6030:12, 1
1A30:09	SubIndex 009	9. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x13 (DMC_DriveStatus __Warning))	UINT32	RO	0x6030:13, 1
1A30:0A	SubIndex 010	10. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x14 (DMC_DriveStatus __Error))	UINT32	RO	0x6030:14, 1
1A30:0B	SubIndex 011	11. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x15 (DMC_DriveStatus __Moving positive))	UINT32	RO	0x6030:15, 1
1A30:0C	SubIndex 012	12. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x16 (DMC_DriveStatus __Moving negative))	UINT32	RO	0x6030:16, 1
1A30:0D	SubIndex 013	13. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A30:0E	SubIndex 014	14. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x1C (DMC_DriveStatus __Digital input 1))	UINT32	RO	0x6030:1C, 1
1A30:0F	SubIndex 015	15. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x1D (DMC_DriveStatus __Digital input 2))	UINT32	RO	0x6030:1D, 1
1A30:10	SubIndex 016	16. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A30:11	SubIndex 017	17. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x21 (DMC_PositioningStatus __Busy))	UINT32	RO	0x6030:21, 1
1A30:12	SubIndex 018	18. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x22 (DMC_PositioningStatus __In-Target))	UINT32	RO	0x6030:22, 1
1A30:13	SubIndex 019	19. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x23 (DMC_PositioningStatus __Warning))	UINT32	RO	0x6030:23, 1
1A30:14	SubIndex 020	20. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x24 (DMC_PositioningStatus __Error))	UINT32	RO	0x6030:24, 1
1A30:15	SubIndex 021	21. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x25 (DMC_PositioningStatus __Calibrated))	UINT32	RO	0x6030:25, 1
1A30:16	SubIndex 022	22. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x26 (DMC_PositioningStatus __Accelerate))	UINT32	RO	0x6030:26, 1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A30:17	SubIndex 023	23. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x27 (DMC_PositioningStatus __Decelerate))	UINT32	RO	0x6030:27, 1
1A30:18	SubIndex 024	24. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x28 (DMC_PositioningStatus __Ready to execute))	UINT32	RO	0x6030:28, 1
1A30:19	SubIndex 025	25. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A30:1A	SubIndex 026	26. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x31 (DMC_Set position))	UINT32	RO	0x6030:31, 64
1A30:1B	SubIndex 027	27. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x32 (DMC_Set velocity))	UINT32	RO	0x6030:32, 16
1A30:1C	SubIndex 028	28. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x33 (DMC_Actual dirve time))	UINT32	RO	0x6030:33, 32
1A30:1D	SubIndex 029	29. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x34 (DMC_Actual position lag))	UINT32	RO	0x6030:34, 64
1A30:1E	SubIndex 030	30. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x35 (DMC_Actual velocity))	UINT32	RO	0x6030:35, 16
1A30:1F	SubIndex 031	31. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x36 (DMC_Actual position))	UINT32	RO	0x6030:36, 64
1A30:20	SubIndex 032	32. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x37 (DMC_Error id))	UINT32	RO	0x6030:37, 32
1A30:21	SubIndex 033	33. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x38 (DMC_Input cycle counter))	UINT32	RO	0x6030:38, 8
1A30:22	SubIndex 034	34. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A30:23	SubIndex 035	35. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3A (DMC_Latch value))	UINT32	RO	0x6030:3A, 64
1A30:24	SubIndex 036	36. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3B (DMC_Cyclic info data 1))	UINT32	RO	0x6030:3B, 16
1A30:25	SubIndex 037	37. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3C (DMC_Cyclic info data 2))	UINT32	RO	0x6030:3C, 16
1A30:26	SubIndex 038	38. PDO Mapping entry (64 bits align)	UINT32	RO	0x0000:00, 64

Index 1A31 DMC TxPDO-Map Inputs 32-bit

Index (hex)	Name	Meaning	Data type	Flags	Default
1A31:0	DMC TxPDO-Map Inputs 32 Bit	PDO Mapping TxPDO 16	UINT8	RO	0x42 ($2A_{dec}$)
1A31:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A31:02	SubIndex 002	2. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x02 (DMC_FeedbackStatus __Latch extern valid))	UINT32	RO	0x6030:02, 1
1A31:03	SubIndex 003	3. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x03 (DMC_FeedbackStatus __Set counter done))	UINT32	RO	0x6030:03, 1
1A31:04	SubIndex 004	4. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A31:05	SubIndex 005	5. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x0D (DMC_FeedbackStatus __Status of extern latch))	UINT32	RO	0x6030:0D, 1
1A31:06	SubIndex 006	6. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A31:07	SubIndex 007	7. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x11 (DMC_DriveStatus __Ready to enable))	UINT32	RO	0x6030:11, 1
1A31:08	SubIndex 008	8. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x12 (DMC_DriveStatus __Ready))	UINT32	RO	0x6030:12, 1
1A31:09	SubIndex 009	9. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x13 (DMC_DriveStatus __Warning))	UINT32	RO	0x6030:13, 1
1A31:0A	SubIndex 010	10. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x14 (DMC_DriveStatus __Error))	UINT32	RO	0x6030:14, 1
1A31:0B	SubIndex 011	11. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x15 (DMC_DriveStatus __Moving positive))	UINT32	RO	0x6030:15, 1
1A31:0C	SubIndex 012	12. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x16 (DMC_DriveStatus __Moving negative))	UINT32	RO	0x6030:16, 1
1A31:0D	SubIndex 013	13. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A31:0E	SubIndex 014	14. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x1C (DMC_DriveStatus __Digital input 1))	UINT32	RO	0x6030:1C, 1
1A31:0F	SubIndex 015	15. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x1D (DMC_DriveStatus __Digital input 2))	UINT32	RO	0x6030:1D, 1
1A31:10	SubIndex 016	16. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A31:11	SubIndex 017	17. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x21 (DMC_PositioningStatus __Busy))	UINT32	RO	0x6030:21, 1
1A31:12	SubIndex 018	18. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x22 (DMC_PositioningStatus __In-Target))	UINT32	RO	0x6030:22, 1
1A31:13	SubIndex 019	19. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x23 (DMC_PositioningStatus __Warning))	UINT32	RO	0x6030:23, 1
1A31:14	SubIndex 020	20. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x24 (DMC_PositioningStatus __Error))	UINT32	RO	0x6030:24, 1
1A31:15	SubIndex 021	21. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x25 (DMC_PositioningStatus __Calibrated))	UINT32	RO	0x6030:25, 1
1A31:16	SubIndex 022	22. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x26 (DMC_PositioningStatus __Accelerate))	UINT32	RO	0x6030:26, 1
1A31:17	SubIndex 023	23. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x27 (DMC_PositioningStatus __Decelerate))	UINT32	RO	0x6030:27, 1
1A31:18	SubIndex 024	24. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x28 (DMC_PositioningStatus __Ready to execute))	UINT32	RO	0x6030:28, 1
1A31:19	SubIndex 025	25. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A31:1A	SubIndex 026	26. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x31 (DMC_Set position))	UINT32	RO	0x6030:31, 32
1A31:1B	SubIndex 027	27. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A31:1C	SubIndex 028	28. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x32 (DMC_Set velocity))	UINT32	RO	0x6030:32, 16
1A31:1D	SubIndex 029	29. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x33 (DMC_Actual dirve time))	UINT32	RO	0x6030:33, 32
1A31:1E	SubIndex 030	30. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x34 (DMC_Actual position lag))	UINT32	RO	0x6030:34, 32
1A31:1F	SubIndex 031	31. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A31:20	SubIndex 032	32. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x35 (DMC_Actual velocity))	UINT32	RO	0x6030:35, 16

Index (hex)	Name	Meaning	Data type	Flags	Default
1A31:21	SubIndex 033	33. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x36 (DMC_Actual position))	UINT32	RO	0x6030:36, 32
1A31:22	SubIndex 034	34. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A31:23	SubIndex 035	35. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x37 (DMC_Error id))	UINT32	RO	0x6030:37, 32
1A31:24	SubIndex 036	36. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x38 (DMC_Input cycle counter))	UINT32	RO	0x6030:38, 8
1A31:25	SubIndex 037	37. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A31:26	SubIndex 038	38. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3A (DMC_Latch value))	UINT32	RO	0x6030:3A, 32
1A31:27	SubIndex 039	39. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A31:28	SubIndex 040	40. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3B (DMC_Cyclic info data 1))	UINT32	RO	0x6030:3B, 16
1A31:29	SubIndex 041	41. PDO Mapping entry (object 0x6030 (DMC inputs), entry 0x3C (DMC_Cyclic info data 2))	UINT32	RO	0x6030:3C, 16
1A31:2A	SubIndex 042	42. PDO Mapping entry (64 bits align)	UINT32	RO	0x0000:00, 64

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:04	Subindex 004	4. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:05	Subindex 005	5. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:06	Subindex 006	6. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:07	Subindex 007	7. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:08	Subindex 008	8. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x03 (3_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:05	Subindex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:07	Subindex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:08	Subindex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:09	Subindex 009	9. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0A	Subindex 010	10. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0B	Subindex 011	11. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0C	Subindex 012	12. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0003D090 (250000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x4808 (18440_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0001E848 (125000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 230], 0x1C33:06, and 0x1C33:09 [▶ 230] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 3: DC - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 0x1C32:02 [▶ 229]	UINT32	RW	0x0003D090 (250000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00001C52 (7250_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous with SM 3 event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 229] or 0x1C33:08)	UINT16	RO	0x4808 (18440_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 229]	UINT32	RO	0x0001E848 (125000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00001C52 (7250_{dec})
1C33:08	Command	as 0x1C32:08 [▶ 229]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00001C52 (7250_{dec})
1C33:0B	SM event missed counter	as 0x1C32:11 [▶ 229]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 229]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 229]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 229]	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 for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index spacing of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0004 (4_{dec})

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 001	Encoder profile number	UINT32	RW	0x00000201 (513_{dec})
F010:02	SubIndex 002	Servo drive profile number	UINT32	RW	0x000002E6 (742_{dec})
F010:03	SubIndex 003		UINT32	RW	0x00000000 (0_{dec})
F010:04	SubIndex 004	Profile number travel distance control	UINT32	RW	0x000002EE (750_{dec})

Index FB40 Memory interface

Index (hex)	Name	Meaning	Data type	Flags	Default
FB40:0	Memory interface	Maximum subindex	UINT8	RO	0x03 (3_{dec})
FB40:01	Address	reserved	UINT32	RW	0x00000000 (0_{dec})
FB40:02	Length	reserved	UINT16	RW	0x0000 (0_{dec})
FB40:03	Data	reserved	OCTET-STRING[8]	RW	{0}

7.2 EL72x1-9014 (DS402)

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information [▶ 32] 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

NOTICE

Risk of damage to the device!

We strongly advise not to change settings in the CoE objects while the axis is active, since this could impair the control.

7.2.1 Configuration data

Index 2002 Amplifier Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2002:0	Amplifier Settings	Maximum subindex	UINT8	RO	0x66 (102_{dec})
2002:11	Device type	1: Servo drive (cannot be changed)	UINT32	RW	0x00000001 (1_{dec})
2002:12*	Current loop integral time	Integral component of current controller **Unit:** 0.1 ms This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x000A (10_{dec})
2002:13*	Current loop proportional gain	Proportional component of current controller **Unit:** 0.1 V/A This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0064 (100_{dec})
2002:14	Velocity loop integral time	Integral component of velocity controller **Unit:** 0.1 ms	UINT32	RW	0x00000032 (50_{dec})
2002:15	Velocity loop proportional gain	Proportional component of velocity controller **Unit:** mA / (rad/s)	UINT32	RW	0x00000096 (150_{dec})
2002:17	Position loop proportional gain	Proportional component position controller **Unit:** (rad/s) / rad	UINT32	RW	0x0000000A (10_{dec})
2002:19	Nominal DC link voltage	Nominal DC link voltage **Unit**: mV	UINT32	RW	0x0000BB80 (48000_{dec})
2002:1A	Min DC link voltage	Minimum DC link voltage **Unit**: mV	UINT32	RW	0x00001A90 (6800_{dec})
2002:1B	Max DC link voltage	Maximum DC link voltage **Unit**: mV	UINT32	RW	0x0000EA60 (60000_{dec})
2002:29	Amplifier I2T warn level	I²T model warning threshold **Unit**: %	UINT8	RW	0x50 (80_{dec})
2002:2A	Amplifier I2T error level	I²T model error threshold **Unit**: %	UINT8	RW	0x69 (105_{dec})
2002:2B	Amplifier temperature warn level	Overtemperature warning threshold **Unit**: 0.1 °C	UINT16	RW	0x0320 (800_{dec})
2002:2C	Amplifier temperature error level	Overtemperature error threshold **Unit**: 0.1 °C	UINT16	RW	0x03E8 (1000_{dec})
2002:31	Velocity limitation	Velocity limitation **Unit**: rpm	UINT32	RW	0x00040000 (262144_{dec})
2002:32	Short circuit brake duration max	Max. duration of armature short circuit brake **Unit**: ms	UINT16	RW	0x03E8 (1000_{dec})
2002:33	Stand still window	Standstill window **Unit**: rpm	UINT16	RW	0x0000 (0_{dec})
2002:41	Low-pass filter frequency	Low-pass filter frequency **Unit**: Hz The following values can be set: 0 Hz = off 160 Hz 320 Hz	UINT16	RW	0x0140 (320_{dec})
2002:49	Halt ramp deceleration	Halt ramp deceleration **Unit**: 0.1 rad / s²	UINT32	RW	0x0000F570 (62832_{dec})

*) see index 0x2059 FB OCT Nameplate

Index (hex)	Name	Meaning	Data type	Flags	Default
2002:54	Feature bits	The adjustable motor current values can be interpreted as peak values or rms values. The feature bit enables the conversion. Peak value → Bit 0 = 0 (default with EL7201-001x) Rms value → Bit 0 = 1 (default with EL7211-001x) In the case of the EL7201-001x the output current can be increased with the help of the ZB8610 fan cartridge. normal output current → Bit 1 = 0 (default) increased output current → Bit 1 = 1 From these, the following combinations can be set: 0_{dec} → normal output current interpreted as peak value 1_{dec} → normal output current interpreted as rms value 2_{dec} → increased output current interpreted as peak value 3_{dec} → increased output current interpreted as rms value	UINT32	RW	0x00000000 (0_{dec})
2002:55	Select info data 1	Selection "Info data 1" Optional display of additional information in the cyclic process data. Permitted values: 1_{dec}: Torque current (filtered 1ms) [1000th of rated current] 2_{dec}: DC link voltage (mV) 4_{dec}: PCB temperature (0.1 °C) 5_{dec}: Errors: Bit0: ADC Error Bit1: Overcurrent Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder Bit8: Watchdog 6_{dec}: Warnings: Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder 7_{dec}: I2T Motor [%] 8_{dec}: I2T Amplifier [%] 10_{dec}: Input Level: Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit8: STO Input Level (STO variant, only)	UINT8	RW	0x00 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
2002:56	Select info data 2	Selection "Info data 2" Optional display of additional information in the cyclic process data. Permitted values: 1_{dec}: Torque current (filtered 1ms) [1000th of rated current] 2_{dec}: DC link voltage (mV) 4_{dec}: PCB temperature (0.1 °C) 5_{dec}: Errors: Bit0: ADC Error Bit1: Overcurrent Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder Bit8: Watchdog 6_{dec}: Warnings: Bit2: Undervoltage Bit3: Overvoltage Bit4: Overtemperature Bit5: I2T Amplifier Bit6: I2T Motor Bit7: Encoder 7_{dec}: I2T Motor [%] 8_{dec}: I2T Amplifier [%] 10_{dec}: Input Level: Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit8: STO Input Level (STO variant, only)	UINT8	RW	0x00 (0_{dec})
2002:57	Velocity feed forward gain	Velocity pre-control	UINT8	RW	0x64 (100_{dec})
2002:65	Invert direction of rotation	Invert direction of rotation	BOOLEAN	RW	0x00 (0_{dec})
2002:66	Enable cogging torque compensation	Enable cogging torque compensation	BOOLEAN	RW	0x00 (0_{dec})

Index 2003 Motor Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2003:0	Motor Settings	Maximum subindex	UINT8	RO	0x2D (45_{dec})
2003:11*	Max current	Peak current **Unit**: mA The adjustable motor current values can be interpreted as peak values or rms values. The feature bit (0x2002:54 [▶ 233]) enables the conversion. Peak value → Bit 0 = 0 (default with EL7201-001x) Rms value → Bit 0 = 1 (default with EL7211-001x) This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00001770 (6000_{dec})
2003:13*	Motor pole pairs	Number of pole pairs This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT8	RW	0x03 (3_{dec})
2003:15*	Commutation offset	Commutation offset (between electrical zero position and mechanical single-turn zero position) **Unit**: ° This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	INT16	RW	0x0000 (0_{dec})
2003:16*	Torque constant	Torque constant **Unit**: mNm / A This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00000000 (0_{dec})
2003:18*	Rotor moment of inertia	Mass moment of inertia of the motor **Unit**: g cm^2 This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT32	RW	0x00000000 (0_{dec})
2003:19*	Winding inductance	Inductance **Unit**: 0.1 mH This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x000E (14_{dec})
2003:29	Motor I2T warn level	I2T model warning threshold **Unit**: %	UINT8	RW	0x50 (80_{dec})
2003:2A	Motor I2T error level	I2T model error threshold **Unit**: %	UINT8	RW	0x69 (105_{dec})
2003:2B*	Motor Temperature warn level	Overtemperature warning threshold **Unit**: 0.1 °C This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0000 (0_{dec})
2003:2C*	Motor Temperature error level	Overtemperature error threshold **Unit**: 0.1 °C This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0000 (0_{dec})
2003:2D*	Motor thermal time constant	Thermal time constant **Unit**: 0.1 s This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 153])	UINT16	RW	0x0028 (40_{dec})

*) see index 0x2059 FB OCT Nameplate

Index 2004 Brake Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2004:0	Brake Settings	Maximum subindex	UINT8	RO	0x14 (20_{dec})
2004:01	Enable manual override	Manual release of the motor holding brake	BOOLEAN	RW	0x00 (0_{dec})
2004:02	Manual brake status	0: Release Release the brake 1: Apply Apply the brake	BOOLEAN	RW	0x00 (0_{dec})
2004:11*	Release delay	Time the holding brake requires for opening (releasing) after the voltage was applied This value is affected by automatic scanning (see Automatic scanning of the electronic type plates [▶ 153]).	UINT16	RW	0x0000 (0_{dec})
2004:12*	Application delay	Time the holding brake requires for closing (holding) after the voltage was switched off This value is affected by automatic scanning (see Automatic scanning of the electronic type plates [▶ 153]).	UINT16	RW	0x0000 (0_{dec})
2004:13	Emergency application timeout	Time the amplifier waits, until the rotation speed reaches the stand still window after stop request (set rotation speed 0 or Torque off). If the set waiting time is exceeded, the holding brake is triggered independently of the rotation speed. **Note:** For rotatory axes and the setting „torque off“ in error case, this parameter has to be set at least to the „coast to the stop“ time of the axis. For suspended (hanging) axes and the setting „torque off“ in error case, this parameter has to be set to an appropriate short time, to prevent the axis/load from drop/fall.	UINT16	RW	0x0000 (0_{dec})
2004:14*	Brake moment of inertia	Mass moment of inertia of the brake **Unit**: g cm^2 This value is affected by automatic scanning (see Automatic scanning of the electronic type plates [▶ 153]).	UINT16	RW	0x0000 (0_{dec})

*) see index 0x2059 FB OCT Nameplate

Index 2010 Feedback Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2010:0	Feedback Settings	Maximum subindex	UINT8	RO	0x17 (23_{dec})
2010:01	Invert feedback direction	Inverting the count direction	BOOLEAN	RW	0x00 (0_{dec})
2010:02	Referenced	Can be set to TRUE by the user. Is automatically set to FALSE as soon as a motor with a deviating serial number is recognized or if the feedback is single-turn.	BOOLEAN	RW	0x00 (0_{dec})
2010:0D	Offset position actual value source	Permitted values 0: Offset disabled No position offset is used. 1: Encoder memory The positon offset from the feedback memory used (0x2058:20 [▶ 246]). 2: Drive memory The positon offset from the servo drive is used (0x2010:17)	UINT8	RW	0x00 (0_{dec})
2010:11	Device type	3: OCT (not changeable)	UINT32	RW	0x00000003 (3_{dec})
2010:12	Single-turn bits	Number of singlei-turn-bits [▶ 137]	UINT8	RW	0x14 (20_{dec})
2010:13	Multi-turn bits	Number of multi-turn-bits [▶ 137]	UINT8	RW	0x0C (12_{dec})
2010:14	Observer bandwidth	Bandwidth of observer [Hz]	UINT16	RW	0x01F4 (500_{dec})
2010:15	Observer feed-forward	Load ratio[%] 100 % = free of load 50 % = moments of inertia of input and output are equal	UINT8	RW	0x01 (0_{dec})
2010:17	Positon offset	The position offset is subtracted from the raw position of the encoder. It can only be written with the axis stopped.	UINT32	RW	0x00000000 (0_{dec})

Index 2018 OCT Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2018:0	OCT Settings	Maximum subindex	UINT8	RO	0x03 (3_{dec})
2018:01	Enable auto config	Configuration takes place automatically after the reading of the electronic type plate (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})
2018:02	Reconfig identical motor	When replacing identical motors, reconfiguration takes place automatically after reading the electronic type plate. Enable autoconfig must be activated. (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})
2018:03	Reconfig non-identical motor	When replacing non-identical motors, reconfiguration takes place automatically after reading the electronic type plate. Enable autoconfig must be activated. (see Automatic scanning of the electronic type plates [▶ 153])	BOOLEAN	RW	0x00 (0_{dec})

7.2.2 Configuration data (vendor-specific)

Index 2020 Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
2020:0	Vendor data	Maximum subindex	UINT8	RO	0x15 (21_{dec})
2020:11	Amplifier peak current	Peak current of the amplifier (peak value) **Unit**: mA	UINT32	RW	0x00001F40 (8000_{dec})
2020:12	Amplifier rated current	Rated current of the amplifier (peak value) **Unit**: mA	UINT32	RW	0x00000FA0 (4000_{dec})
2020:13	Amplifier thermal time constant	Thermal time constant of the amplifier **Unit**: 0.1 ms	UINT16	RW	0x0023 (35_{dec})
2020:14	Amplifier overcurrent threshold	Threshold value for short-circuit detection **Unit**: mA	UINT32	RW	0x00002EE0 (12000_{dec})
2020:15	Max rotary field frequency	Max. rotary field frequency– **Unit:** Hz	UINT16	RW	0x0257 (599_{dec})

7.2.3 Command object

Index FB00 command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	Command	Maximum subindex	UINT8	RO	0x03 (3_{dec})
FB00:01	Request	reserved	OCTET-STRING[2]	RW	{0}
FB00:02	Status	reserved	UINT8	RO	0x00 (0_{dec})
FB00:03	Response	reserved	OCTET-STRING[4]	RO	{0}

7.2.4 Input/output data

Index 2001 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
2001:0	Outputs	Maximum subindex	UINT8	RO	0x11 (17_{dec})
2001:11	Torque offset	Torque value offset The value is specified in 1000th of the *rated current* . Equation for index 0x2002:54 [▶ 233] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x2003:16 [▶ 236]) Equation for index 0x2002:54 [▶ 233] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x2003:16 [▶ 236])	INT16	RO	0x0000 (0_{dec})

Index 2008 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
2008:0	Inputs	Maximum subindex	UINT8	RO	0x02 (2_{dec})
2008:01	Info data1	Synchronous information (selection via subindex 0x2002:55 [▶ 233])	UINT16	RO	0x0000 (0_{dec})
2008:01	Info data2	Synchronous information (selection via subindex 0x2002:56 [▶ 233])	UINT16	RO	0x0000 (0_{dec})

Index 603E TxPDO Data invalid

Index (hex)	Name	Meaning	Data type	Flags	Default
603E:0	TxPDO Data invalid	Max. subindex	UINT8	RO	0x02 (2_{dec})
603E:02	Position actual value	0: The current position is valid 1: The current position is invalid	BOOLEAN	RO P	0

Index 6040 Controlword

Index (hex)	Name	Meaning	Data type	Flags	Default
6040:0	Controlword	DS402 Controlword [▶ 150] Bit 0: Switch on Bit 1: Enable voltage Bit 2: reserved Bit 3: Enable operation Bit 4 - 6: reserved Bit 7: Fault reset Bit 8 - 15: reserved	UINT16	RO	0x0000 (0_{dec})

Index 6041 Statusword

Index (hex)	Name	Meaning	Data type	Flags	Default
6041:0	Statusword	DS402 Statusword [▶ 150] Bit 0: Ready to switch on Bit 1: Switched on Bit 2: Operation enabled Bit 3: Fault Bit 4: reserved Bit 5: reserved Bit 6: Switch on disabled Bit 7: Warning Bit 8 + 9: reserved Bit 10: TxPDOToggle (selection/deselection via 0x60DA [▶ 243]) Bit 11: Internal limit active Bit 12: (Target value ignored) Bit 13 - 15: reserved	UINT16	RO	0x0000 (0_{dec})

Index 605E Fault reaction option code

Index (hex)	Name	Meaning	Data type	Flags	Default
605E:0	Fault reaction option code	0: Disable drive function, motor is free to rotate 1: Slow down by slow down ramp	ENUM16BIT	RW	0

Index 6060 Modes of operation

Index (hex)	Name	Meaning	Data type	Flags	Default
6060:0	Modes of operation	permitted values: 0x08: Cyclic synchronous position mode (CSP) 0x09: Cyclic synchronous velocity mode (CSV) 0x0A: Cyclic synchronous torque mode (CST) 0x0B: Cyclic synchronous torque mode with commutation angle (CSTCA)	UINT8	RW	0x00 (0_{dec})

Index 6061 Modes of operation display

Index (hex)	Name	Meaning	Data type	Flags	Default
6061:0	Modes of operation display	permitted values: 8: Cyclic synchronous position mode (CSP) 9: Cyclic synchronous velocity mode (CSV) 10: Cyclic synchronous torque mode (CST) 11: Cyclic synchronous torque mode with commutation angle (CSTCA)	UINT8	RO	0x00 (0_{dec})

Index 6064 Position actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
6064:0	Position actual value	Position **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})

Index 6065 Following error window

Index (hex)	Name	Meaning	Data type	Flags	Default
6065:0	Following error window	Following error monitor: Following error window **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141] 0xFFFFFFFF (-1_{dec}) = following error monitor off Any other value = following error monitor on	UINT32	RO	0xFFFFFFFF (-1_{dec})

Index 6066 Following error time out

Index (hex)	Name	Meaning	Data type	Flags	Default
6066:0	Following error time out	Following error monitor: Timeout **Unit:** ms If the following error is larger than the following error window for a time that exceeds the timeout, this leads to an error reaction	UINT16	RO	0x0000 (0_{dec})

Index 606C Velocity actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
606C:0	Velocity actual value	Display of the current velocity value	INT32	RO	0x00000000 (0_{dec})

Index 6071 Target torque

Index (hex)	Name	Meaning	Data type	Flags	Default
6071:0	Target torque	This object shall indicate the configured input value for the torque controller. The value is specified in 1000th of the *rated current*. Equation for index 0x2002:54 [▶ 233] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x2003:16 [▶ 236]) Equation for index 0x2002:54 [▶ 233] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x2003:16 [▶ 236])	INT16	RO	0x0000 (0_{dec})

Index 6072 Max torque

Index (hex)	Name	Meaning	Data type	Flags	Default
6072:0	Max torque	This object limits the target torque for the torque controller (bipolar limit). The value is specified in 1000th of the *rated current*. Equation for index 0x2002:54 [▶ 233] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x2003:16 [▶ 236]) Equation for index 0x2002:54 [▶ 233] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x2003:16 [▶ 236])	UINT16	RW	0x7FFF (32767_{dec})

Index 6075 Motor rated current

Index (hex)	Name	Meaning	Data type	Flags	Default
6075:0	Motor rated current	Motor rated current **Unit**: mA	UINT32	RW	0x000003E8 (1000_{dec})

Index 6077 Torque actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
6077:0	Torque actual value	This object shall provide the actual value of the torque. The value is specified in 1000th of the *rated current*. Equation for index 0x2002:54 [▶ 233] = 0 : M = ((Torque actual value / 1000) x (rated current / √2)) x torque constant (0x2003:16 [▶ 236]) Equation for index 0x2002:54 [▶ 233] = 1 : M = ((Torque actual value / 1000) x rated current)) x torque constant (0x2003:16 [▶ 236])	INT16	RO	0x0000 (0_{dec})

Index 6079 DC link circuit voltage

Index (hex)	Name	Meaning	Data type	Flags	Default
6079:0	DC link circuit voltage	DC link voltage g **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})

Index 607A Target position

Index (hex)	Name	Meaning	Data type	Flags	Default
607A:0	Target position	This object shall provide the actual position. **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	UINT32	RO	0x00000000 (0_{dec})

Index 6080 Max motor speed

Index (hex)	Name	Meaning	Data type	Flags	Default
6080:0	Max motor speed	Velocity limitation **Unit**: rpm	UINT32	RW	0x00040000 (262144_{dec})

Index 608F Position encoder resolution

Index (hex)	Name	Meaning	Data type	Flags	Default
608F:0	Position encoder resolution	This object represents the configured encoder increments and the number of motor revolutions. The position encoder resolution is calculated according to the following equation: Position encoder resolution = (encoder increments / motor revolutions)	UINT8	RO	0x02 (2_{dec})
608F:01	SubIndex 001	Encoder increments	UINT32	RO	0x00000000 (0_{dec})
608F:02	SubIndex 002	Motor revolutions	UINT32	RO	0x00000000 (0_{dec})

Index 6090 Velocity Encoder Resolution

Index (hex)	Name	Meaning	Data type	Flags	Default
6090:0	Velocity Encoder Resolution	Display of configured encoder increments/s and motor revolutions/s. The velocity encoder resolution is calculated according to the following formula: Velocity Encoder Resolution = (encoder_increments / s) / (motor_revolutions / s)	UINT32	RO	0x00000000 (0_{dec})

Index 60B2 Torque offset

Index (hex)	Name	Meaning	Data type	Flags	Default
60B2:0	Torque offset	Torque value offset	UINT16	RO	0x0000 (0_{dec})

Index 60B8 Touch probe function

Index (hex)	Name	Meaning	Data type	Flags	Default
60B8:0	Touch probe function	Touch probe function byte	UINT16	RO	0x0000 (0_{dec})

Index 60B9 Touch probe status

Index (hex)	Name	Meaning	Data type	Flags	Default
60B9:0	Touch probe status	Touch probe status byte	UINT16	RO	0x0000 (0_{dec})

Index 60BA Touch probe 1 positive edge

Index (hex)	Name	Meaning	Data type	Flags	Default
60BA:0	Touch probe 1 positive edge	Positive position value of TP 1 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})

Index 60BB Touch probe 1 negative edge

Index (hex)	Name	Meaning	Data type	Flags	Default
60BB:0	Touch probe 1 negative edge	Negative position value of TP 1 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})

Index 60BC Touch probe 2 positive edge

Index (hex)	Name	Meaning	Data type	Flags	Default
60BC:0	Touch probe 2 positive edge	Positive position value of TP 2 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})

Index 60BD Touch probe 2 negative edge

Index (hex)	Name	Meaning	Data type	Flags	Default
60BD:0	Touch probe 2 negative edge	Negative position value of TP 2 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})

Index 60C2 Interpolation time period

Index (hex)	Name	Meaning	Data type	Flags	Default
60C2:0	Interpolation time period	Maximum subindex x	UINT8	RO	0x02 (2_{dec})
60C2:01	Interpolation time period value	This object shall indicate the configured interpolation cycle time. The interpolation time period (sub-index 0x01) value shall be given in $10^{(interpolation\ time\ index)}$ (second). The interpolation time index (sub-index 0x02) shall be dimensionless.	UINT8T8	RO	0x00 (0_{dec})
60C2:02	Interpolation time index		INT8	RO	0x00 (0_{dec})

Index 60D9 Supported functions

Index (hex)	Name	Meaning	Data type	Flags	Default
60D9:0	Supported functions	This object shall provide information on the supported functions in the device.	UINT3232	RO	0x00000000 (0_{dec})

Index 60DA Function settings

Index (hex)	Name	Meaning	Data type	Flags	Default
60DA:0	Function settings	This object shall enable/disable supported functions in the device. Bit 0: Enable TxPDOToggle-Bit in Statusword: Bit 10 Bit 1: Enable input cycle counter in Statusword: Bit 13 Bit 2-31: reserved	UINT32	RW	0x00000000 (0_{dec})

Index 60EA Commutation angle

Index (hex)	Name	Meaning	Data type	Flags	Default
60EA:0	Commutation angle	Electrical commutation angle (for the CSTCA mode) **Unit**: 5.49 * 10-3 °	UINT16	RO	0x0000 (0_{dec})

Index 60F4 Following error actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
60F4:0	Following error actual value	Following error **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 141]	INT32	RO	0x00000000 (0_{dec})

Index 60FF Target velocity

Index (hex)	Name	Meaning	Data type	Flags	Default
60FF:0	Target velocity	This object shall indicate the configured target velocity The velocity scaling can be found in object 0x6090 (Velocity encoder resolution)	INT32	RO	0x00000000 (0_{dec})

Index 6403 Motor catalogue number

Index (hex)	Name	Meaning	Data type	Flags	Default
6403:0	Motor catalogue number	This is the order code from the electronic type plate of the motor, e.g. AM8121-0F20-0000	STRING	RO	

Index 6502 Supported drive modes

Index (hex)	Name	Meaning	Data type	Flags	Default
6502:0	Supported drive modes	This object shall provide information on the supported drive modes. (DS402 Object 0x6502) Only modes *CSV*, *CST*, *CSTCA and CSP* are supported Bit 0: PP Bit 1: VL Bit 2: PV Bit 3: TQ Bit 4: R Bit 5: HM Bit 6: IP Bit 7: CSP Bit 8: CSV Bit 9: CST Bit 10: CSTCA Bit 11-15: reserved Bit 16-31: Manufacturer-specific	UINT32	RO	0x00000000 (0_{dec})

7.2.5 Information / diagnosis data

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x37 (55_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages. A maximum of 50 messages can be stored	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[28]	RO	{0}
...	...	...	...	...	...
10F3:37	Diagnosis Message 050	Message 50	OCTET-STRING[28]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Time stamp	UINT64	RO	

Index 2019 OCT Memory interface

Index (hex)	Name	Meaning	Data type	Flags	Default
2019:0	OCT Memory interface	Maximum subindex	UINT8	RO	0x06 (6_{dec})
2019:01	cmd	3_{dec}: Enc Write 6_{dec}: Enc Read Direct 7_{dec}: Enc Read Indirect (Default) 9_{dec}: Enc Reset 10_{dec}: IP Write 15_{dec}: IP Read 16_{dec}: Write encoder position offset	INT16	RW	0x0007 (7_{dec})
2019:02	Len		INT16	RW	0x0000 (0_{dec})
2019:03	Adr		UINT32	RW	0x00000000 (0_{dec})
2019:04	Offset		UINT32	RW	0x00000000 (0_{dec})
2019:05	Ctrl/Status	0_{dec}: Init (Default) 1_{dec}: Execute 2_{dec}: Busy 3_{dec}: Done 4_{dec}: Error	INT16	RW	0x0000 (0_{dec})
2019:06	Data		OCTET-STRING[32]	RW	0x00000000 (0_{dec})

Index 2030 Amplifier Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
2030:0	Amplifier Diag data	Maximum subindex	UINT8	RO	0x11 (17_{dec})
2030:11	Amplifier I2T temperature	I2T model load **Unit**: %	UINT8	RO	0x00 (0_{dec})

Index 2031 Motor Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
2031:0	Motor Diag data	Maximum subindex	UINT8	RO	0x13 (19_{dec})
2031:11	Motor I2T temperature	I2T model load **Unit**: %	UINT8	RO	0x00 (0_{dec})
2031:13	Motor temperature	Temperature utilization ratio **Unit**: °	UINT16	RO	0x0000 (0_{dec})

Index 2040 Amplifier Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
2040:0	Amplifier Info data	Maximum subindex	UINT8	RO	0x12 (18_{dec})
2040:11	Amplifier temperature	Internal terminal temperature **Unit**: 0.1 °C	UINT16	RO	0x0000 (0_{dec})
2040:12	DC link voltage	DC link voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})

Index 2041 Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
2041:0	Info data	Maximum subindex	UINT8	RO	0x11 (17_{dec})
2041:11	Auxiliary voltage (10 V)	Auxiliary voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})

Index 2058 OCT Info data

(these data are always read in automatically from the electronic type plate of the motor and serve purely informative purposes)

Index (hex)	Name	Meaning	Data type	Flags	Default
2058:0	OCT Info data	Maximum subindex	UINT8	RO	0x20 (32_{dec})
2058:11	Encoder Type	Feedback type 2: rotary encoder, unipolar counting	UINT16	RO	0x0000 (0_{dec})
2058:12	Resolution	Resolution of the feedback **Unit**: Steps per revolution	UINT32	RO	0x00000000 (0_{dec})
2058:13	Range	Working range of the feedback. On leaving this range there is an overflow of the position. **Unit**: Revolutions	UINT32	RO	0x00000000 (0_{dec})
2058:14	Type Code Name	Name of the feedback	STRING	RO	
2058:15	Serial No	Serial number of the feedback	STRING	RO	
2058:16	Firmware Revision No	Revision of the firmware	STRING	RO	
2058:17	Firmware Date	Date of the firmware	STRING	RO	
2058:18	EEPROM Size	EEPROM size	UINT16	RO	0x0000 (0_{dec})
2058:19	Temperature	Temperature **Unit**: 0.1°	INT16	RO	0x0000 (0_{dec})
2058:1A	LED Current	Current of the feedback LED **Unit**: 0.1 mA	UINT16	RO	0x0000 (0_{dec})
2058:1B	Supply voltage	Supply voltage of the feedback **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2058:1C	Life- time	Operating hour counter **Unit**: Minutes	UINT32	RO	0x00000000 (0_{dec})
2058:1D	Received Signal Strength Indicator	Received signal strength at the terminal **Unit**: %	UINT16	RO	0x0000 (0_{dec})
2058:1E	Slave Received Signal Strength Indicator	Received signal strength at the encoder **Unit**: %	UINT16	RO	0x0000 (0_{dec})
2058:1F	Line delay	Running time of the signal in the cable **Unit**: ns	UINT16	RO	0x0000 (0_{dec})
2058:20	Encoder position offset	Position offset stored in the motor feedback	UINT16	RO	0x0000 (0_{dec})

Index 2059 OCT Nameplate

The parameters described in this index are always read from the electronic type plate of the connected motor. These parameters automatically lead to the parameters marked with an asterisk (*) in this chapter, if automatic scanning of the electronic type plate is switched on (index 0x2018 [▶ 238]).

Index (hex)	Name	Meaning	Data type	Flags	Default
2059:0	OCT Nameplate	Maximum subindex	UINT8	RO	0x24 (36_{dec})
2059:01	Motor vendor	Motor vendor	STRING	RO	
2059:02	Electric motor type	Motor type	STRING	RO	
2059:03	Serial No	Serial number	STRING	RO	
2059:04	Order code	Order number (In case of Autoconfig a check is made on the basis of this index as to whether the motor is identical to the predecessor)	STRING	RO	
2059:05	Motor construction	Type of motor	STRING	RO	
2059:06	Pole pairs	Number of pole pairs	UINT32	RO	0x00000000 (0_{dec})
2059:07	Standstill current (rms)	Effective holding current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
2059:08	Rated current (rms)	Effective rated current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
2059:09	Peak current (rms)	Effective peak current **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
2059:0A	Nominal voltage (rms)	Effective nominal voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2059:0B	Max voltage (rms)	Maximum voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2059:0C	Max winding du/dt	Maximum permissible voltage rise on the winding **Unit**: kV/s	UINT32	RO	0x00000000 (0_{dec})
2059:0D	Max torque	Maximum torque **Unit**: mNm	UINT16	RO	0x0000 (0_{dec})
2059:0E	Torque constant	Torque constant **Unit**: mNm / A	UINT16	RO	0x0000 (0_{dec})
2059:0F	EMK (rms)	Reverse voltage **Unit**: mV / (rpm)	UINT32	RO	0x00000000 (0_{dec})
2059:10	Winding resistance Ph-Ph 20°C	Coil resistance **Unit**: mOhm	UINT16	RO	0x0000 (0_{dec})
2059:11	Ld Ph-Ph	Inductance in the direction of flow **Unit**: 0.1 mH	UINT16	RO	0x0000 (0_{dec})
2059:12	Lq Ph-Ph	Inductance in the torque-forming direction **Unit**: 0.1 mH	UINT16	RO	0x0000 (0_{dec})
2059:13	Max speed	Maximum speed **Unit**: rpm	UINT32	RO	0x00000000 (0_{dec})
2059:14	Moment of inertia	Mass moment of inertia **Unit**: g cm^2	UINT16	RO	0x0000 (0_{dec})
2059:15	T motor warn limit	Motor temperature warning threshold **Unit**: 0.1 °C	UINT16	RO	0x0000 (0_{dec})
2059:16	T motor shut down	Motor temperature error threshold **Unit**: 0.1°C	UINT16	RO	0x0000 (0_{dec})
2059:17	Time constant i2t	Time constant I2T model **Unit**: s	UINT16	RO	0x0000 (0_{dec})
2059:18	Motor thermal constant	Thermal time constant of the motor **Unit**: s	UINT16	RO	0x0000 (0_{dec})
2059:1B	Brake type	Brake type • no Brake • holding Brake	STRING	RO	
2059:1C	Min brake voltage	Minimum brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2059:1D	Max brake voltage	Maximum brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2059:1E	Min brake monitor current	Minimum current for the monitoring of the brake **Unit**: mA	UINT32	RO	0x00000000 (0_{dec})
2059:1F	Brake holding torque	Brake holding torque **Unit**: mNm	UINT32	RO	0x00000000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
2059:20	Brake T on	Time until the brake is applied **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
2059:21	Brake T off	Time until the brake is released **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
2059:22	Brake reduced holding voltage	Reduced brake voltage **Unit**: mV	UINT32	RO	0x00000000 (0_{dec})
2059:23	Brake time to red. holding volt.	Time from which the brake holds with reduced voltage **Unit**: ms	UINT16	RO	0x0000 (0_{dec})
2059:24	Motor temp sensor connection	Temperature sensor connection feedback port (not changeable)	STRING	RO	

7.2.6 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 CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00000192 (402_{dec})

Index 1001 Error register

Index (hex)	Name	Meaning	Data type	Flags	Default
1001:0	Error register		UINT8	RO	0x00 (0_{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	EL72x1-x01x

Index 1009 Hardware version

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

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware-Version des EtherCAT-Slaves	STRING	RO	01

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x1C213052 (471937106_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 1600 DS402 RxPDO-Map Controlword

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DS402 RxPDO-Map Controlword	PDO Mapping RxPDO 1	UINT8	RO	0x01 (1_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6040:00, 16

Index 1601 DS402 RxPDO-Map Target velocity

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DS402 RxPDO-Map Target velocity	PDO Mapping RxPDO 2	UINT8	RO	0x01 (1_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60FF:00, 32

Index 1602 DS402 RxPDO-Map Target torque

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	DS402 RxPDO-Map Target torque	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6071:00, 16

Index 1603 DS402 RxPDO-Map Commutation angle

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	DS402 RxPDO-Map Commutation angle	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60EA:00, 16

Index 1604 DS402 RxPDO-Map Torque limitation

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	DS402 RxPDO-Map Torque limitation	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6072:00, 16

Index 1605 DS402 RxPDO-Map Torque offset

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	DS402 RxPDO-Map Torque offset	PDO Mapping RxPDO 6	UINT8	RO	0x01 (1_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x2001:11, 16

Index 1606 DS402 RxPDO-Map Target position

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	DS402 RxPDO-Map Target position	PDO Mapping RxPDO 7	UINT8	RO	0x01 (1_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x607A:00, 32

Index 1607 DS402 RxPDO-Map Touch probe function

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	DS402 RxPDO-Map Touch probe function	PDO Mapping RxPDO 8	UINT8	RO	0x01 (1_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60B8:00, 16

Index 1608 DS402 RxPDO-Map Modes of operation

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	DS402 RxPDO-Map Modes of operation	PDO Mapping RxPDO 9	UINT8	RO	0x01 (1_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6060:00, 8

Index 1A00 DS402 TxPDO-Map Statusword

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	DS402 TxPDO-Map Statusword	PDO Mapping TxPDO 1	UINT8	RO	0x01 (1_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6041:00, 16

Index 1A01 DS402 TxPDO-Map Position actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	DS402 TxPDO-Map Position actual value	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6064:00, 32

Index 1A02 DS402 TxPDO-Map Velocity actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	DS402 TxPDO-Map Velocity actual value	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x606C:00, 32

Index 1A03 DS402 TxPDO-Map Torque actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	DS402 TxPDO-Map Torque actual value	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x6077:00, 16

Index 1A04 DS402 TxPDO-Map Following error actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	DS402 TxPDO-Map Following error actual value	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60F4:00, 32

Index 1A05 DS402 TxPDO-Map Touch probe status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	DS402 TxPDO-Map Touch probe status	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60B9:00, 16

Index 1A06 DS402 TxPDO-Map Touch probe 1 positive edge

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	DS402 TxPDO-Map Touch probe 1 positive edge	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60BA:00, 32

Index 1A07 DS402 TxPDO-Map Touch probe 1 negative edge

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	DS402 TxPDO-Map Touch probe 1 negative edge	PDO Mapping TxPDO 8	UINT8	RO	0x01 (1_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60BB:00, 32

Index 1A08 DS402 TxPDO-Map Touch probe 2 positive edge

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	DS402 TxPDO-Map Touch probe 2 positive edge	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60BC:00, 32

Index 1A09 DS402 TxPDO-Map Touch probe 2 negative edge

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	DS402 TxPDO-Map Touch probe 2 negative edge	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry	UINT32	RO	0x60BD:00, 32

Index 1A0A DS402 TxPDO-Map Data invalid

Index	Name	Meaning	Data type	Flags	Default
1A0A:0	DS402 TxPDO-Map Data invalid	PDO Mapping TxPDO 11	UINT8	RO	0x03 (3_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A0A:02	SubIndex 002	2. PDO Mapping entry (object 0x603E (Data invalid), entry 0x02 (Position actual value))	UINT32	RO	0x603E:02, 1
1A0A:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1A0B DS402 TxPDO-Map Info data 1

Index	Name	Meaning	Data type	Flags	Default
1A0B:0	DS402 TxPDO-Map Info data 1	PDO Mapping TxPDO 12	UINT8	RO	0x01 (1_{dec})
1A0B:01	SubIndex 001	1st PDO Mapping entry (object 0x2008 (inputs), entry 0x01 (Info data 1))	UINT32	RO	0x2008:01, 16

Index 1A0C DS402 TxPDO-Map Info data 2

Index	Name	Meaning	Data type	Flags	Default
1A0C:0	DS402 TxPDO-Map Info data 2	PDO Mapping TxPDO 13	UINT8	RO	0x01 (1_{dec})
1A0C:01	SubIndex 001	1st PDO Mapping entry (object 0x2008 (inputs), entry 0x02 (Info data 2))	UINT32	RO	0x2008:02, 16

Index 1A0E DS402 TxPDO-Map Modes of operation display

Index	Name	Meaning	Data type	Flags	Default
1A0E:0	DS402 TxPDO-Map Modes of operation display	PDO Mapping TxPDO 14	UINT8	RO	0x01 (1_{dec})
1A0E:01	SubIndex 001	1. PDO Mapping entry (object 0x6061 (Modes of operation display))	UINT32	RO	0x6061:00, 8

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:04	Subindex 004	4. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:05	Subindex 005	5. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:06	Subindex 006	6. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:07	Subindex 007	7. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:08	Subindex 008	8. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C13 TxPDO assign

Index	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:05	Subindex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:07	Subindex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:08	Subindex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:09	Subindex 009	9. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0A	Subindex 010	10. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter

Index (Hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 3: DC-Mode - Synchronous with SYNC1 event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0003D090 (250000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x4808 (18440_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0001E848 (125000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 254], 0x1C33:06, and 0x1C33:09 [▶ 254] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 3: DC - Synchronous with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	as 0x1C32:02 [▶ 253]	UINT32	RW	0x0003D090 (250000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00001C52 (7250_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous with SM 3 event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 253] or 0x1C33:08)	UINT16	RO	0x4808 (18440_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 253]	UINT32	RO	0x0001E848 (125000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00001C52 (7250_{dec})
1C33:08	Command	as 0x1C32:08 [▶ 253]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00001C52 (7250_{dec})
1C33:0B	SM event missed counter	as 0x1C32:11 [▶ 253]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 253]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 253]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 253]	BOOLEAN	RO	0x00 (0_{dec})

Index F008 Code word

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

8 Error correction

8.1 Diagnostics - basic principles of diag messages

DiagMessages designates a system for the transmission of messages from the EtherCAT Slave to the EtherCAT Master/TwinCAT. The messages are stored by the device in its own CoE under 0x10F3 and can be read by the application or the System Manager. An error message referenced via a code is output for each event stored in the device (warning, error, status change).

Definition

The *DiagMessages* system is defined in the ETG (EtherCAT Technology Group) in the guideline ETG.1020, chapter 13 “Diagnosis handling”. It is used so that pre-defined or flexible diagnostic messages can be conveyed from the EtherCAT Slave to the Master. In accordance with the ETG, the process can therefore be implemented supplier-independently. Support is optional. The firmware can store up to 250 DiagMessages in its own CoE.

Each DiagMessage consists of

- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

The DiagMessages are explained in text form in the ESI/XML file belonging to the EtherCAT device: on the basis of the Text ID contained in the DiagMessage, the corresponding plain text message can be found in the languages contained in the ESI/XML. In the case of Beckhoff products these are usually German and English.

Via the entry *NewMessagesAvailable* the user receives information that new messages are available.

DiagMessages can be confirmed in the device: the last/latest unconfirmed message can be confirmed by the user.

In the CoE both the control entries and the history itself can be found in the CoE object 0x10F3:

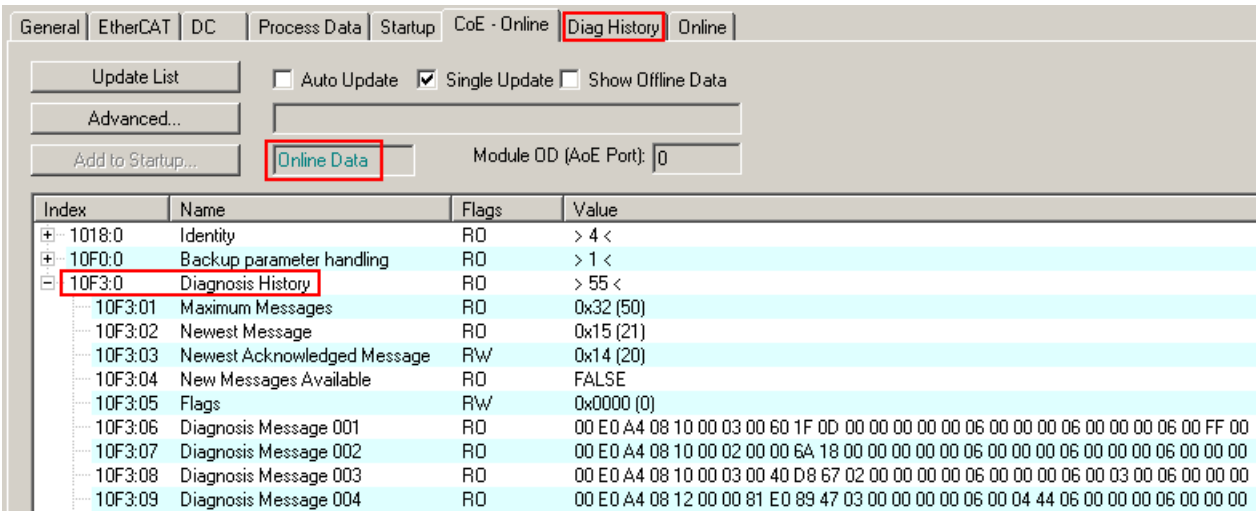


Fig. 214: DiagMessages in the CoE

The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

The DiagMessages system is to be used above all during the commissioning of the plant. The diagnostic values e.g. in the StatusWord of the device (if available) are helpful for online diagnosis during the subsequent continuous operation.

TwinCAT System Manager implementation

From TwinCAT 2.11 DiagMessages, if available, are displayed in the device's own interface. Operation (collection, confirmation) also takes place via this interface.

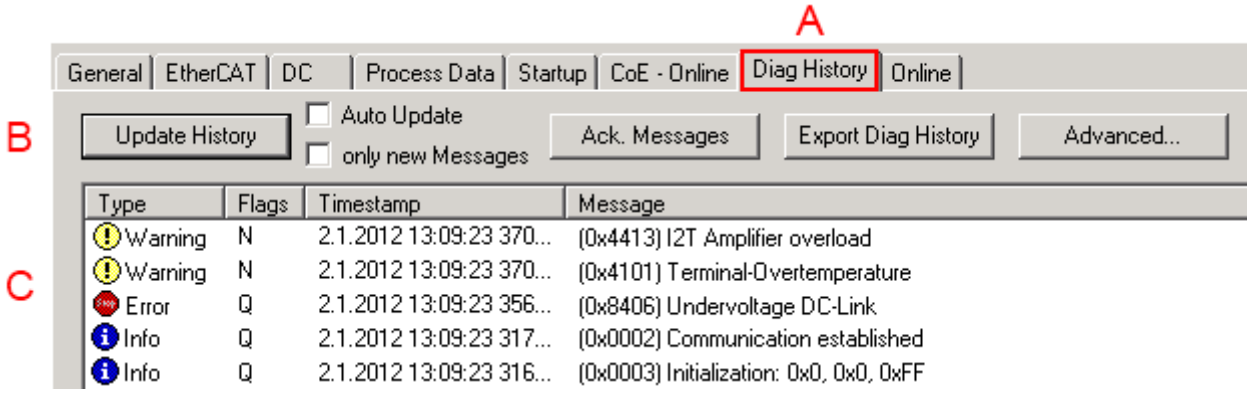


Fig. 215: Implementation of the DiagMessage system in the TwinCAT System Manager

The operating buttons (B) and the history read out (C) can be seen on the Diag History tab (A). The components of the message:

- Info/Warning/Error
- Acknowledge flag (N = unconfirmed, Q = confirmed)
- Time stamp
- Text ID
- Plain text message according to ESI/XML data

The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

From TwinCAT 3.1 build 4022 onwards, DiagMessages sent by the terminal are shown by the TwinCAT ADS Logger. Given that DiagMessages are represented IO- comprehensive at one place, commissioning will be simplified. In addition, the logger output could be stored into a data file – hence DiagMessages are available long-term for analysis.

DiagMessages are actually only available locally in CoE 0x10F3 in the terminal and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT Terminals are set by default to report the presence of a DiagMessage as emergency via EtherCAT; the event logger can then retrieve the DiagMessage. The function is activated in the terminal via 0x10F3:05, so such terminals have the following entry in the StartUp list by default:

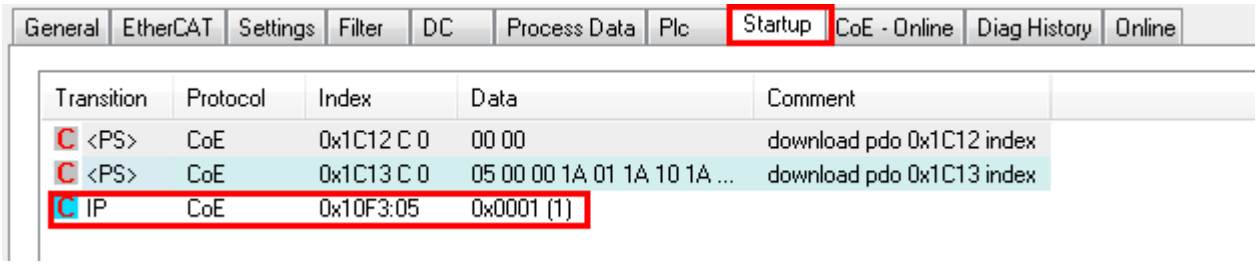


Fig. 216: Startup List

If the function is to be deactivated because, for example, many messages come in or the EventLogger is not used, the StartUp entry can be deleted or set to 0. The value can then be set back to 1 later from the PLC via CoE access if required.

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

The time stamp is obtained from the local clock of the terminal at the time of the event. The time is usually the distributed clock time (DC) from register x910.

Please note: When EtherCAT is started, the DC time in the reference clock is set to the same time as the local IPC/TwinCAT time. From this moment the DC time may differ from the IPC time, since the IPC time is not adjusted. Significant time differences may develop after several weeks of operation without a EtherCAT restart. As a remedy, external synchronization of the DC time can be used, or a manual correction calculation can be applied, as required: The current DC time can be determined via the EtherCAT master or from register x901 of the DC slave.

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 2: reserved 1: Info 4: Warning 8: Error	0: System 1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: I/O general 7: reserved	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Overview of text IDs

Specific text IDs are listed in the device documentation.

Text ID	Type	Place	Text Message	Additional comment
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Connection established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1000	Information	System	Information: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1012	Information	System	EtherCAT state change Init - PreOp	
0x1021	Information	System	EtherCAT state change PreOp - Init	
0x1024	Information	System	EtherCAT state change PreOp - Safe-Op	
0x1042	Information	System	EtherCAT state change SafeOp - PreOp	
0x1048	Information	System	EtherCAT state change SafeOp - Op	
0x1084	Information	System	EtherCAT state change Op - SafeOp	
0x1100	Information	General	Detection of operation mode completed: 0x%X, %d	Detection of the mode of operation ended
0x1135	Information	General	Cycle time o.k.: %d	Cycle time OK
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved manually
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved automatically
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted
0x117F	Information	General	Information: 0x%X, 0x%X, 0x%X	Information
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation.
0x1300	Information	Encoder	Position set: %d, %d	Position set - StartInputhandler
0x1303	Information	Encoder	Encoder Supply ok	Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully completed
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command
0x1400	Information	Drive	Drive is calibrated: %d, %d	Drive is calibrated
0x1401	Information	Drive	Actual drive state: 0x%X, %d	Current drive status
0x1705	Information		CPU usage returns in normal range (< 85%%)	Processor load is back in the normal range
0x1706	Information		Channel is not in saturation anymore	Channel is no longer in saturation
0x1707	Information		Channel is not in overload anymore	Channel is no longer overloaded
0x170A	Information		No channel range error anymore	A measuring range error is no longer active
0x170C	Information		Calibration data saved	Calibration data were saved
0x170D	Information		Calibration data will be applied and saved after sending the command “0x5AFE”	Calibration data are not applied and saved until the command "0x5AFE" is sent.

Text ID	Type	Place	Text Message	Additional comment
0x2000	Information	System	%s: %s	
0x2001	Information	System	%s: Network link lost	Network connection lost
0x2002	Information	System	%s: Network link detected	Network connection found
0x2003	Information	System	%s: no valid IP Configuration - Dhcp client started	Invalid IP configuration
0x2004	Information	System	%s: valid IP Configuration (IP: %d.%d.%d.%d) assigned by Dhcp server %d.%d.%d.%d	Valid IP configuration, assigned by the DHCP server
0x2005	Information	System	%s: Dhcp client timed out	DHCP client timeout
0x2006	Information	System	%s: Duplicate IP Address detected (%d.%d.%d.%d)	Duplicate IP address found
0x2007	Information	System	%s: UDP handler initialized	UDP handler initialized
0x2008	Information	System	%s: TCP handler initialized	TCP handler initialized
0x2009	Information	System	%s: No more free TCP sockets available	No free TCP sockets available.

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d.%d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values.

Text ID	Type	Place	Text Message	Additional comment
				• The I2T-model of the motor is incorrectly parameterized.
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x%X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. • The EtherCAT connection was interrupted during operation. • The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	• There is no voltage applied to the power contacts. • A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel. %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel. %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x%X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.

Text ID	Type	Place	Text Message	Additional comment
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

9 Appendix

9.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

9.2 Firmware compatibility

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

Note

- The newest possible firmware for the respective hardware should be implemented
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTICE

Risk of damage to the device!

Observe the notes on the firmware update in the EtherCAT System Manual in chapter “Firmware Update”.
If a device is set to BOOTSTRAP mode for firmware update, it may not check whether the new firmware is suitable during download.
This may result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL7201-9014			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 07	11	EL7201-9014-0026	2016/06
	12	EL7201-9014-0027	2016/11
	13	EL7201-9014-0028	2017/09
	14		2017/11
	15		2018/02
	16	EL7201-9014-0029	2018/05
	17		2018/05
	18		2018/12
	19	EL7201-9014-0030	2020/02
	20	EL7201-9014-0031	2020/05
	21		2020/09
	22		2020/11
	23	EL7201-9014-0032	2021/01
	24		2021/06
	25		2022/03
08*	26		2022/12
	27*		2323/06

EL7211-9014			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 09	11	EL7211-9014-0026	2016/06
	12	EL7211-9014-0027	2016/11
	13	EL7211-9014-0028	2017/09
	14		2017/11
	15		2018/02
	16	EL7211-9014-0029	2018/05
	17		2018/06
	18		2018/12
	19	EL7211-9014-0030	2020/02
	20	EL7211-9014-0031	2020/05
	21		2020/09
	22		2020/11
	23	EL7211-9014-0032	2021/01
	24		2021/06
	25		2022/03
10*	26		2022/12
	27*		2023/06

EL7221-9014			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 07	12	EL7221-9014-0027	2016/11
	13	EL7221-9014-0028	2017/07
	14		2017/11
	15		2018/02
	16	EL7221-9014-0029	2018/05
	17		2018/06
	18		2018/12
	19	EL7221-9014-0030	2020/02
	20	EL7221-9014-0031	2020/05
	21		2020/09
	22		2020/11
	23	EL7221-9014-0032	2021/01
	24		2021/06
06 - 08	25		2022/03
09*	26		2022/12
	27*		2023/06

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

9.3 Firmware Update

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

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website.

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

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

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

Storage locations

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

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

- 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.

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_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

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

9.3.1 Device description ESI file/XML

NOTICE

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:

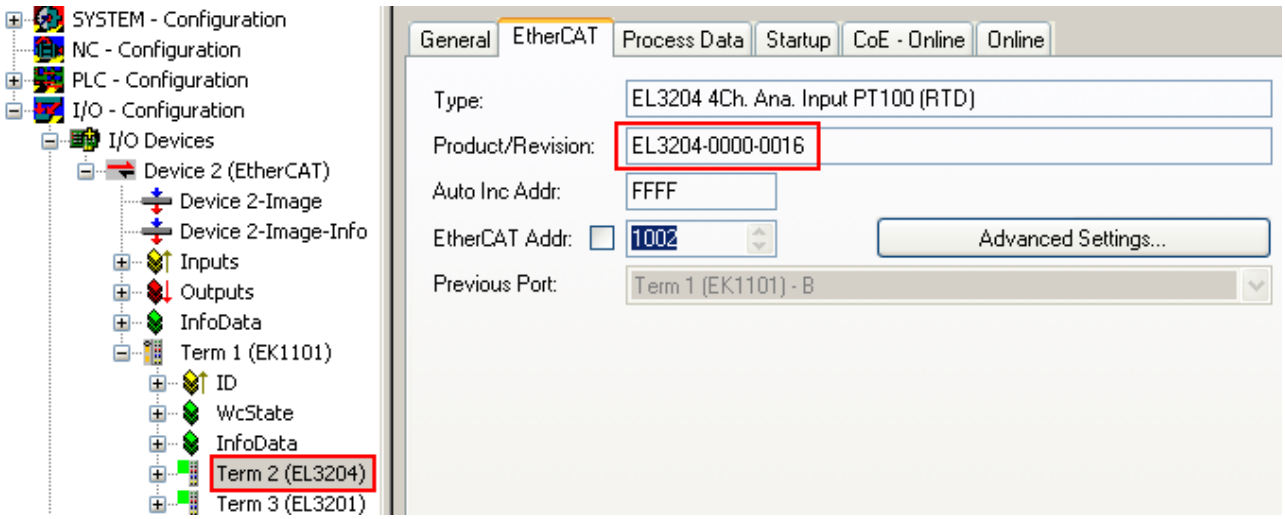


Fig. 217: Device identifier consisting of name EL3204-0000 and revision -0016

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

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:

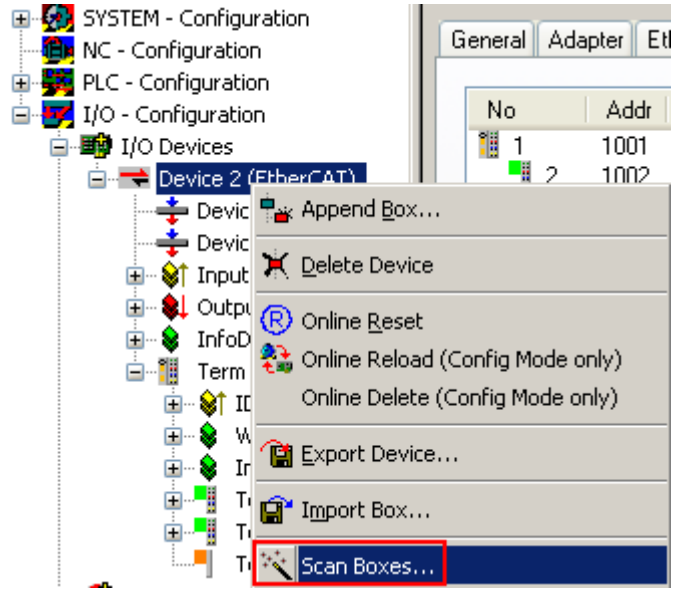


Fig. 218: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 219: Configuration is identical

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

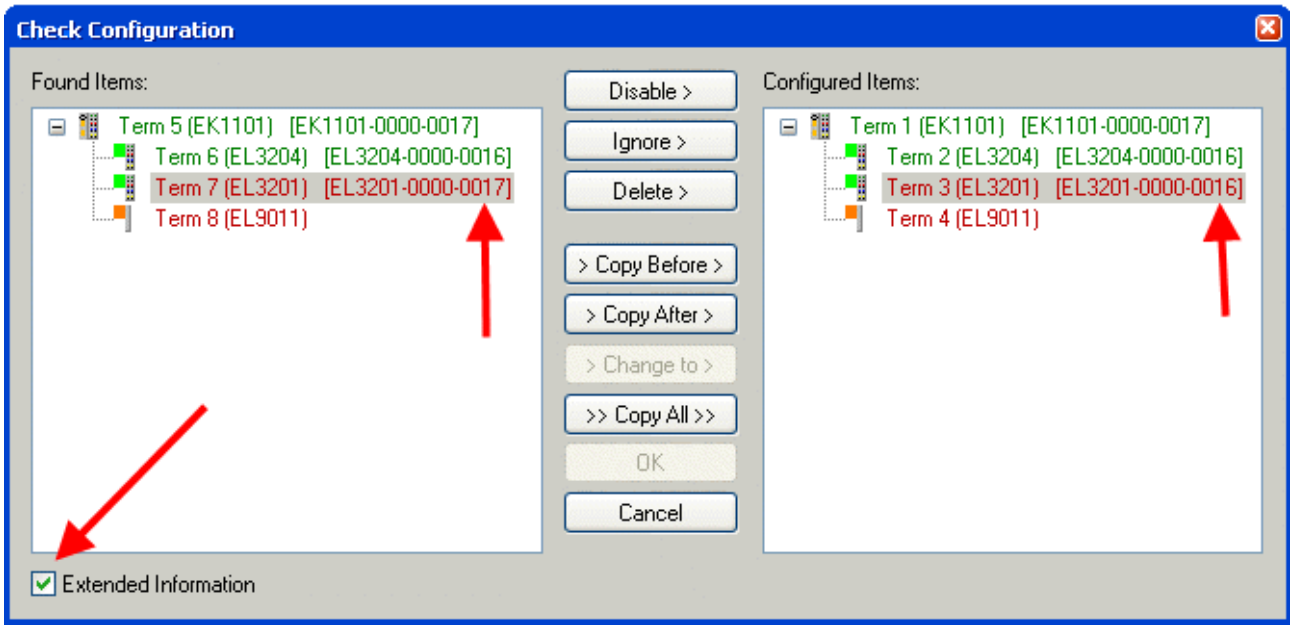


Fig. 220: Change dialog

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*

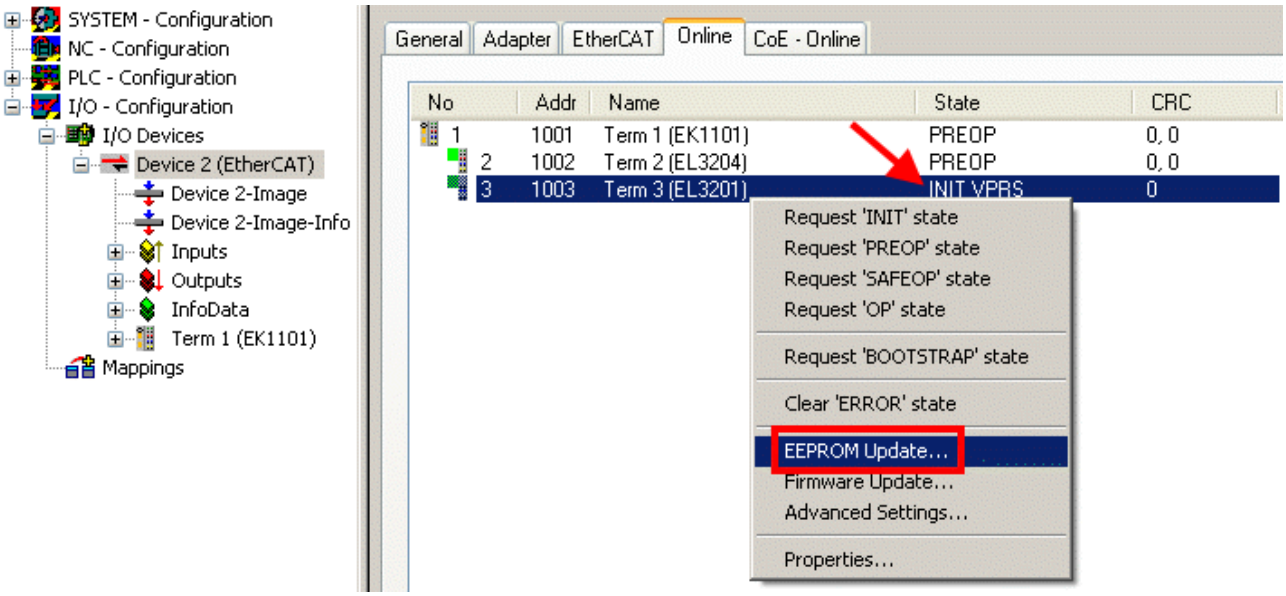


Fig. 221: EEPROM Update

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.

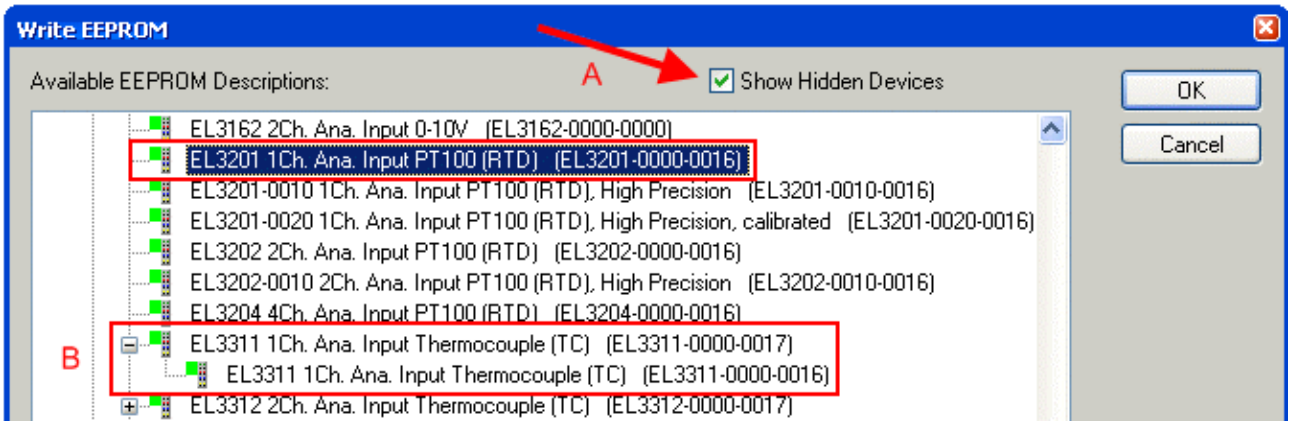


Fig. 222: Selecting the new ESI

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

The change only takes effect after a restart.

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.

9.3.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT 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

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.

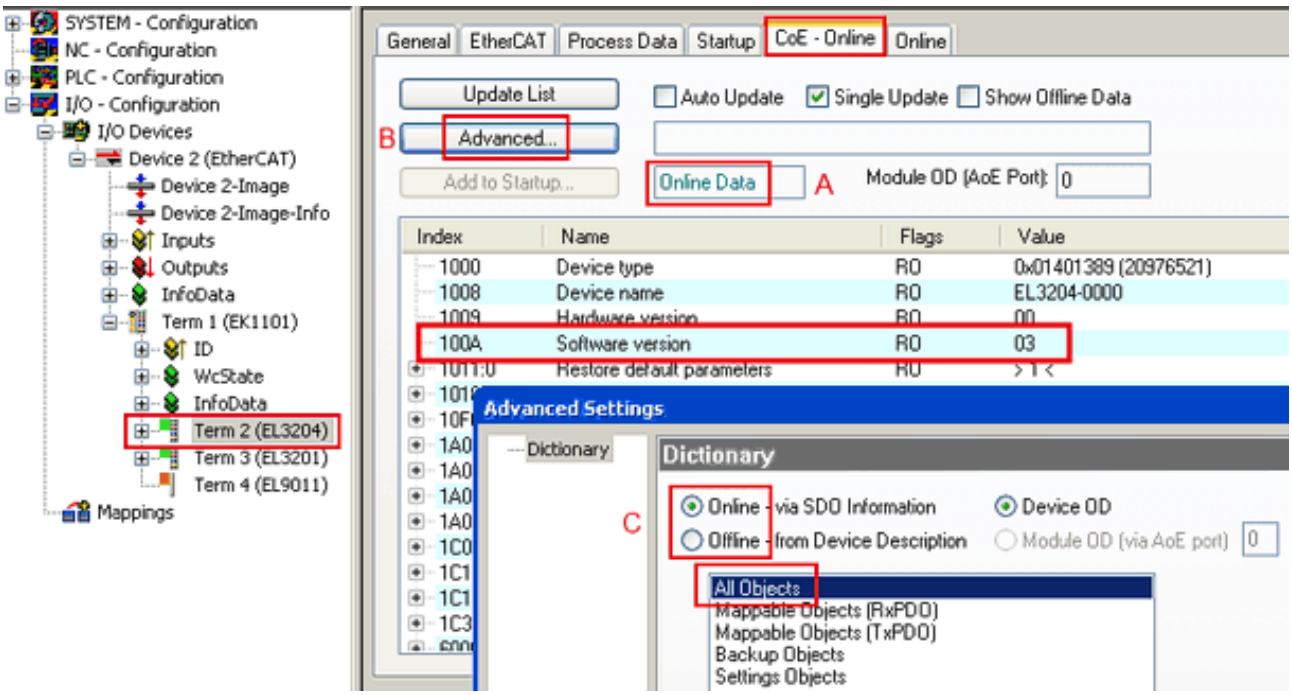


Fig. 223: Display of EL3204 firmware version

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

9.3.3 Updating controller firmware *.efw

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.*

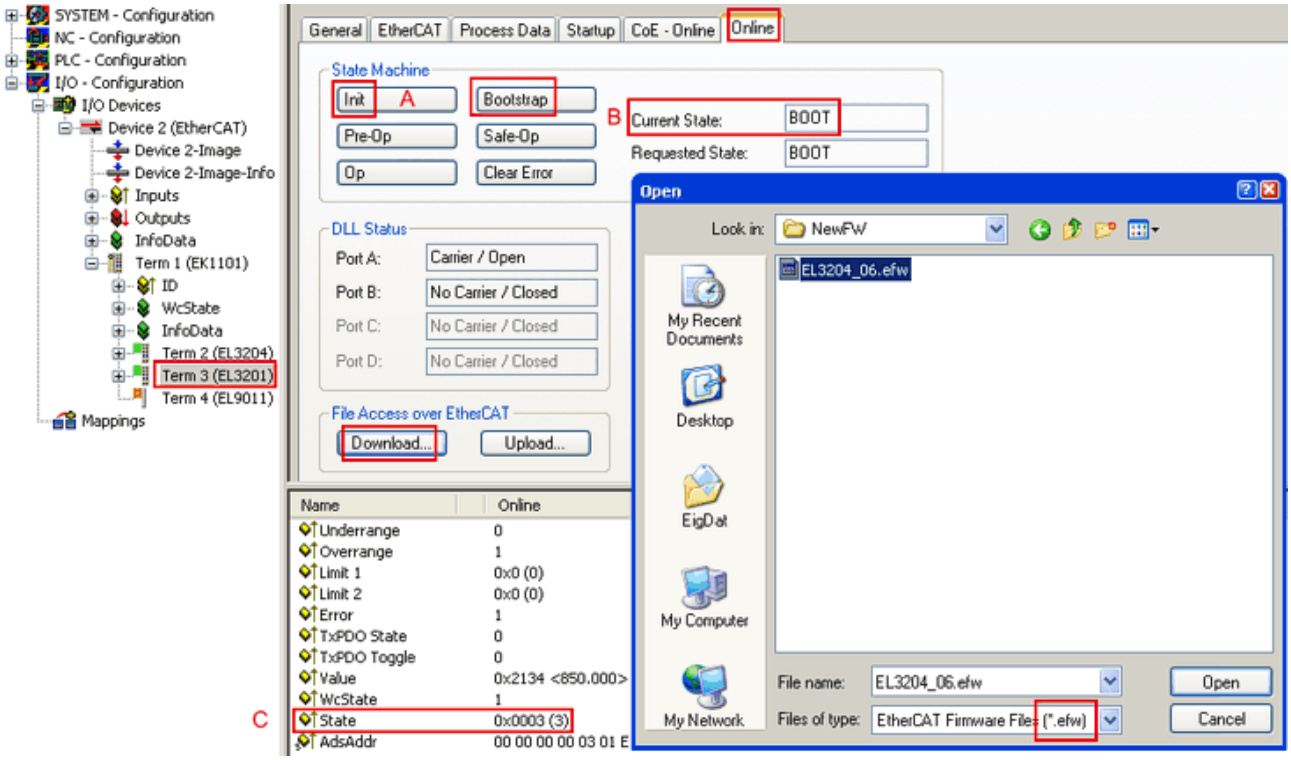
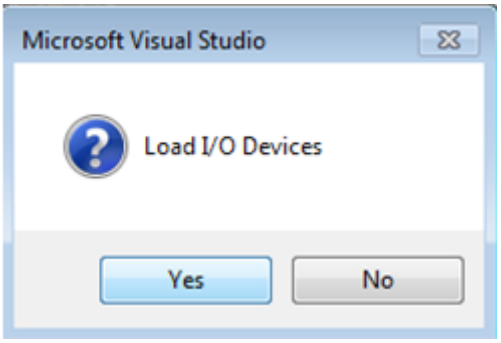
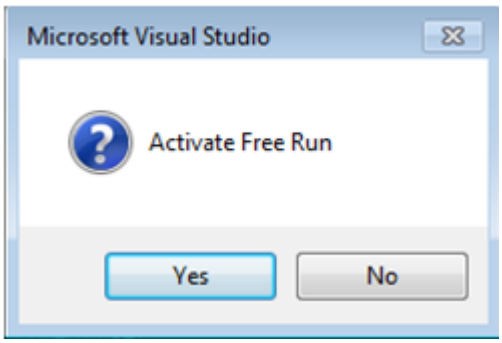


Fig. 224: Firmware Update

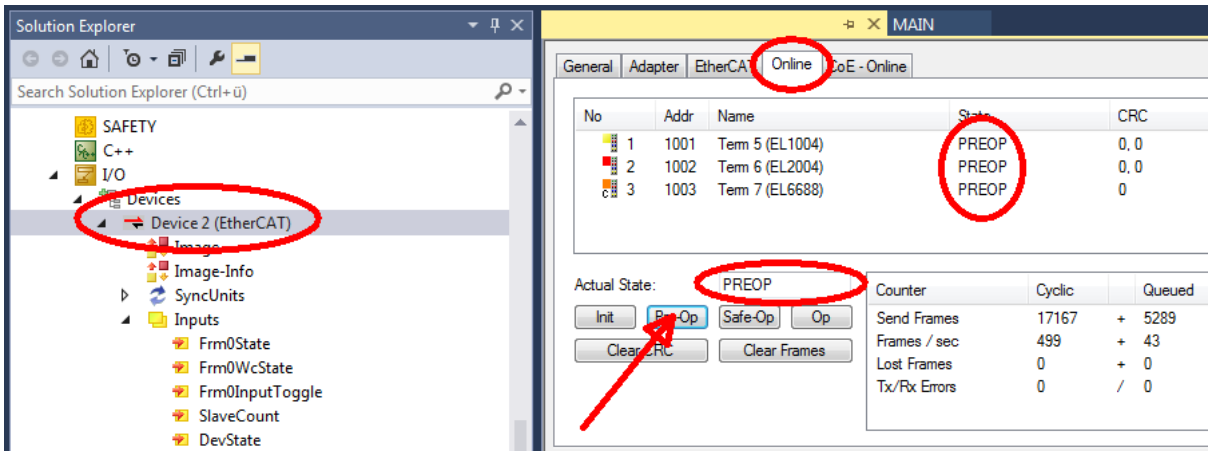
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

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



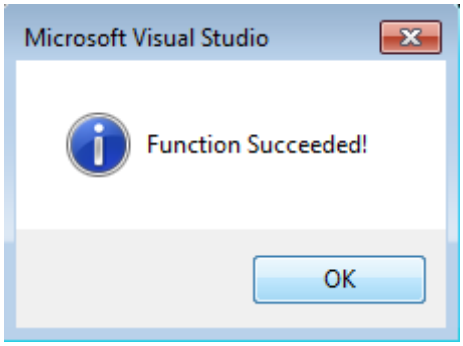


- Switch EtherCAT Master to PreOP



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

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



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

9.3.4 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 TwinCAT 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.

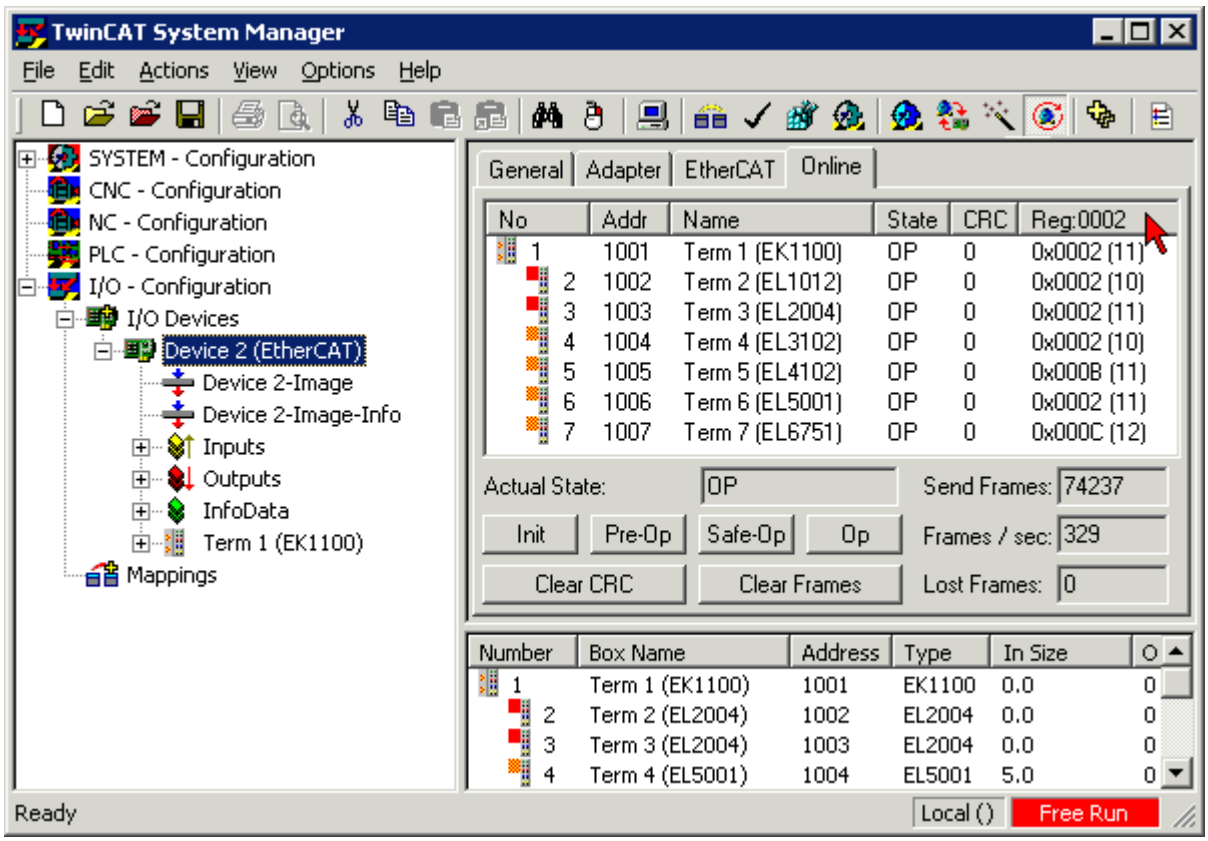


Fig. 225: FPGA firmware version definition

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

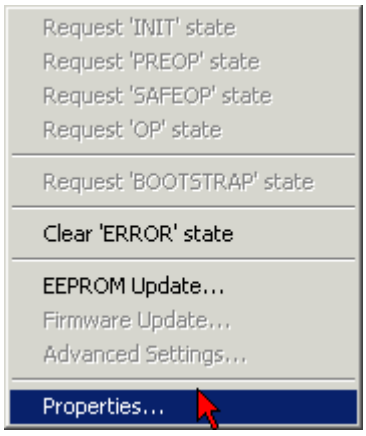


Fig. 226: Context menu *Properties*

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.

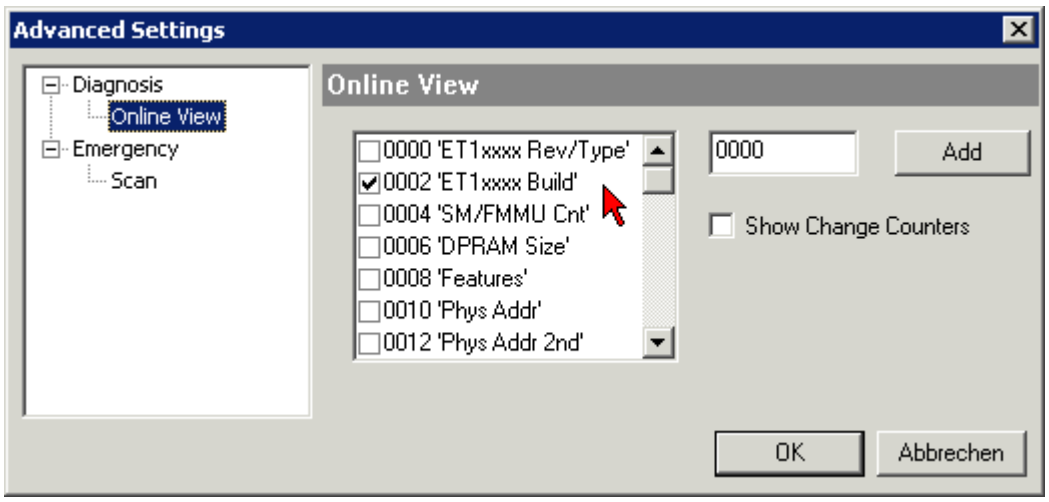


Fig. 227: Dialog *Advanced Settings*

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.

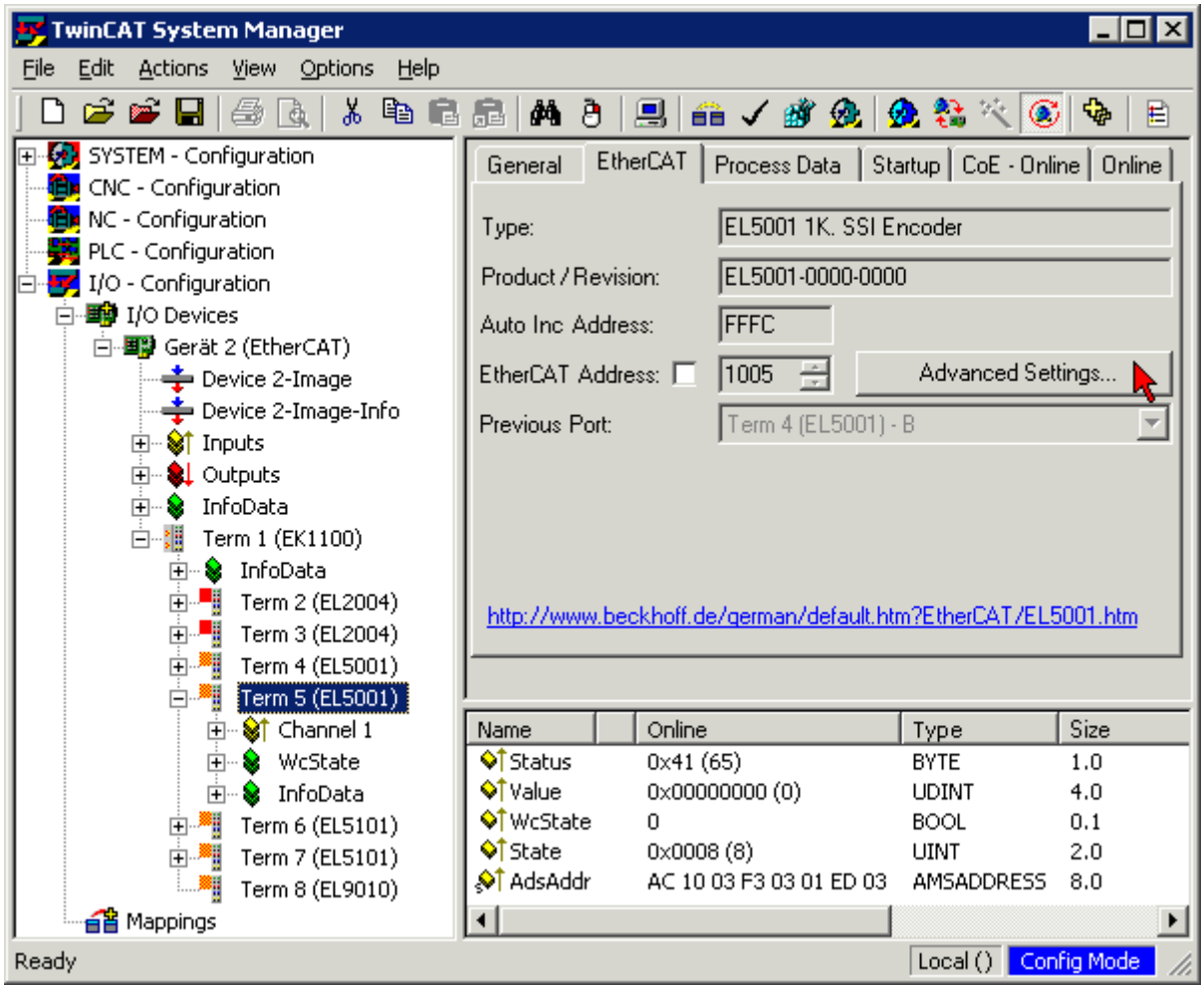
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

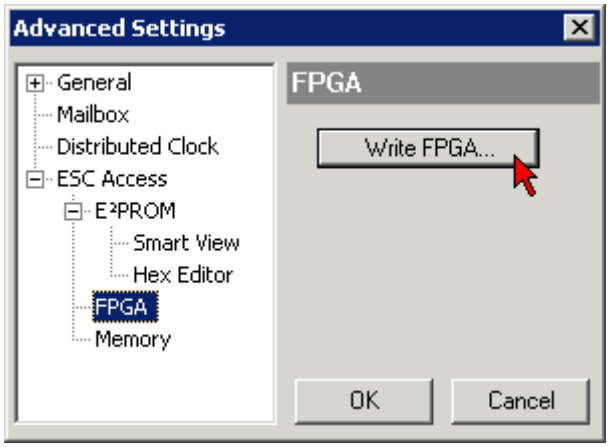
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

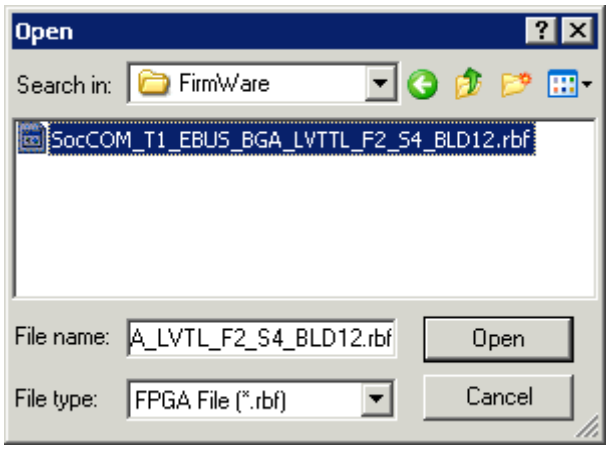
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



- 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

NOTICE

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!

9.3.5 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.

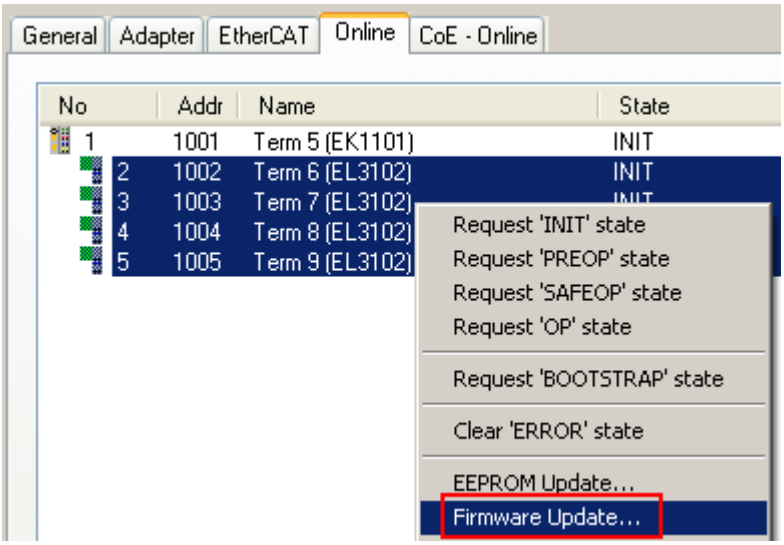


Fig. 228: Multiple selection and firmware update

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

9.4 Restoring the delivery state

Some EtherCAT devices (“Slaves”, “SubDevices”) save parameters or data locally in a persistent (non-volatile) manner. To reset them, the CoE object *Restore default parameters*, subindex 001 can be used via an EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the “Restore default parameters” PDO*)

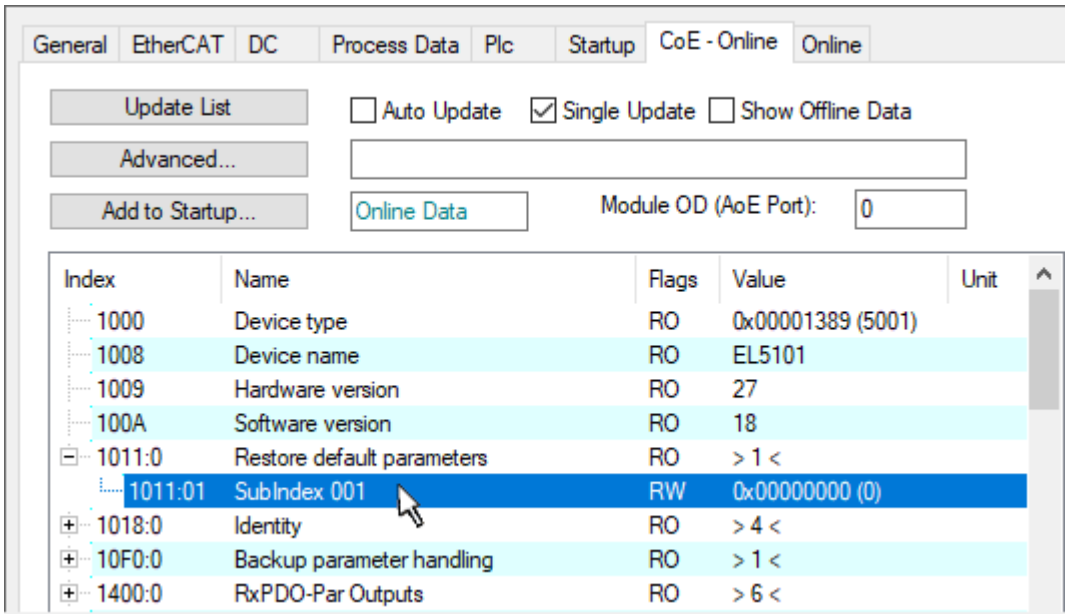


Fig. 229: Selecting the *Restore default parameters* PDO

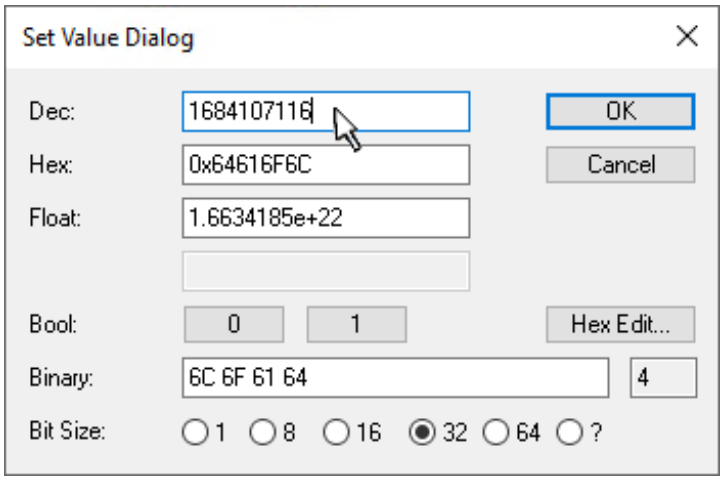


Fig. 230: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to open the Set Value dialog.

The aim is to reset function-controlling “parameters” (settings, values) that are persistently stored in the device and can be changed by the user to the default state.

Volatile (not power-fail-safe) “parameters” are reset

- by a RePower On/Off
- by an EtherCAT status change through INIT

unless otherwise described in the device documentation. Depending on the device and FW version, one or two reset functions are available; please refer to the respective device documentation for availability.

NOTICE

Changing existing checksums

Any existing checksums for the parameter ranges will (of course) change as a result of the reset.

Reset level 1: Function reset “load”

It is assumed that the function reset is used more often, possibly several times, e.g. during commissioning.

Enter the reset value **1684107116** in the *Dec* field or alternatively the value **0x64616F6C** in the *Hex* field (ASCII: “load”) and confirm with OK (Fig. *Entering the restore value in the Set Value dialog*).

- All changeable CoE entries are reset to the default values.
  Exception: objects protected by a write protection password
- Depending on the size of the object directory, this process can take from a few ms to > 1 second.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state, i.e. EtherCAT data exchange is taking place. Ensure error-free EtherCAT transmission.
- The reset does not result in a separate confirmation. A changeable object can be modified beforehand for checking purposes.
- This reset procedure can also be included as the first entry in the startup list of the slave, e.g. in the PREOP->SAFEOP state transition or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.
- In this way, backup objects are reset to the delivery state.

NOTICE

Depending on the device, individual values are not reset

Depending on the device, some values are intentionally not reset on “load” if the values are necessary for commissioning or essential device functions, for example, and determining them (after a deletion) would require considerable effort.

Alternative restore value

With some older terminals (FW created around 2007 or earlier), the backup objects can be changed with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect!

Reset level 2: Data reset “wipe”

It is assumed that a data reset is rarely used, e.g. when a product is decommissioned.

Enter the reset value **2003398757** in the *Dec* field or alternatively the value **0x77697065** in the *Hex* field (ASCII: “wipe”) and confirm with OK (Fig. *Entering the restore value in the Set Value dialog*).

- The aim of this function is a complete reset (data deletion) to the factory settings so that no user-specific settings can be read afterwards
- It affects the following parameter areas (if available on the device)
  - CoE (parameters)
  - SoE (parameters)
  - FoE (files on the device)
  - Register values (if stored persistently)
  - User memory such as NOVRAM (if available)
  - Additional data areas, if available
- Values/data are deleted even if they are protected, for example with a user-defined write-protection password.
- The following are not reset:
  - FW version and EtherCAT revision status; any updates remain installed on the device
  - Electrical health data such as Device Statistics (CoE: 0x1020), all types of operating hours counters, possibly histograms or peak values relating to ambient temperature or load conditions, …
  - Vendor calibration data protected by the vendor password

9.5 Certificates

ZERTIFIKAT ◆ CERTIFICATE ◆ 認証証書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

Product Service

CERTIFICATE

No. Z10 16 06 62386 027

Holder of Certificate: **Beckhoff Automation GmbH & Co. KG**
Hülshorstweg 20
33415 Verl
GERMANY

Factory(ies): 62386

Certification Mark:

Product: **Safety components**

Model(s): **EL72x1-9014**

Parameters:

Safety function:	STO (acc. EN 61800-5-2)
Supply voltage:	24VDC (-15%/+20%)
Protection class:	IP 20
Ambient temperature:	0°C ... +55°C

Tested according to:
EN ISO 13849-1:2015 (Cat 3, PL d)
EN 61000-6-2:2005
EN 61000-6-4:2007/A1:2011

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

Test report no.: BV88583T

Valid until: 2021-06-29

685473

Date, 2016-06-30 (Peter Weiss)

Page 1 of 1

A1 / 04.11

TÜV SÜD Product Service GmbH · Zertifizierstelle · Ridlerstraße 65 · 80339 München · Germany

Fig. 231: EL72x1-9014_Certificate

9.6 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: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

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

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

Hotline:	+49 5246 963 157
e-mail:	support@beckhoff.com
web:	www.beckhoff.com/support

Service

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

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

Hotline:	+49 5246 963 460
e-mail:	service@beckhoff.com
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

Trademark statements

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

Third-party trademark statements

CANopen and CANopen FD are registered trademarks of CAN in AUTOMATION - International Users and Manufacturers Group e.V.

ControlNet®, DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

EnDat is a trademark of Dr. Johannes Heidenhain GmbH.

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

More Information:
www.beckhoff.com/EL7xxx

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