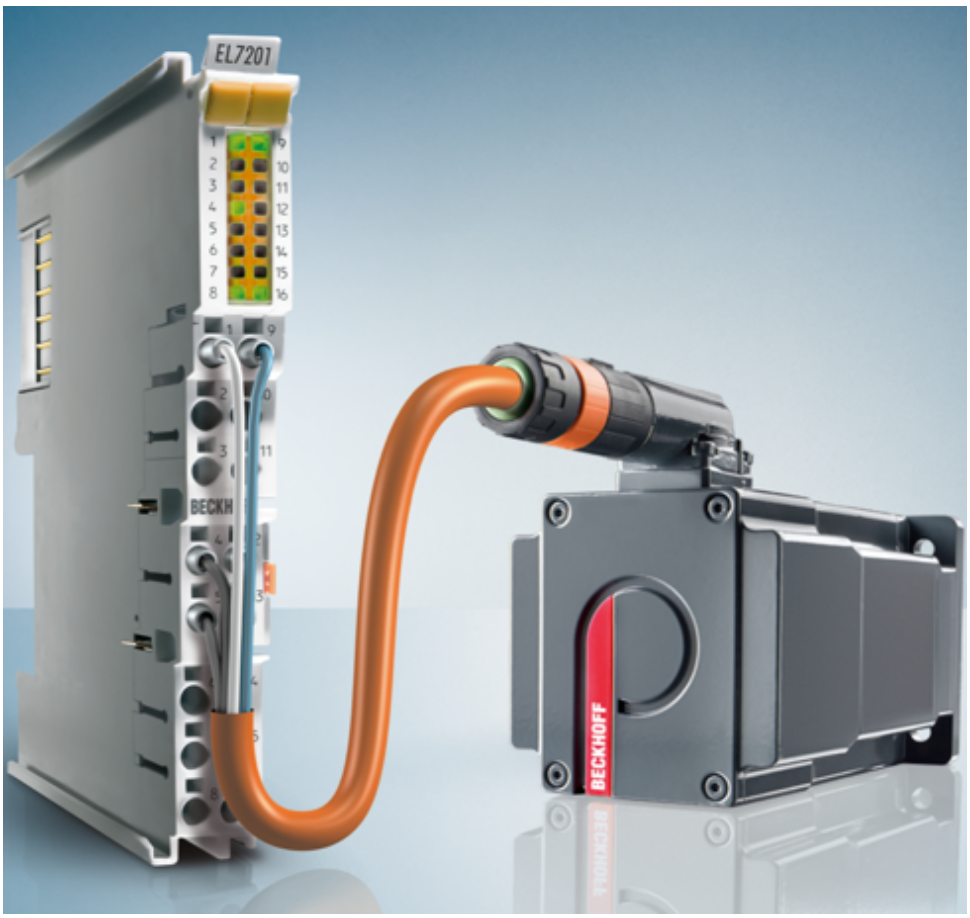


Documentation

EL72x1-0010

Servo Motor Terminals with OCT (One Cable Technology)

Version: 2.0
Date: 2015-11-30

BECKHOFF

1 Product overview Servomotor terminal

EL7201-0010 [▶ 12] Servomotor terminal 50 V_{DC} with OCT, 2.8 A_{rms}

EL7211-0010 [▶ 12] Servomotor terminal 50 V_{DC} with OCT, 4.5 A_{rms}

Table of contents

2 Foreword

2.1 Notes on the documentation

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.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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

Patent Pending

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

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.

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

Copyright

2.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

 DANGER	**Serious risk of injury!** Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.
 WARNING	**Risk of injury!** Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.
 CAUTION	**Personal injuries!** Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.
 Attention	**Damage to the environment or devices** Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.
 Note	**Tip or pointer** This symbol indicates information that contributes to better understanding.

2.3 Documentation issue status

Version	Comment
2.0	- Migration - Update structure - Update revision status
1.4	- Update chapter "Technical data" - Addenda chapter "Installation instructions for enhanced mechanical load capacity" - Update structure - Update revision status
1.3	- Addenda EL7211-0010 - Update structure
1.2	- Addenda chapter "Limit switches"
1.1	- Update MDP object description
1.0	- First public issue - Corrections and addenda
0.2	- Corrections and addenda
0.1	- Provisional documentation for EL72x1-0010

2.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

Notes

- the elements named above make up the **technical designation**
- The **order designation**, conversely, is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type
  - version
- The **revision** shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff website.
  The revision has been applied to the IP20 terminals on the outside since 2014/01, see fig. 1.

- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification number

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

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

Serial number is the name generally given to the 8-digit number that is printed on the device or attached to it on a sticker. This serial number indicates the as-built status on delivery and thus ambiguously marks a whole production lot.

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

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

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

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

Syntax: D ww yy x y z u

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

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

Unique serial number/ID

Beyond that there are some series in which each individual module has its own unique, sequential serial number.

See also the further documentation in the area

- IP67: EtherCAT Box
- Safety: TwinSafe

Examples of markings:

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

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

Fig. 3: CU2016 switch with batch number

Fig. 4: EL3202-0020 with batch numbers 26131006 and unique D-number 204418

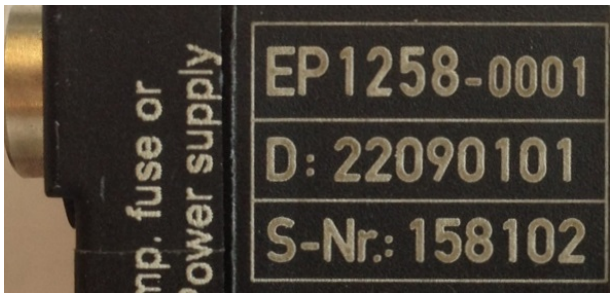


Fig. 5: EP1258-00001 IP67 EtherCAT Box with batch number 22090101 and serial number 158102

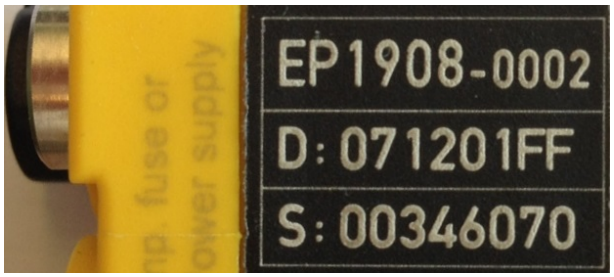


Fig. 6: EP1908-0002 IP76 EtherCAT Safety Box with batch number 071201FF and serial number 00346070

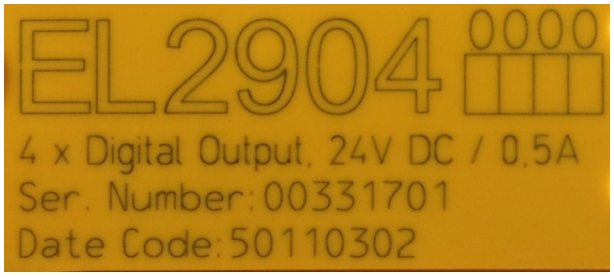


Fig. 7: EL2904 IP20 safety terminal with batch number/date code 50110302 and serial number 00331701

3 Product overview

3.1 Introduction

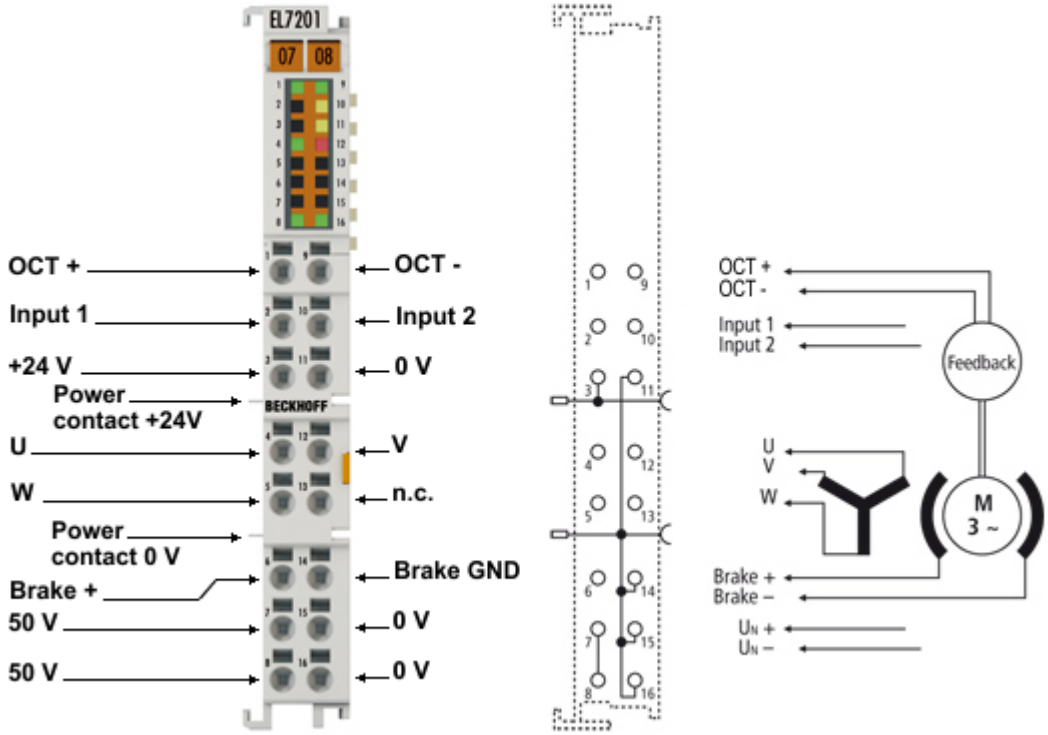


Fig. 8: EL7201

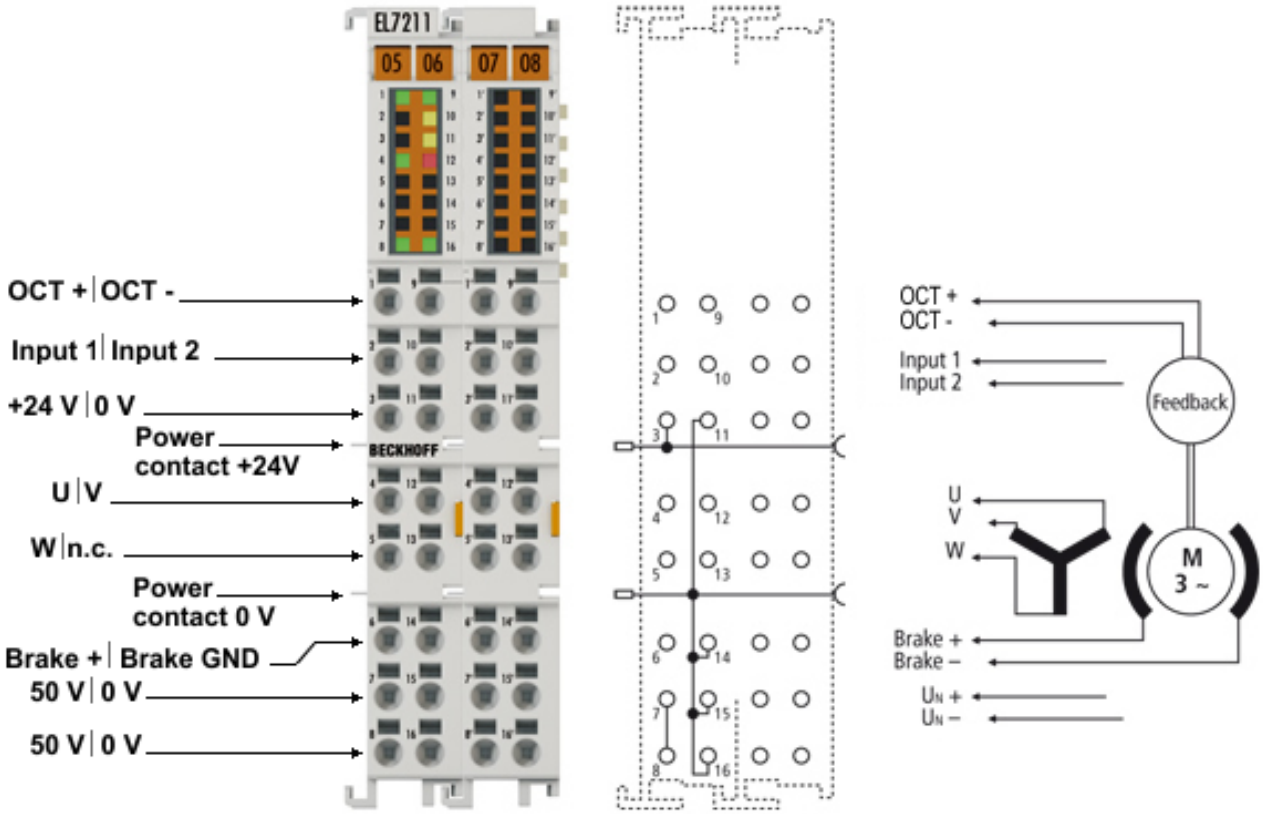


Fig. 9: EL7211

Servomotor terminals with OCT

The servomotor EtherCAT Terminals EL7201-0010 (50 V_{DC}, 2.8 A_{rms}) and EL7211-0010 (50 V_{DC}, 4.5 A_{rms}) with integrated absolute value interface offer high servo performance with a very compact design. The EL72x1-0010 was 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 a 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.

Note	**Recommended TwinCAT version** In order to be able to utilize the full power of the EL72x1-0010, we recommend using the EL72x1-0010 with TwinCAT 2.11 R3 or higher!

Note	**Mandatory hardware** The EL72x1-0010 must be operated with a real-time capable computer and distributed clocks.

Note	**Approved motors** Trouble-free operation can only be guaranteed with motors approved by Beckhoff.

Quick links

Connection instructions

- Chapter "Mounting and wiring",
  - LEDs and pin assignment [▶ 40]
  - Shielding concept [▶ 37]
  - Notes on current measurement via Hall sensor [▶ 38]

Configuration instructions

- Chapter "Commissioning",
  - Configuration of the main parameters [▶ 85]
- Chapter "Configuration with the TwinCAT System Manager",
  - Object description and parameterization [▶ 165]

Application example

- Chapter "Commissioning",
  - Application example [▶ 104]

3.2 Technical data

Prerequisites

Technical data	EL7201-0010	EL7211-0010
Number of outputs	3 motor phases, 2 motor holding brake	
Number of inputs	2 (4) DC link voltage, 2 absolute feedback, 2 digital inputs	
DC link supply voltage	8...50 V_{DC}	
Supply voltage	24 V_{DC} via the power contacts / via the E-bus	
Output current	2.8 A_{rms}	4.5 A_{rms}
Peak current	5.7 A_{rms} for 1 second	9 A_{rms} for 1 second
Rated power	170 W	276 W
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	16kHz	
Current controller frequency	double PWM switching frequency	
Velocity controller frequency	16 kHz	
Diagnostics 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. 55 mA + holding brake	
Supports NoCoeStorage [▶ 23] 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	
Fusing (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	
Weight	approx. 60 g	approx. 95 g
Permissible ambient temperature range during operation	0°C ... + 55°C	
Permissible ambient temperature range during storage	-25°C ... + 85°C	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 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)
Mounting [▶ 29]	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 31] 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	
Installation position	see note [▶ 36]	
Approval	CE	

3.3 Technology

The very compact servomotor terminal EL72x1-0010 integrates a complete servo drive for servomotors up to 170 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 EL72x1-0010 servo terminal supports control of permanent magnet synchronous motors. These consist of 3 coils which are offset by 120° and a permanent magnet rotor.

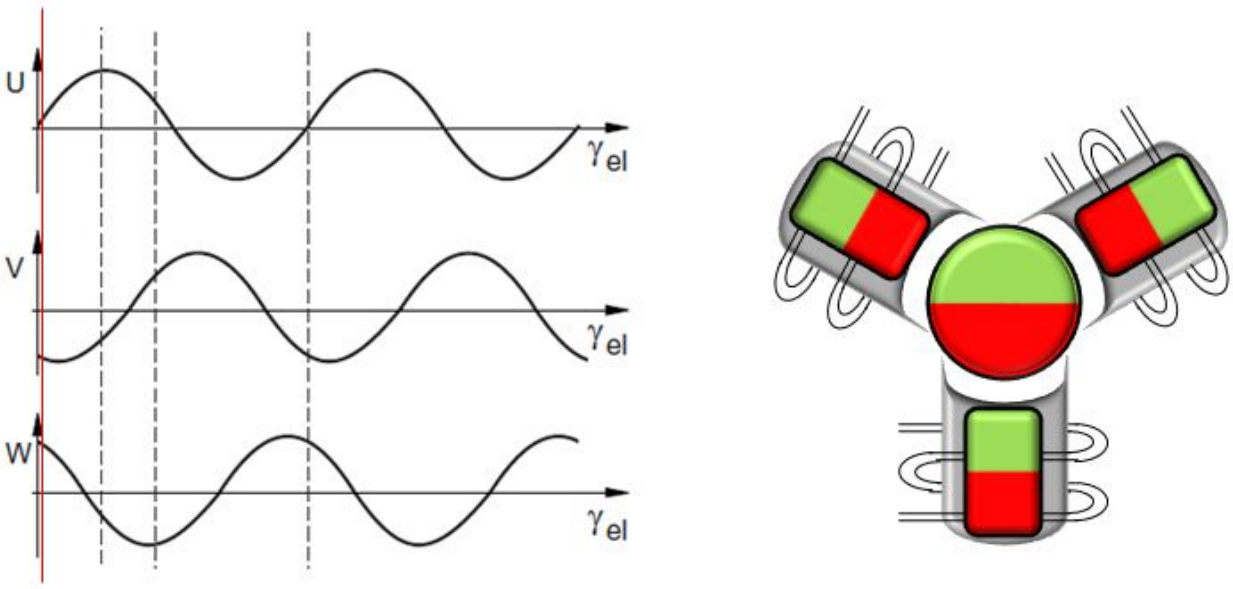


Fig. 10: Three synchronous motor coils, each offset by 120°

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 EL72x1-0010 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 EL72x1-0010 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 50 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 EL7201-0010 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.

Connection to the control system

A further big advantage of the EL72x1-0010 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 EL72x1-0010 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 EL72x1-0010 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 EL72x1-0010, 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 EL72x1-0010 servomotor terminal.

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.

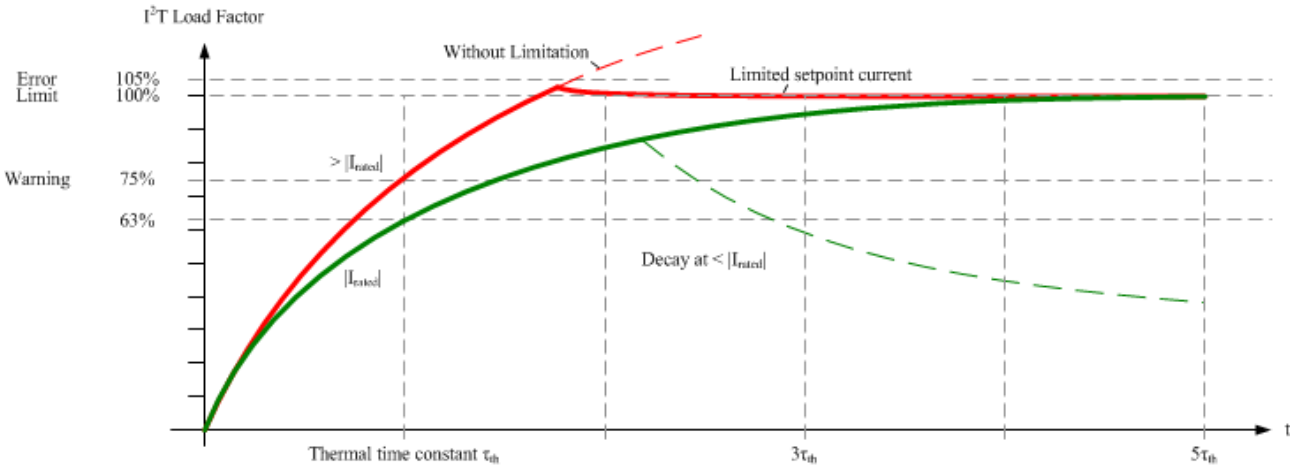


Fig. 11: Limitation to the rated motor current

3.4 Start

For commissioning:

- mount the EL72x1-0010 as described in the chapter Mounting and wiring [▶ 29]
- configure the EL72x1-0010 in TwinCAT as described in the chapter Commissioning [▶ 44].

4 Basics communication

4.1 EtherCAT basics

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

4.2 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.

Note

Recommended cables

Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff web-site!

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 2A 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.

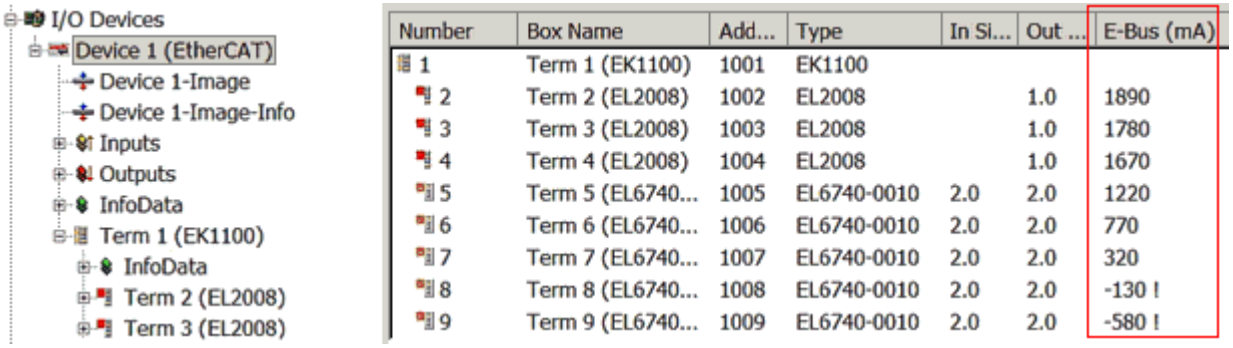


Fig. 12: System manager current calculation

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

4.3 General notes for setting the watchdog

ELxxxx terminals are equipped with a safety feature (watchdog) that switches off the outputs after a specifiable time e.g. in the event of an interruption of the process data traffic, depending on the device and settings, e.g. in OFF state.

The EtherCAT slave controller (ESC) in the EL2xxx terminals features 2 watchdogs:

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

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset after each successful EtherCAT process data communication with the terminal. If no EtherCAT process data communication takes place with the terminal for longer than the set and activated SM watchdog time, e.g. in the event of a line interruption, the watchdog is triggered and the outputs are set to FALSE. The OP state of the terminal is unaffected. The watchdog is only reset after a successful EtherCAT process data access. Set the monitoring time as described below.

The SyncManager watchdog monitors correct and timely process data communication with the ESC from the EtherCAT side.

PDI watchdog (Process Data Watchdog)

If no PDI communication with the EtherCAT slave controller (ESC) takes place for longer than the set and activated PDI watchdog time, this watchdog is triggered.
PDI (Process Data Interface) is the internal interface between the ESC and local processors in the EtherCAT slave, for example. The PDI watchdog can be used to monitor this communication for failure.

The PDI watchdog monitors correct and timely process data communication with the ESC from the application side.

The settings of the SM- and PDI-watchdog must be done for each slave separately in the TwinCAT System Manager.

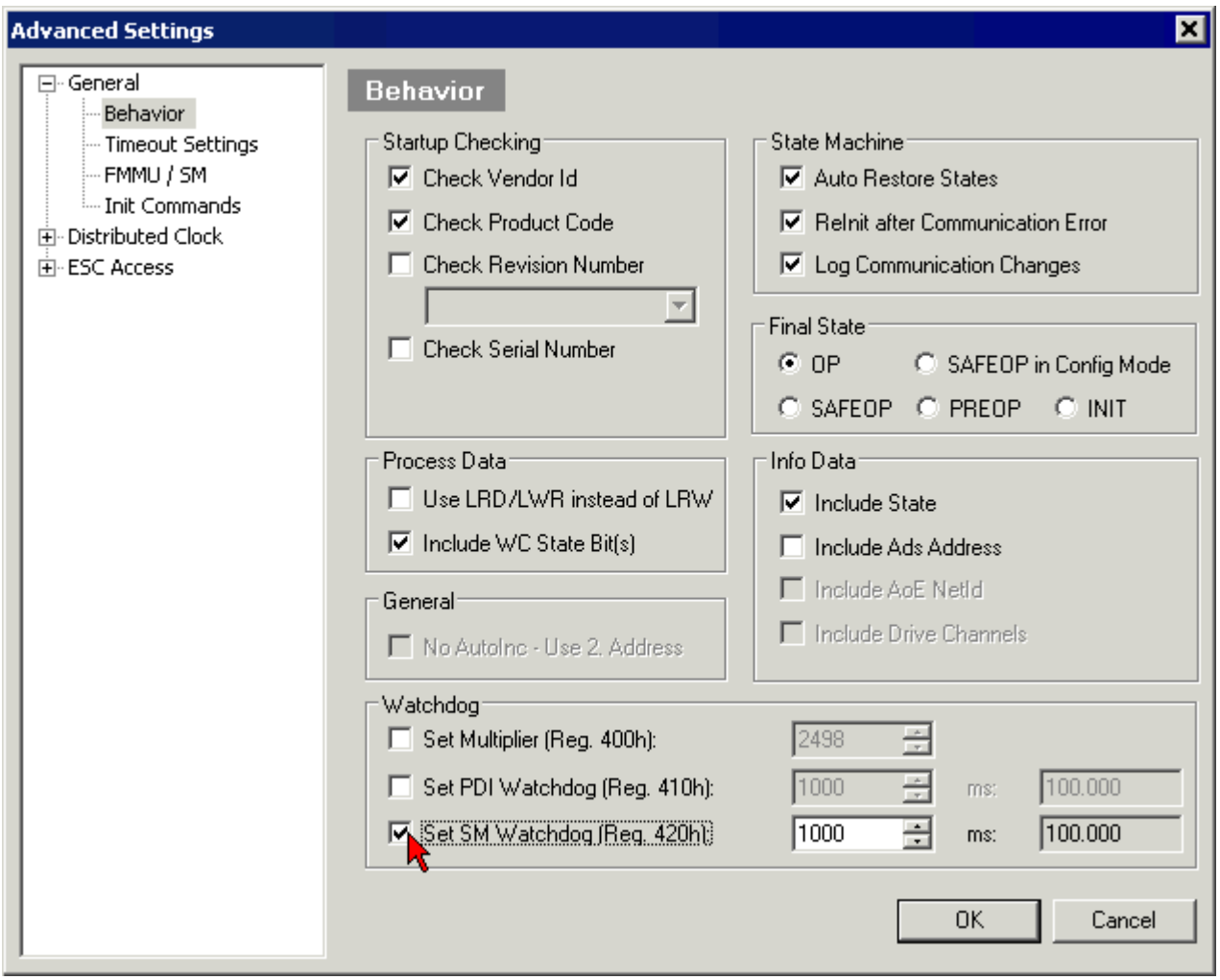


Fig. 13: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the multiplier is valid for both watchdogs.
- each watchdog has its own timer setting, the outcome of this in summary with the multiplier is a resulting time.
- Important: the multiplier/timer setting is only loaded into the slave at the start up, if the checkbox is activated.
  If the checkbox is not activated, nothing is downloaded and the ESC settings remain unchanged.

Multiplier

Multiplier

Both watchdogs receive their pulses from the local terminal cycle, divided by the watchdog multiplier:

1/25 MHz * (watchdog multiplier + 2) = 100 µs (for default setting of 2498 for the multiplier)

The standard setting of 1000 for the SM watchdog corresponds to a release time of 100 ms.

The value in multiplier + 2 corresponds to the number of basic 40 ns ticks representing a watchdog tick. The multiplier can be modified in order to adjust the watchdog time over a larger range.

Example "Set SM watchdog"

This checkbox enables manual setting of the watchdog times. If the outputs are set and the EtherCAT communication is interrupted, the SM watchdog is triggered after the set time and the outputs are erased. This setting can be used for adapting a terminal to a slower EtherCAT master or long cycle times. The default SM watchdog setting is 100 ms. The setting range is 0..65535. Together with a multiplier with a range of 1..65535 this covers a watchdog period between 0..~170 seconds.

Calculation

Multiplier = 2498 → watchdog base time = 1 25 MHz * (2498 + 2) = 0.0001 seconds = 100 µs
SM watchdog = 10000 → 10000 * 100 µs = 1 second watchdog monitoring time

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

 CAUTION	**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.

4.4 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 and
- Operational
- Boot

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

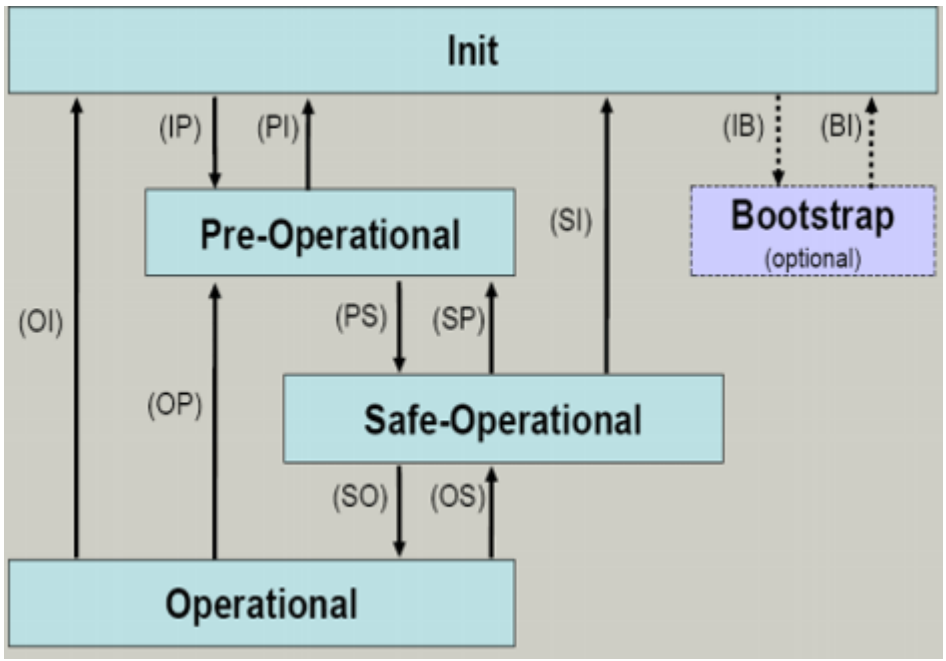


Fig. 14: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

Note	**Outputs in SAFEOP state** The default set watchdog [▶ 19] monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

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

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

Boot

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

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

4.5 CoE Interface

General description

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

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

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

The order is specified in 2 levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

- 0x4000: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

Note	**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:

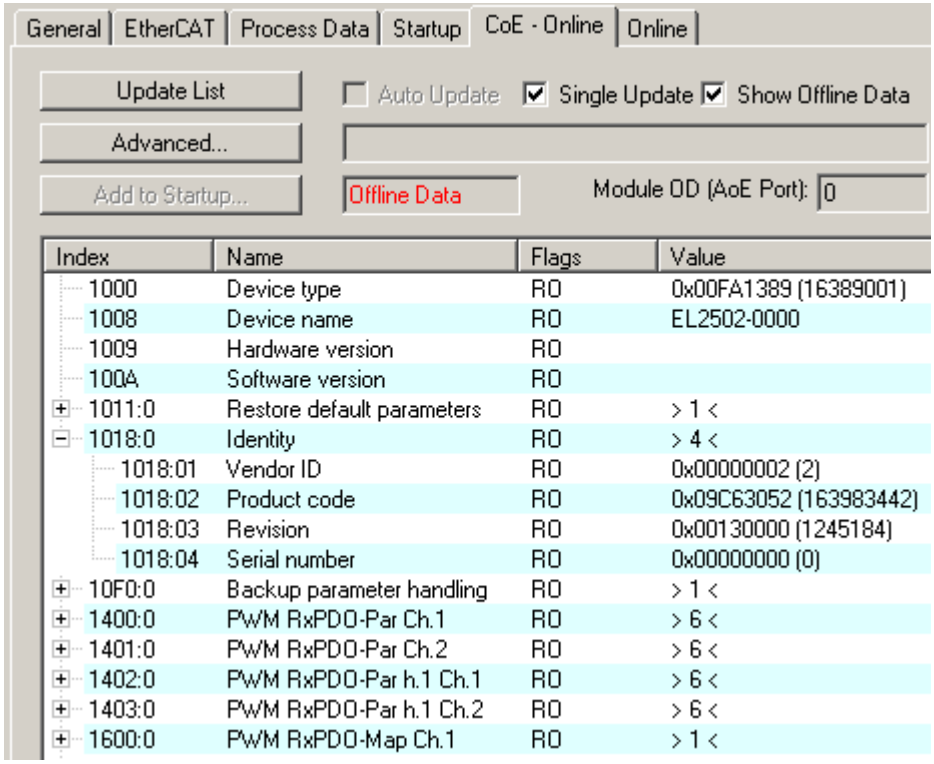


Fig. 15: "CoE Online " tab

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

Data management

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

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

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.

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

Note	**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 parameterised 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.

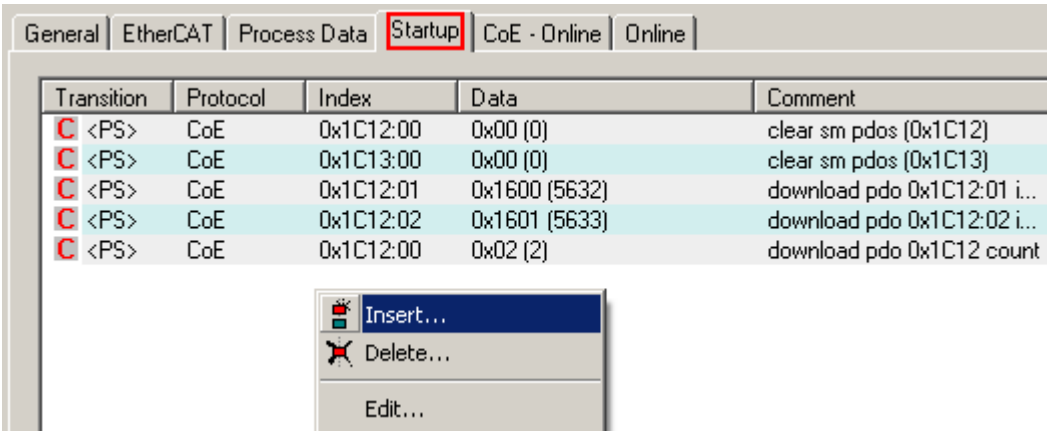


Fig. 16: Startup list in the TwinCAT System Manager

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

Online/offline list

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

In both cases a CoE list as shown in Fig. *"'CoE online' tab"* is displayed. The connectivity is shown as offline/online.

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

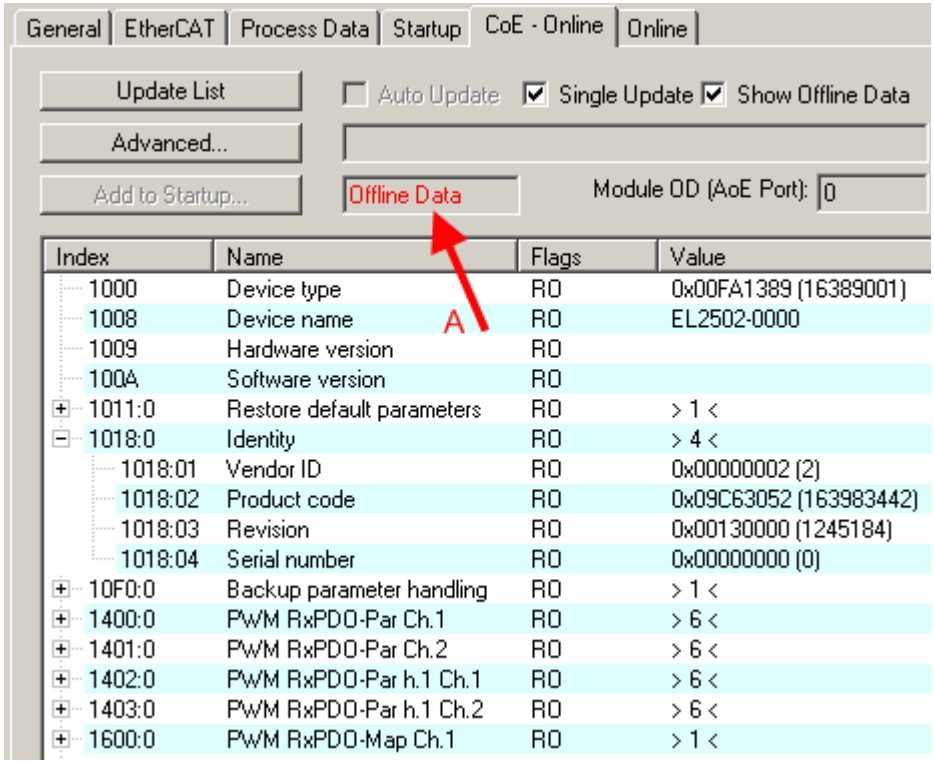


Fig. 17: Offline list

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

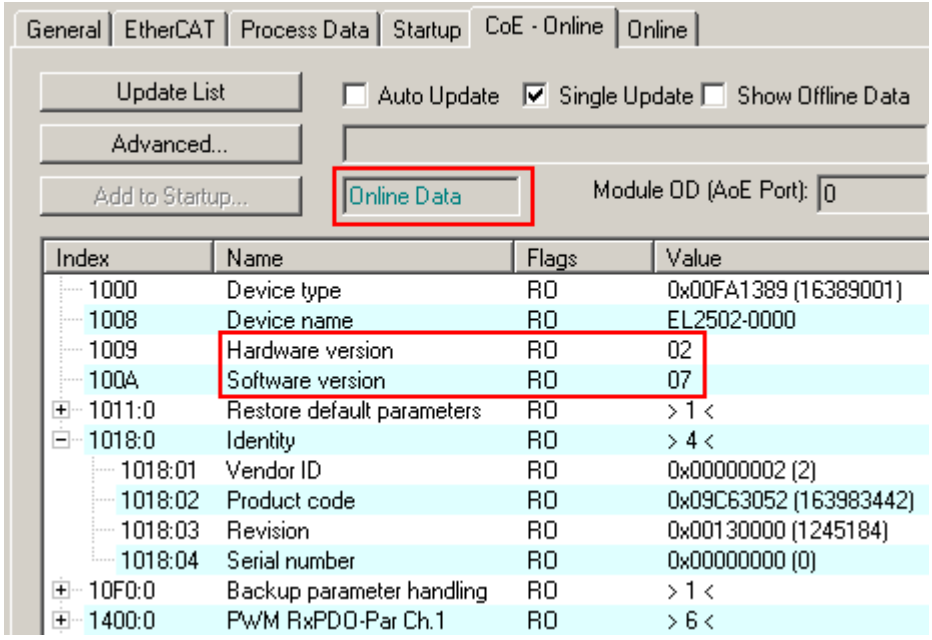


Fig. 18: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

4.6 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.

5 Installation

5.1 Installation on mounting rails

WARNING	**Risk of electric shock and damage of device!** Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Assembly

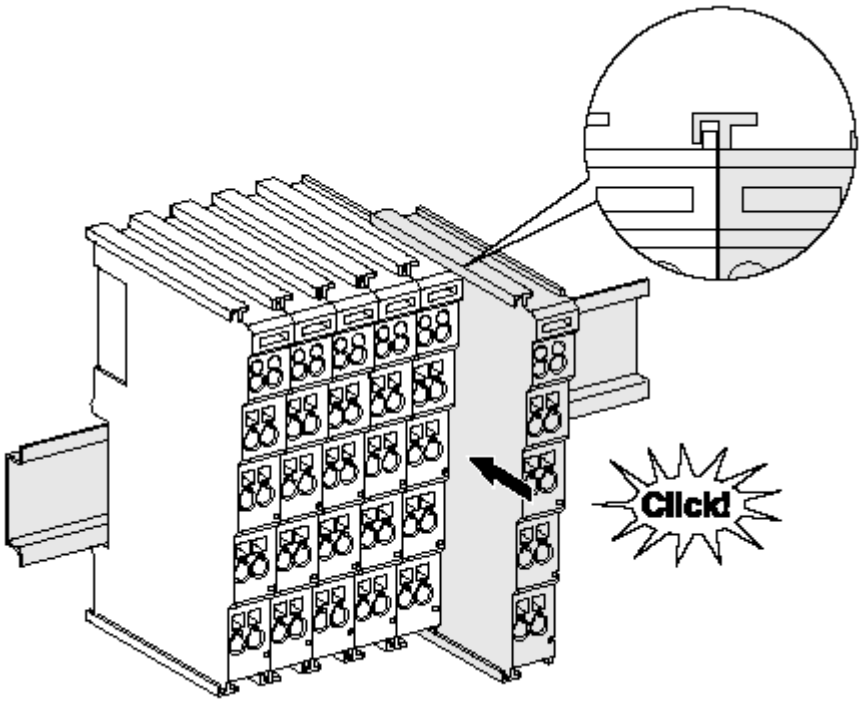


Fig. 19: Attaching on mounting rail

The Bus Coupler and Bus Terminals are attached to commercially available 35 mm mounting rails (DIN rails 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. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the Terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

Note	**Fixing of mounting rails** The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

Disassembly

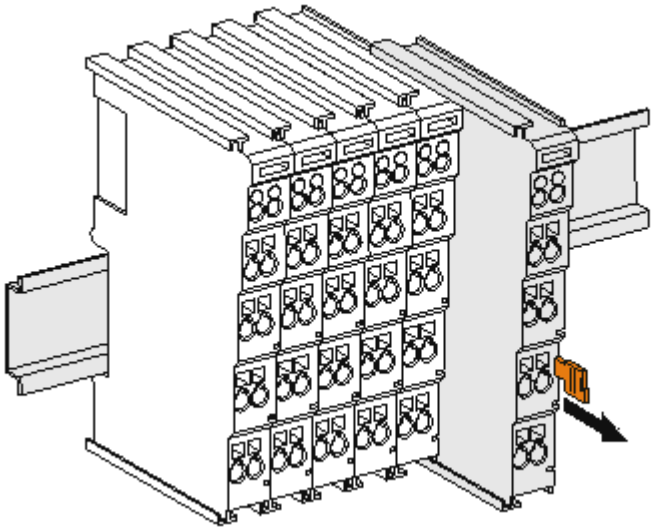


Fig. 20: Disassembling of terminal

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

1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the K-Bus/E-Bus deal with the transfer of the data 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 terminals on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

Note	**Power Contacts** During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (KL91xx, KL92xx or EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

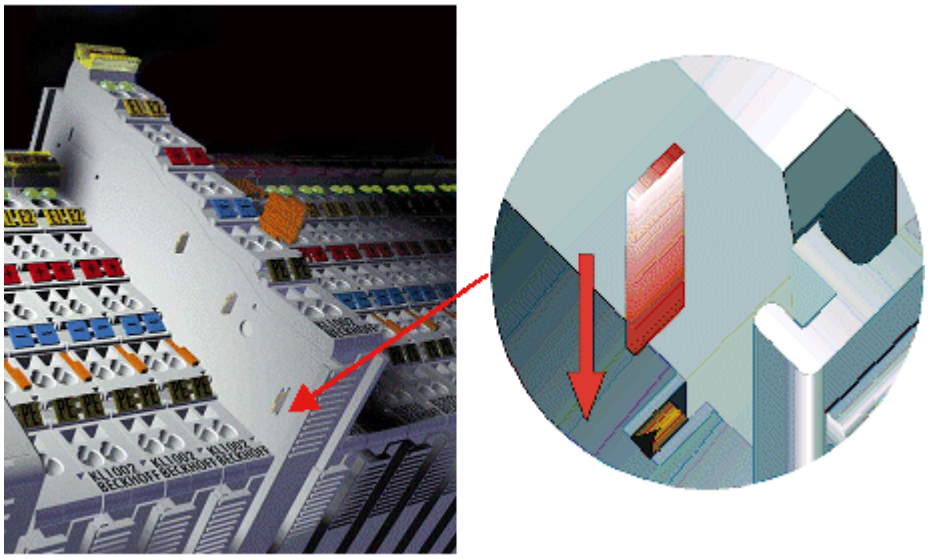


Fig. 21: Power contact on left side

Attention	**Possible damage of the device** Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

WARNING	**Risk of electric shock!** The PE power contact must not be used for other potentials!

5.2 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

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

- Any installation position is permitted
- 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 with) or 32 terminals (24 mm mounting with)
- 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.

5.3 Connection system

 **WARNING**	**Risk of electric shock and damage of device!** Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Overview

The Bus Terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of KLxxxx and ELxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of KSxxxx and ESxxxx 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

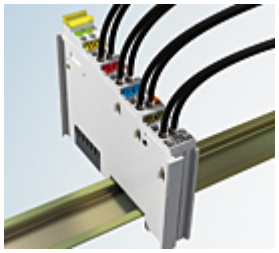


Fig. 22: Standard wiring

The terminals of KLxxxx and ELxxxx series have been tried and tested for years.
They feature integrated screwless spring force technology for fast and simple assembly.

Pluggable wiring

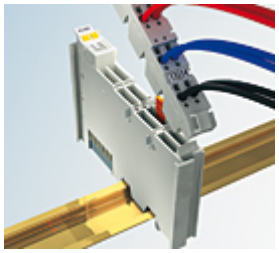


Fig. 23: Pluggable wiring

The terminals of KSxxxx and ESxxxx series feature a pluggable connection level.
The assembly and wiring procedure for the KS series is the same as for the KLxxxx and ELxxxx series.
The KS/ES series terminals enable 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 KSxxxx and ESxxxx series has been retained as known from KLxxxx and ELxxxx series.

High Density Terminals (HD Terminals)

Fig. 24: High Density Terminals

The Bus Terminals from these series with 16 connection 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.

Note	**Wiring HD Terminals** The High Density (HD) Terminals of the KLx8xx and ELx8xx series doesn't support steady wiring.

Ultrasonically "bonded" (ultrasonically welded) conductors

Note	**Ultrasonically “bonded" conductors** It is also possible to connect the Standard and High Density Terminals with ultrasonically "bonded" (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [▶ 34] below!

Wiring

Terminals for standard wiring ELxxxx / KLxxxx and terminals for steady wiring ESxxxx / KSxxxx

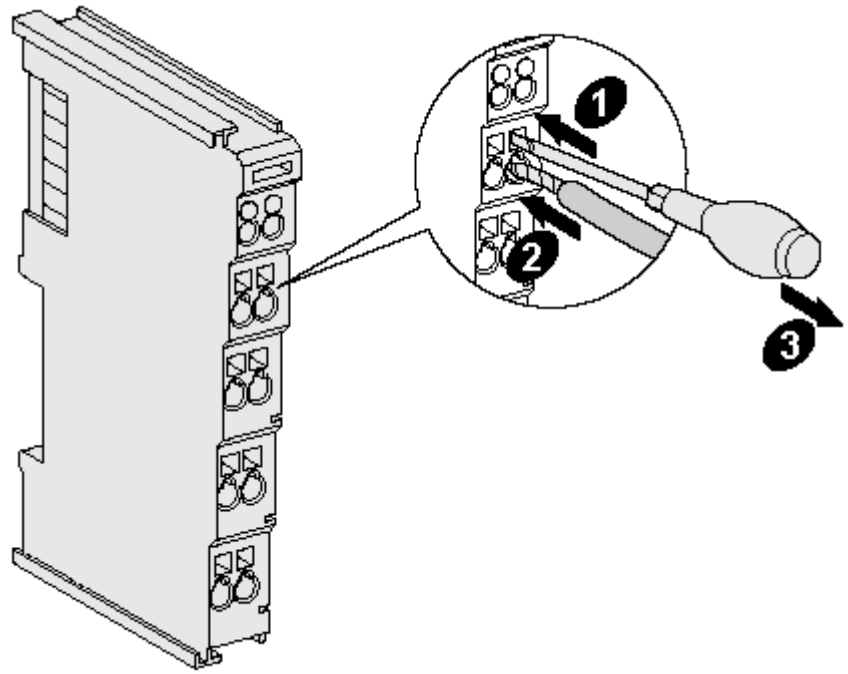


Fig. 25: Mounting a cable on a terminal connection

Up to eight connections enable the connection of solid or finely stranded cables to the Bus Terminals. The terminals are implemented in spring force technology. Connect the cables as follows:

1. Open a spring-loaded terminal by slightly pushing with a screwdriver or a rod into the square opening above the terminal.
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal closes automatically when the pressure is released, holding the wire securely and permanently.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width	0.08 ... 2,5 mm^2	0.08 ... 2.5 mm^2
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals ELx8xx, KLx8xx (HD)

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 contact 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 (conductors with a wire end sleeve)	0.14... 0.75 mm^2
Wire size width (single core wires)	0.08 ... 1.5 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm^2 (see notice [▶ 33]!)
Wire stripping length	8 ... 9 mm

Shielding

Note	**Shielding** Analog sensors and actors should always be connected with shielded, twisted paired wires.

5.4 Mounting of Passive Terminals

Note	**Hint for mounting passive terminals** EtherCAT Bus Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called Passive Terminals. The Passive Terminals have no current consumption out of the E-Bus To ensure an optimal data transfer, you must not directly string together more than 2 Passive Terminals!

Examples for mounting passive terminals (highlighted)

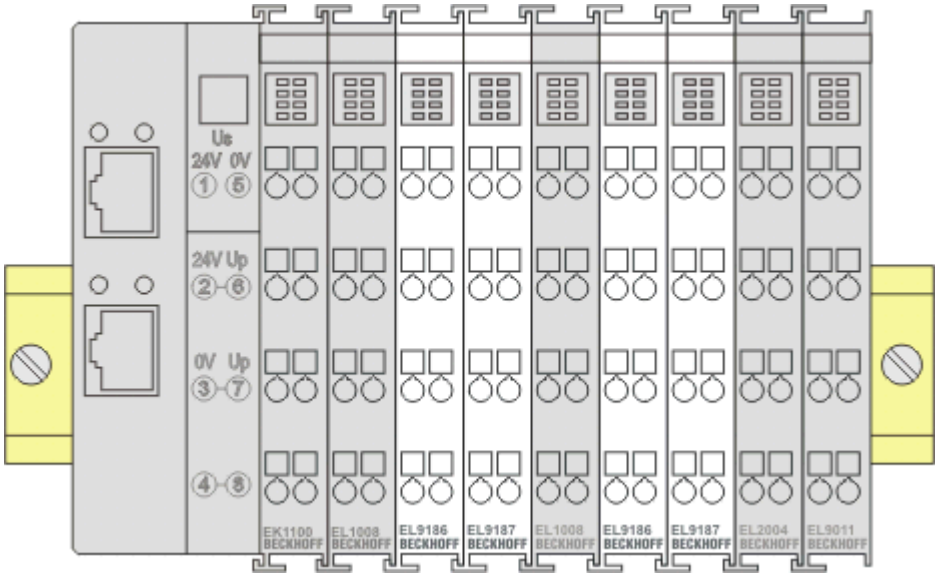


Fig. 26: Correct configuration

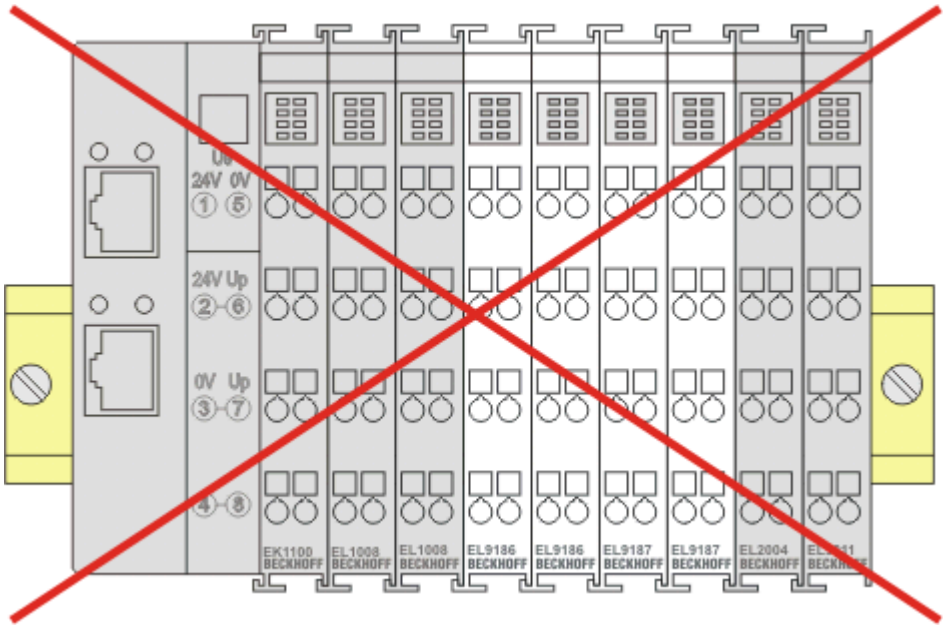


Fig. 27: Incorrect configuration

5.5 Installation positions

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

Optimum installation position (standard)

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

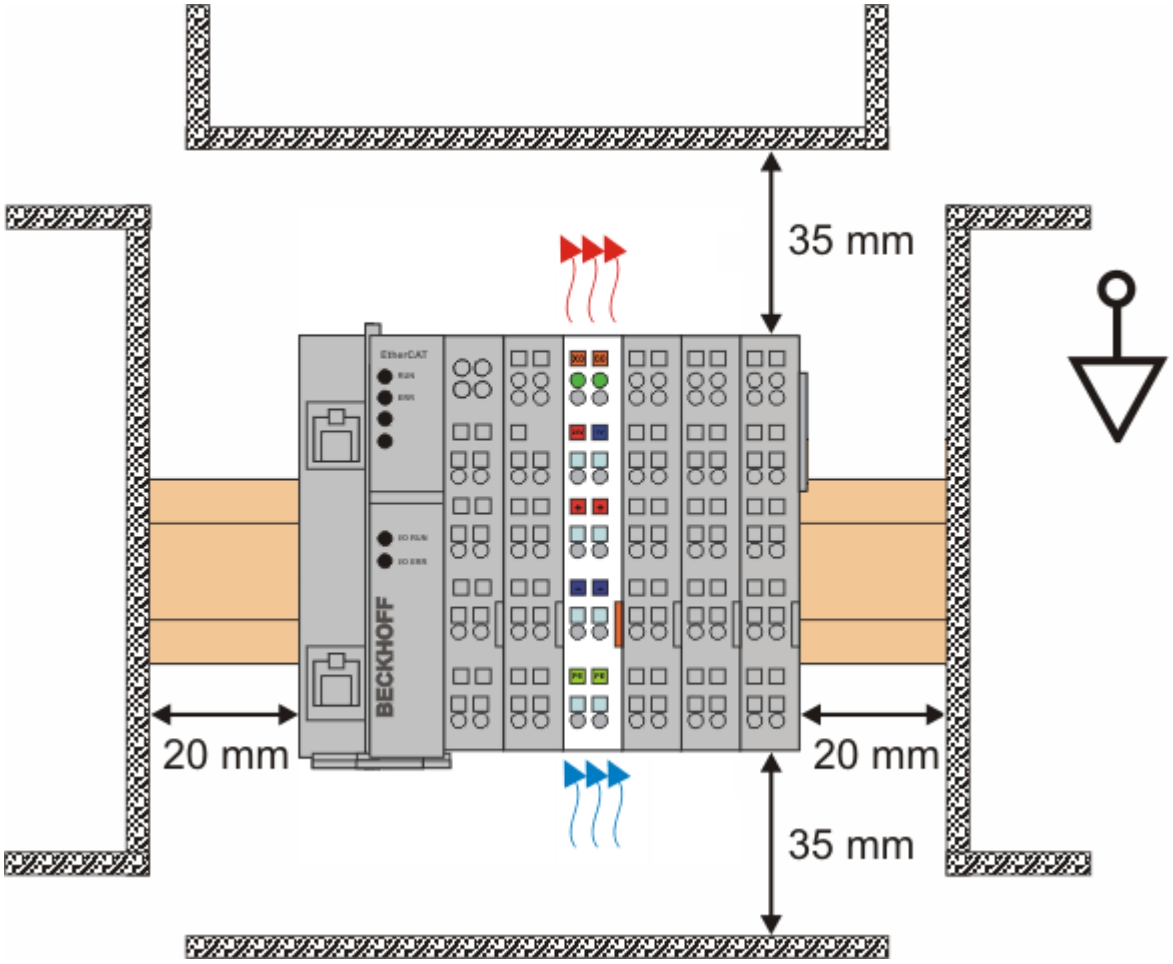


Fig. 28: Recommended distances for standard installation position

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

Other installation positions

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

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

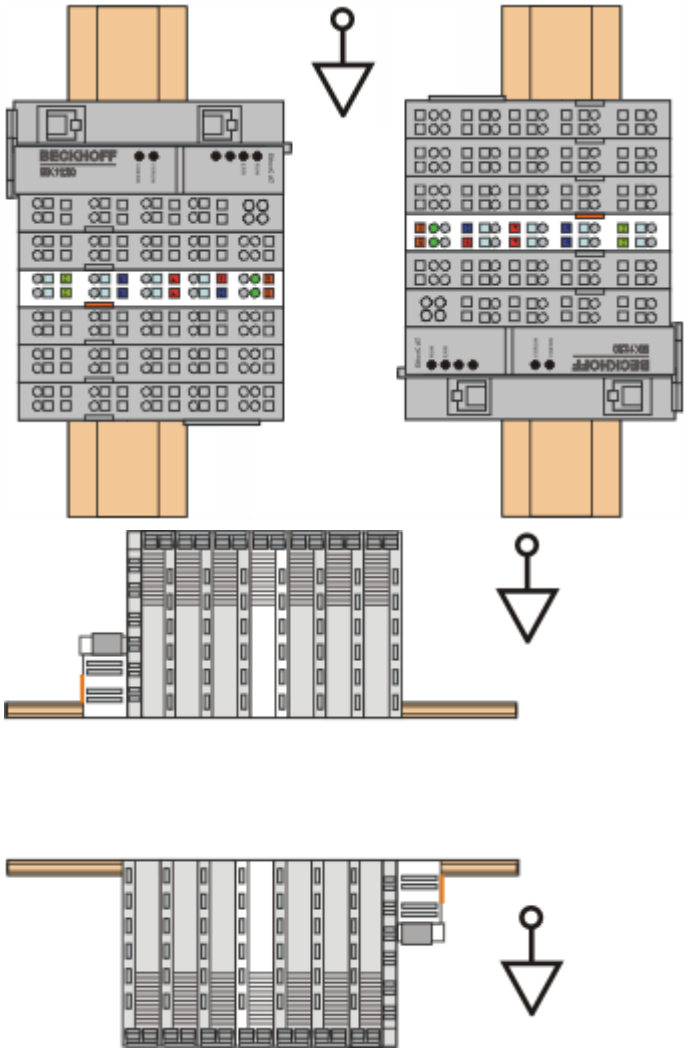


Fig. 29: Other installation positions

5.6 Shielding concept

Together with the shield busbar, the prefabricated cables from Beckhoff Automation offer optimum protection against electromagnetic interference.

Connection of the motor cable to the shield busbar

Fasten the shield busbar supports 1 to the DIN rail 2. The DIN rail 2 must be in contact with the metallic rear wall of the control cabinet over a wide area. Install the shield busbar 3 as shown below.

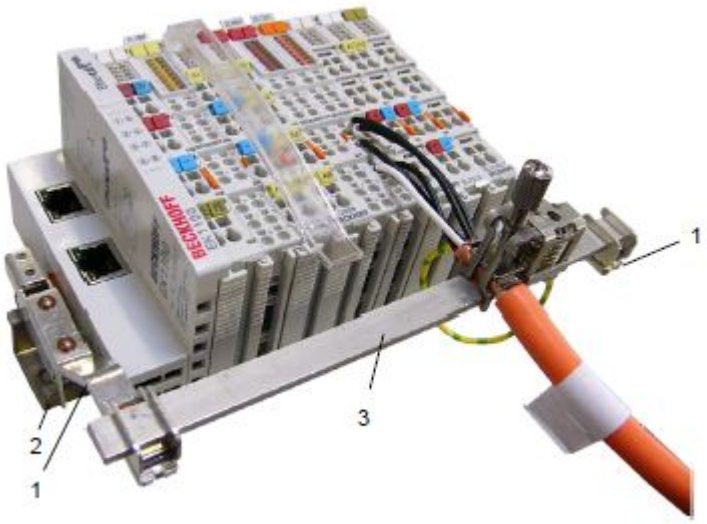


Fig. 30: Shield busbar

Connect the cores 4 of the motor cable 5, then attach the copper-sheathed end 6 of the motor cable 5 with the shield clamp 7 to the shield busbar 3. Tighten the screw 8 to the stop.
Fasten the PE clamp 9 to the shield busbar 3. Clamp the PE core 10 of the motor cable 5 under the PE clamp 9.

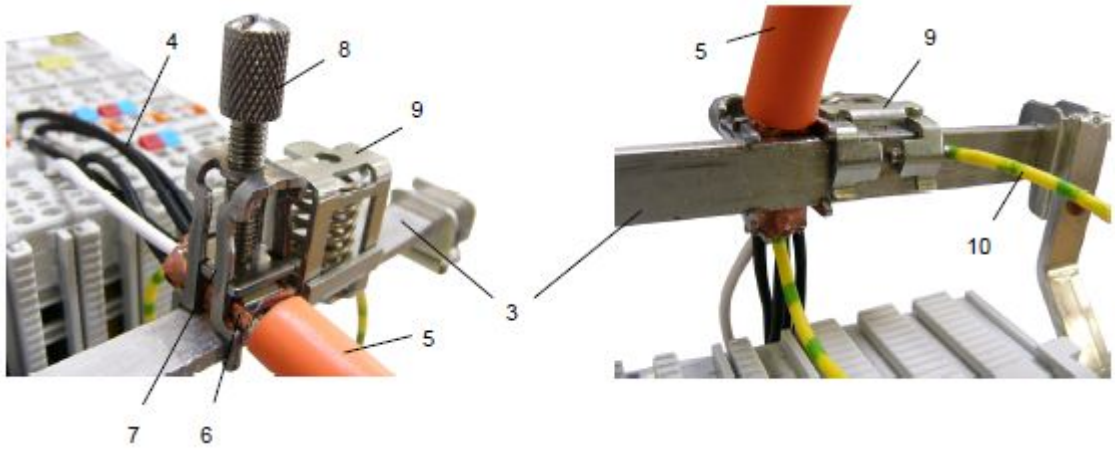


Fig. 31: Shield connection

Connection of the feedback cable

The shield of the feedback cable is connected via the metallic plug fastener when screwing the feedback plug onto the AM3100.
On the terminal side the shield can also be connected.. Connect the cores of the feedback cable and attach the copper-sheathed end of the feedback cable to the shield busbar 3 with the shield clamp 7. The motor cable and the feedback cable can be connected to the shield clamp 7 with the screw 8.

5.7 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. 32: 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

Note	**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.

5.8 EL72x1-0010 - LEDs and connection

EL7201-0010

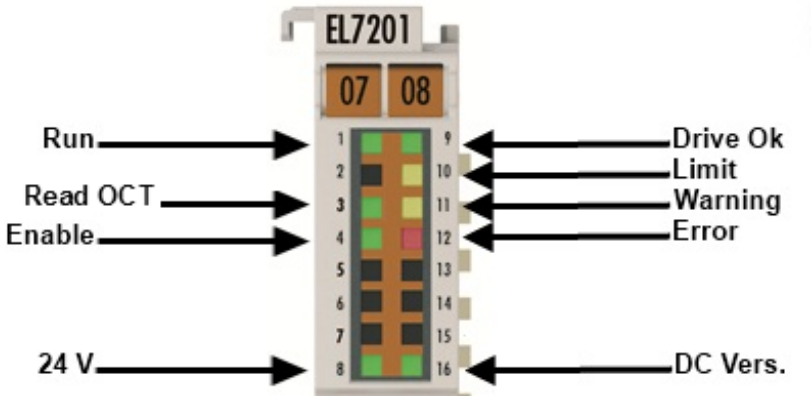


Fig. 33: EL7201-0010 - LEDs

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine [▶ 77]: **INIT** = initialization of the terminal
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates [▶ 192] of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager [▶ 77] 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 [▶ 171] / DS402 [▶ 151]) (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 [▶ 171] / DS402 [▶ 151]) (warning) The “Warning” threshold value is exceeded. I²T model Temperature (80°C) exceeded Voltage
Enable	green	on	The LED is linked with the bits 1 and 2 of status word (MDP742 [▶ 171] / DS402 [▶ 151]) (if "Switched on" or "Operation enabled") Driver stage enabled
Error	red	on	The LED is linked with bit 3 of the status word (MDP742 [▶ 171] / DS402 [▶ 151]) (fault) The “Error” threshold value is exceeded. Overcurrent 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.

Connection

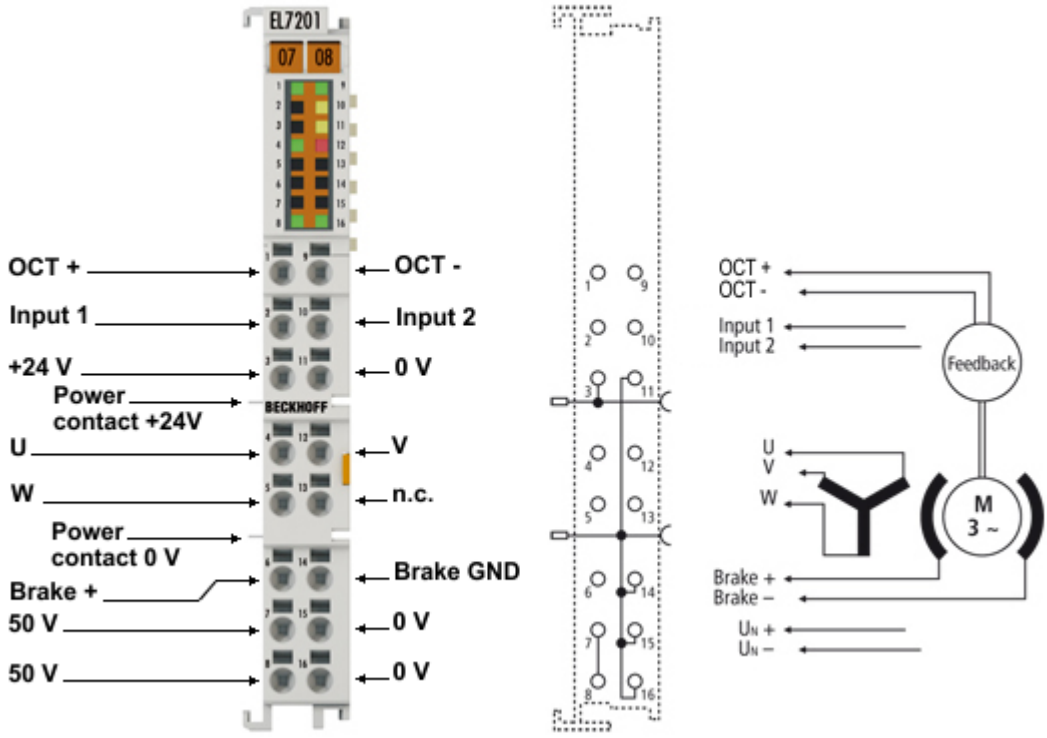


Fig. 34: EL7201-0010 - Connection

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	50 V	DC link supply + (8...50 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	n.c.	not connected
14	Brake GND	Motor brake 0 V
15	0 V	DC link 0 V supply
16		

EL7211-0010

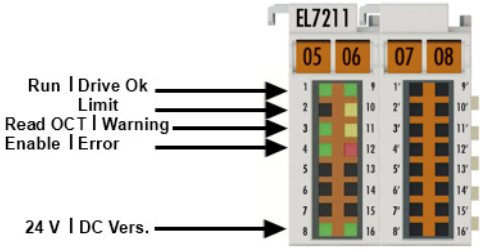


Fig. 35: EL7211-0010 - LEDs

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine [▶ 77]: **INIT** = initialization of the terminal
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates [▶ 192] of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager [▶ 77] 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 [▶ 171] / DS402 [▶ 151]) (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 [▶ 171] / DS402 [▶ 151]) (warning) The “Warning” threshold value is exceeded. I²T model Temperature (80°C) exceeded Voltage
Enable	green	on	The LED is linked with the bits 1 and 2 of status word (MDP742 [▶ 171] / DS402 [▶ 151]) (if "Switched on" or "Operation enabled") Driver stage enabled
Error	red	on	The LED is linked with bit 3 of the status word (MDP742 [▶ 171] / DS402 [▶ 151]) (fault) The “Error” threshold value is exceeded. Overcurrent 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.

Connection

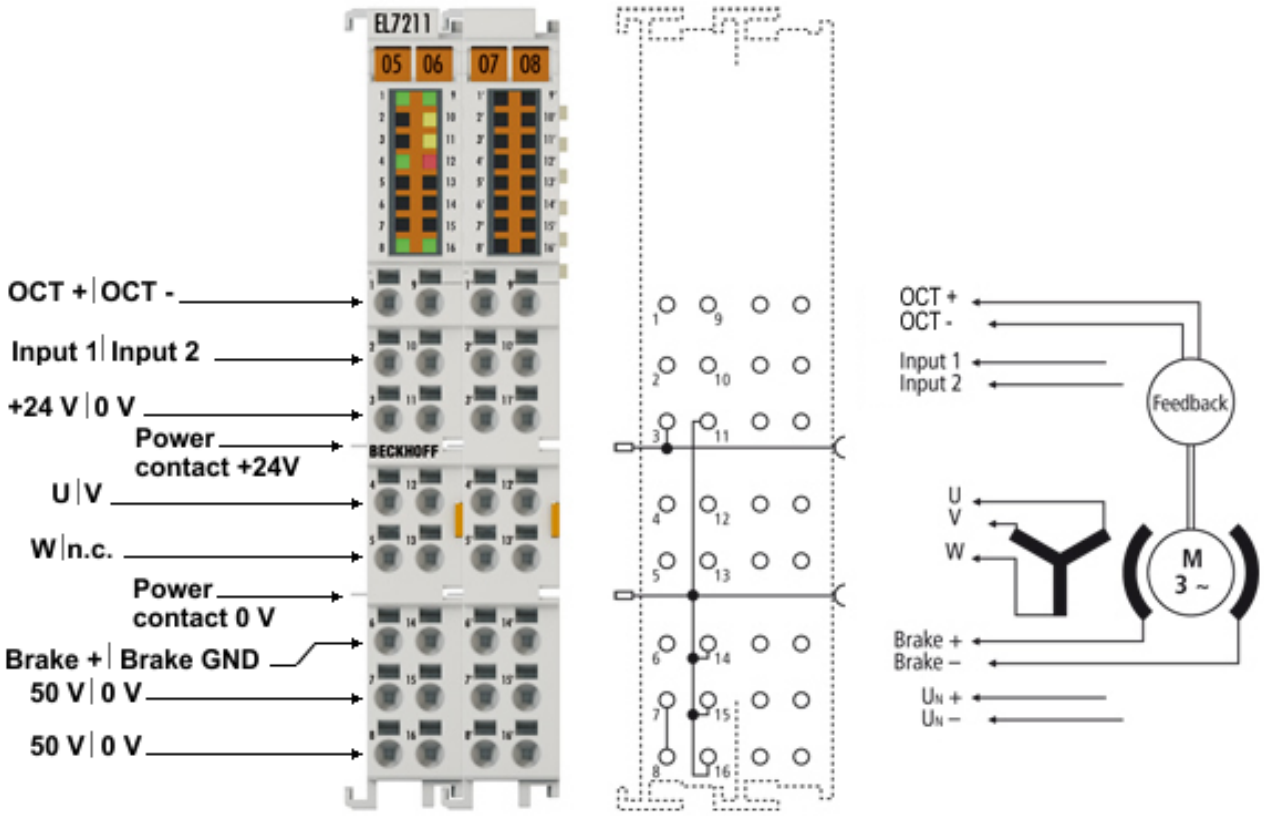


Fig. 36: EL7211-0010 - Connection

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	50 V	DC link supply + (8...50 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	n.c.	not connected
14	Brake GND	Motor brake 0 V
15	0 V	DC link 0 V supply
16		
1' - 16'		n.c.

6 Commissioning

6.1 TwinCAT 2.1x

6.1.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways. One option is described here.

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

Fig. 37: System Manager option

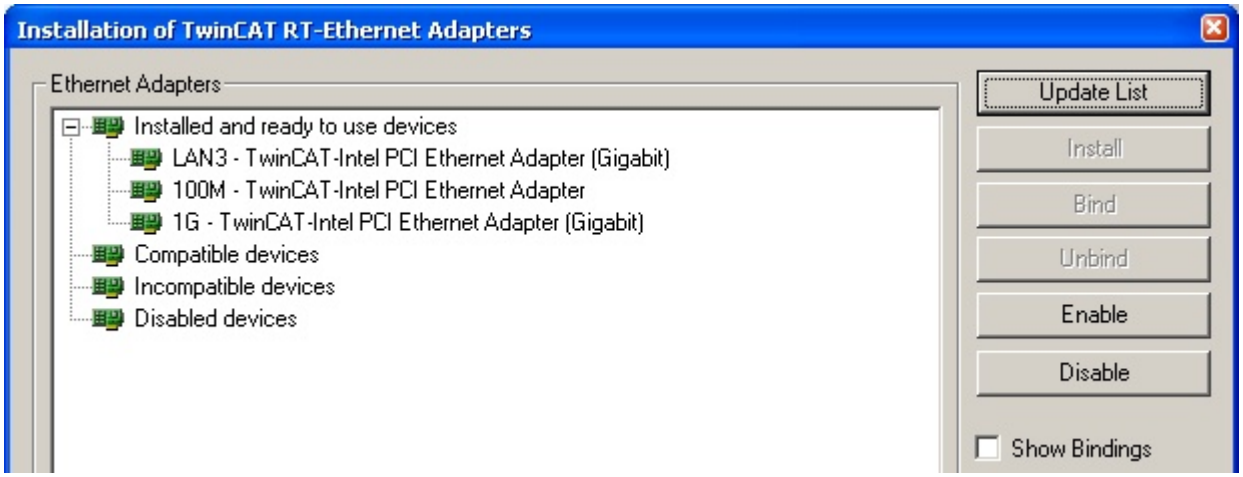


Fig. 38: Overview of network interfaces

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, the compatible Ethernet ports can be viewed in the System Manager via EtherCAT properties.

Fig. 39: EtherCAT device properties

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

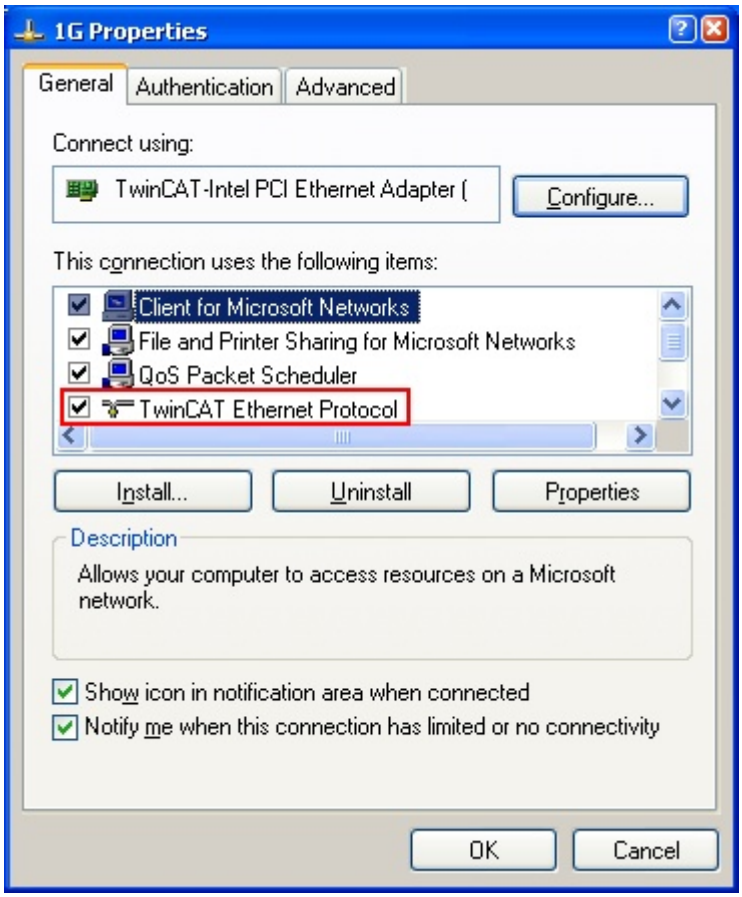


Fig. 40: Windows properties of the network interface

Other possible settings are to be avoided:

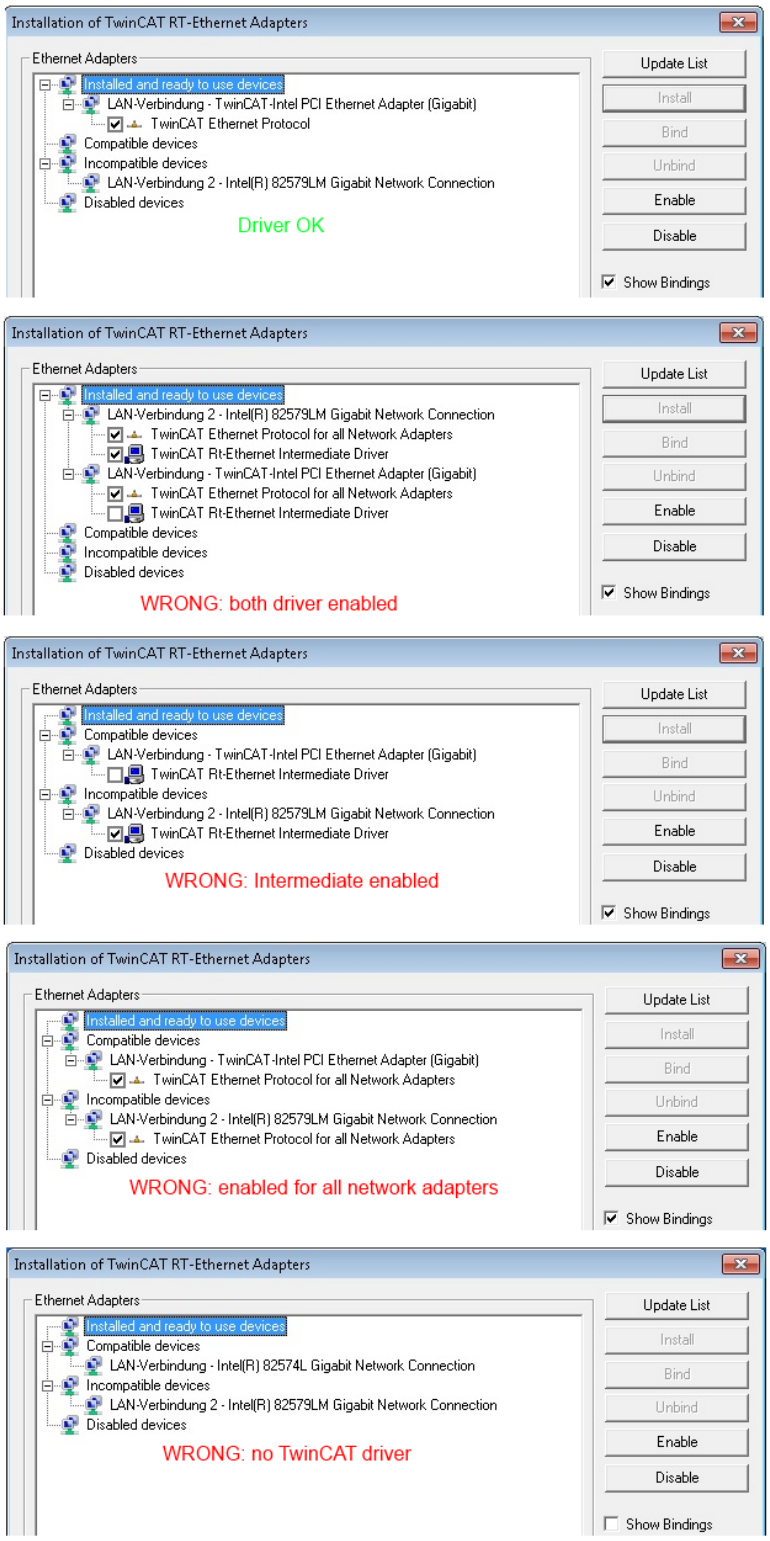


Fig. 41: Incorrect driver settings for the Ethernet port

IP address of the port used

Note	**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.

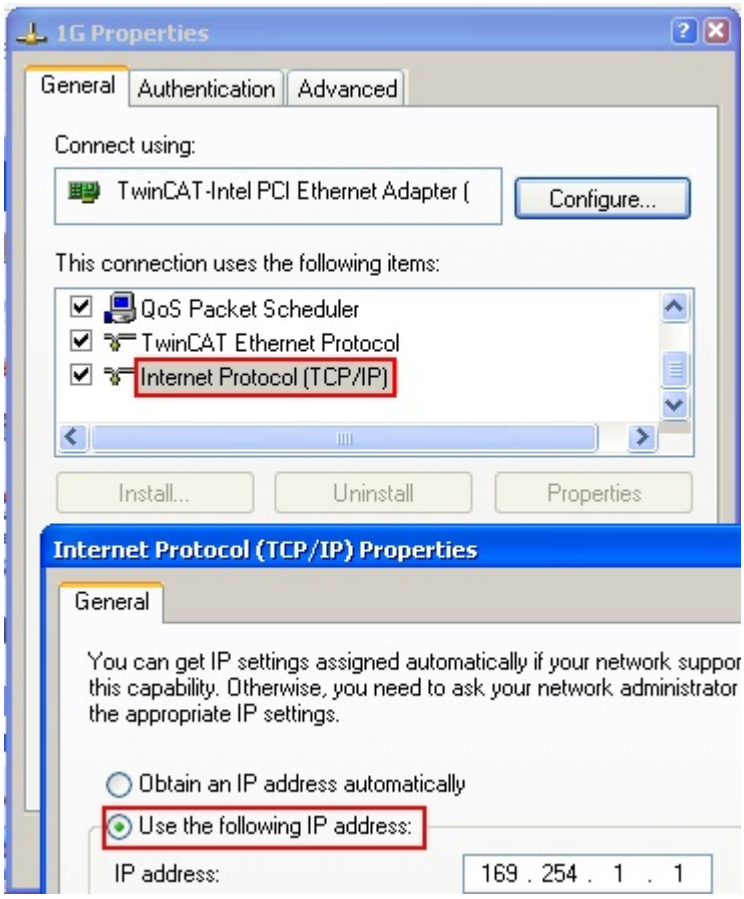


Fig. 42: TCP/IP setting for the Ethernet port

6.1.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

The ESI files for Beckhoff EtherCAT devices are available on the Beckhoff website.

The ESI files should be stored in the TwinCAT installation directory (default TwinCAT2: C:\TwinCAT\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 (Option -> “Update EtherCAT Device Descriptions”)

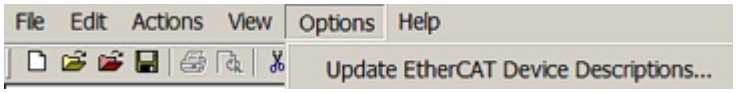


Fig. 43: For TwinCAT 2.11 and higher, the System Manager can search for current Beckhoff ESI files automatically, if an online connection is available

Note	**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 4 properties, which determine the full device identifier. The EL2521-0025-1018 ID consists of

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

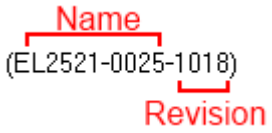


Fig. 44: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 8].

Online description

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

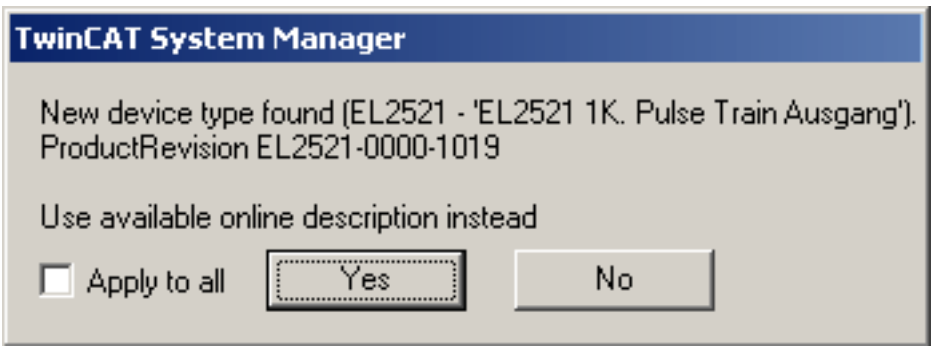


Fig. 45: OnlineDescription information window

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

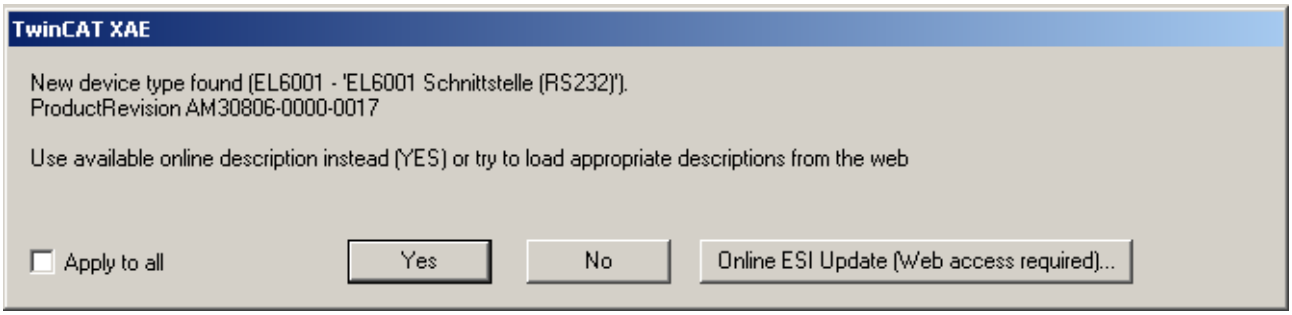


Fig. 46: Information window OnlineDescription, TwinCAT 3.x

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.

Attention	**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 'Configuration creation – manual'

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. The route via the ESI files is therefore recommended.

The System Manager creates a new file "OnlineDescription0000...xml" its ESI directory, which contains all ESI descriptions that were read online.

```
OnlineDescriptionCache00000002.xml
```

Fig. 47: File OnlineDescription.xml created by the System Manager

If slaves are added manually to the configuration at a later stage, slaves created in the manner described above are indicated by an arrow, see Fig. *"Arrow indicates ESI recorded from OnlineDescription"*, EL2521.

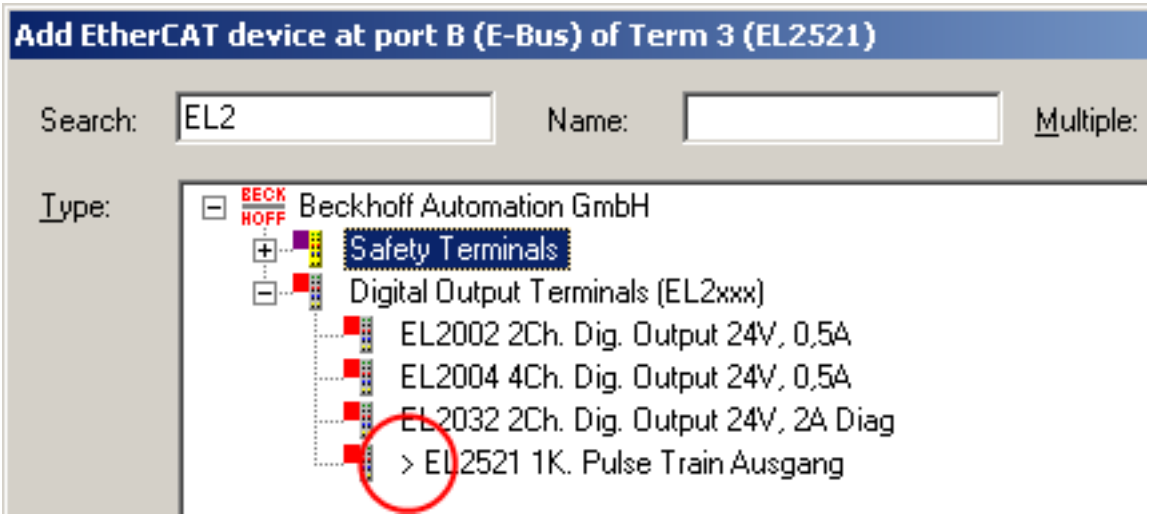


Fig. 48: Arrow indicates ESI recorded from OnlineDescription

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

Note	**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\EtherCAT-Cache.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. 49: Information window for faulty ESI file

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

6.1.3 Offline configuration creation (master: TwinCAT 2.x)

Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals). 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.

Note	**Installation of the latest ESI-XML 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. The ESIs for Beckhoff EtherCAT devices are provided on the Beckhoff website. The ESI files should be saved in the TwinCAT installation directory (default: C:\TwinCAT\IO\EtherCAT). The files are read (once) when a new System Manager window is 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 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet (Option -> “Update EtherCAT Device Descriptions”)

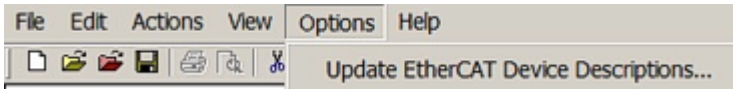


Fig. 50: Updating of the ESI directory

The following conditions must be met before a configuration can be set up:

- the EtherCAT device must be created/defined in the System Manager [▶ 52]
- the EtherCAT slaves must be defined [▶ 54]

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

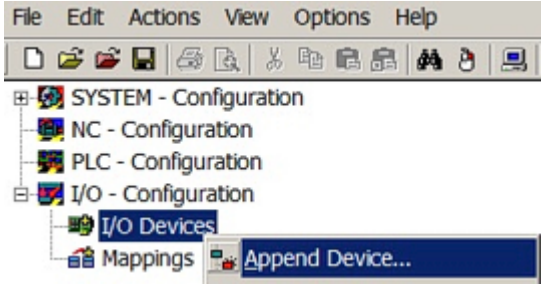


Fig. 51: Append EtherCAT device

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”.

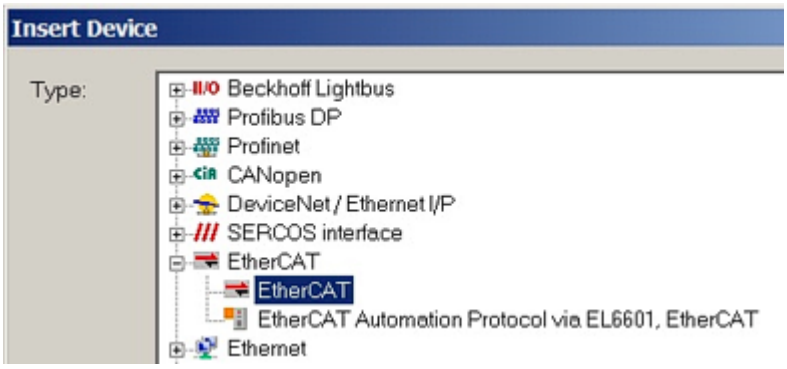


Fig. 52: Selecting the EtherCAT connection (TwinCAT 2.11)

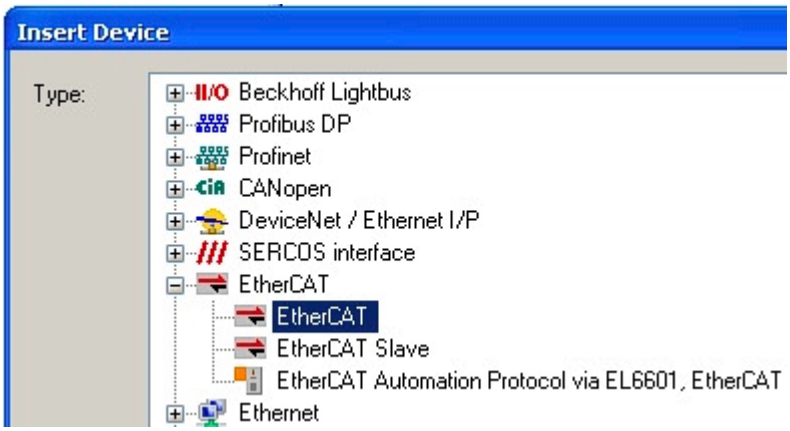


Fig. 53: Selecting the EtherCAT connection (TwinCAT 2.11 R2)

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

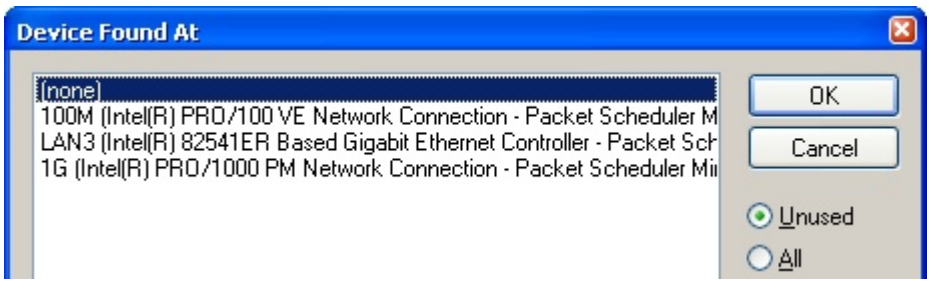


Fig. 54: Selecting the Ethernet port

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 properties dialog"*).

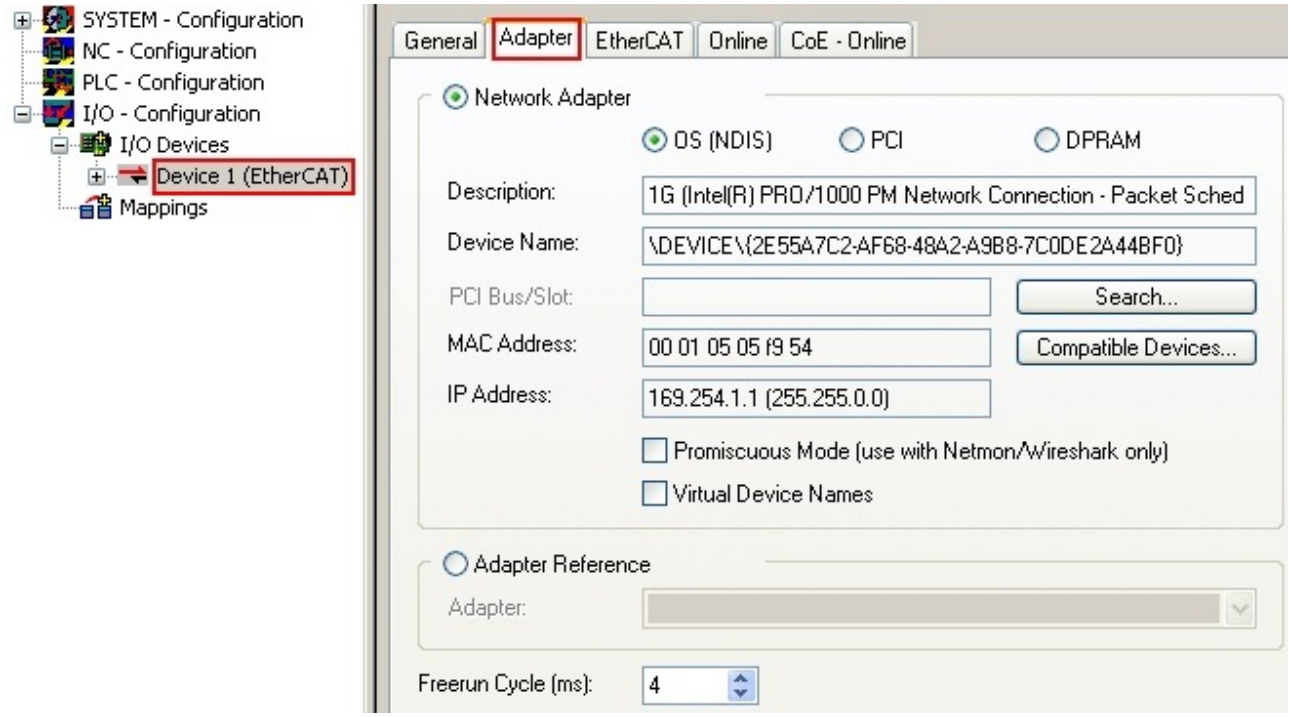


Fig. 55: EtherCAT properties dialog

Note	**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.

Defining EtherCAT slaves

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

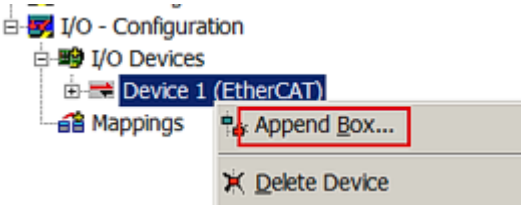


Fig. 56: Appending EtherCAT devices

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

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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: EK couplers, EP boxes, devices with RJ45/M8/M12 connector
- “E-Bus”: LVDS “terminal bus”: EL/ES terminals, various modular modules

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

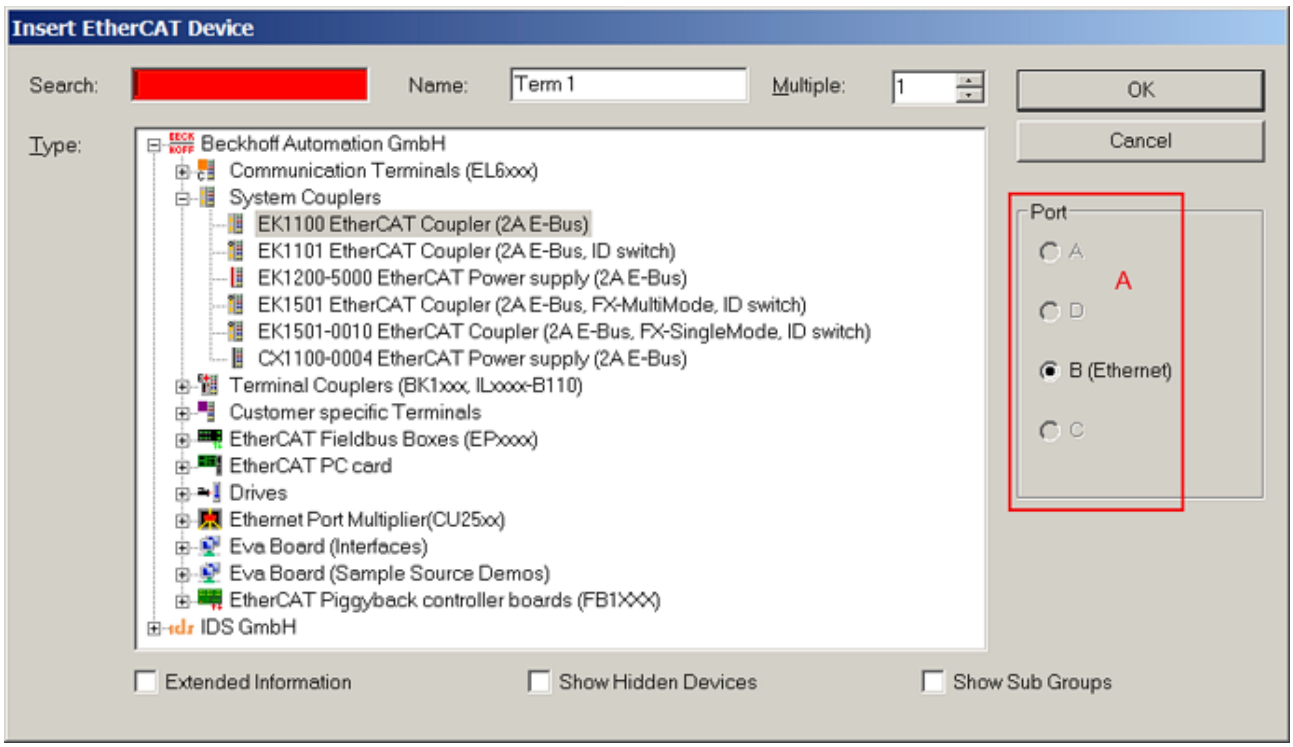


Fig. 57: Selection dialog for new EtherCAT device

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”.

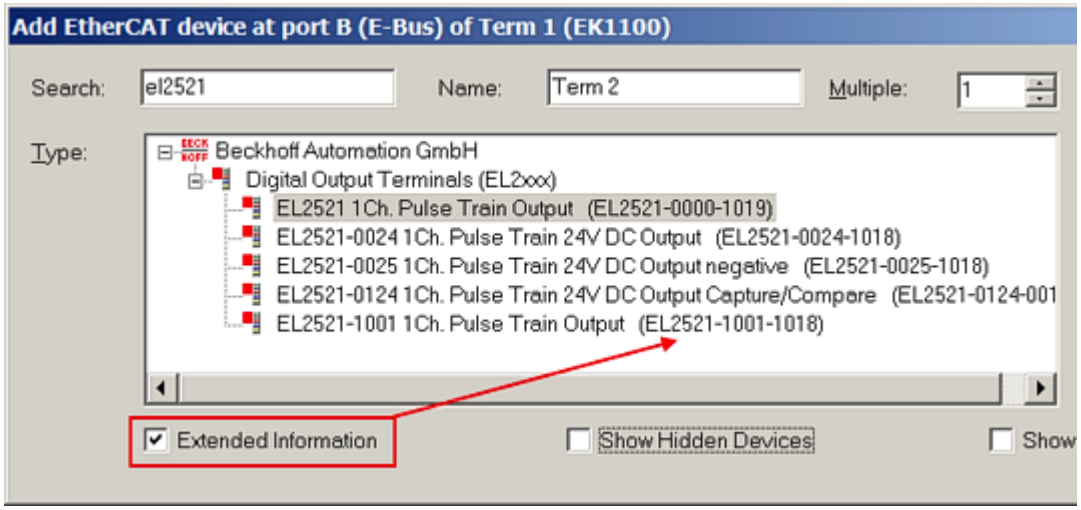


Fig. 58: Display of device revision

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”*.

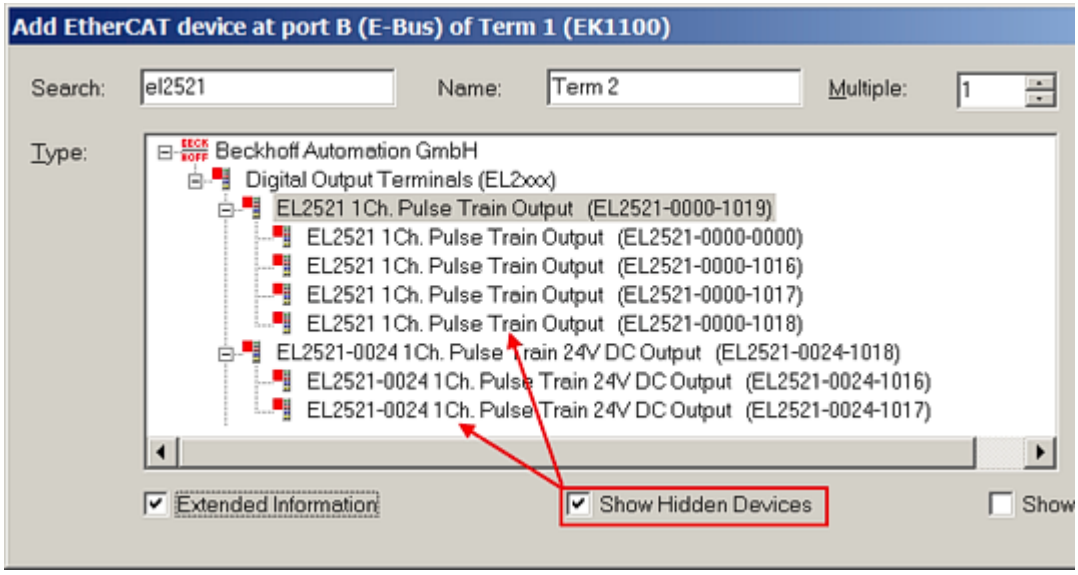


Fig. 59: Display of previous revisions

Note	**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: **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.

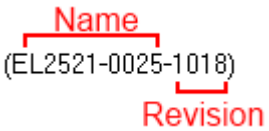


Fig. 60: Name/revision of the terminal

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

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

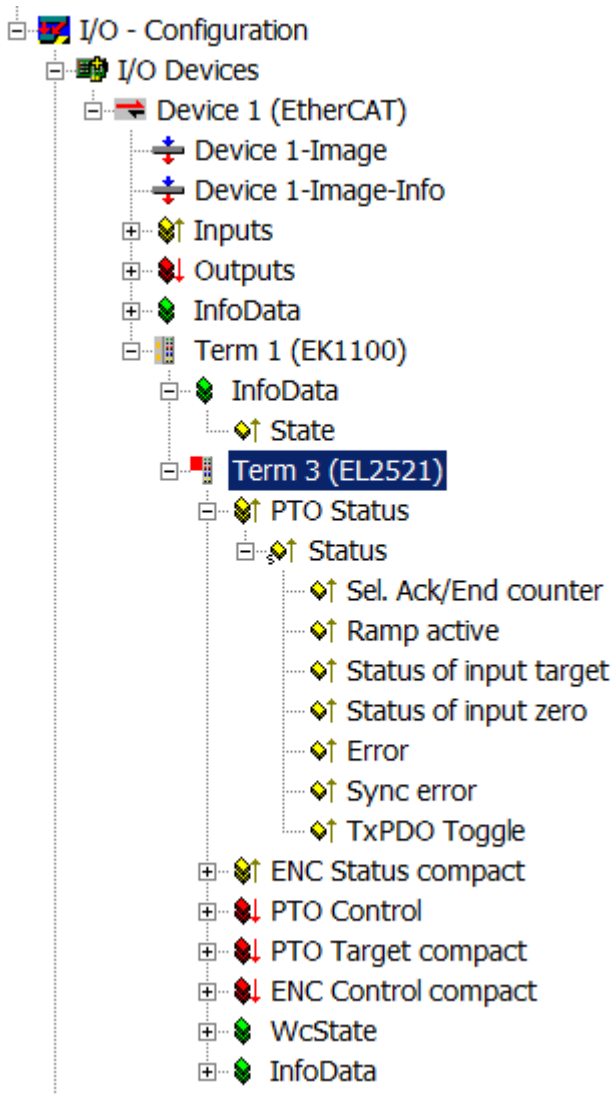


Fig. 61: EtherCAT terminal in the TwinCAT tree

6.1.4 Online configuration creation ‘scanning’ (master: TwinCAT 2.x)

Distinction between Online and Offline

Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals). 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.

Note	**Installation of the latest ESI-XML 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. The ESIs for Beckhoff EtherCAT devices are provided on the Beckhoff website. The ESI files should be saved in the TwinCAT installation directory (default: C:\TwinCAT\IO\EtherCAT). The files are read (once) when a new System Manager window is 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 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet (Option -> “Update EtherCAT Device Descriptions”)

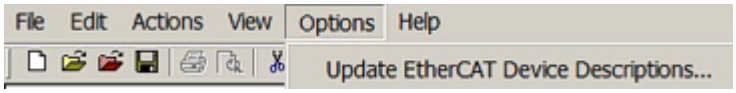


Fig. 62: Updating ESI directory

The following conditions must be met before a configuration can be set up:

- 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 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 [▶ 58] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 60]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 63]

The scan with existing configuration [▶ 64] can also be carried out for comparison.

Detecting/scanning of the EtherCAT device

The online device search can be used if the TwinCAT system is in CONFIG mode (blue TwinCAT icon or blue indication in the System Manager).

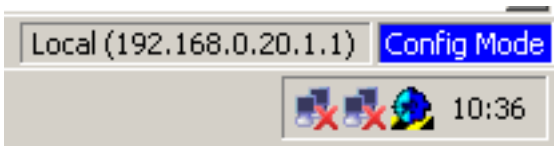


Fig. 63: TwinCAT CONFIG mode display

Note	**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 icon next to the Windows clock always shows the TwinCAT mode of the local IPC. The System Manager window shows the TwinCAT state of the target system.

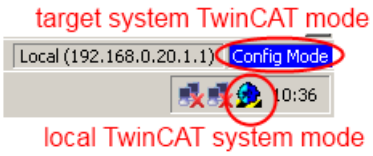


Fig. 64: Differentiation local/target system

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

Fig. 65: Scan Devices

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

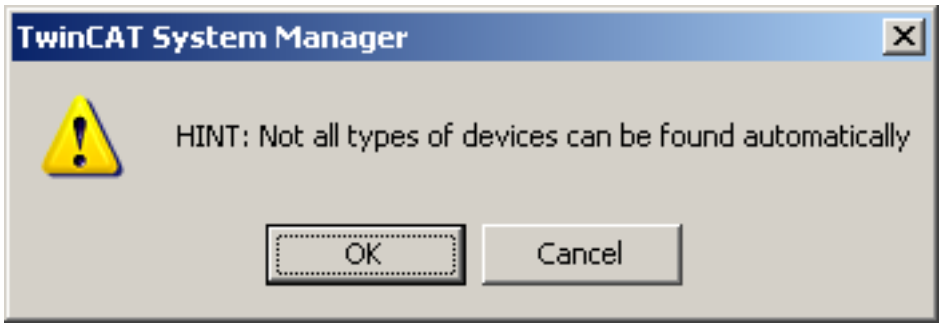


Fig. 66: Note for automatic device scan

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

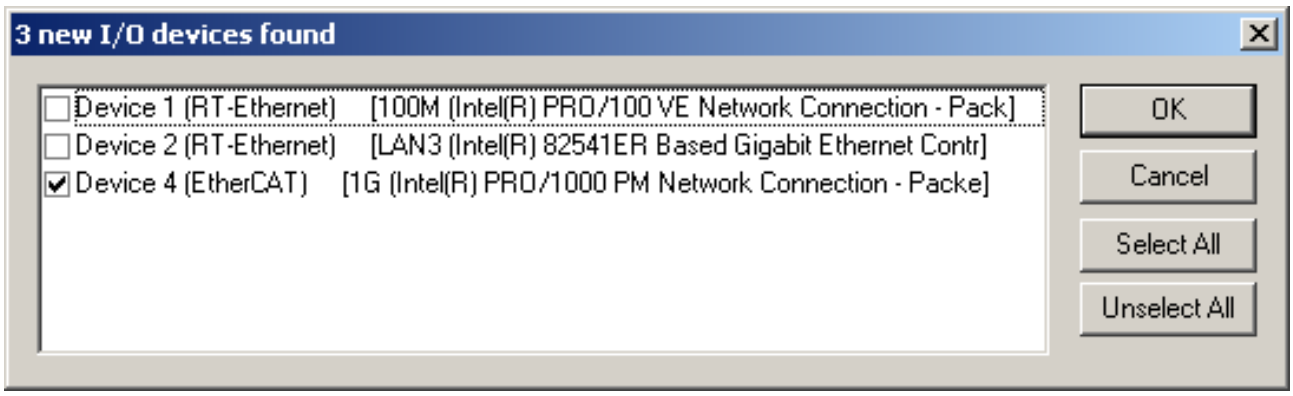


Fig. 67: Detected Ethernet devices

After confirmation with “OK” a device scan is suggested for all selected devices, see Fig. 5.

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

Detecting/Scanning the EtherCAT devices

Note	**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.

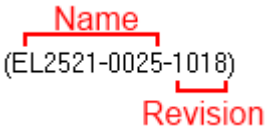


Fig. 68: Example default state

Attention	**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 [▶ 64] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

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

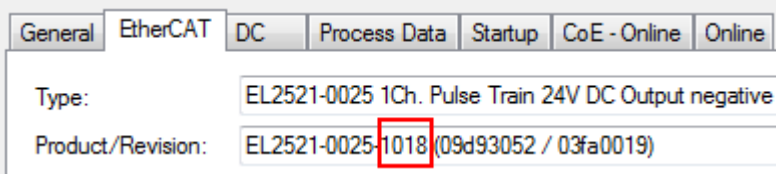


Fig. 69: Installing EthetCAT terminal with revision -1018

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 TwinCAT3 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 [▶ 64] against the initial configuration 'B.tsm' in order to check the built machine.

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

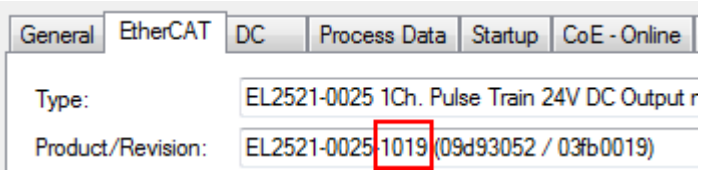


Fig. 70: Detection of EtherCAT terminal with revision -1019

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration 'B2.tsm' created in this way.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. 71: Scan query after automatic creation of an EtherCAT device

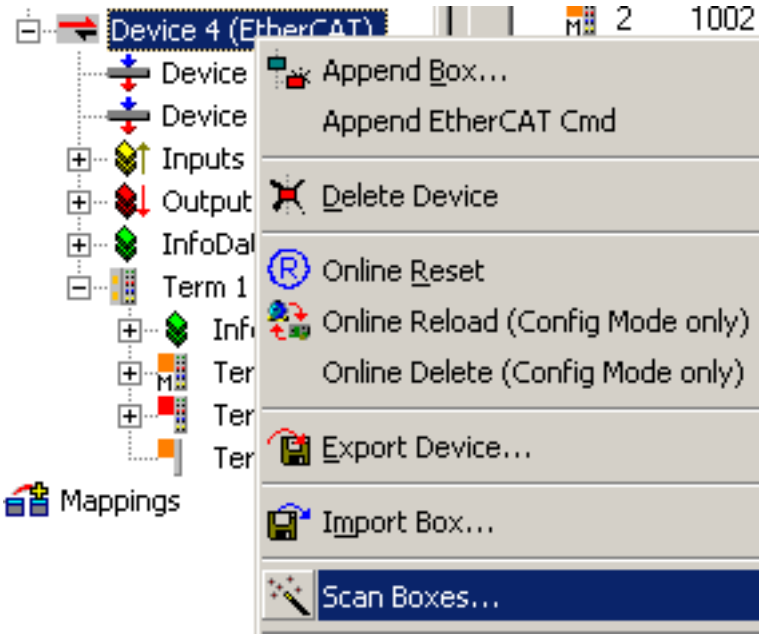


Fig. 72: Manual triggering of a device scan on a specified EtherCAT device

In the System Manager the scan process can be monitored via the progress bar at the bottom of the screen.

Fig. 73: Scan progress

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

Fig. 74: Config/FreeRun query

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

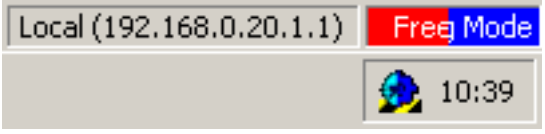


Fig. 75: Config/FreeRun indicator

Fig. 76: TwinCAT kann auch durch einen Button in diesen Zustand versetzt werden

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

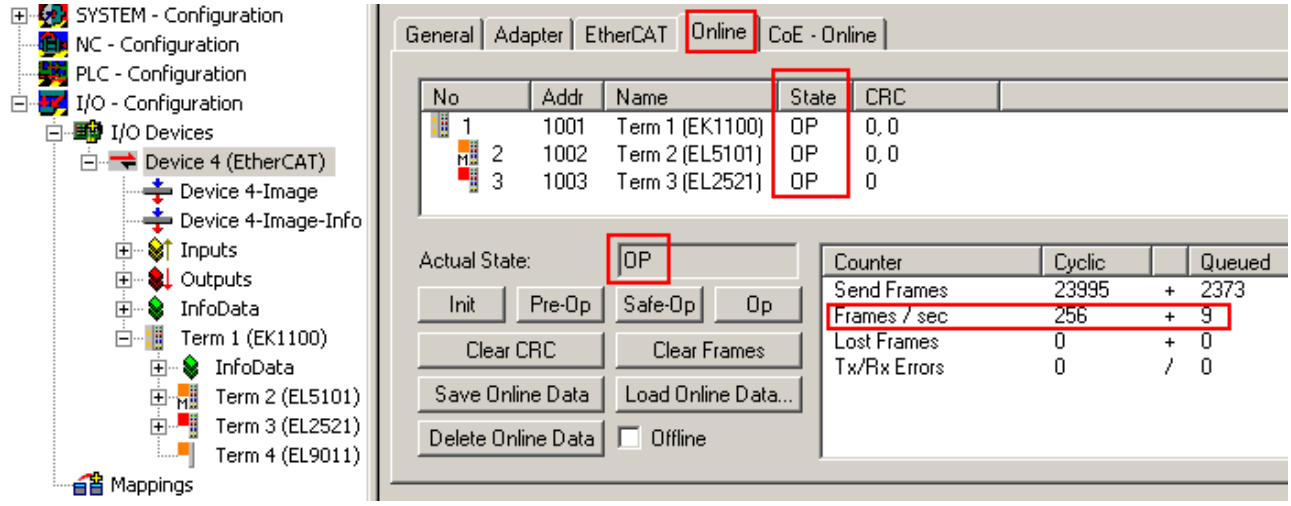


Fig. 77: Online display example

Please note:

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

The configuration is now complete. It can be modified as described under manual procedure [▶ 52].

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter "Notes regarding ESI device description".
- **Device are not detected properly**
  Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description
  The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

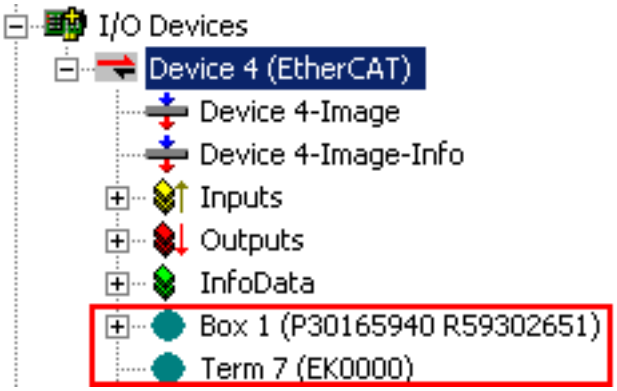


Fig. 78: Faulty identification

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

Scan over existing Configuration

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

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

Fig. 79: Identical configuration

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

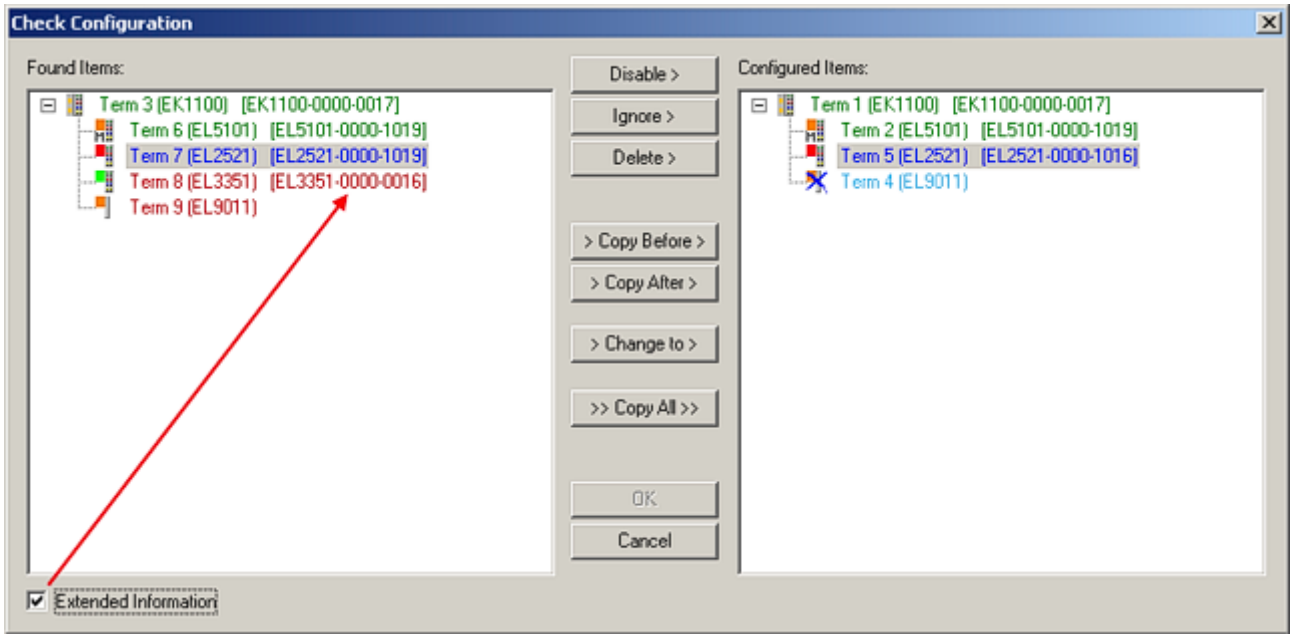


Fig. 80: Correction dialog

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

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

Note

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:

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

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

Example:

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

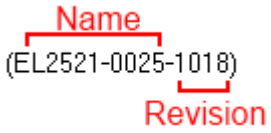


Fig. 81: Name/revision terminal

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.

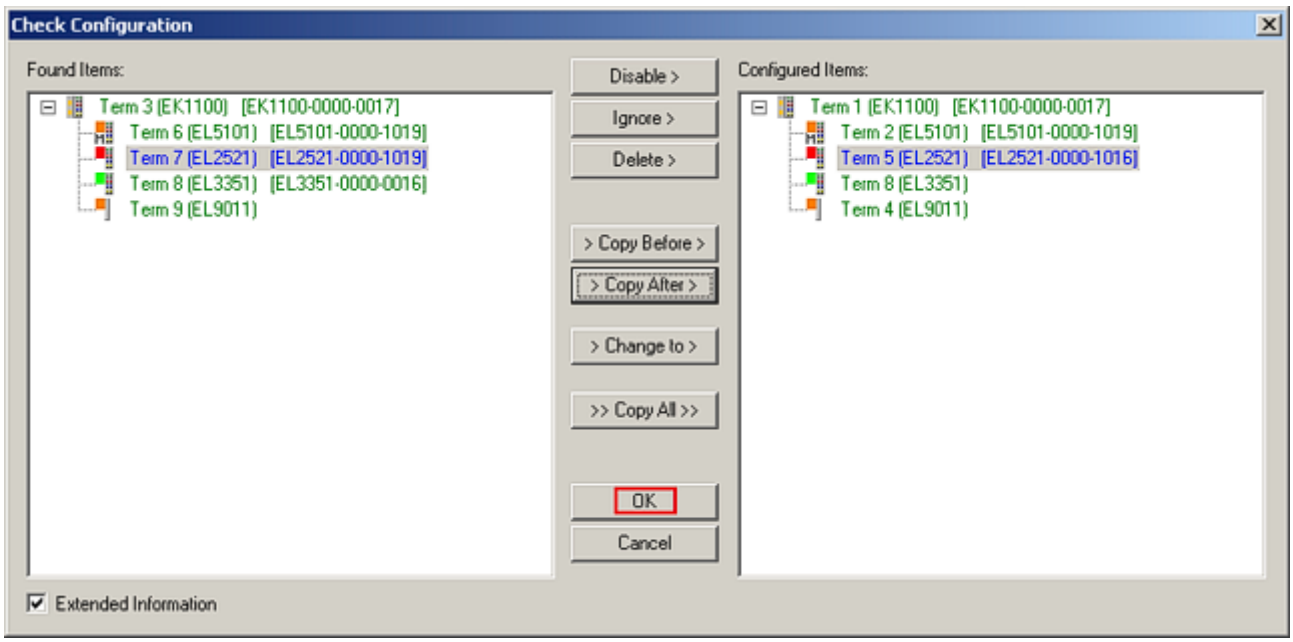


Fig. 82: Correction dialog with modifications

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

Change to compatible device

The TwinCAT System Manager offers a function for the exchange of a device whilst retaining the links in the task: *Change to compatible device.*

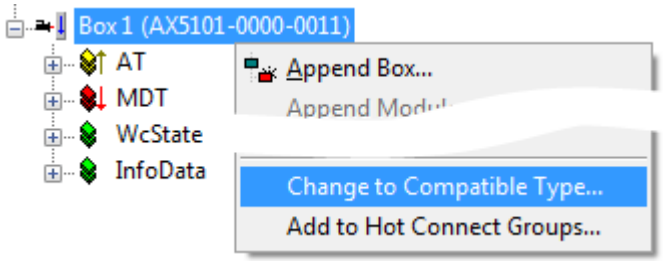


Fig. 83: TwinCAT 2 Dialog ChangeToCompatibleDevice

This function is preferably to be used on AX5000 devices. If called, the System Manager suggests the devices that it finds in the associated sub-folder; in the case of the AX5000, for example, in *\TwiNCAT\IO \EtherCAT\Beckhoff AX5xxx*.

Change to Alternative Type

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

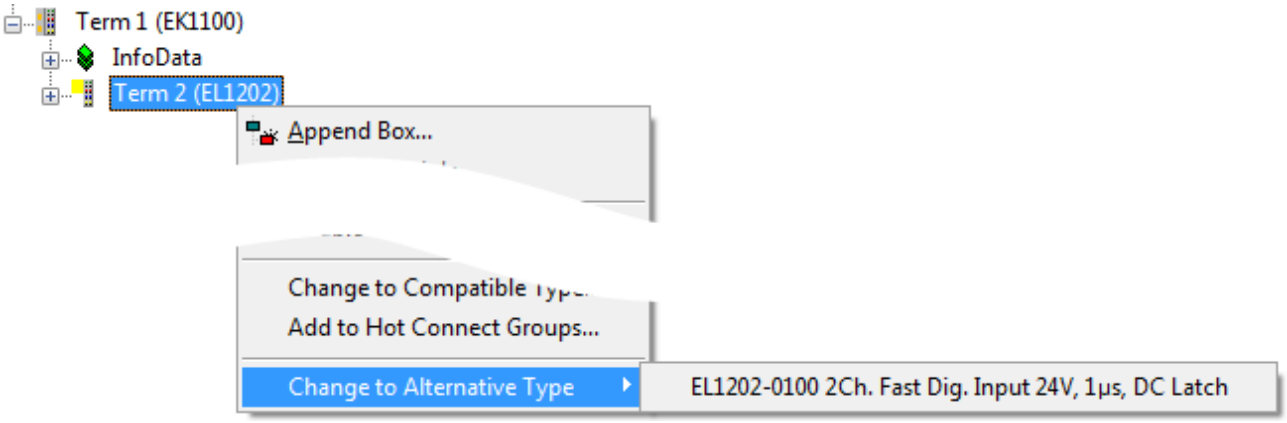


Fig. 84: TwinCAT 2 Dialog ChangeToCompatibleDevice

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).

6.1.5 EtherCAT slave process data settings

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

For Beckhoff EtherCAT EL/ES slaves the following applies in general:

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

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

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

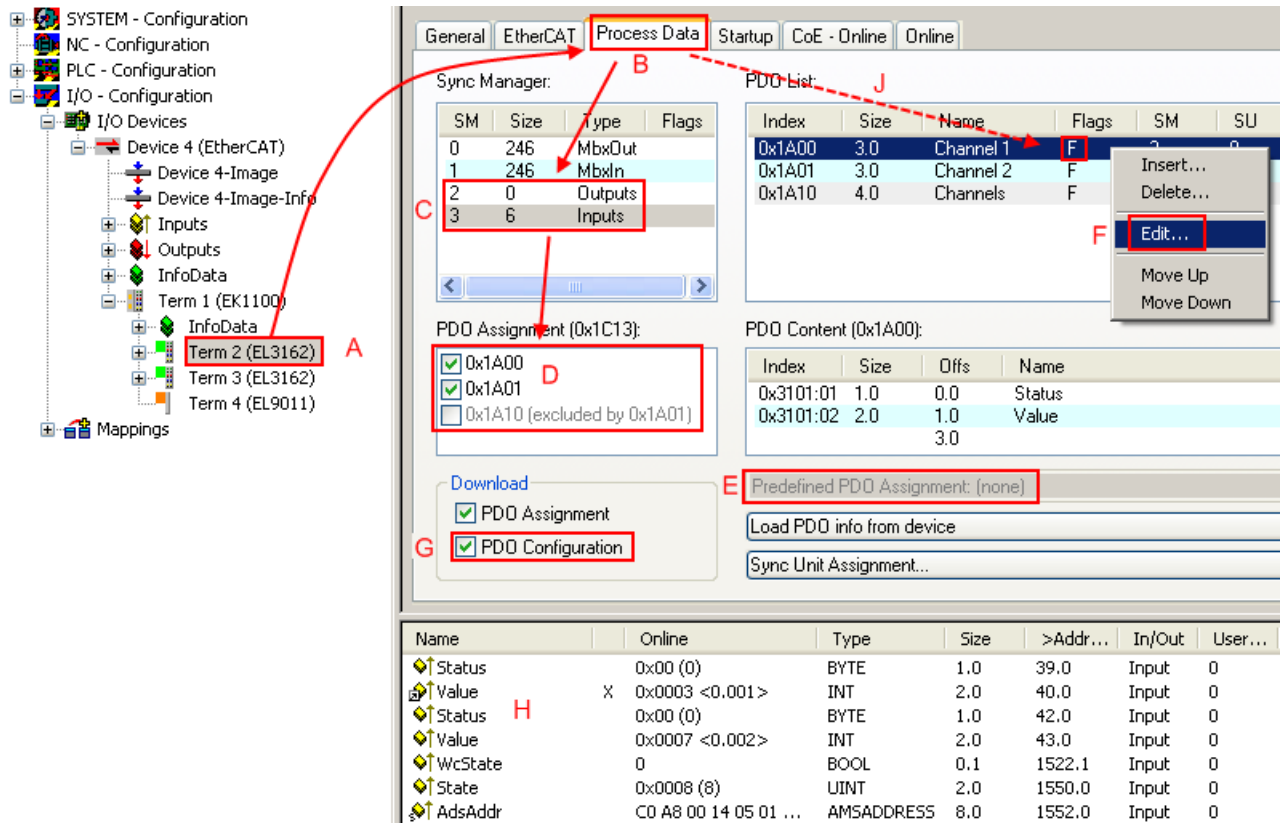


Fig. 85: Configuring the process data

Note

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.

6.1.6 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

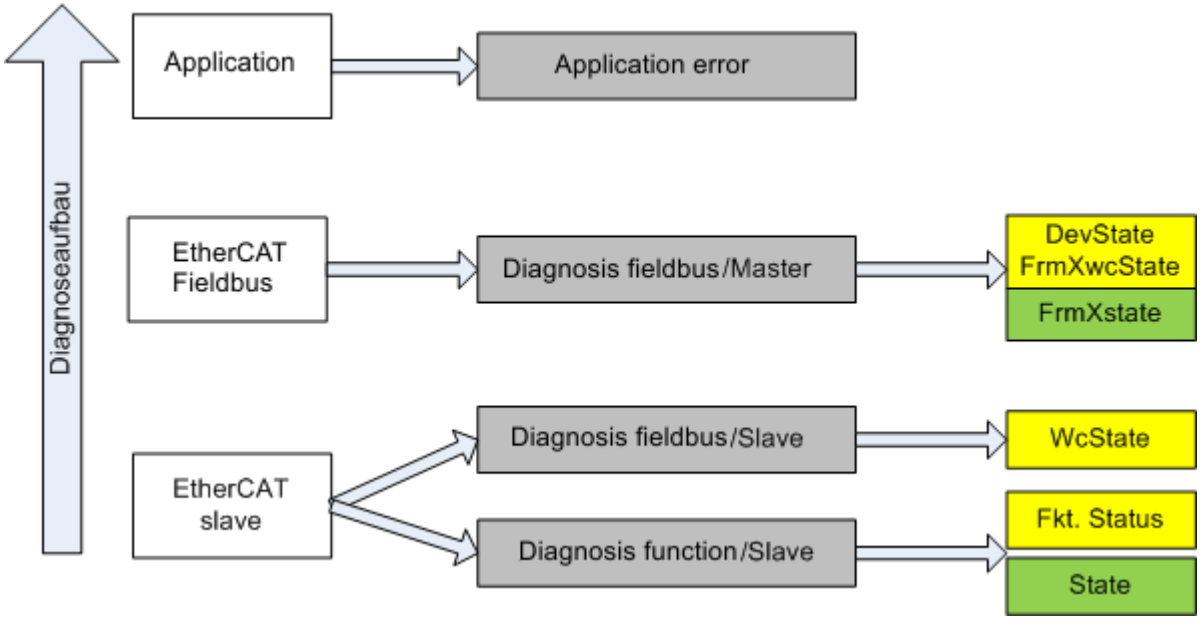


Fig. 86: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

  as well as

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

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

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

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

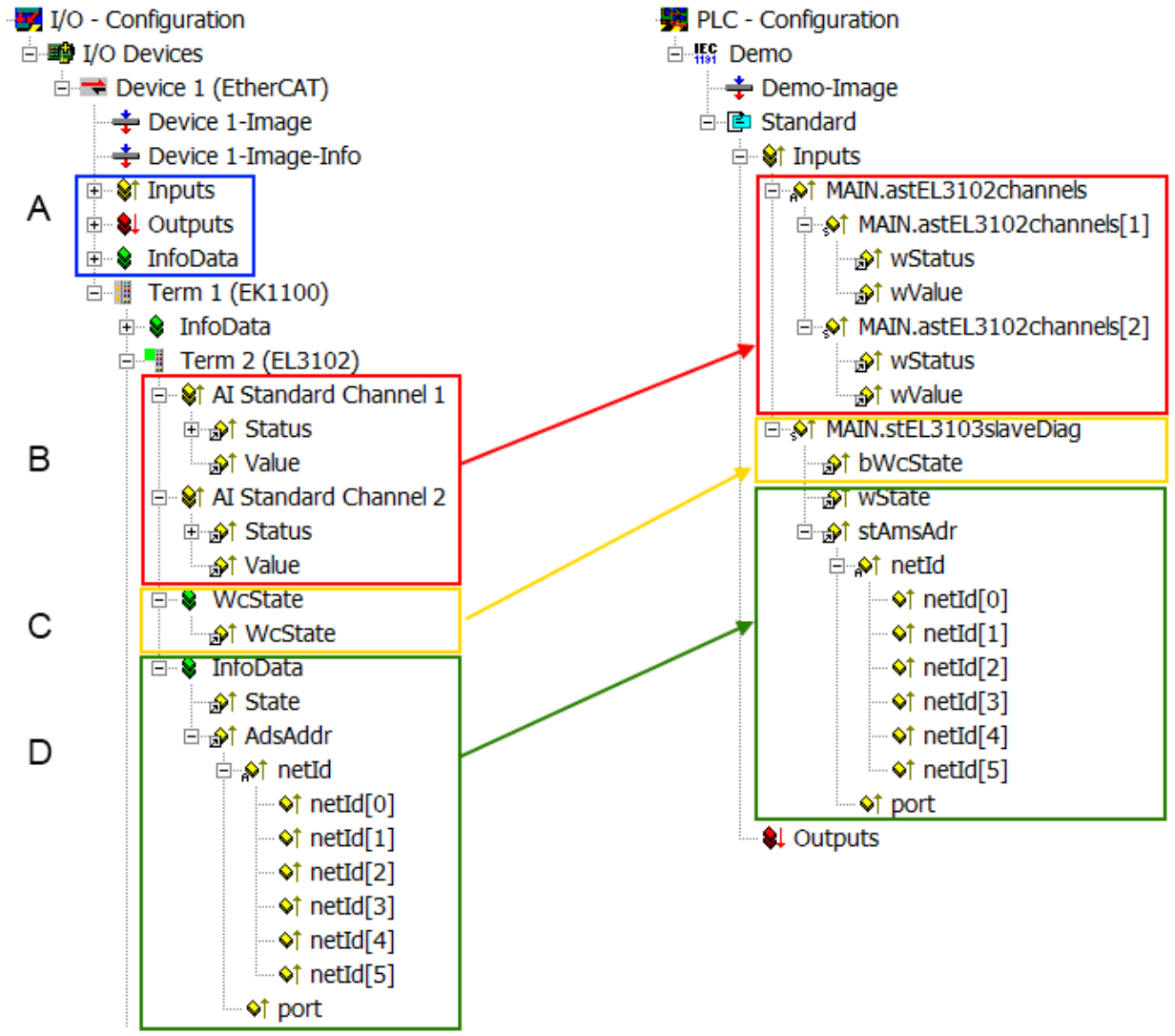


Fig. 87: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • CoE in the Master for communication with/through the Slaves • Functions from *TcEtherCAT.lib* • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status • the bit significations may be found in the device documentation • other devices may supply more information, or none that is typical of a slave	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager 1. at the EtherCAT Slave, and, with identical contents 2. as a collective variable at the EtherCAT Master (see Point A) for linking.	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.

Code	Function	Implementation	Application/evaluation
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it • is only rarely/never changed, except when the system starts up • is itself determined acyclically (e.g. EtherCAT Status)	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. *AdsAddr* The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the *port* (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

Attention

Diagnostic information

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

CoE Parameter Directory

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

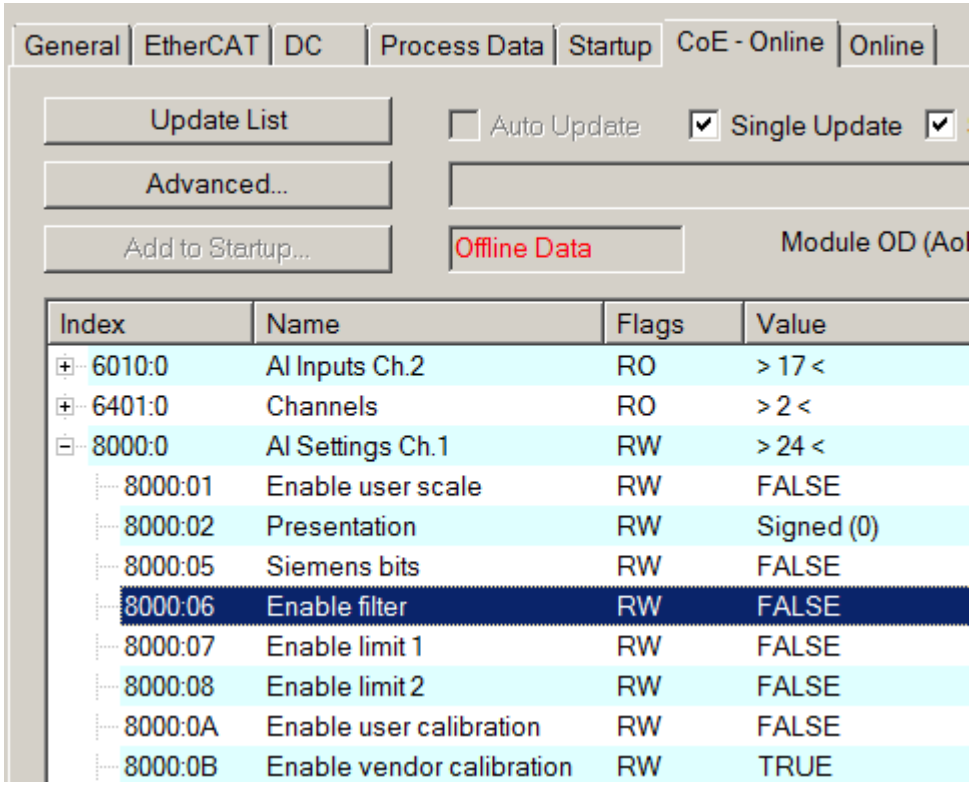


Fig. 88: EL3102, CoE directory

Note	**EtherCAT System Documentation** The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

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

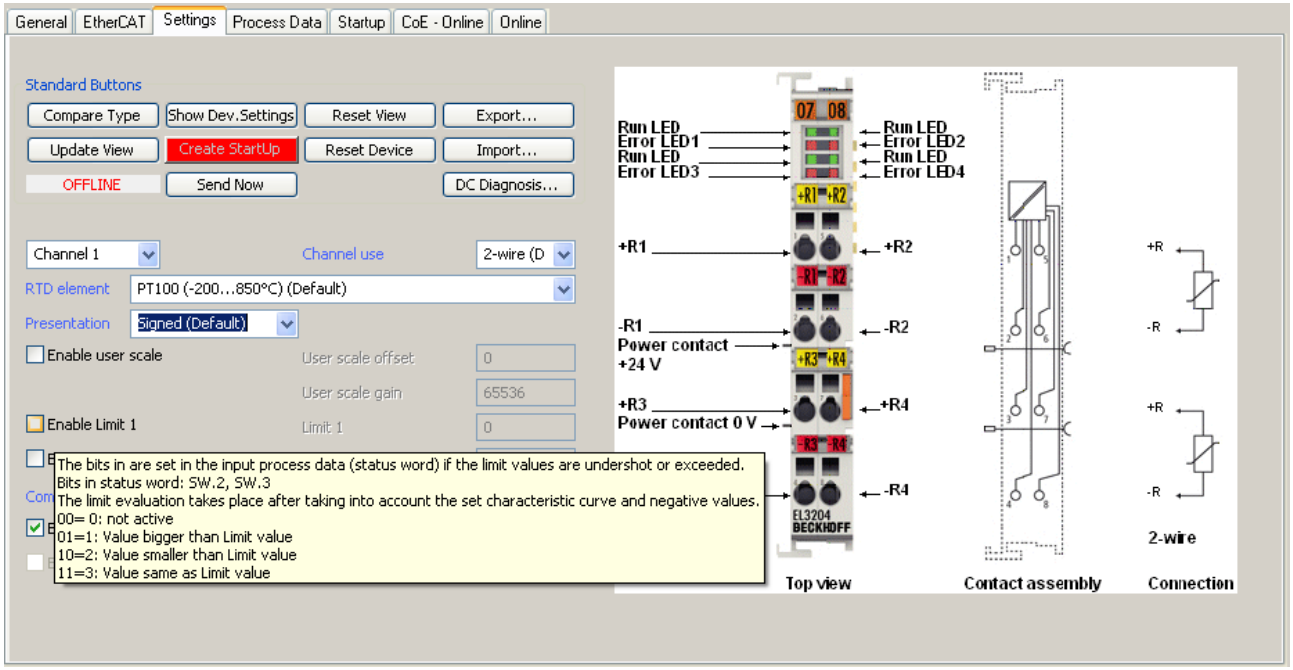


Fig. 89: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP

- SAFEOP
- OP

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

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

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

Standard setting

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

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

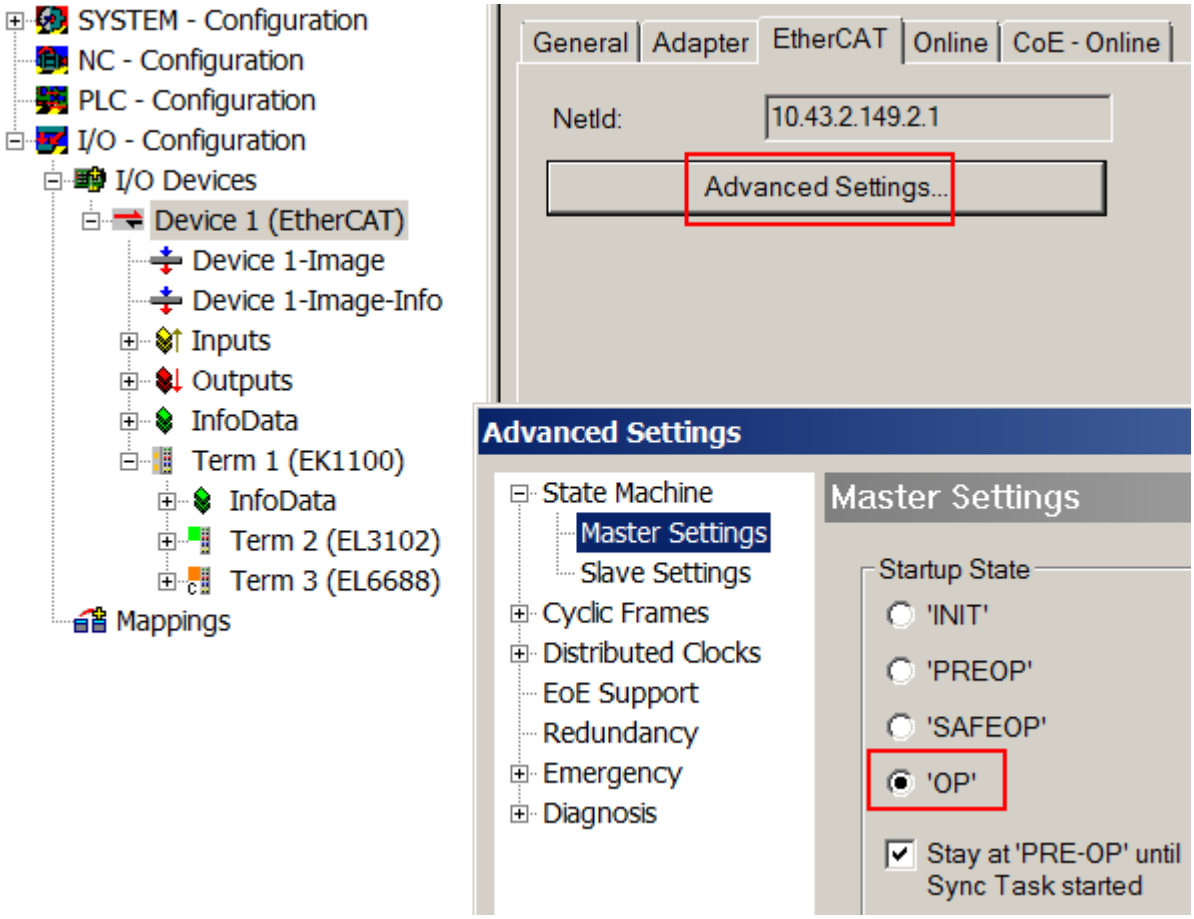


Fig. 90: Default behaviour of the System Manager

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

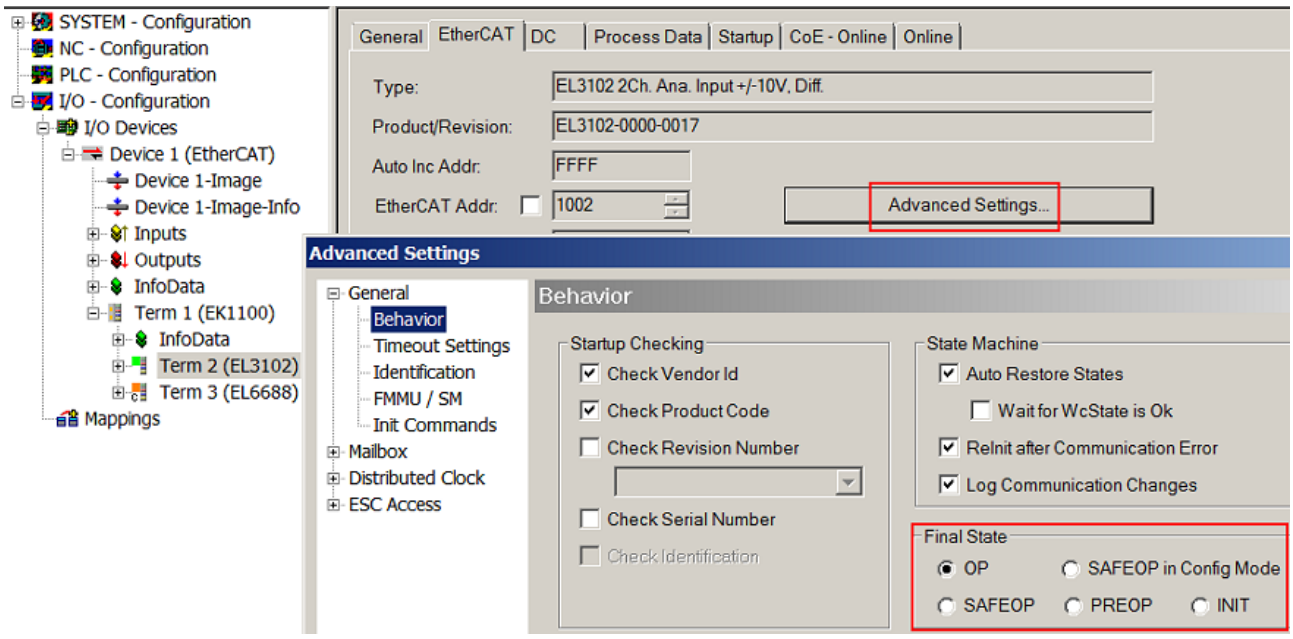


Fig. 91: Default target state in the Slave

Manual Control

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

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

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

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

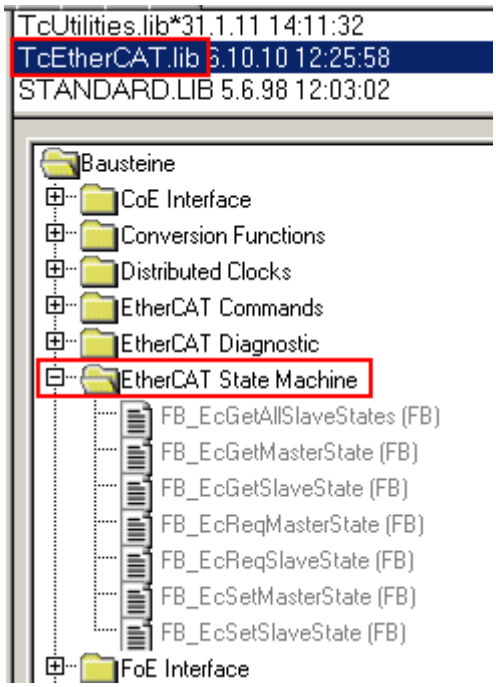
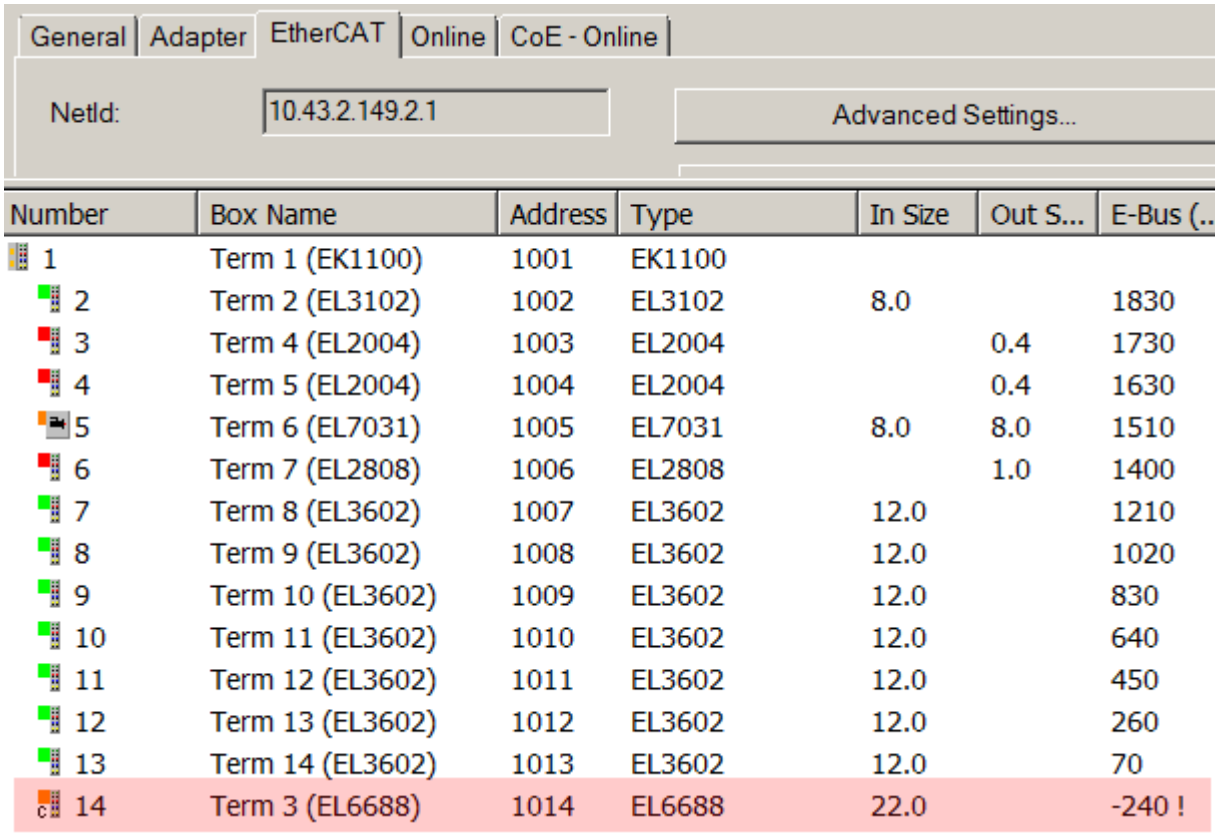


Fig. 92: PLC function blocks

Note regarding E-Bus current

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

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



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

Fig. 93: Illegally exceeding the E-Bus current

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

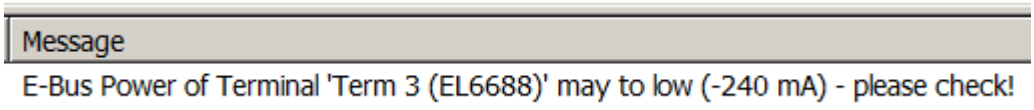


Fig. 94: Warning message for exceeding E-Bus current

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

6.1.7 Configuration by means of the TwinCAT System Manager

(with TwinCAT from version 2.10.0 (Build 1241), using EL5001 from firmware version 0.7 as an example)

In the left-hand window of the TwinCAT System Manager, click on the branch you wish to configure (in the example: EL5001 Terminal 6).

Fig. 95: Branch of EL5001

In the right-hand window of the TwinCAT System manager, various tabs are now available for configuring the terminal.

„General“ tab

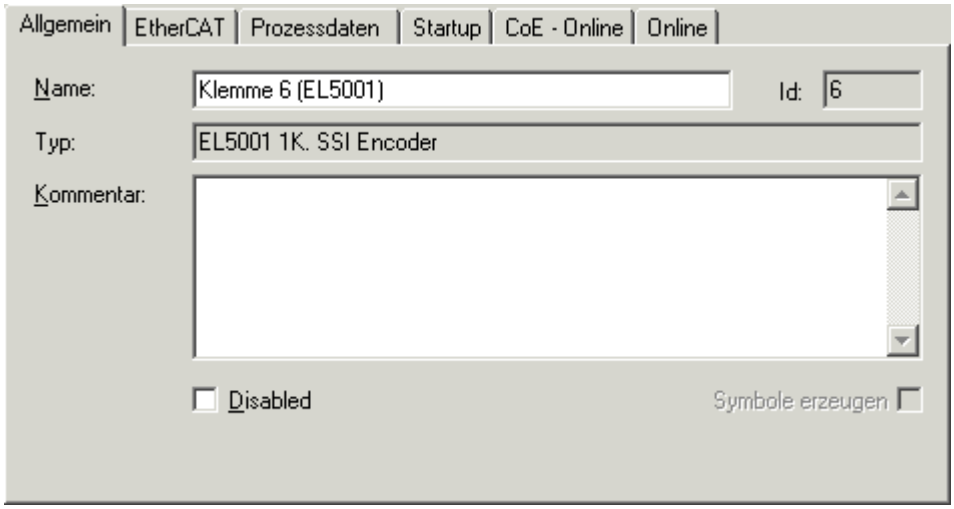


Fig. 96: “General” tab

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. 97: „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 (PDO). 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.

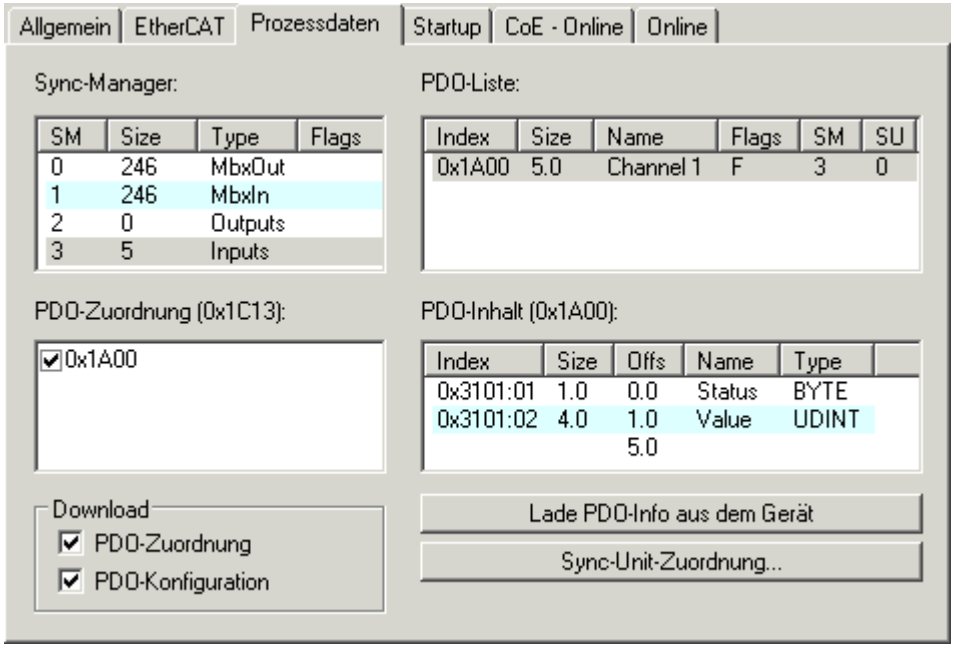


Fig. 98: "Process Data" tab

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 do select a greyed out PDO, the currently selected PDO has to be deselected first.

Note

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 [▶ 84]),

b) and the System Manager has to reload the EtherCAT slaves (button)

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 [▶ 80] 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.

„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.

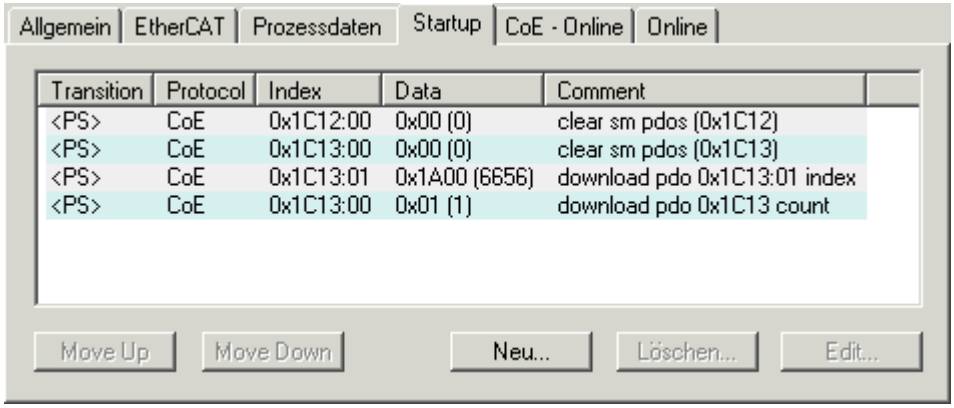


Fig. 99: „Startup“ tab

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.

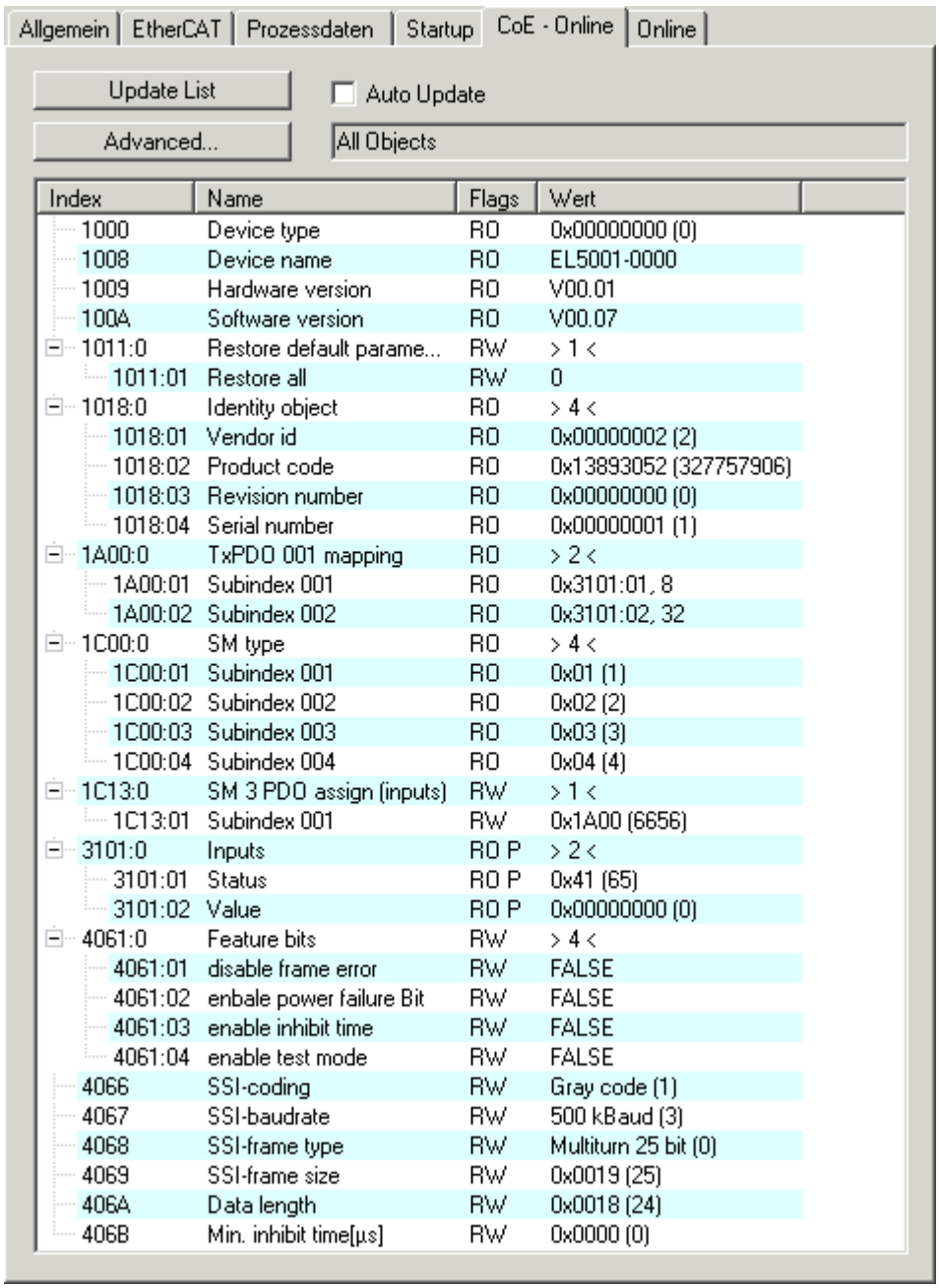


Fig. 100: "CoE – Online" tab

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.

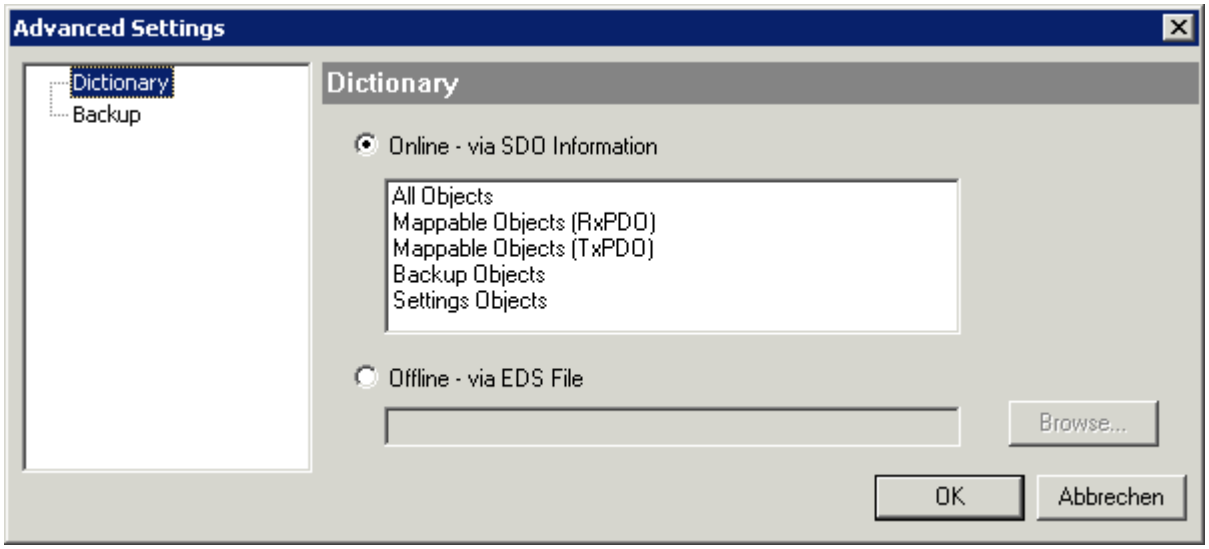


Fig. 101: Dialog “Advanced settings”

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

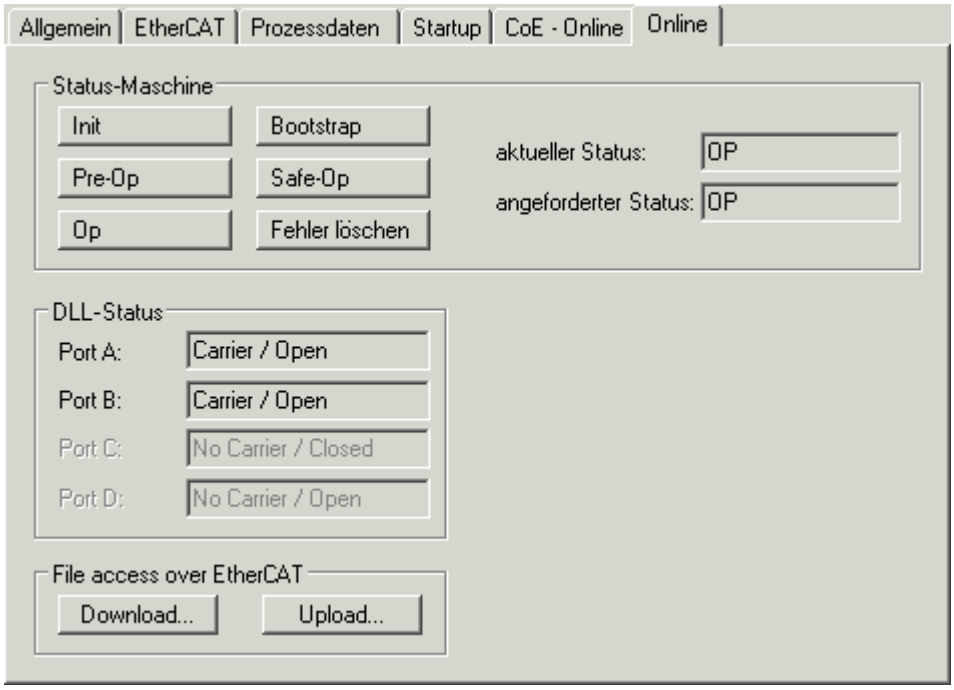


Fig. 102: „Online“ tab

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.

6.2 Start-up and parameter configuration

6.2.1 Integration into the NC configuration

(Master: TwinCAT 2.11 R3)

Note	**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 [▶ 44]").

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.

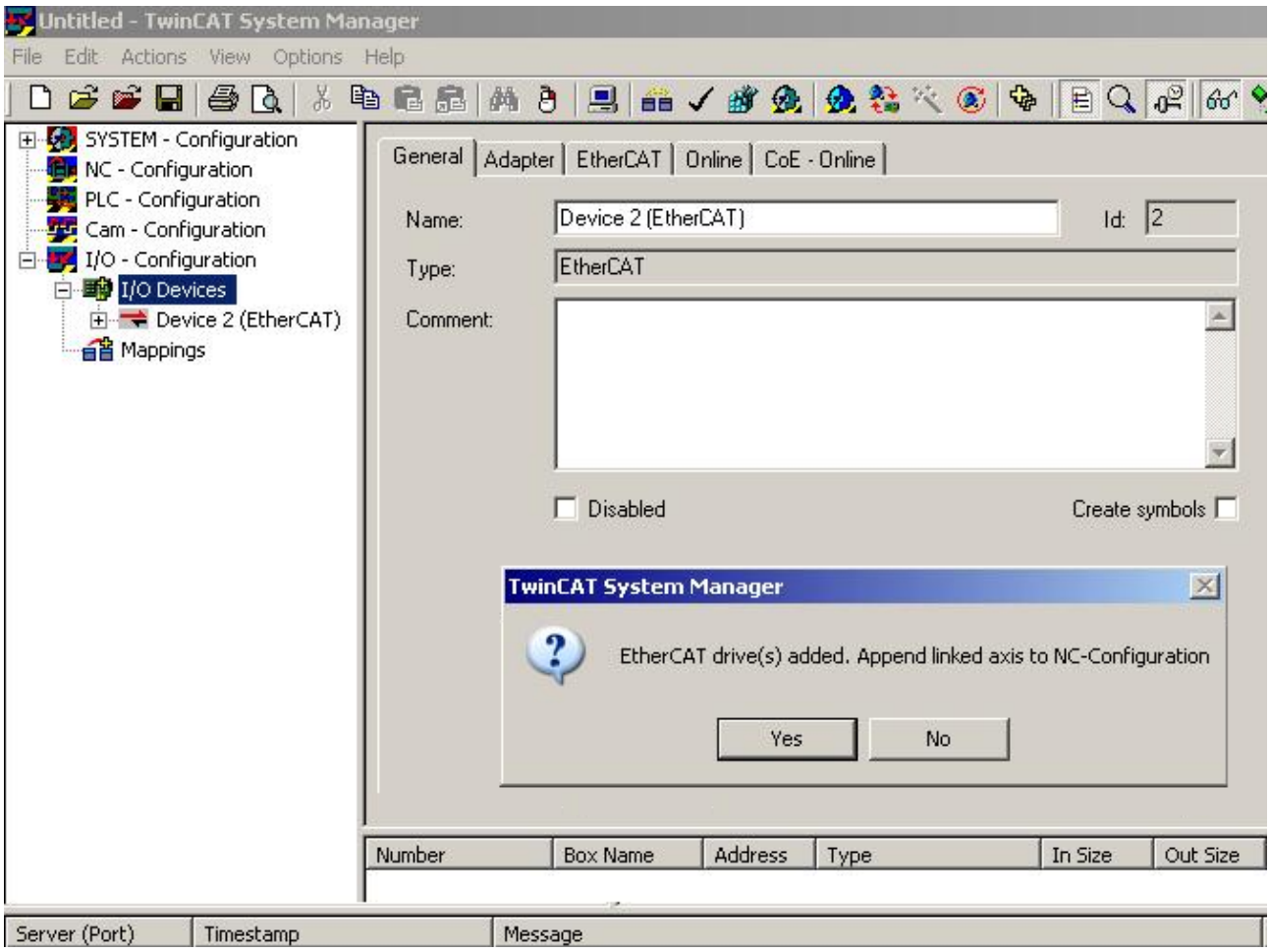


Fig. 103: Axis detected

- 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 [▶ 93]".
  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.

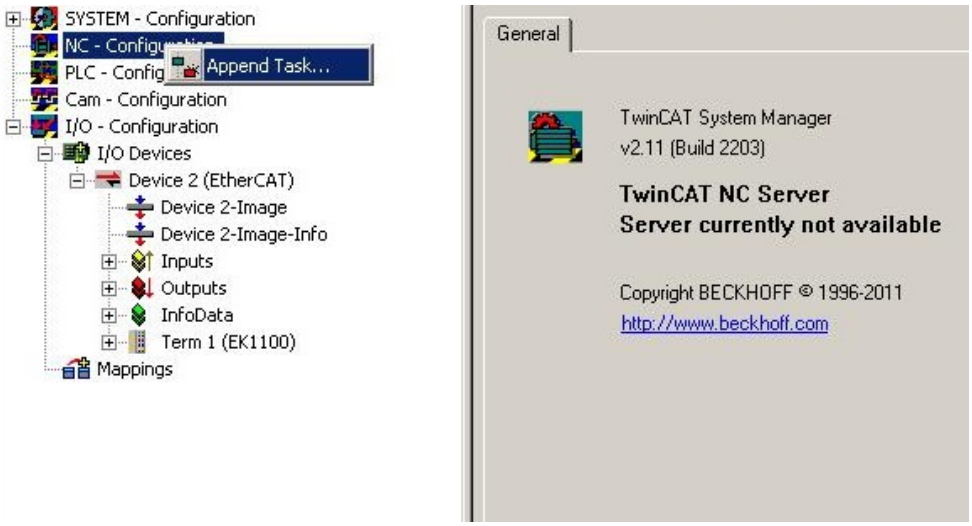


Fig. 104: Adding a new task

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

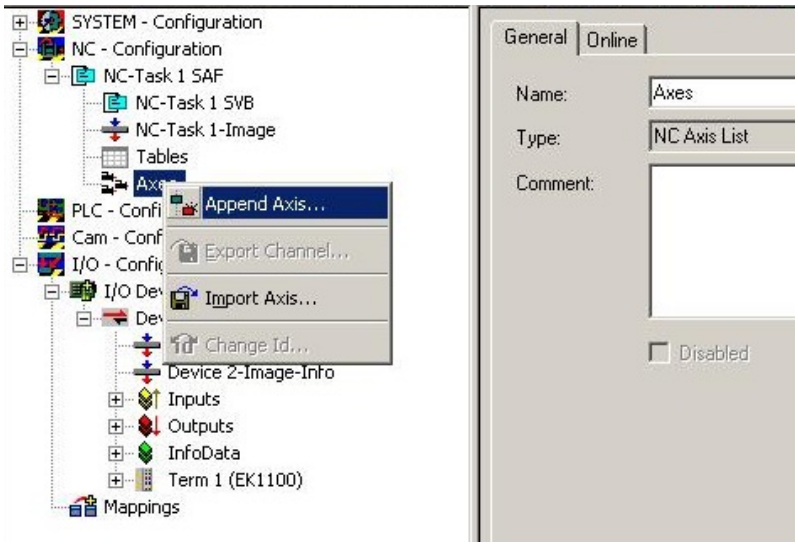


Fig. 105: Adding a new axis

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

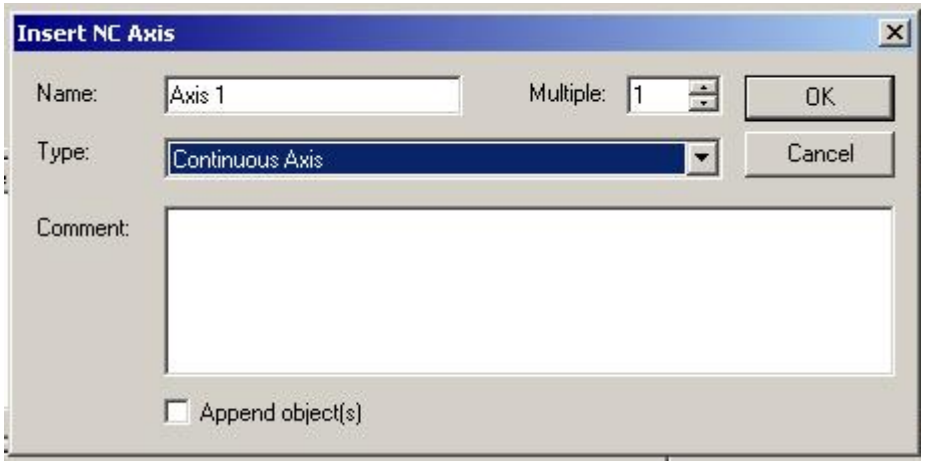


Fig. 106: Selecting and confirming the axis type

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

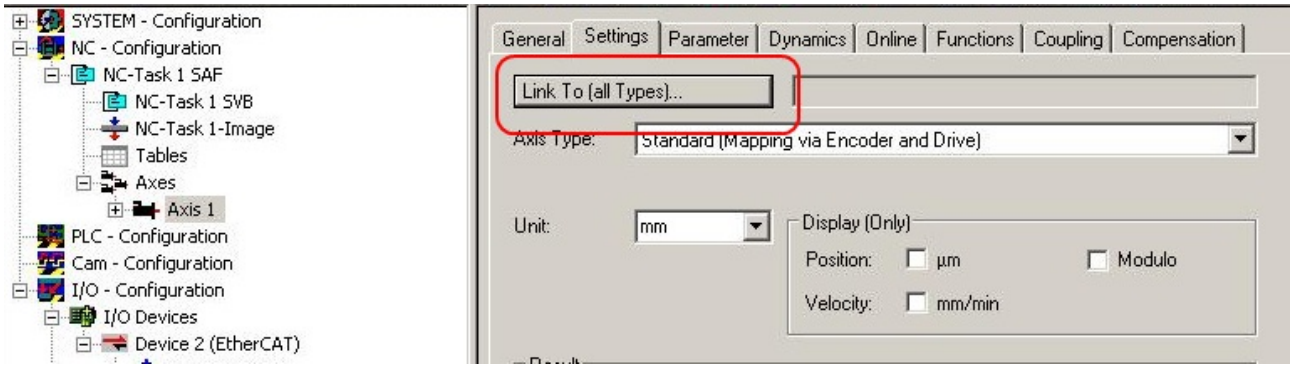


Fig. 107: Linking the axis with the terminal

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

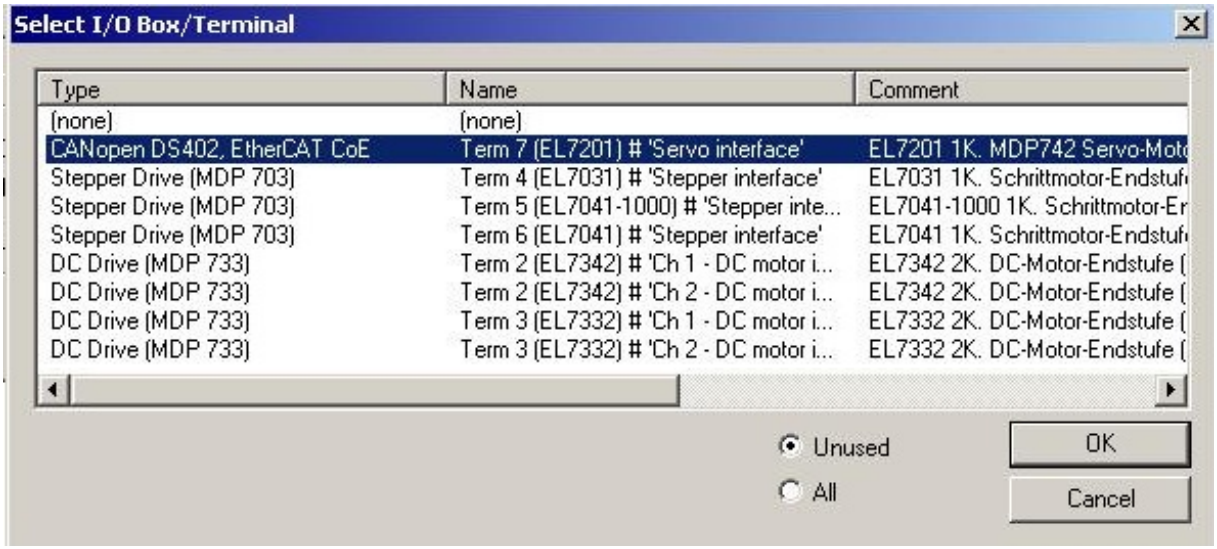


Fig. 108: Selecting the right terminal

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

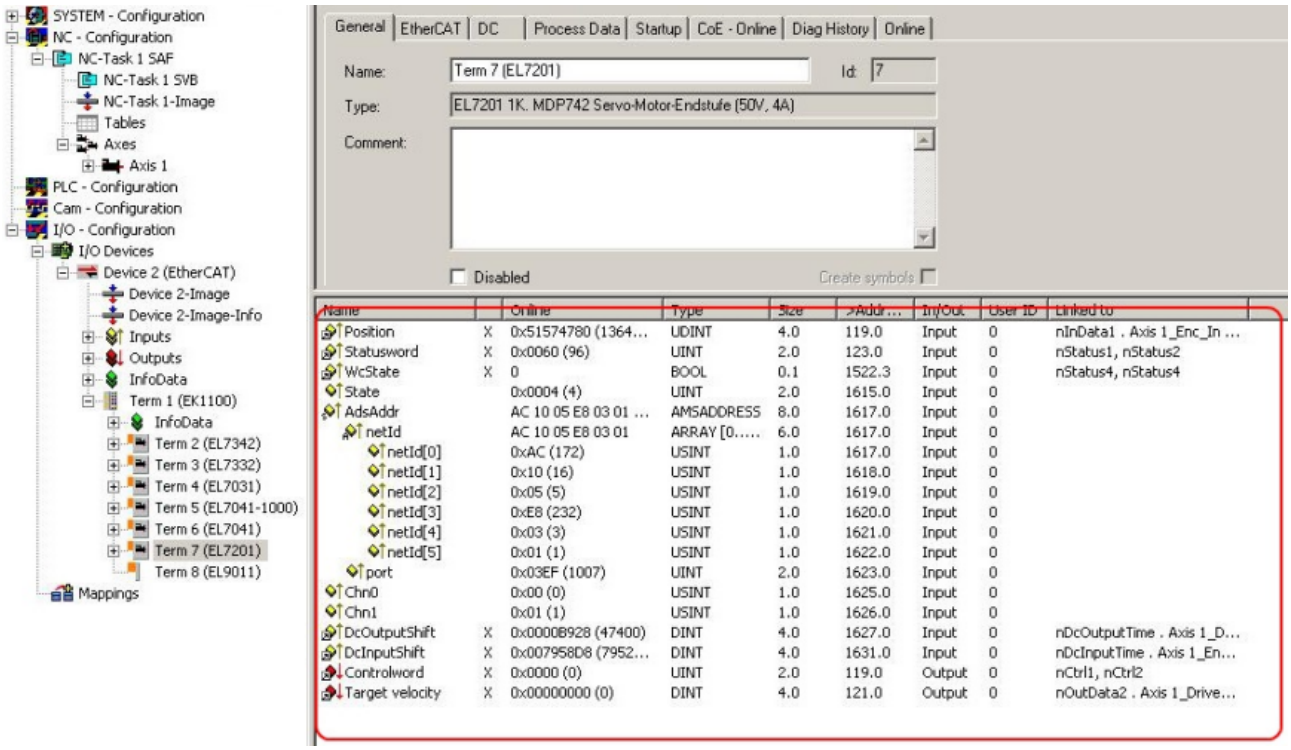


Fig. 109: Automatic linking of all main variables

- Several parameters have to be set before the motor can be started up. The values can be found in sections "CoE settings [▶ 93]" and "NC settings [▶ 97]".
  Please set these parameters before continuing with the motor commissioning procedure.

6.2.2 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.

Note	**Using the Drive Manager from revision -0019** The Drive Manager is only supported from revision -0019 [▶ 191] of the EL72x1-0010. If you use an older version, the settings have to be made manually. See sections "CoE settings [▶ 93]" and "NC settings [▶ 97]"

Table of contents

The TwinCAT Drive Manager is available for download in the AX5000 download package.

The TwinCAT Drive Manager for parameterizing an EL72x1-0010 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 [▶ 52]")
- The terminal must already be integrated in the NC (see section "Integration in the NC configuration [▶ 85]")
- Select the *Configuration* tab for the EL72x1-0010.
- Select the connected voltage under *Power Management*.

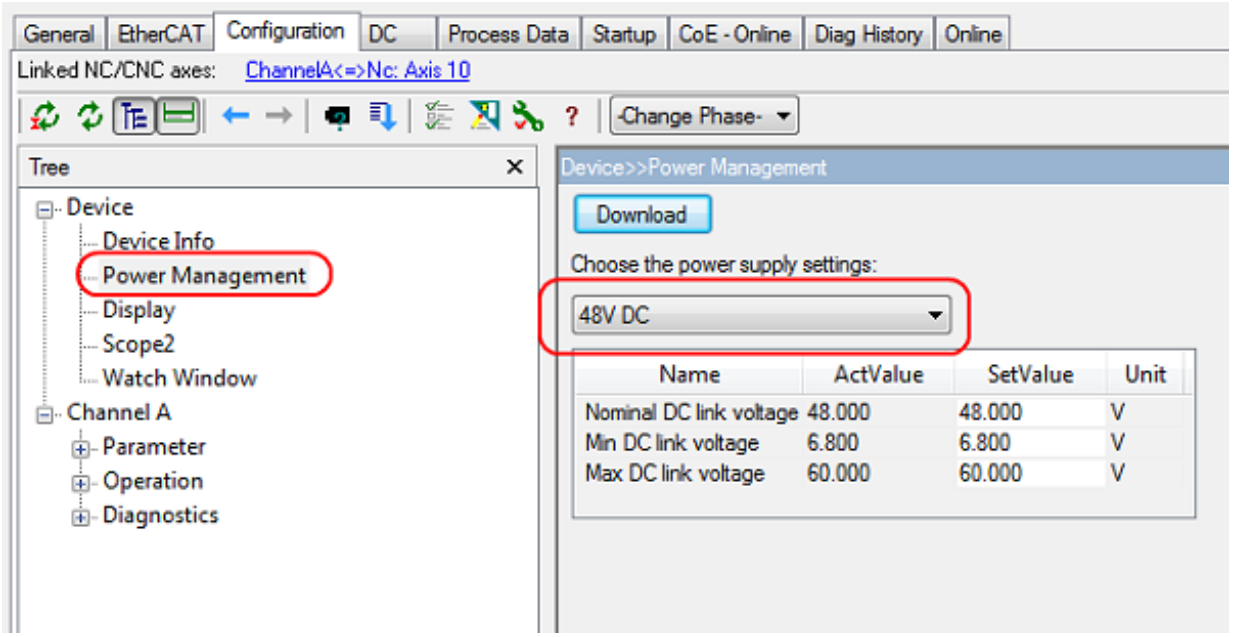


Fig. 110: Selecting the connected voltage

- 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 0x8001 [▶ 165], MDP or Index 0x2018 [▶ 149], DS402)

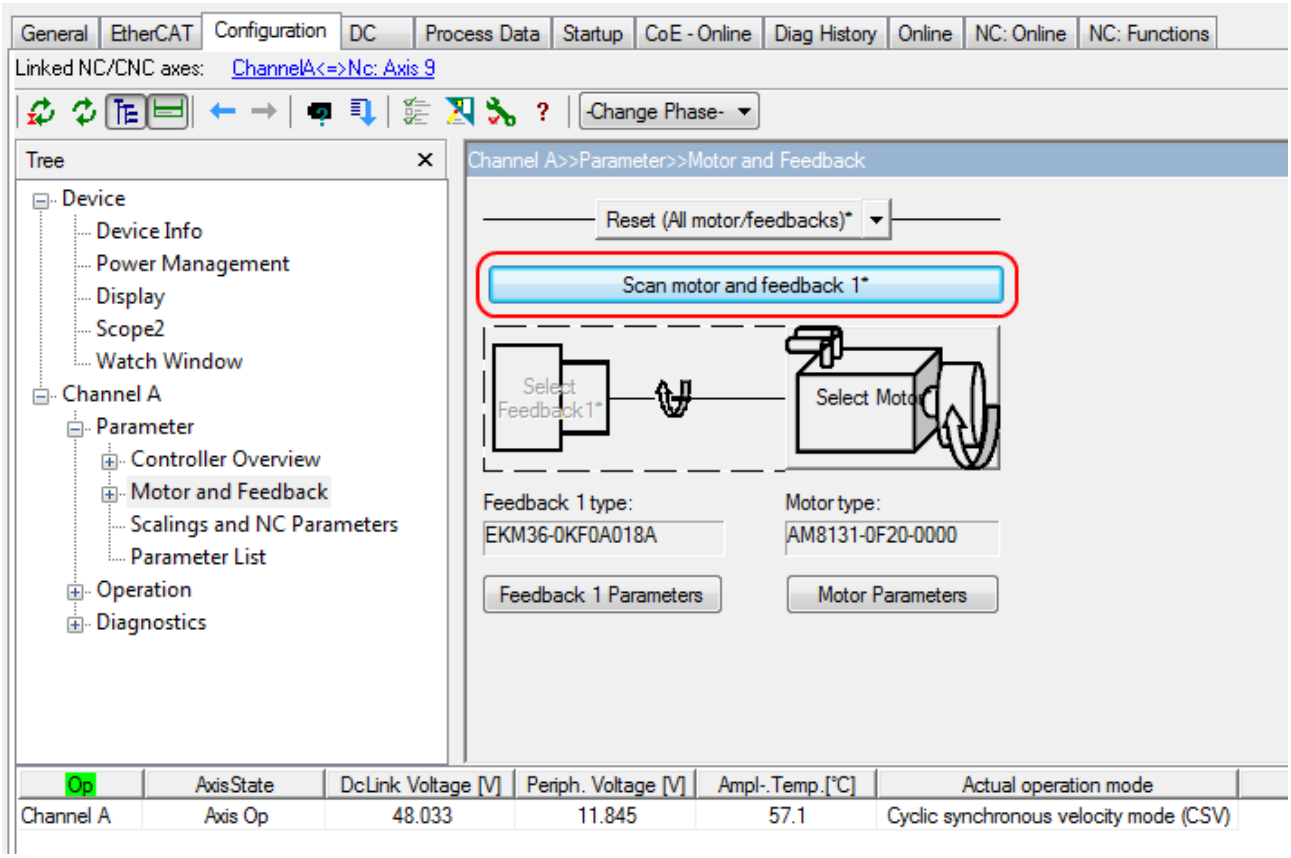


Fig. 111: Automatic scanning of the connected motor

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

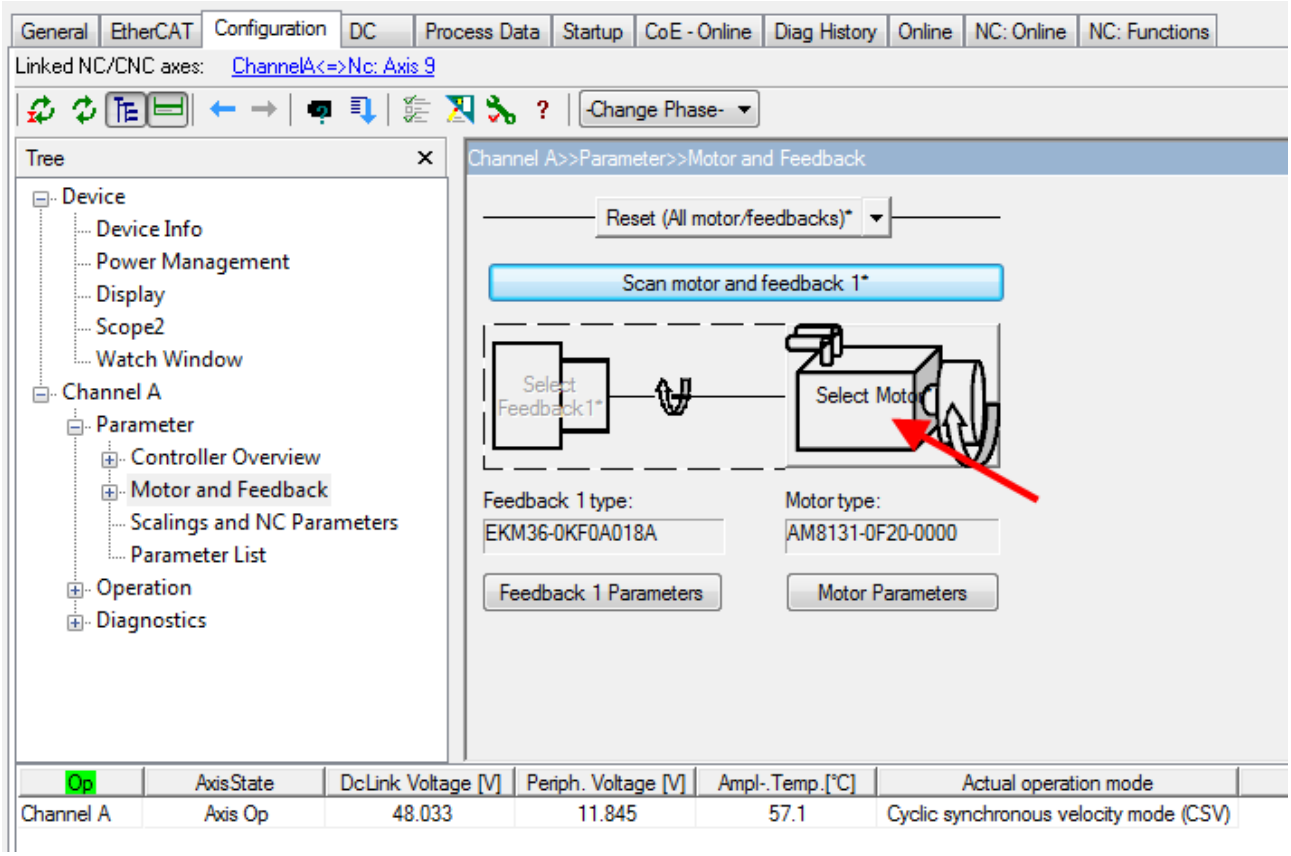


Fig. 112: Selecting the connected motor

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

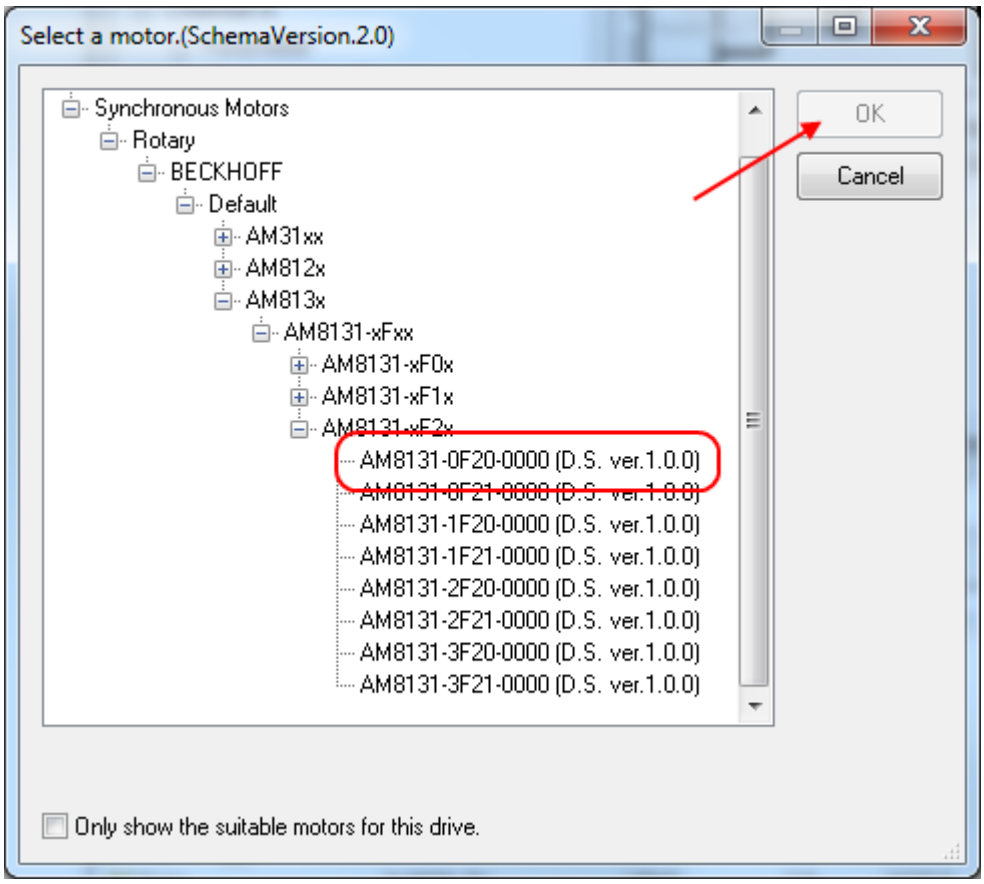


Fig. 113: List of available motors

- 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 [▶ 97]".

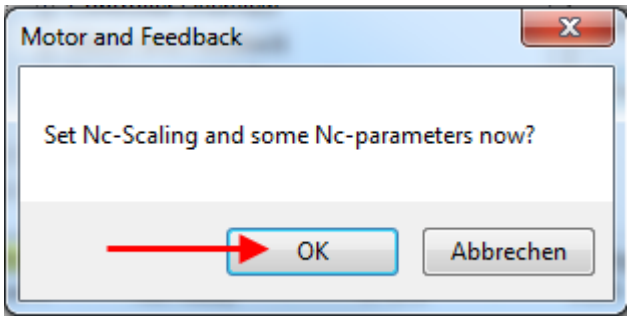


Fig. 114: Confirmation of the automatic NC settings parameters

- 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.

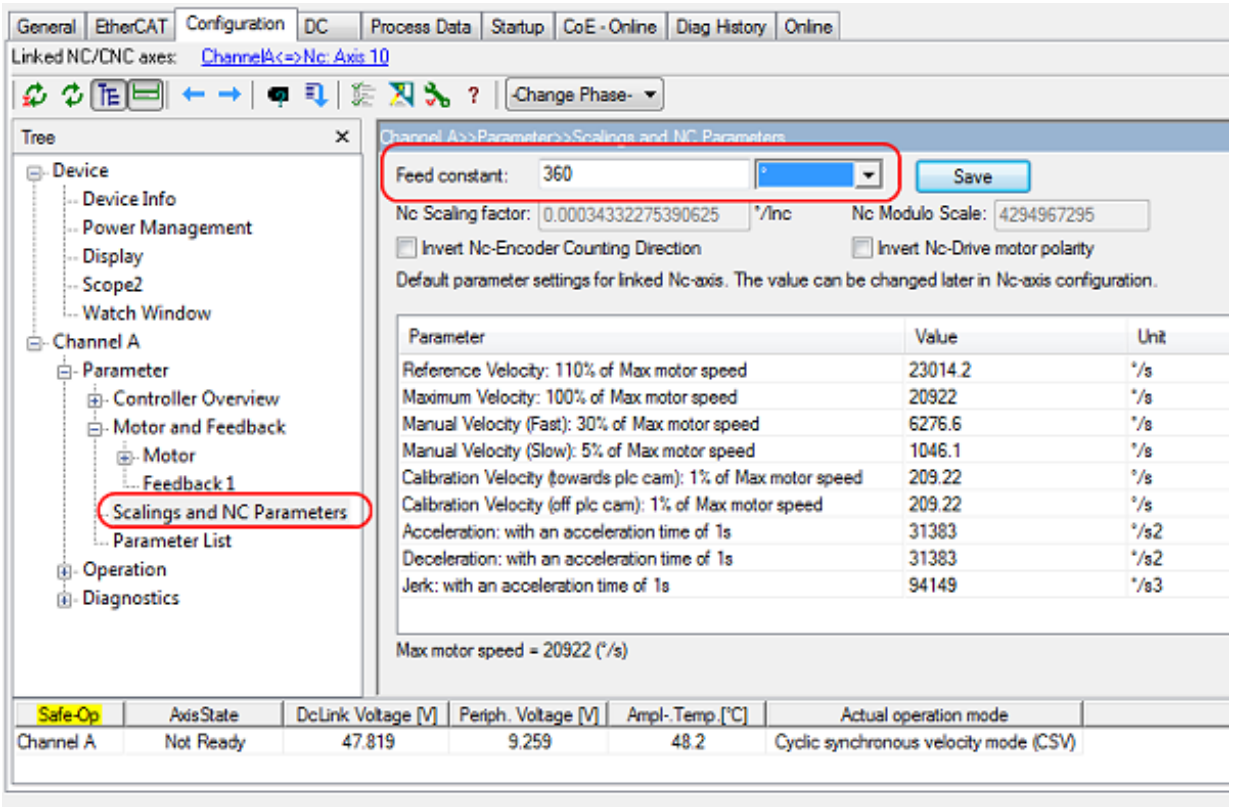


Fig. 115: Adapting the scaling

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 [▶ 102]". Or the NC can be addressed from the PLC. A small http://infosys.beckhoff.com/content/1033/el72x1-0010/Resources/zip/1859339787.zip 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%.

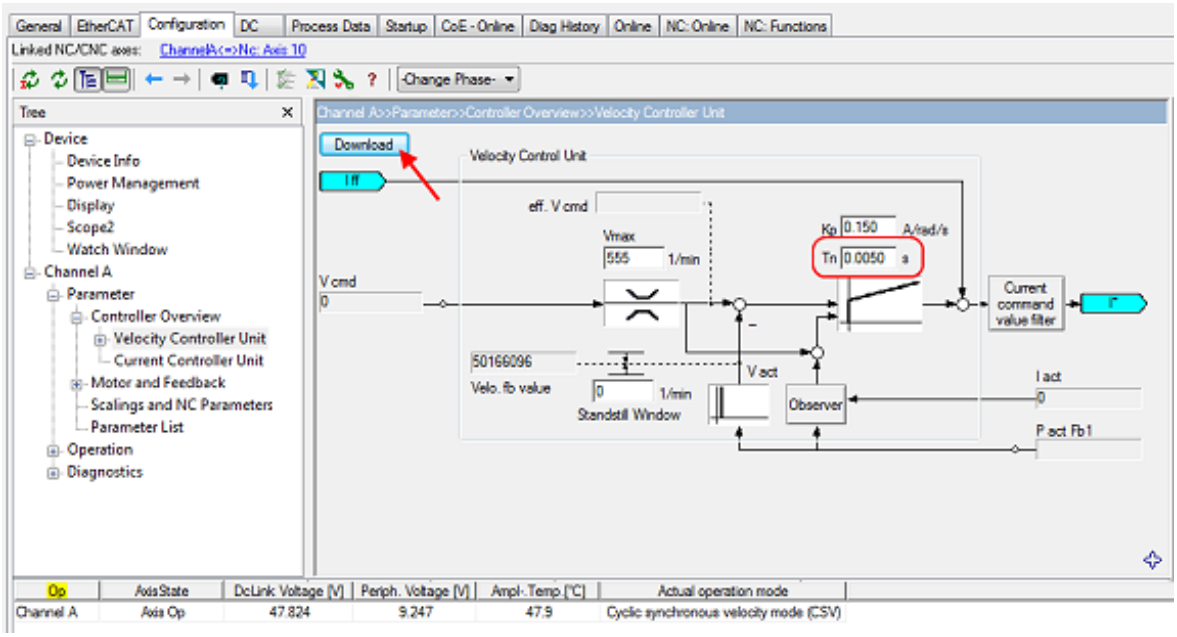


Fig. 116: Adapting Tn

Proportional velocity controller component Kp

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

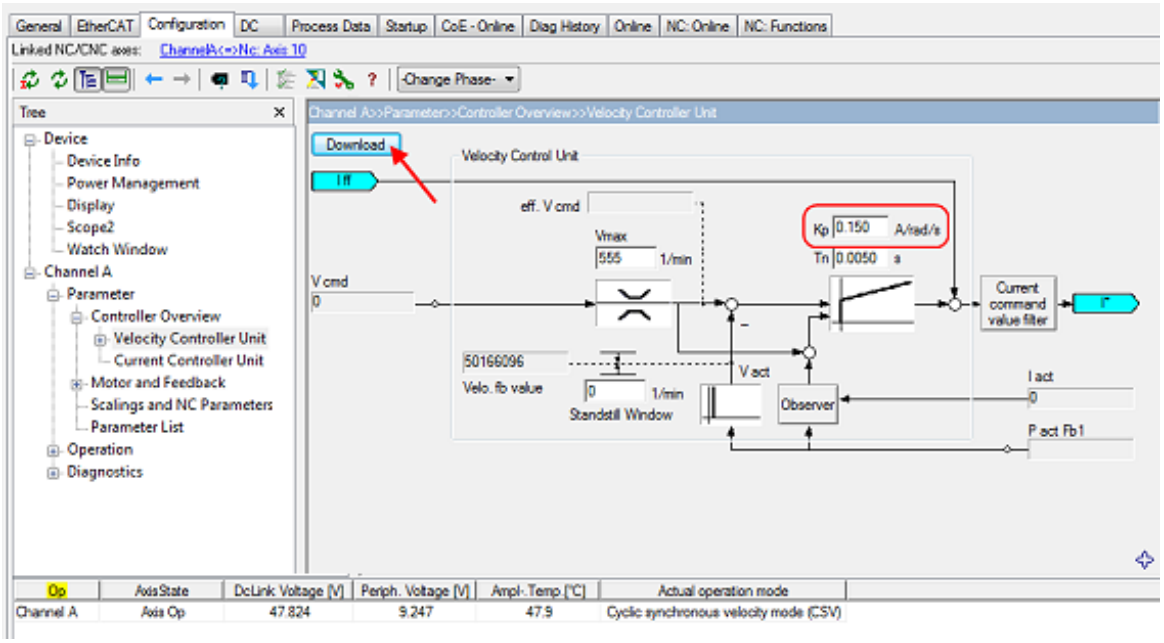


Fig. 117: Adapting Kp

6.2.3 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

Note	**Downloading the EL72x1-0010 motor XML files** The motor XML files are available for download from the Beckhoff website.

To facilitate commissioning of the EL72x1-0010 servo terminal, motor XML files are provided for the servomotors that are supported by the EL72x1-0010. 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 EL72x1-0010 and open the *Startup* tab. Right-click in the empty field and select *Import from XML*...(see Fig. *Importing the motor XML file*).

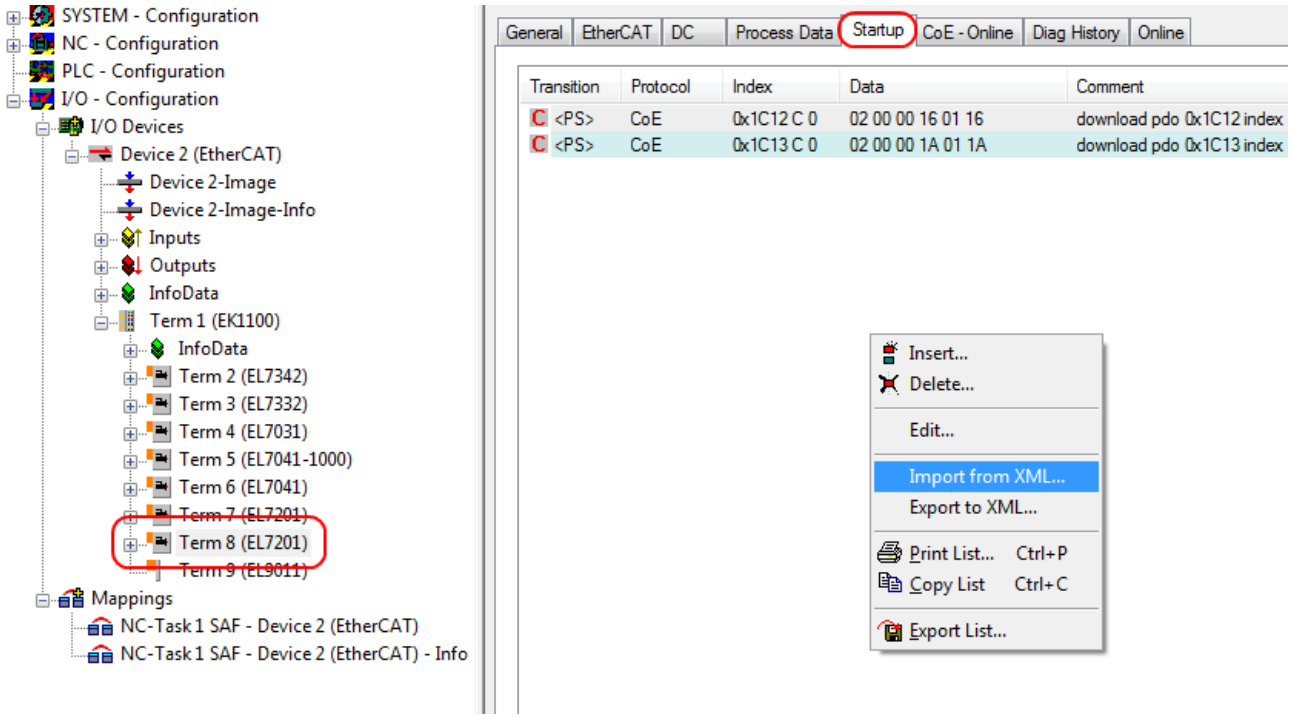


Fig. 118: Importing the motor XML file

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

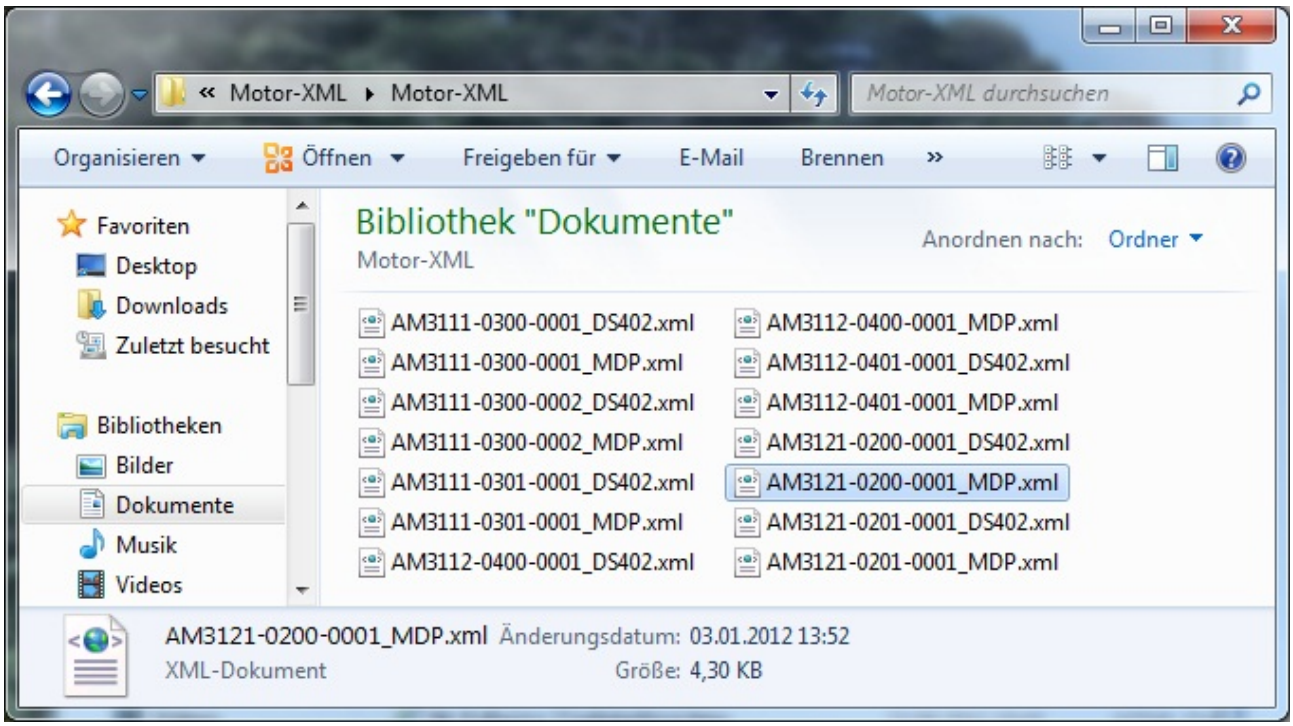


Fig. 119: Selecting the correct motor XML file

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

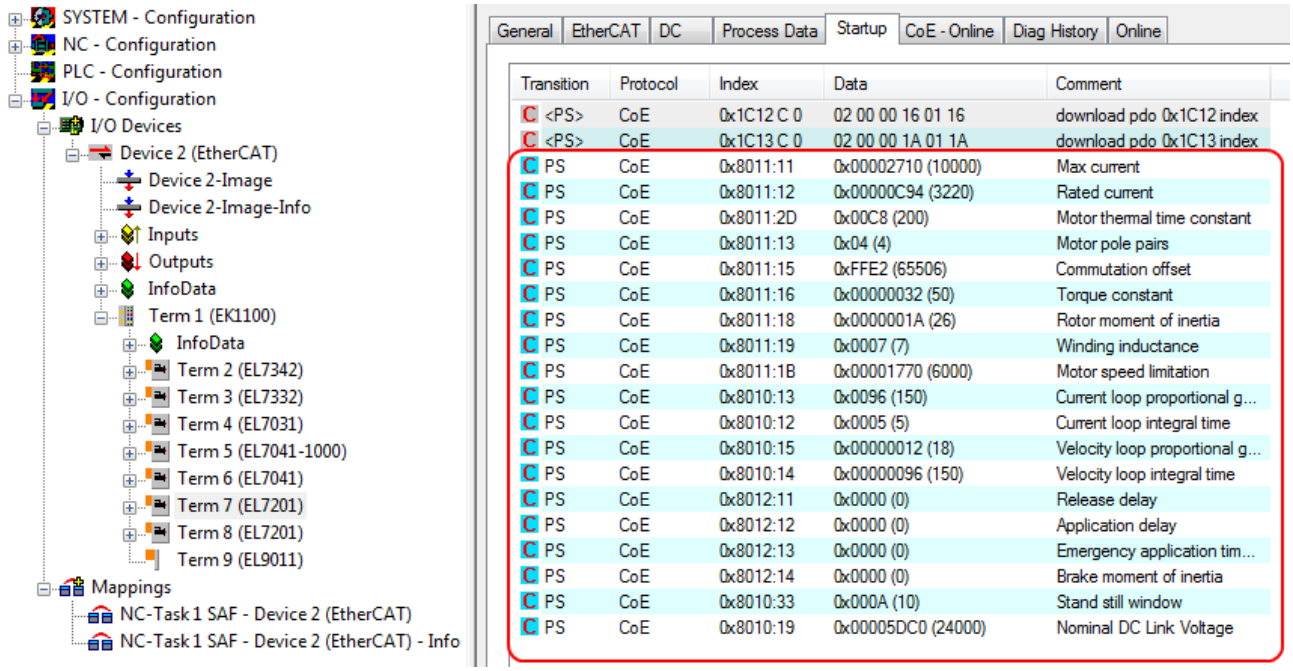


Fig. 120: CoE parameters of the motor XML file

Note	**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

Attention	**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 [▶ 166] (0x2002:19 [▶ 147], DS402 profile) "Nominal DC Link Voltage" of the connected voltage to be set accordingly

Setting further parameters

Single-turn Bits (MDP742: Index 0x8000:12 [▶ 165] / DS402: Index 0x2010:12 [▶ 149]) / Multi-turn Bits (MDP742: Index 0x8000:13 [▶ 165] / DS402: Index 0x2010:13 [▶ 149])

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.

Note	**The motor may overheat!** If the number of single-turn bits is changed, the scaling factor in the NC [▶ 97] has to be adjusted

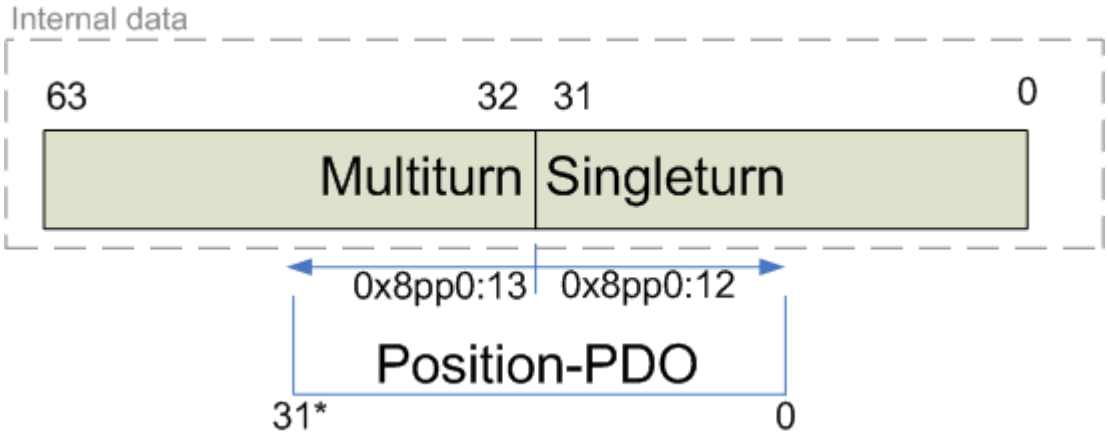


Fig. 121: Multi-turn / single-turn bits:

Torque limitation (MDP742: Index 0x7010:0B [▶ 173] / DS402: Index 0x6072:0 [▶ 152])

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 [▶ 166] / DS402: Index 0x2002:14 [▶ 147])

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 [▶ 166] / DS402: Index 0x2002:15 [▶ 147])

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%.

6.2.4 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

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.

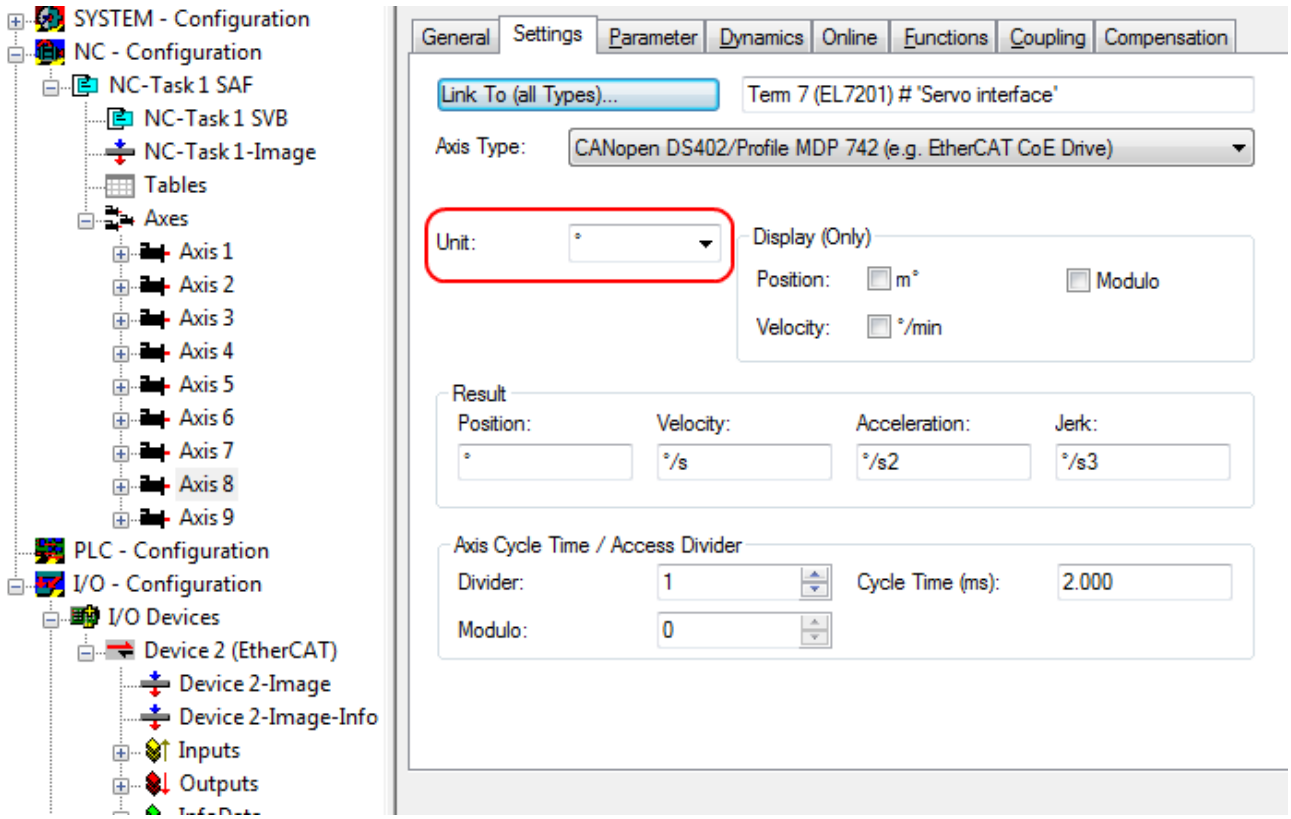


Fig. 122: Definition of the unit

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.

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

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

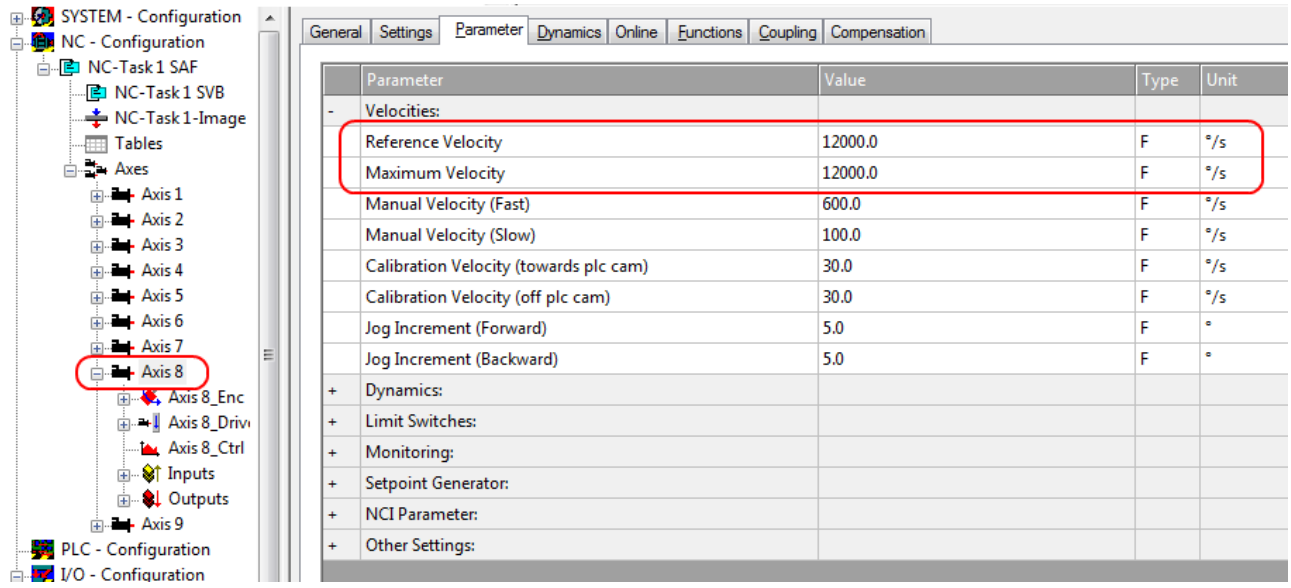


Fig. 123: Adjusting the reference velocity

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 found further below. Open *Further settings*. The dead time compensation should, in theory, be 3 cycles of the NC cycle time, although in practice 4 cycles were found to be preferable. At a cycle time of 2 ms it should therefore be 0.008 s. The dead time compensation can be found under *Advanced Settings in the encoder parameters*.

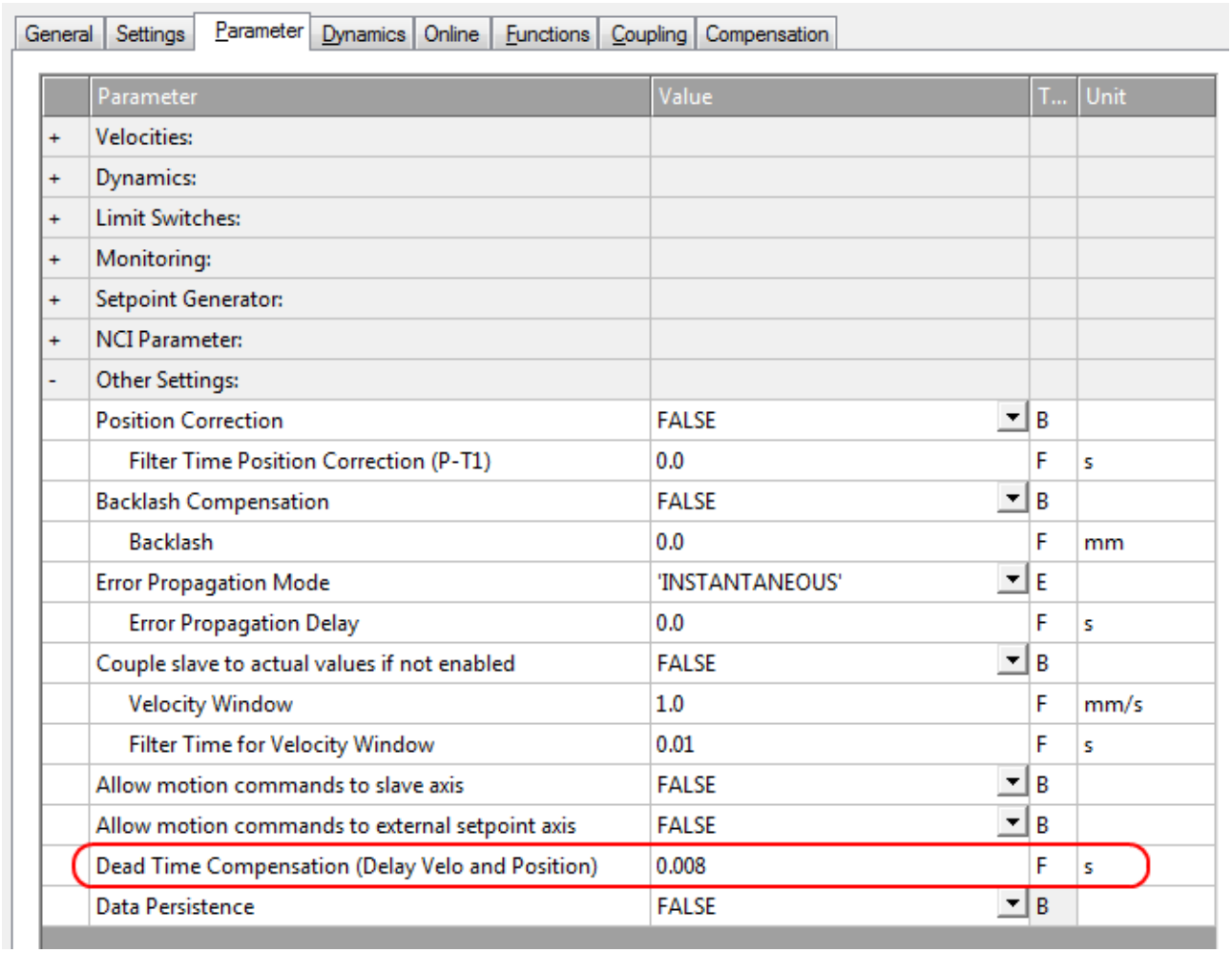


Fig. 124: Dead time compensation

The dead time compensation can only be set in expert mode (Fig. *Expert mode setting*).

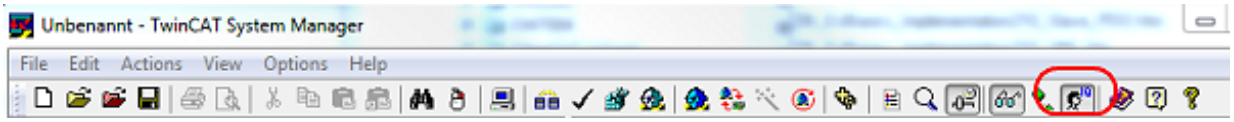


Fig. 125: Expert mode setting

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 EL72x1-0010 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.

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

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

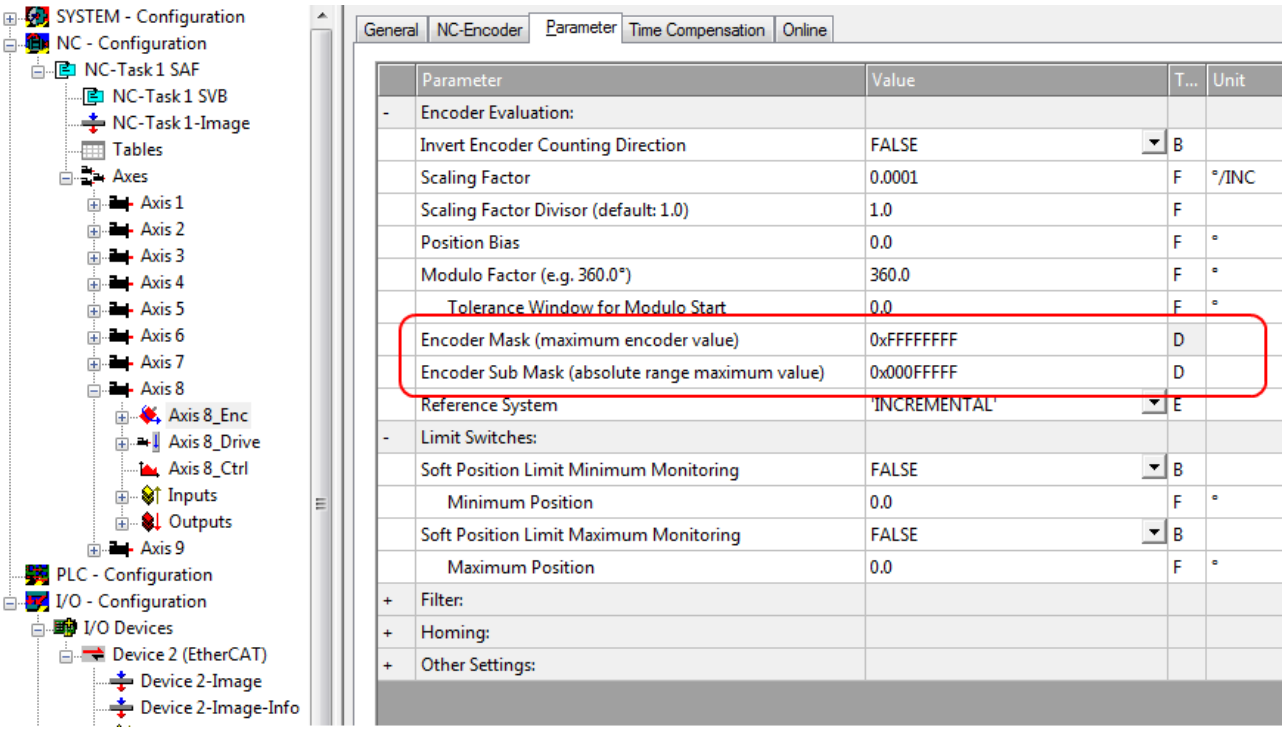


Fig. 126: Setting the encoder mask

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 EL72x1-0010 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

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

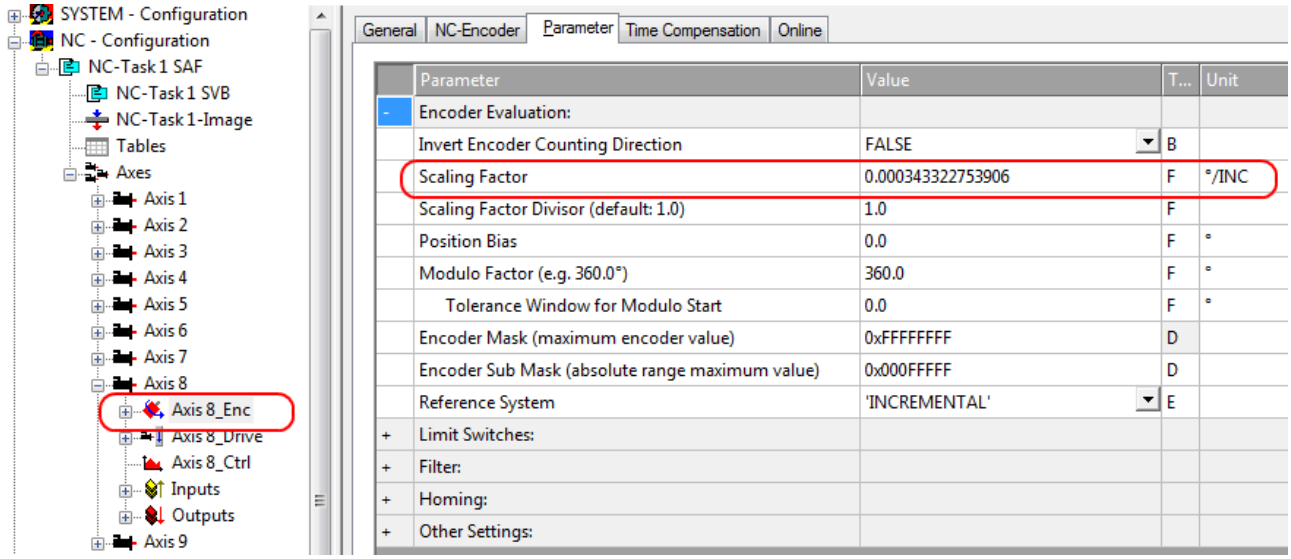


Fig. 127: Setting the Scaling Factor

Scaling output

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

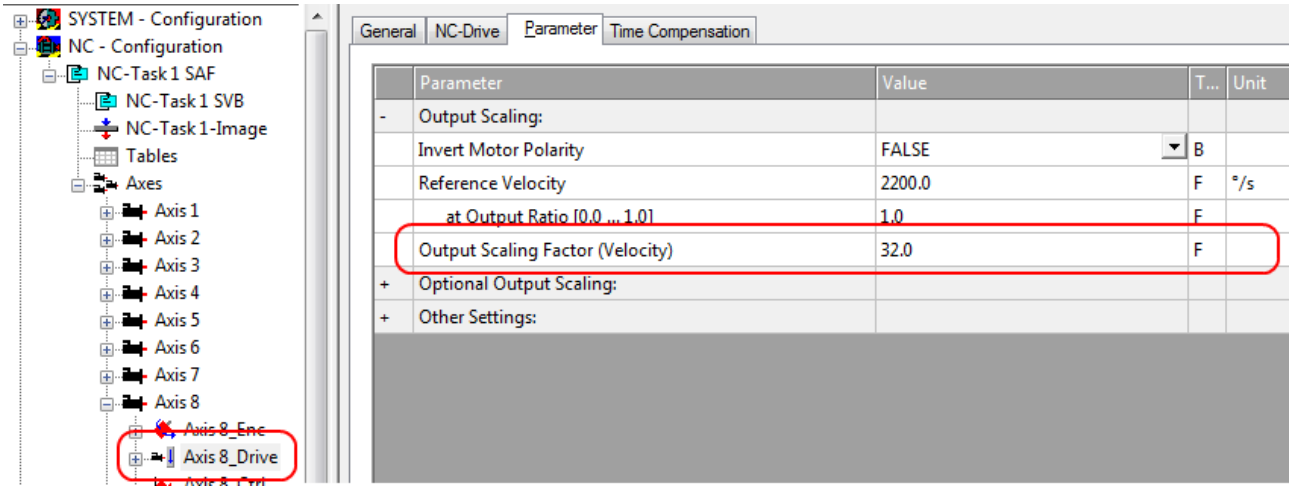


Fig. 128: Output scaling

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.

Attention	**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.

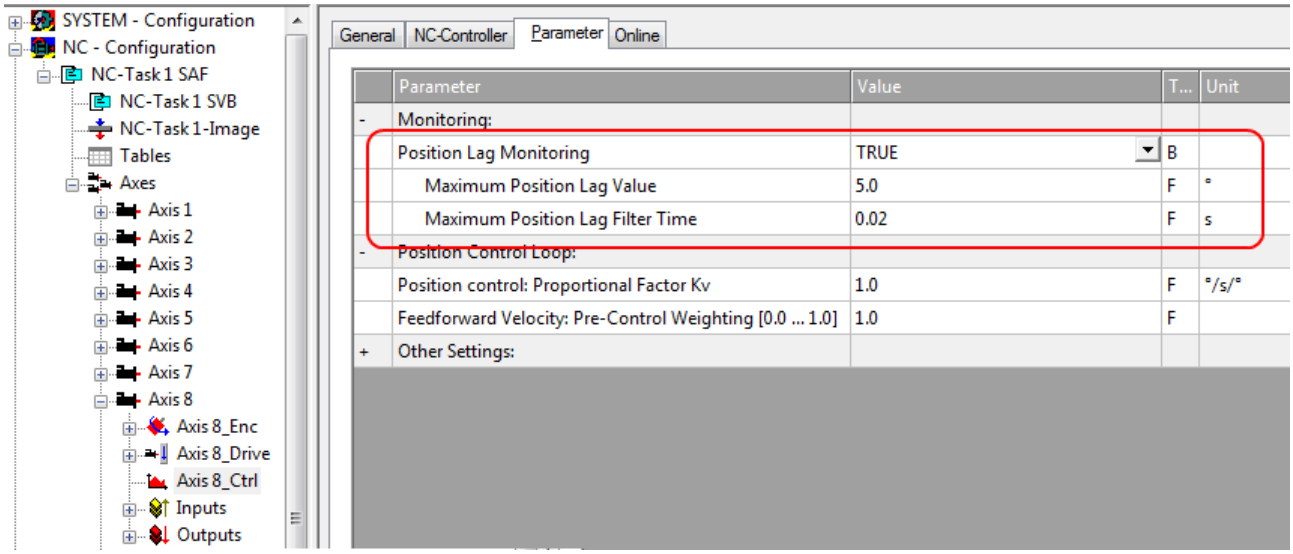


Fig. 129: Lag monitoring

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.

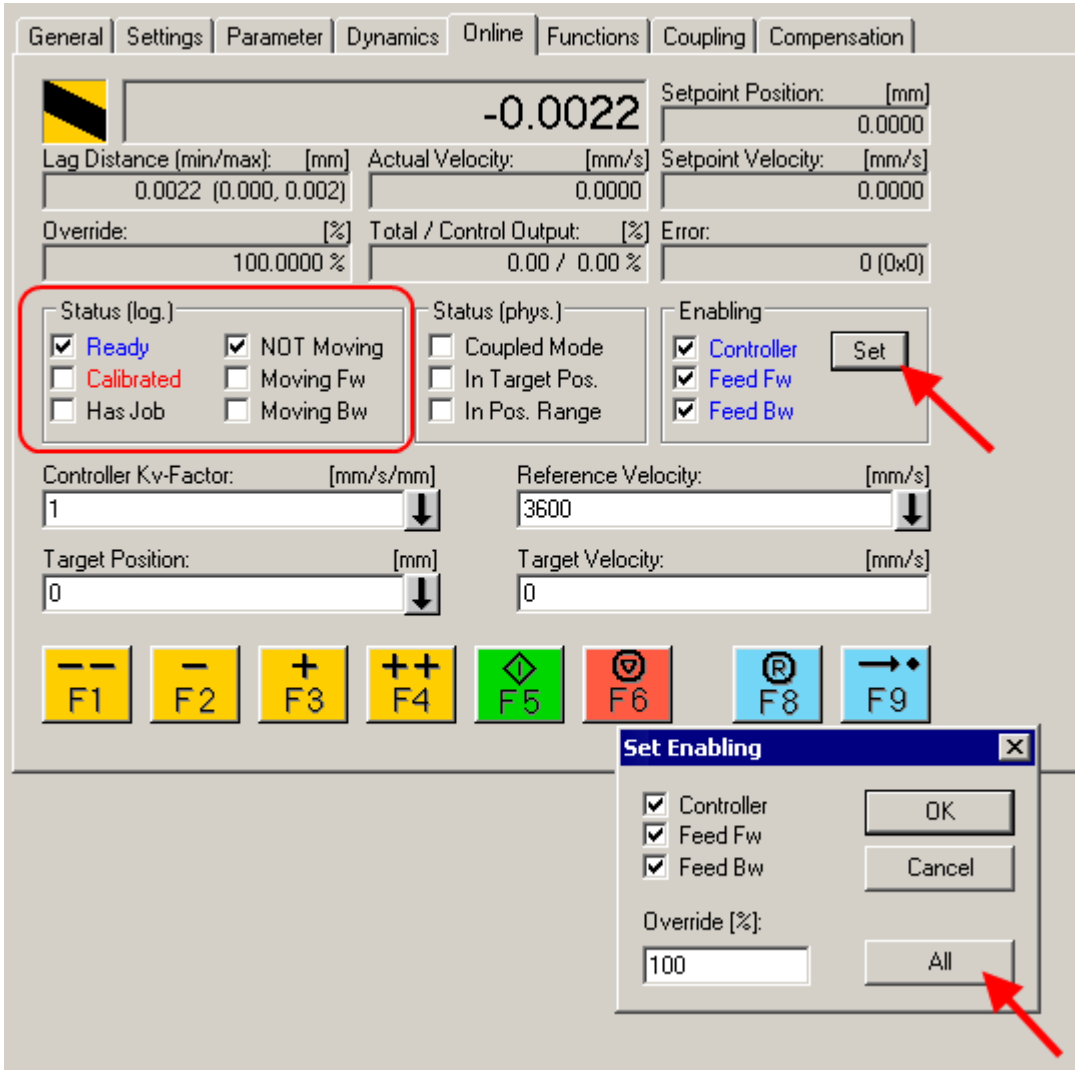


Fig. 130: Enabling an axis

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 [▶ 98]". 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.

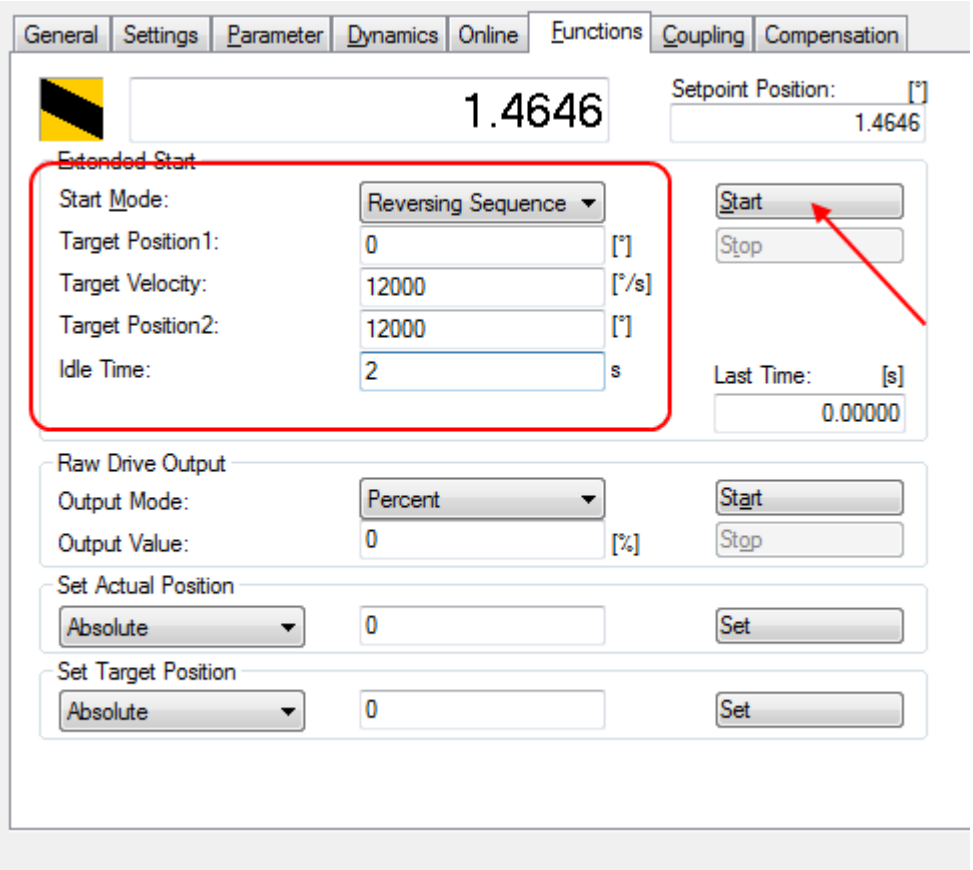


Fig. 131: Reversing Sequence

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

6.2.5 Application example

Note	**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 (http://infosys.beckhoff.com/content/1033/el72x1-0010/Resources/zip/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 2 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*.

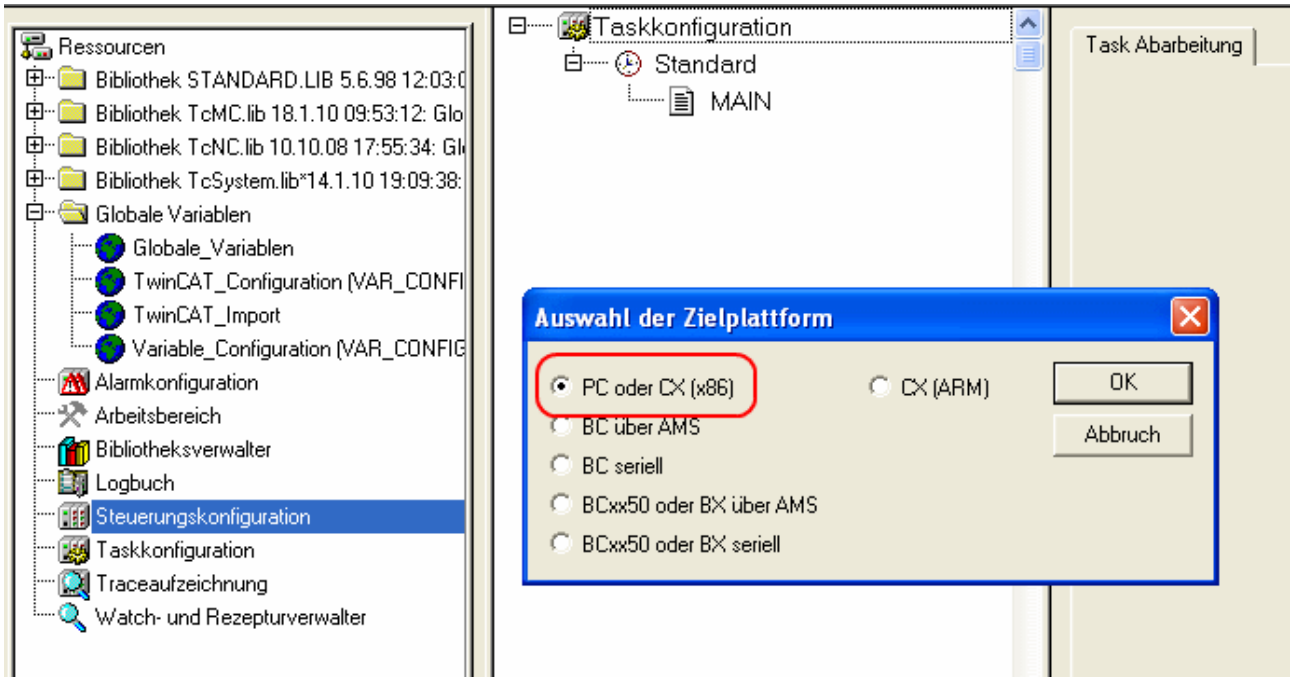


Fig. 132: Selection of the target platform

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.

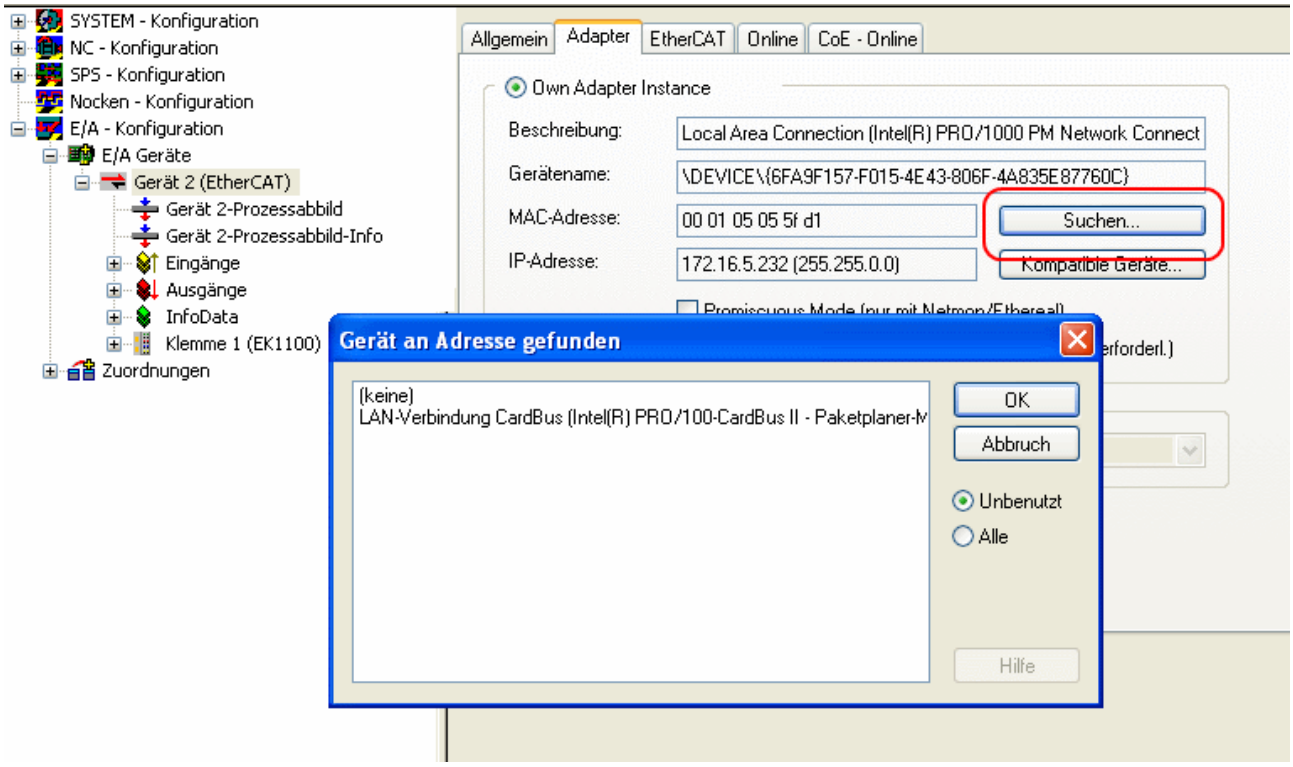


Fig. 133: Selecting the MAC address

- 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.

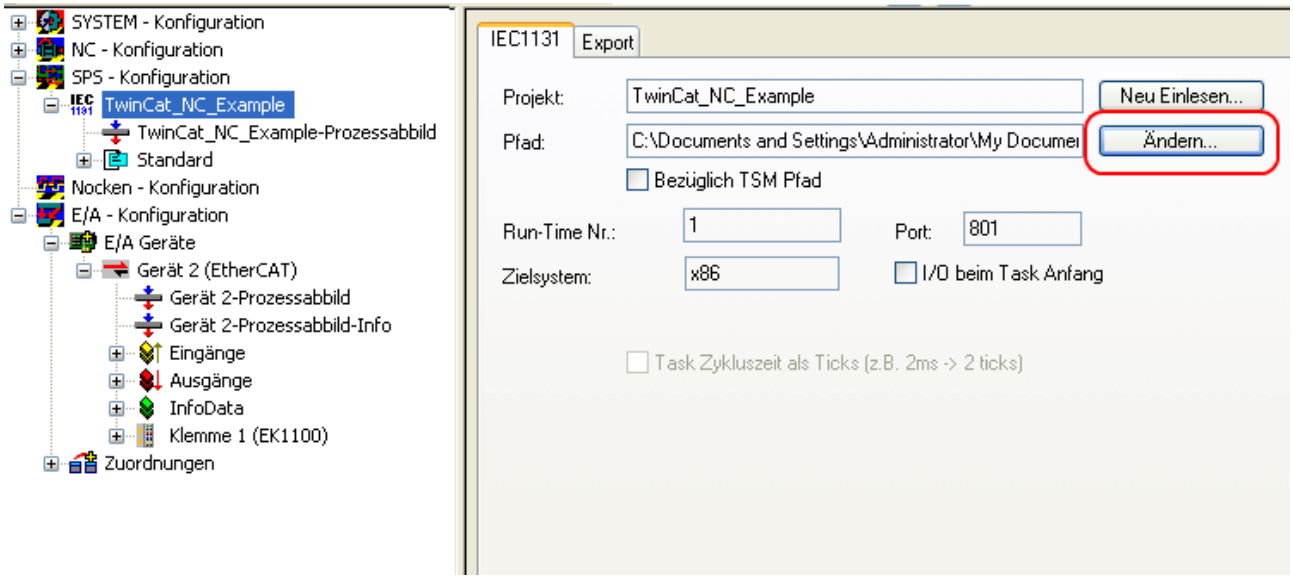


Fig. 134: Changing the PLC path

- 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 [▶ 85]".

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

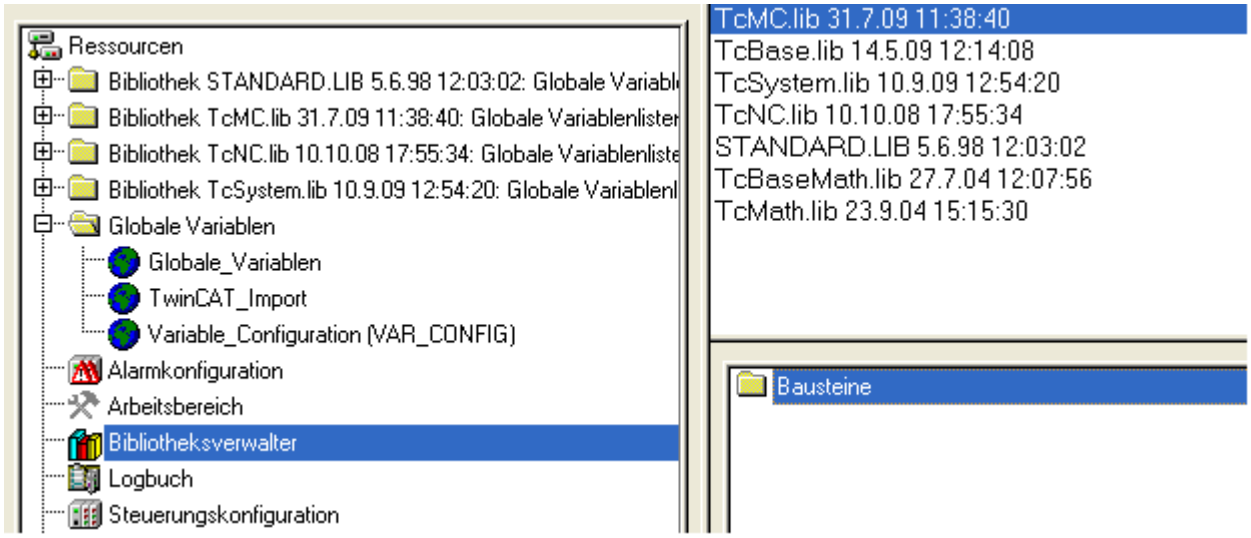


Fig. 135: Required libraries

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.

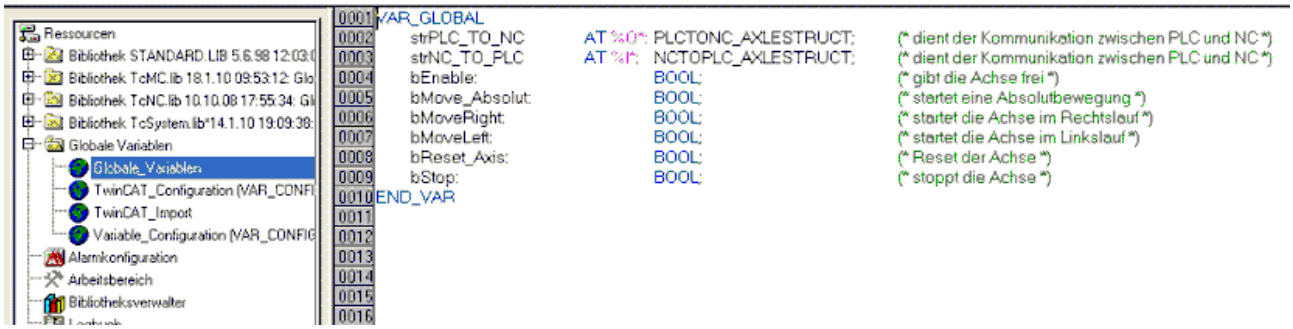


Fig. 136: Global variables

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.

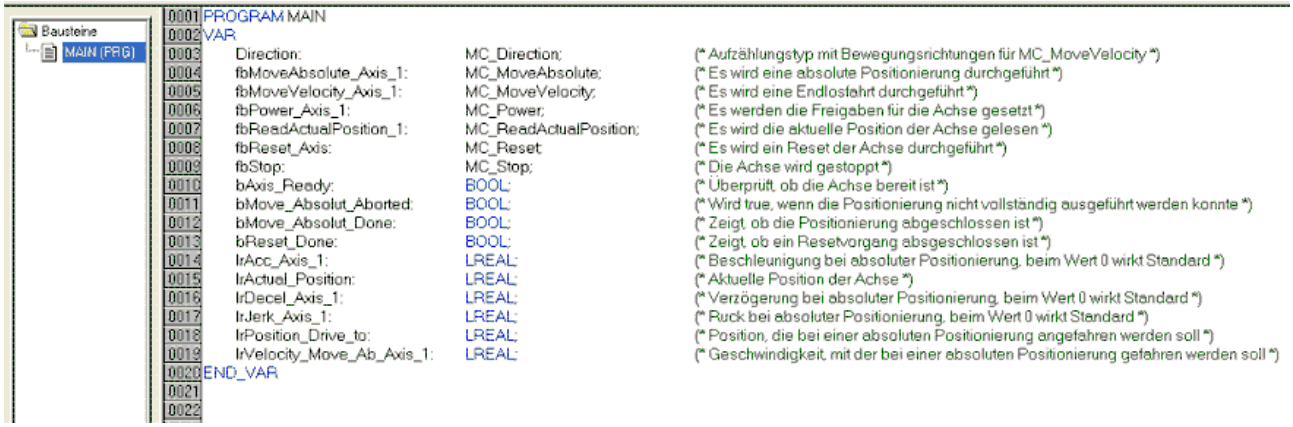
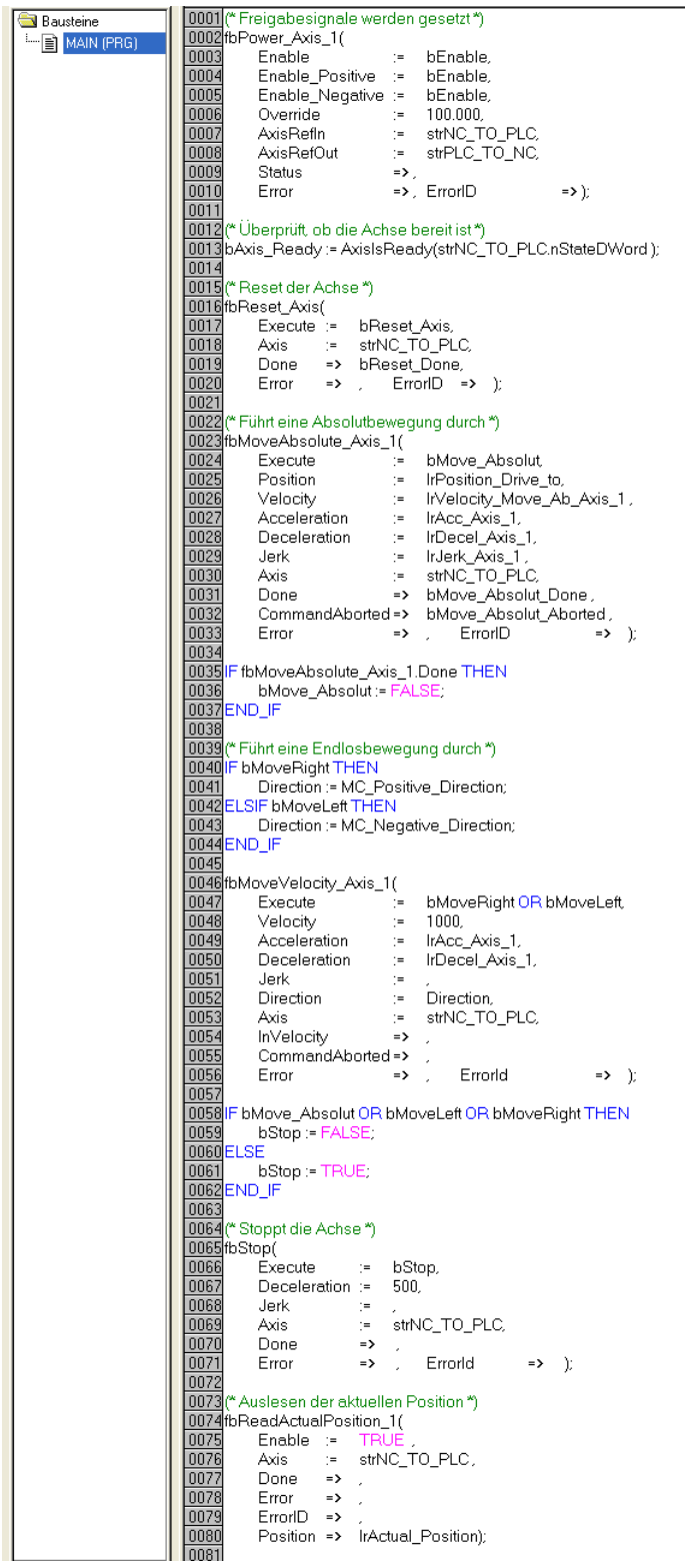


Fig. 137: Local variables

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




```
0001 (* Freigabesignale werden gesetzt *)
0002 fbPower_Axis_1(
0003     Enable              :=  bEnable,
0004     Enable_Positive     :=  bEnable,
0005     Enable_Negative     :=  bEnable,
0006     Override            :=  100.000,
0007     AxisRefIn           :=  strNC_TO_PLC,
0008     AxisRefOut          :=  strPLC_TO_NC,
0009     Status              =>  ,
0010     Error               =>  , ErrorID           => );
0011
0012 (* Überprüft, ob die Achse bereit ist *)
0013 bAxis_Ready := AxisIsReady(strNC_TO_PLC.nStateDWord );
0014
0015 (* Reset der Achse *)
0016 fbReset_Axis(
0017     Execute :=  bReset_Axis,
0018     Axis    :=  strNC_TO_PLC,
0019     Done    =>  bReset_Done,
0020     Error   =>  ,   ErrorID =>  );
0021
0022 (* Führt eine Absolutbewegung durch *)
0023 fbMoveAbsolute_Axis_1(
0024     Execute             :=  bMove_Absolut,
0025     Position            :=  lrPosition_Drive_to,
0026     Velocity            :=  lrVelocity_Move_Ab_Axis_1 ,
0027     Acceleration        :=  lrAcc_Axis_1,
0028     Deceleration        :=  lrDecel_Axis_1,
0029     Jerk                :=  lrJerk_Axis_1 ,
0030     Axis                :=  strNC_TO_PLC,
0031     Done                =>  bMove_Absolut_Done ,
0032     CommandAborted      =>  bMove_Absolut_Aborted ,
0033     Error               =>  ,   ErrorID         =>  );
0034
0035 IF fbMoveAbsolute_Axis_1.Done THEN
0036     bMove_Absolut := FALSE;
0037 END_IF
0038
0039 (* Führt eine Endlosbewegung durch *)
0040 IF bMoveRight THEN
0041     Direction := MC_Positive_Direction;
0042 ELSIF bMoveLeft THEN
0043     Direction := MC_Negative_Direction;
0044 END_IF
0045
0046 fbMoveVelocity_Axis_1(
0047     Execute             :=  bMoveRight OR bMoveLeft,
0048     Velocity            :=  1000,
0049     Acceleration        :=  lrAcc_Axis_1,
0050     Deceleration        :=  lrDecel_Axis_1,
0051     Jerk                :=  ,
0052     Direction           :=  Direction,
0053     Axis                :=  strNC_TO_PLC,
0054     InVelocity          =>  ,
0055     CommandAborted      =>  ,
0056     Error               =>  ,   ErrorId         =>  );
0057
0058 IF bMove_Absolut OR bMoveLeft OR bMoveRight THEN
0059     bStop := FALSE;
0060 ELSE
0061     bStop := TRUE;
0062 END_IF
0063
0064 (* Stoppt die Achse *)
0065 fbStop(
0066     Execute     :=  bStop,
0067     Deceleration :=  500,
0068     Jerk        :=  ,
0069     Axis        :=  strNC_TO_PLC,
0070     Done        =>  ,
0071     Error       =>  ,   ErrorId     =>  );
0072
0073 (* Auslesen der aktuellen Position *)
0074 fbReadActualPosition_1(
0075     Enable  :=  TRUE ,
0076     Axis    :=  strNC_TO_PLC ,
0077     Done    =>  ,
0078     Error   =>  ,
0079     ErrorID =>  ,
0080     Position =>  lrActual_Position);
0081
```

Fig. 138: 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.

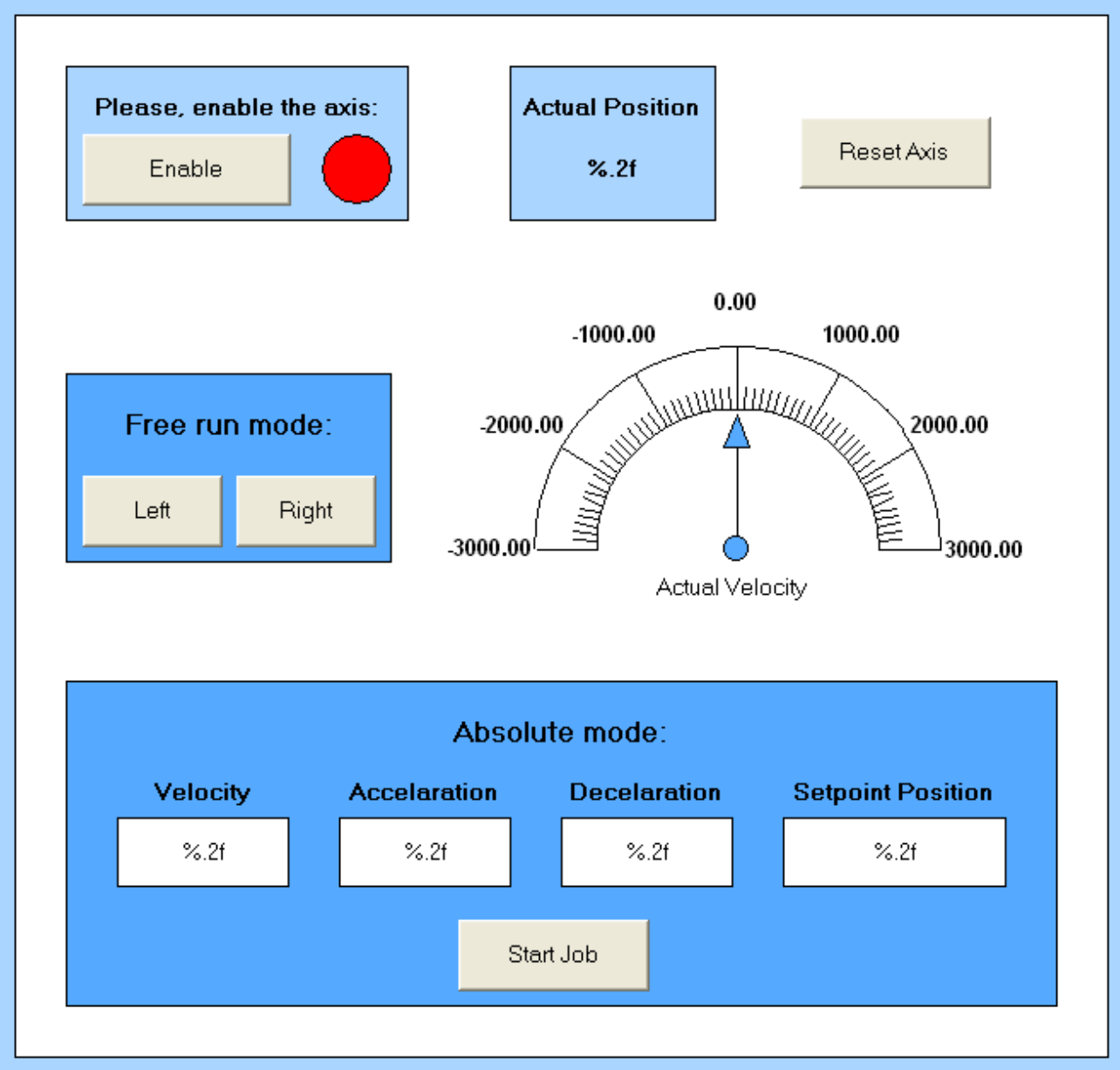


Fig. 139: Visualization

Note	**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.

6.2.6 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 [▶ 173] / DS402 [▶ 151]) (see Fig. *DS402 State Machine*). The respective status messages are output in the status word (MDP742 [▶ 171] / DS402 [▶ 151]).

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

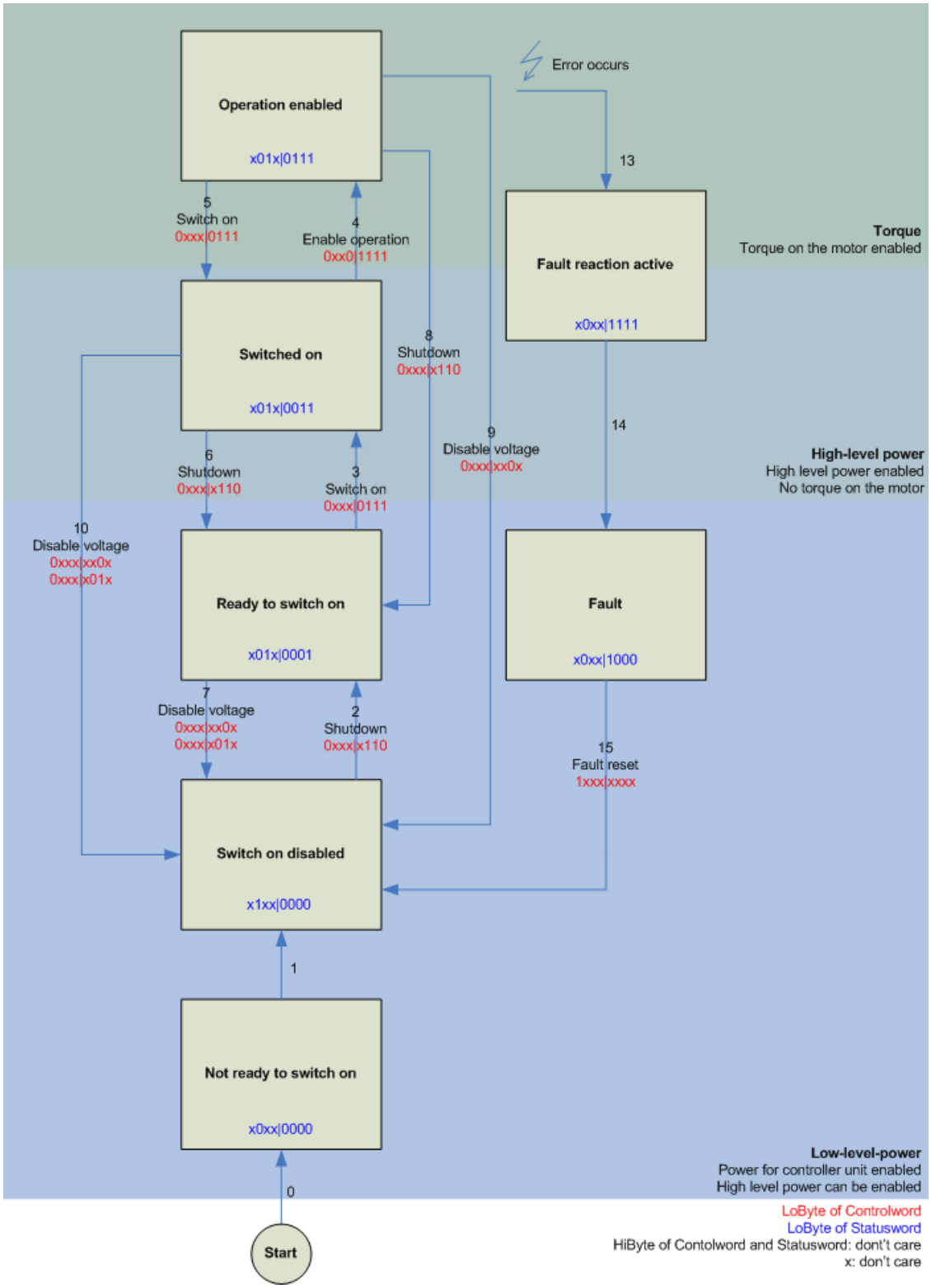


Fig. 140: DS402 State Machine

CST - cyclic synchronous torque

Select *Cyclic synchronous torque mode* in index 0x7010:03 [▶ 173] *Modes of operation* (MDP) or 0x6060:0 [▶ 151] *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 [▶ 137] or DS402 process data [▶ 141]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 171] *Modes of operation display* (MDP) or 0x6061:0 [▶ 151] *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 [▶ 168] *Rated current* (MDP) or index 0x6075:0 [▶ 152] *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 [▶ 173] *Modes of operation* (MDP) or 0x6060:0 [▶ 151]*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 [▶ 137] or DS402 process data [▶ 141]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 171] *Modes of operation display* (MDP) or 0x6061:0 [▶ 151] *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 [▶ 168] *Rated current* (MDP) or index 0x6075:0 [▶ 152] *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 [▶ 173] *Modes of operation* (MDP) or index 0x6060:0 [▶ 151] *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 [▶ 137] or DS402 process data [▶ 141]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 171] *Modes of operation display* (MDP) or 0x6061:0 [▶ 151] *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 [▶ 173] (MDP) or 0x60FF:0 [▶ 154] (DS402). The constant value *Velocity encoder resolution* in CoE object 0x9010:14 [▶ 176] (MDP) or 0x6090:0 [▶ 153] (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 [▶ 173] *Modes of operation* (MDP) or index 0x6060:0 [▶ 151] *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 [▶ 137] or DS402 process data [▶ 141]). The configuration then has to be reloaded in order to activate the selection.
Index 0x6010:03 [▶ 171] *Modes of operation display* (MDP) or 0x6061:0 [▶ 151] *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 [▶ 173] (MDP) or 0x607A:0 [▶ 153] (DS402) to which the motor is to drive. The calculated scaling factor [▶ 100] 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.

6.2.7 Settings for the automatic configuration

(Master TwinCAT 2.11 R3)

The EL72x1-0010 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*).

Note	**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 0x8001:01 (0x2018:01 [▶ 149], DS402 Profile) *Enable autoconfig*.
- In the index 0x8001:02 (0x2018:02 [▶ 149], DS402 Profile) *Reconfig identical motor*, the user can decide in the case of replacing an identical motor whether the terminal should automatically reconfigure 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 0x8001:03 (0x2018:03 [▶ 149], DS402 Profile) *Reconfig non-identical motor*, the user can decide in the case of replacing a non-identical motor whether the terminal should automatically reconfigure the motor (setting = *true*) or whether the motor should be operated with the stored settings (setting = *false*).

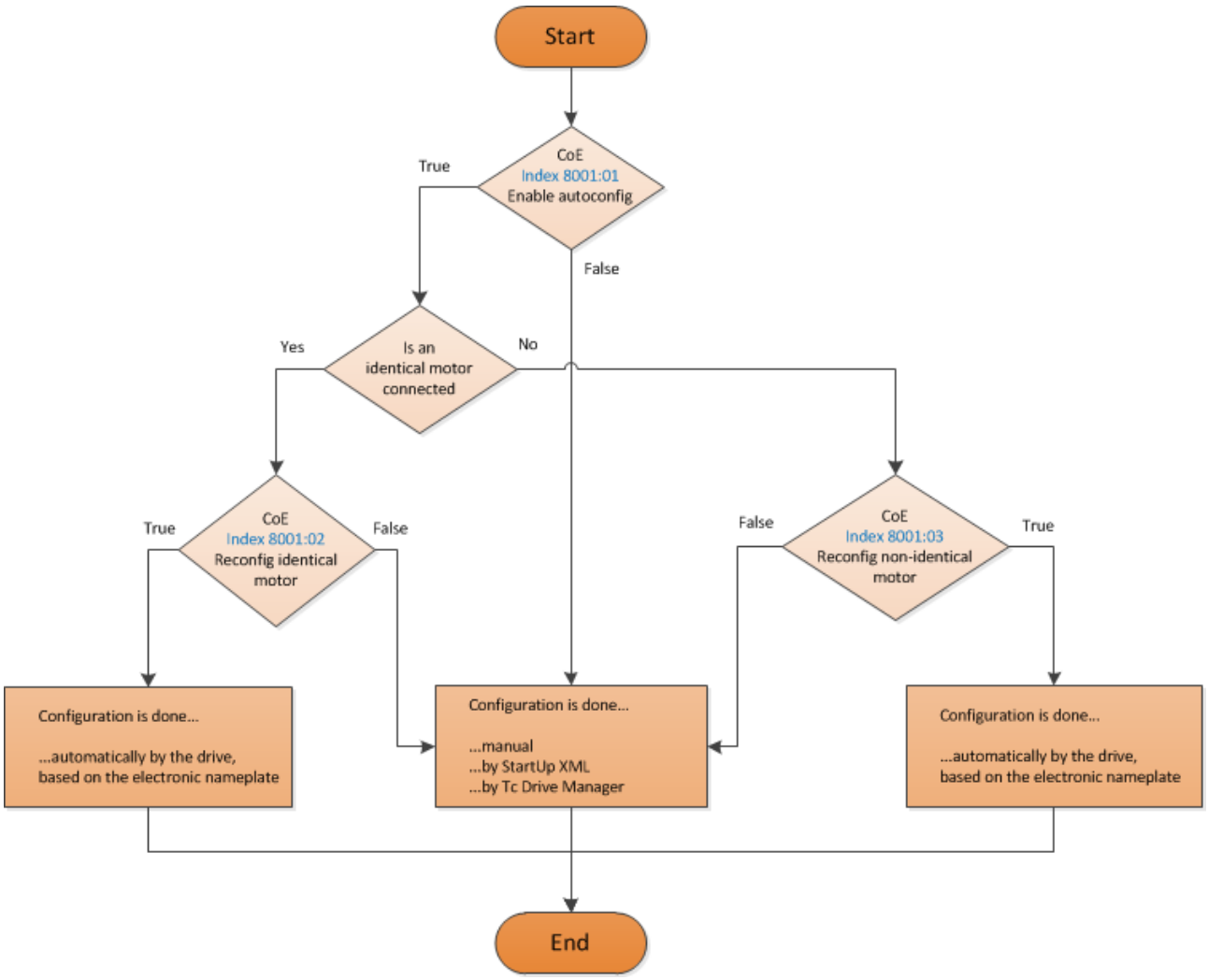


Fig. 141: Flow chart for the automatic configuration

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 [▶ 166]	2002:12 [▶ 147]	Current loop integral time	is calculated according to the symmetrical optimum
8010:13 [▶ 166]	2002:13 [▶ 147]	Current loop proportional gain	is calculated according to the symmetrical optimum
8011:11 [▶ 168]	2003:11 [▶ 148]	Max. current	is adopted directly from the electronic type plate of the connected motor
8011:12 [▶ 168]	2003:12 [▶ 148]	Rated current	is adopted directly from the electronic type plate of the connected motor
8011:13 [▶ 168]	2003:13 [▶ 148]	Motor pole pairs	is adopted directly from the electronic type plate of the connected motor
8011:15 [▶ 168]	2003:15 [▶ 148]	Commutation offset	is always set to -90°
8011:16 [▶ 168]	2003:16 [▶ 148]	Torque constant	is adopted directly from the electronic type plate of the connected motor
8011:18 [▶ 168]	2003:18 [▶ 148]	Rotor moment of inertia	is adopted directly from the electronic type plate of the connected motor
8011:19 [▶ 168]	2003:19 [▶ 148]	Winding inductance	is adopted directly from the electronic type plate of the connected motor
8011:1B [▶ 168]	2003:1B [▶ 148]	Motor speed limitation	Calculation of the max. speed of the connected motor
8011:2B [▶ 168]	2003:2B [▶ 148]	Motor temperature warn level	is adopted directly from the electronic type plate of the connected motor
8011:2C [▶ 168]	2003:2C [▶ 148]	Motor temperature error level	is adopted directly from the electronic type plate of the connected motor
8011:2D [▶ 168]	2003:2D [▶ 148]	Motor thermal time constant	is adopted directly from the electronic type plate of the connected motor
8012:11 [▶ 169]	2004:11 [▶ 149]	Release delay	is adopted directly from the electronic type plate of the connected motor
8012:12 [▶ 169]	2004:12 [▶ 149]	Application delay	is adopted directly from the electronic type plate of the connected motor
8012:14 [▶ 169]	2004:14 [▶ 149]	Brake moment of inertia	is adopted directly from the electronic type plate of the connected motor

Also see about this

- EL72x1-0010-MDP742 - Object description and parameterization [▶ 165]

6.2.8 Configuring the limit switch

Software end position monitoring

The TwinCAT NC can be used to set software end position monitoring for the EL72x1-0010 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.

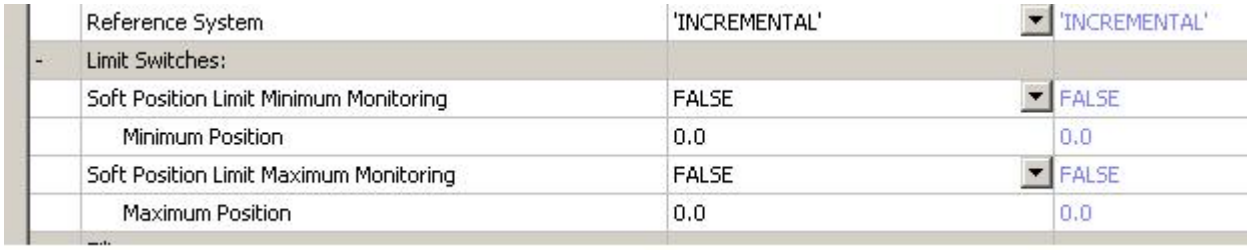


Fig. 142: Pull-down menu for activating end position monitoring

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.

6.2.9 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

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

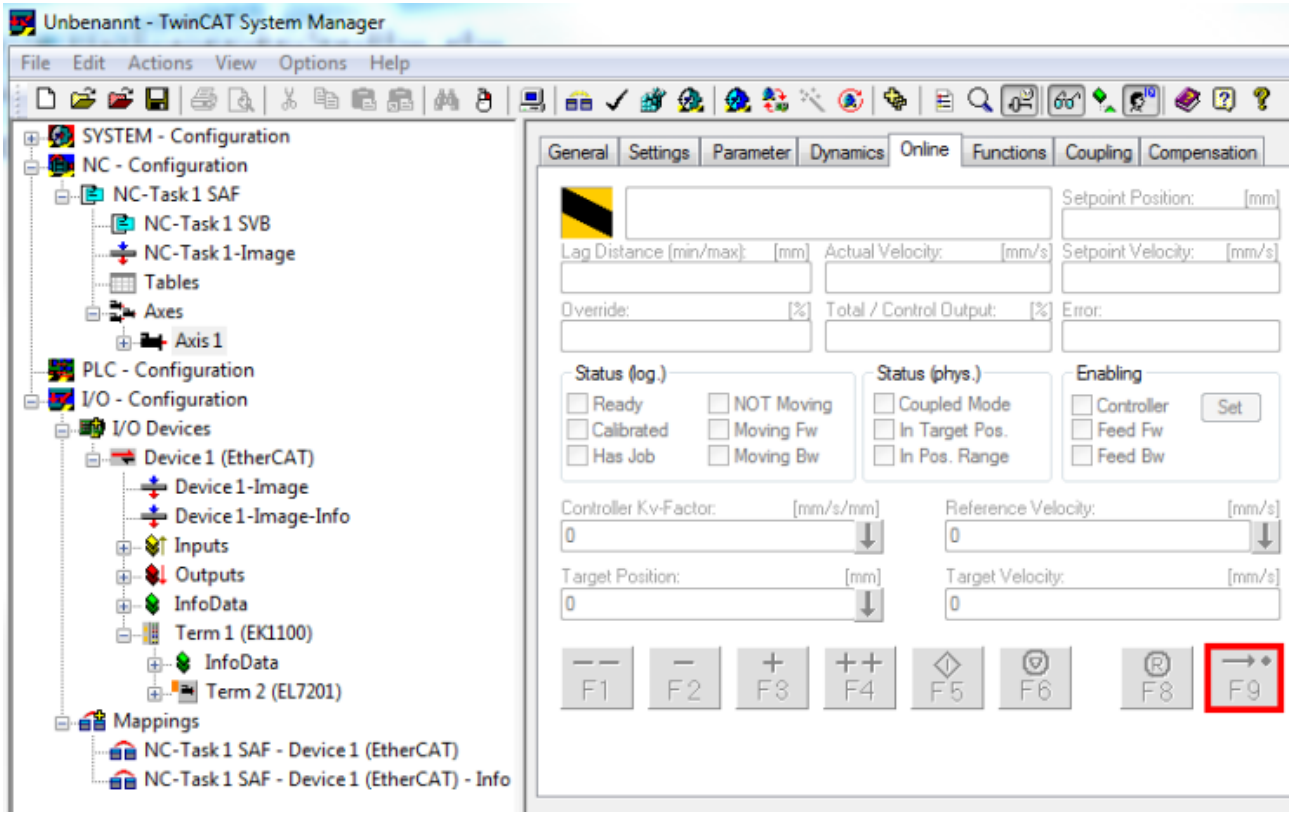


Fig. 143: Online homing in the NC

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. 144: 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.

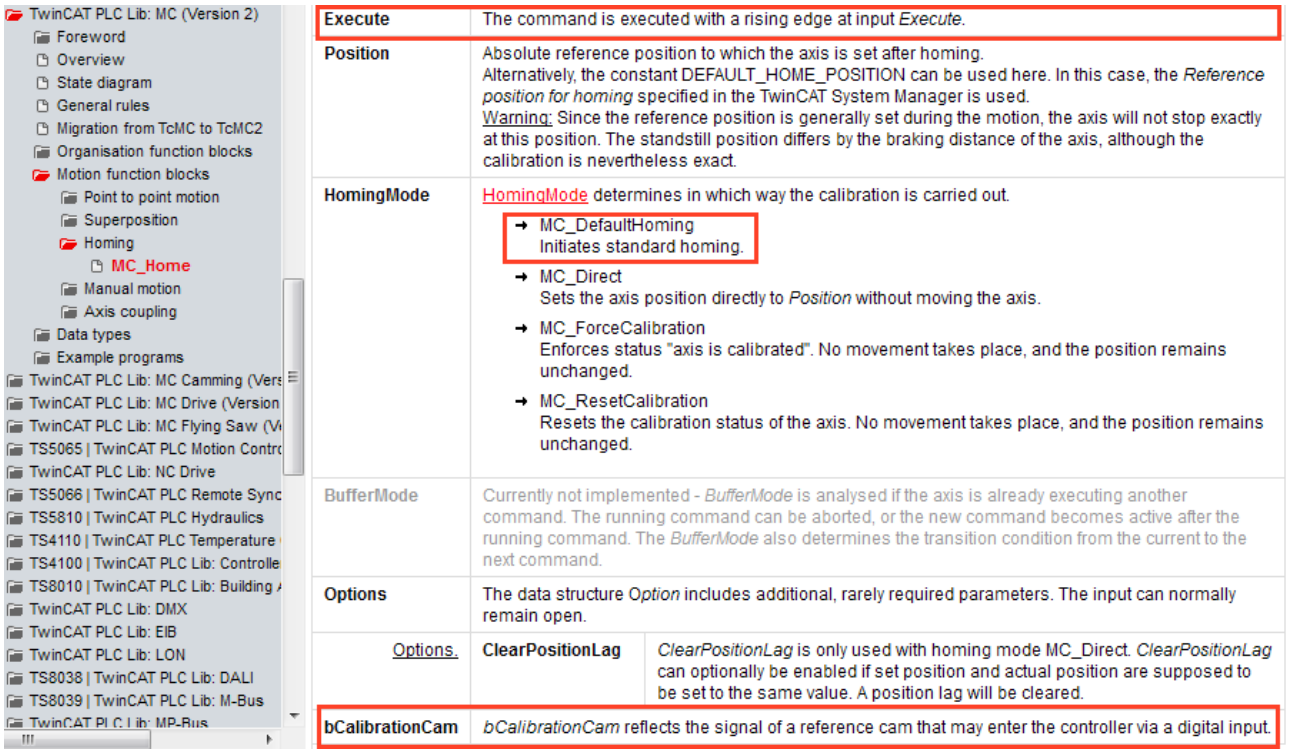


Fig. 145: Extraction from the functional description for MC_Home

Reference modes

- The EL72x1-0010 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.

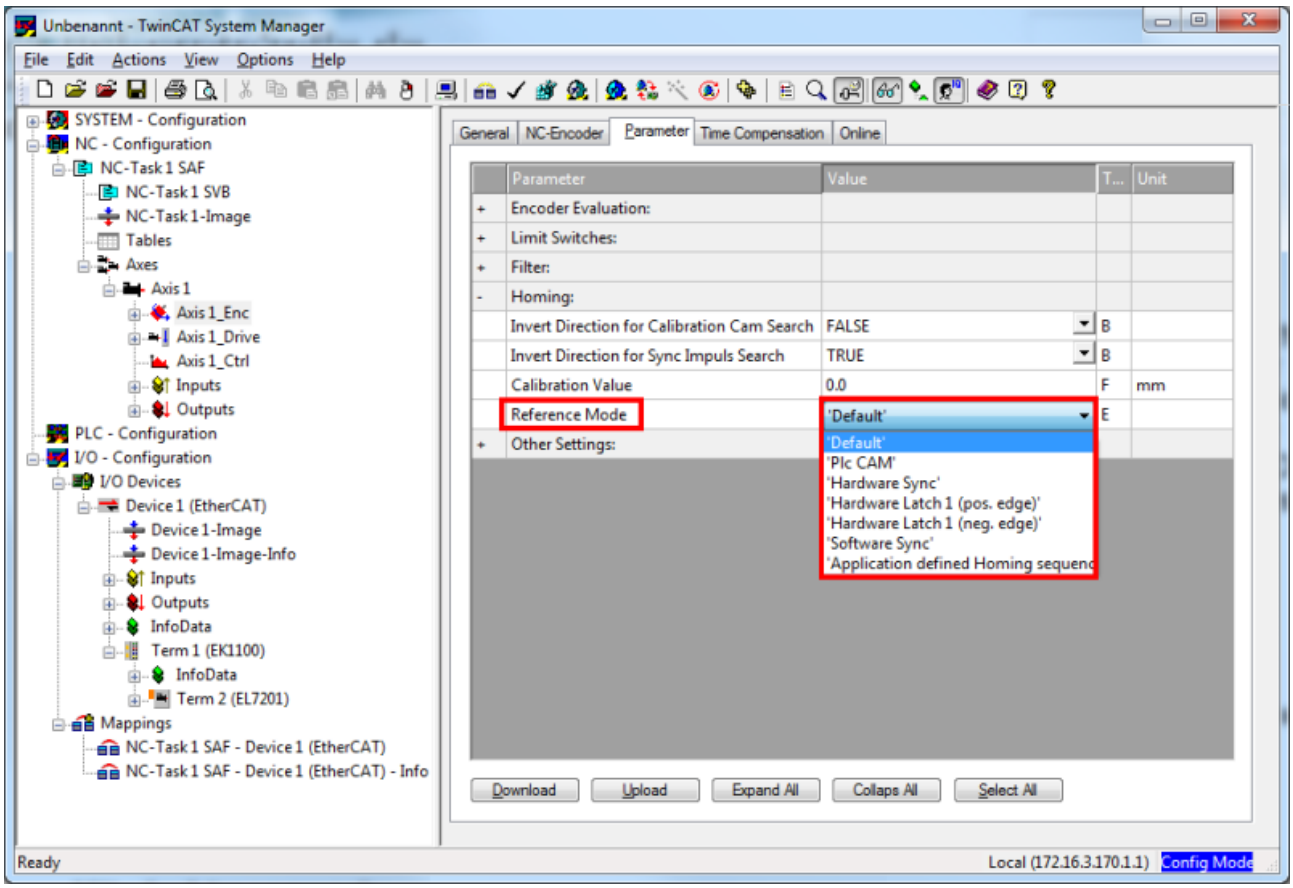


Fig. 146: Selection of the reference modes in the NC

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

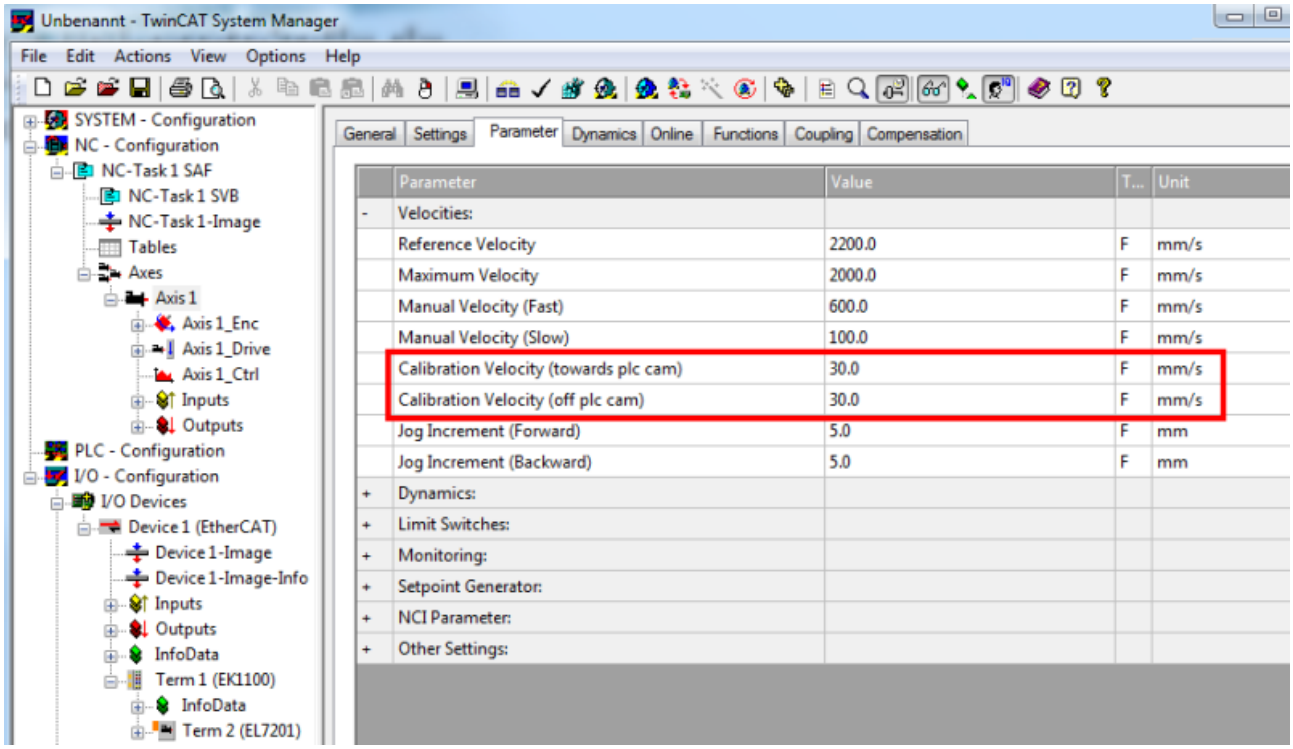


Fig. 147: Setting the reference velocity

6.2.10 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 EL72x1-0010 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 [▶ 137] tab (see Fig. *Touch Probe inputs* and Fig. *Touch Probe outputs*).

The EL72x1-0010 has two 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.

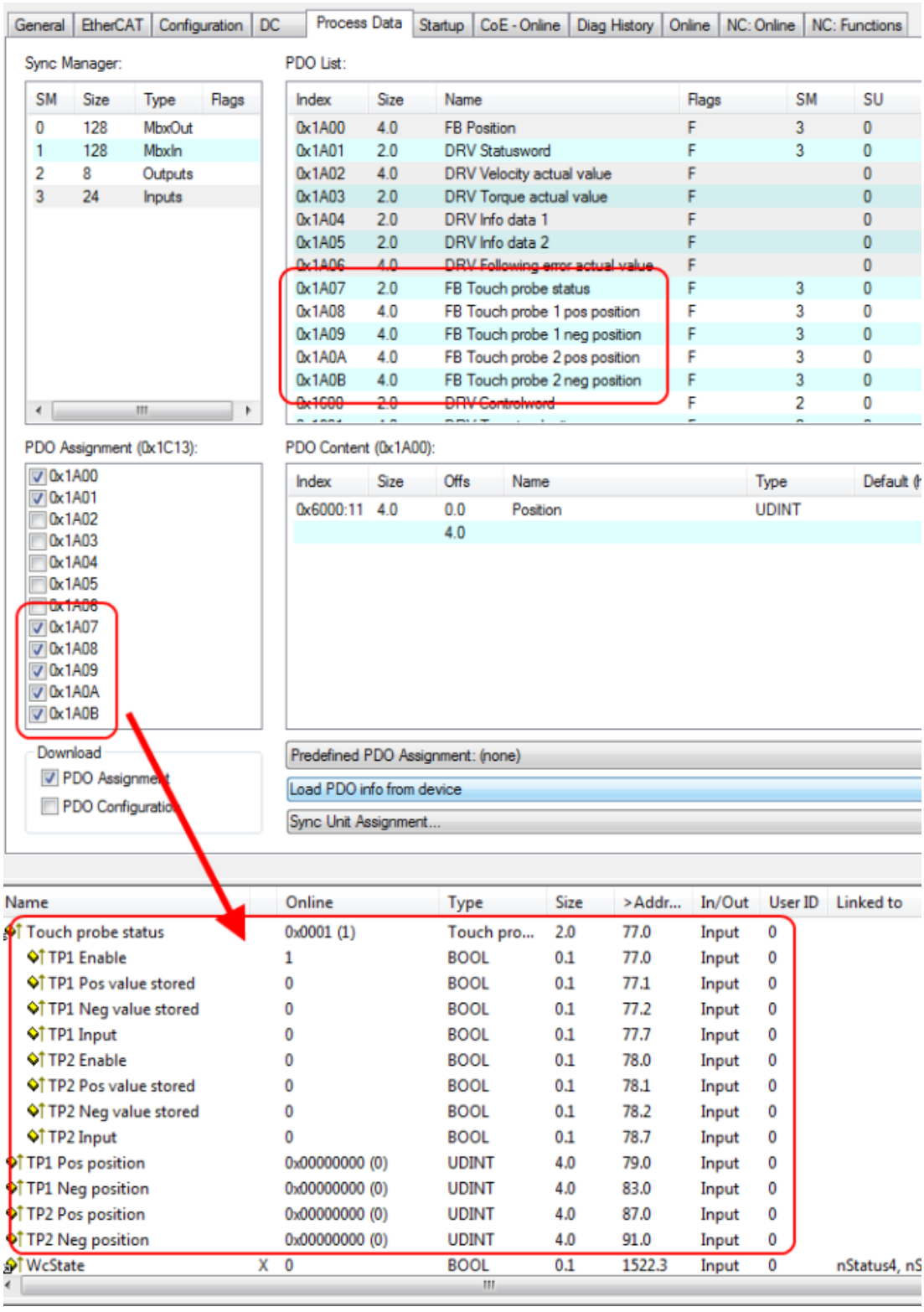


Fig. 148: Touch Probe inputs

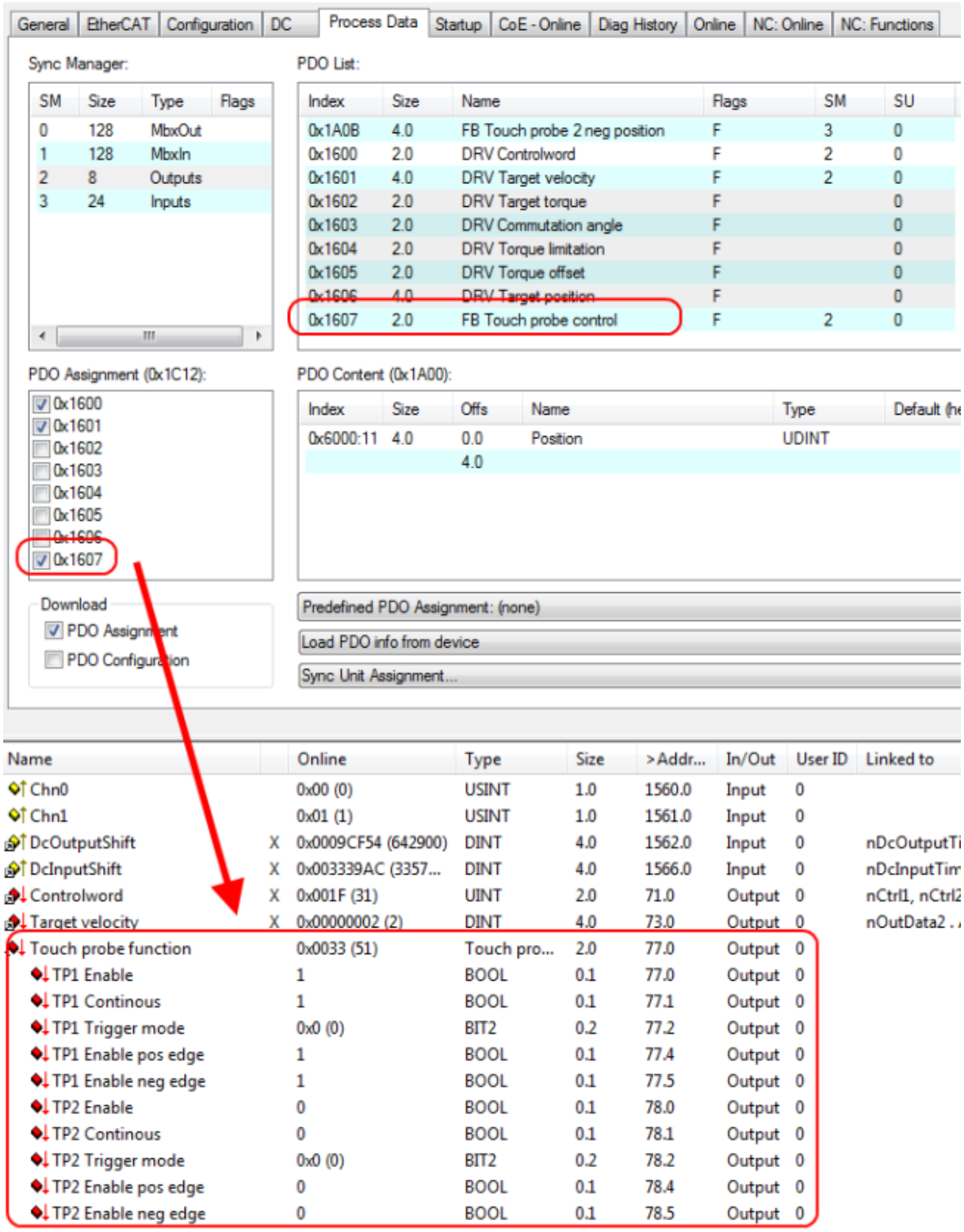


Fig. 149: Touch Probe outputs

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 EL72x1-0010.
- 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.

6.3 Modes of operation

6.3.1 Overview

Operating modes CST, CSTCA, CSV and CSP are supported. The operating mode is set in the CoE list in index 0x7010:03 Modes of operation (MDP) [▶ 173] or index 0x6060:0 Modes of operation (DS402) [▶ 151]. 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 [▶ 121] - cyclic synchronous velocity (velocity control)

In CSV mode the EL72x1-0010 operates with the cyclic velocity interface. A defined velocity can be set via the *Target velocity* variable.

CST [▶ 125] - cyclic synchronous torque (torque control)

In CST mode the EL72x1-0010 operates in the cyclic torque interface. A defined torque can be set via the *Target torque* variable.

CSTCA [▶ 128] - 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 [▶ 132] - cyclic synchronous position (position control)

In the CSP operating mode the EL72x1-0010 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 [▶ 109].

6.3.2 CSV

Modes of operation

CSV - cyclic synchronous velocity (velocity control)

In CSV mode the EL72x1-0010 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 [▶ 52] - manual or - Online scan [▶ 58].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 85].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 93].
- Set the mode of operation in the CoE directory to *Cyclic synchronous velocity mode (CSV), Fig. Selection of the mode of operation*.

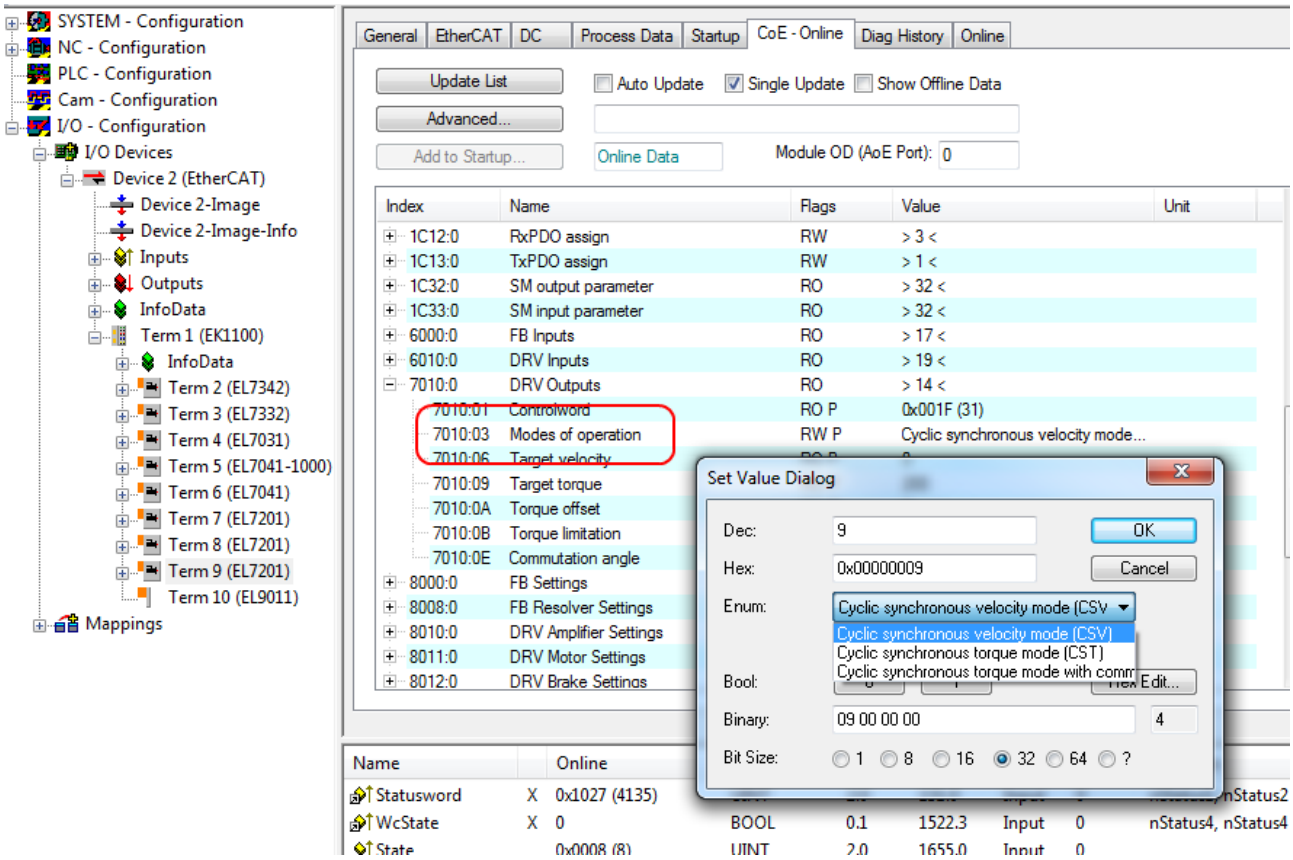


Fig. 150: Selection of the mode of operation

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

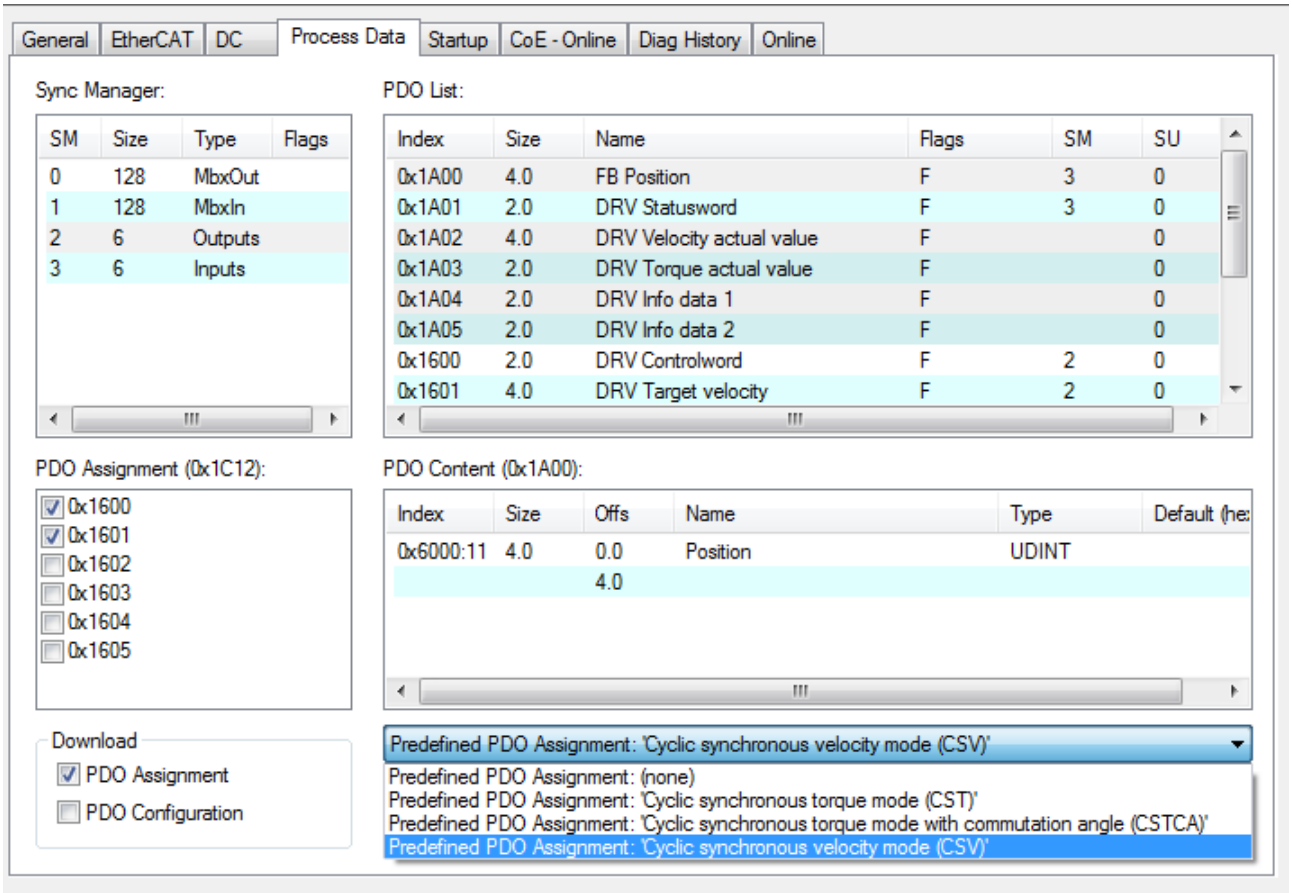


Fig. 151: Selecting a predefined PDO assignment

- 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.

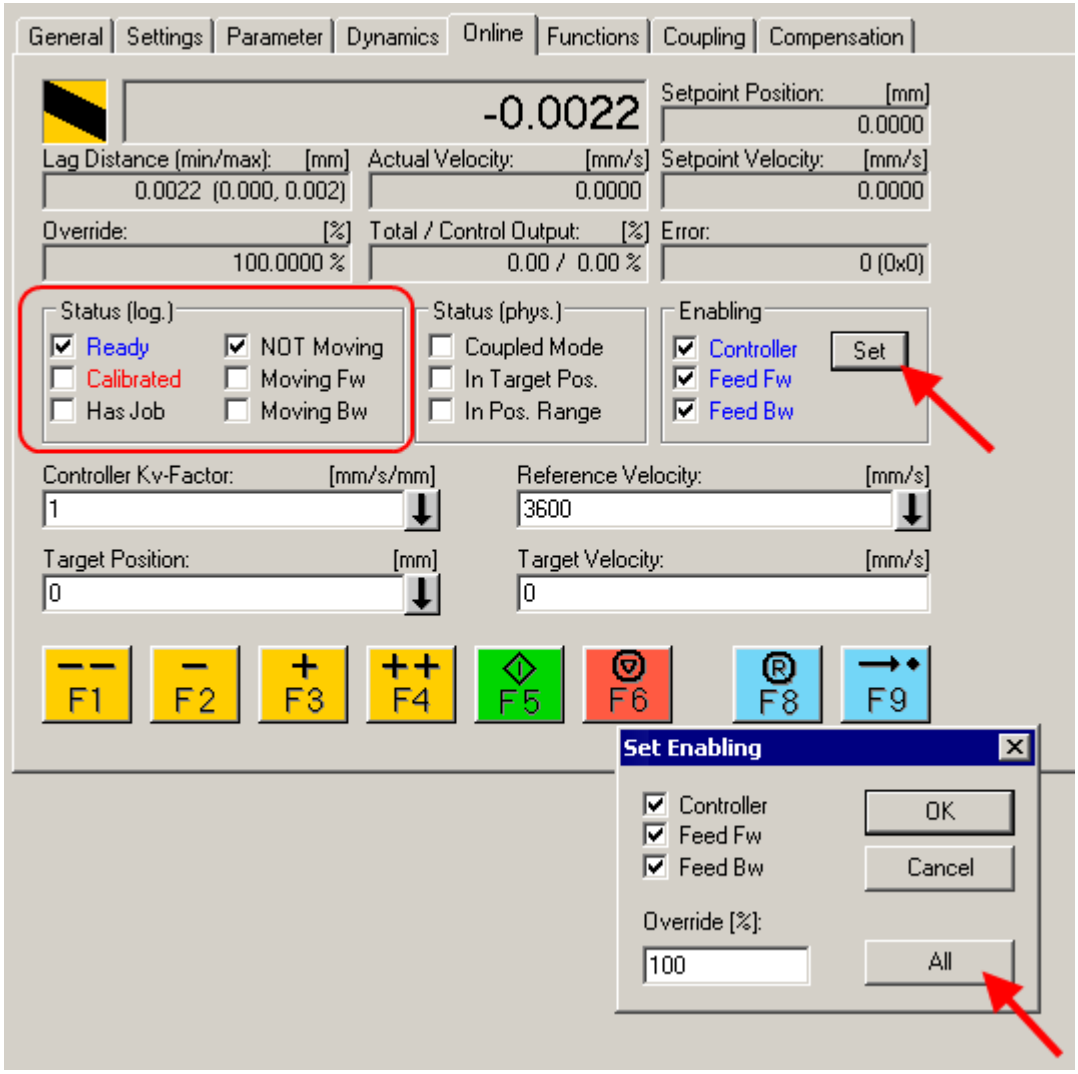


Fig. 152: Set enables

◦ 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 [▶ 109].

- The cyclic variable *Target velocity* (Fig. *Torque specification*) can be used to specify a defined velocity. The value in the index 0x9010:14 *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. 153: Torque specification

6.3.3 CST

CST - cyclic synchronous torque (torque control)

In CST mode the EL72x1-0010 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 [▶ 52] - manual or - Online scan [▶ 58].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 85].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 93].
- Set the mode of operation in the CoE directory to *Cyclic synchronous torque mode (CST)*, Fig. *Selection of the mode of operation*.

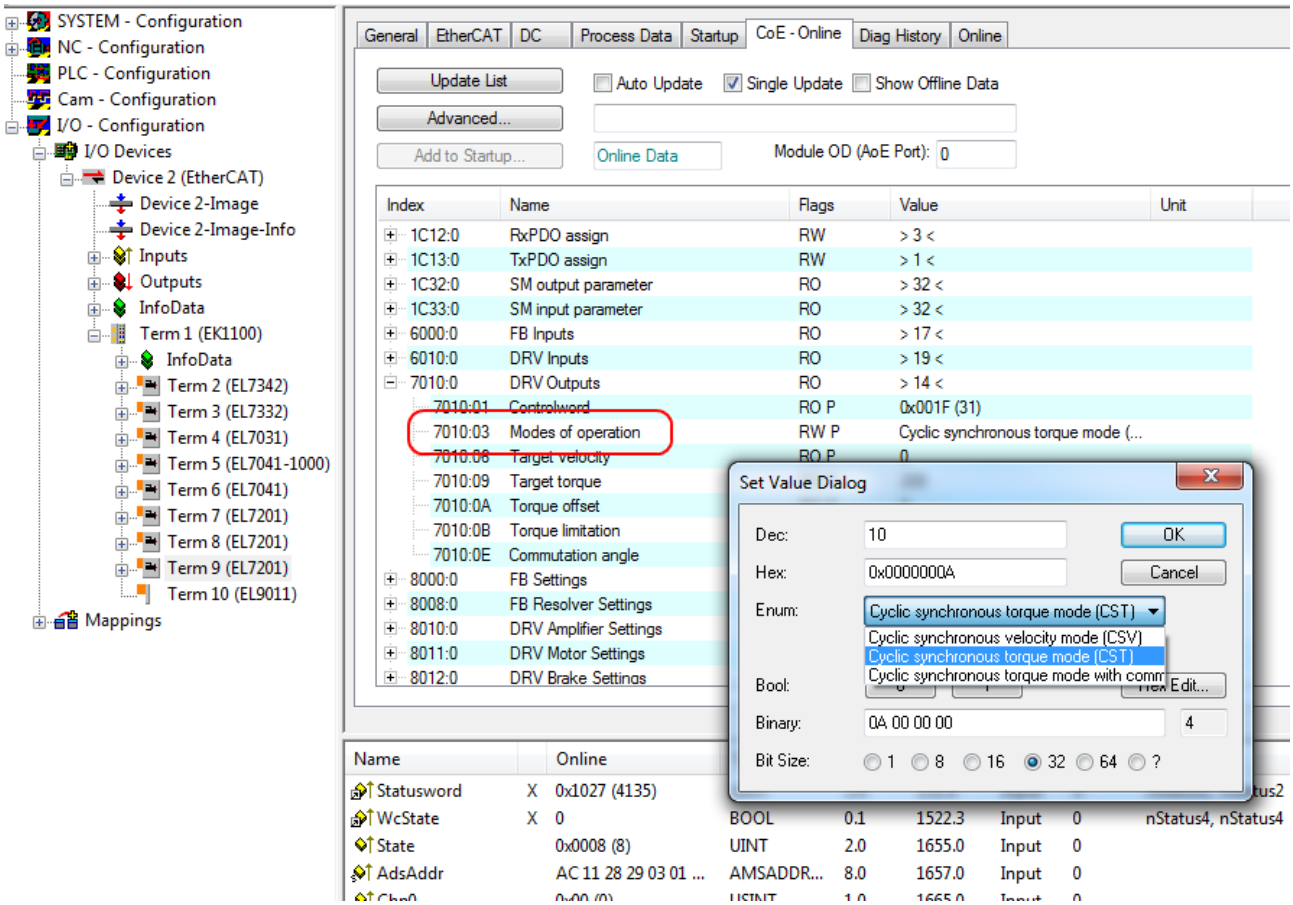


Fig. 154: Selection of the mode of operation

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

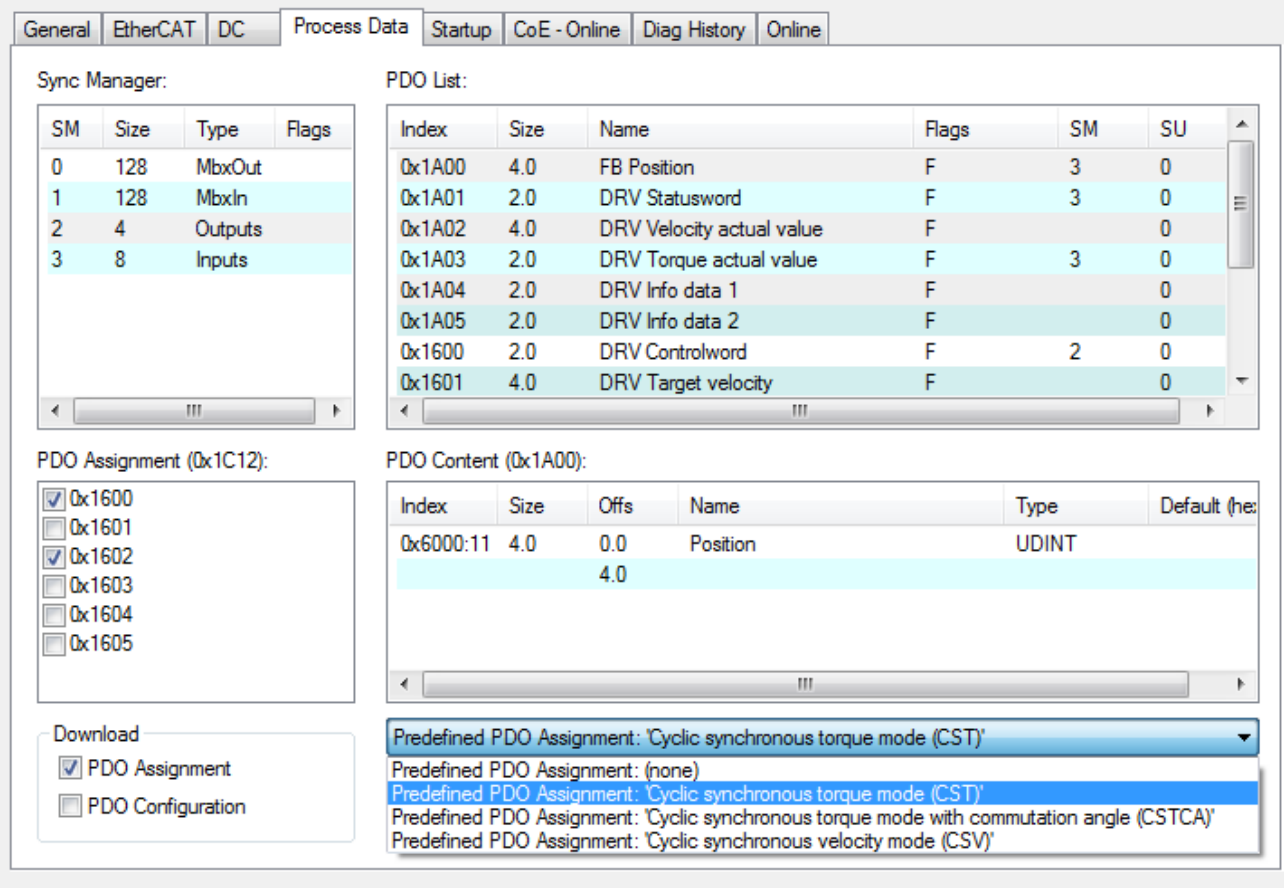


Fig. 155: Selecting a predefined PDO assignment

- 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.

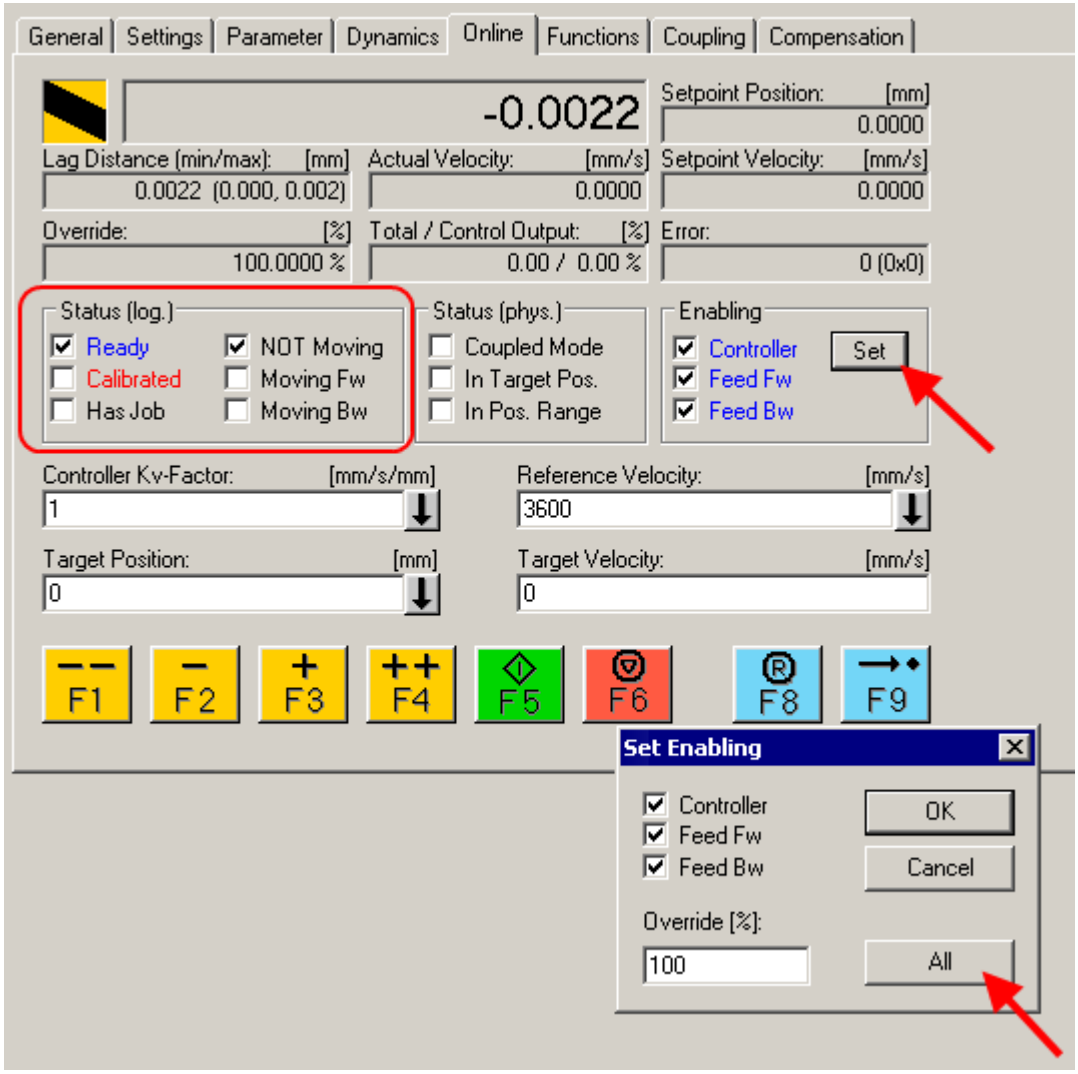
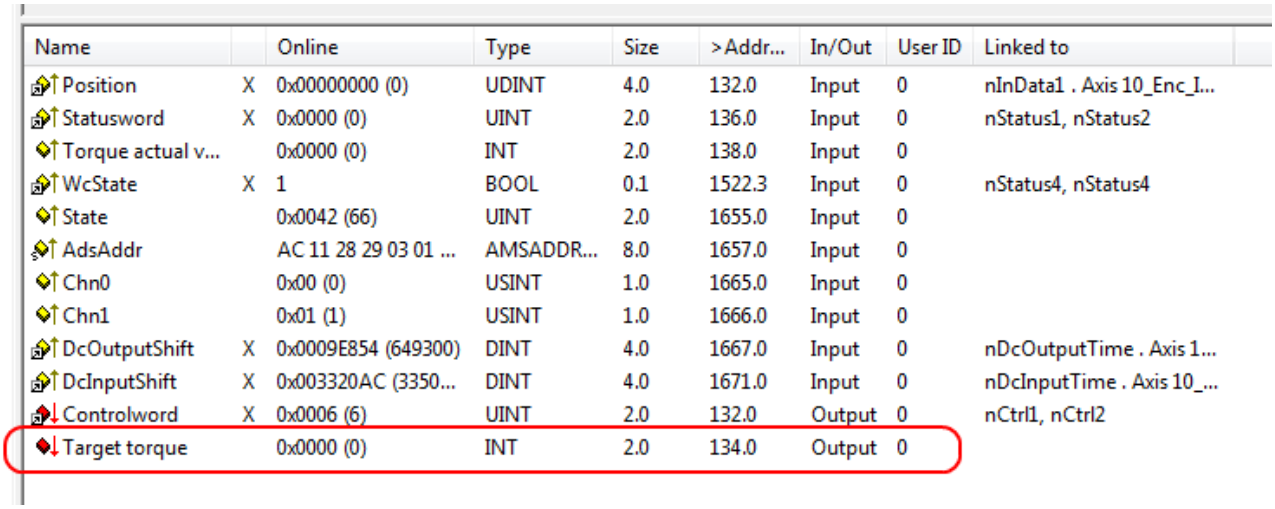


Fig. 156: Set enables

◦ 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 [▶ 109].

- 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 [▶ 168] (*rated current*).

$$M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot torque\ constant\ (datasheet\ motor)$$



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. 157: Torque specification

6.3.4 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 [▶ 52] - manual or - Online scan [▶ 58].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 85].
- Import the motor XML file into the Startup directory as described in the chapter Settings in the CoE [▶ 93].
- 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*

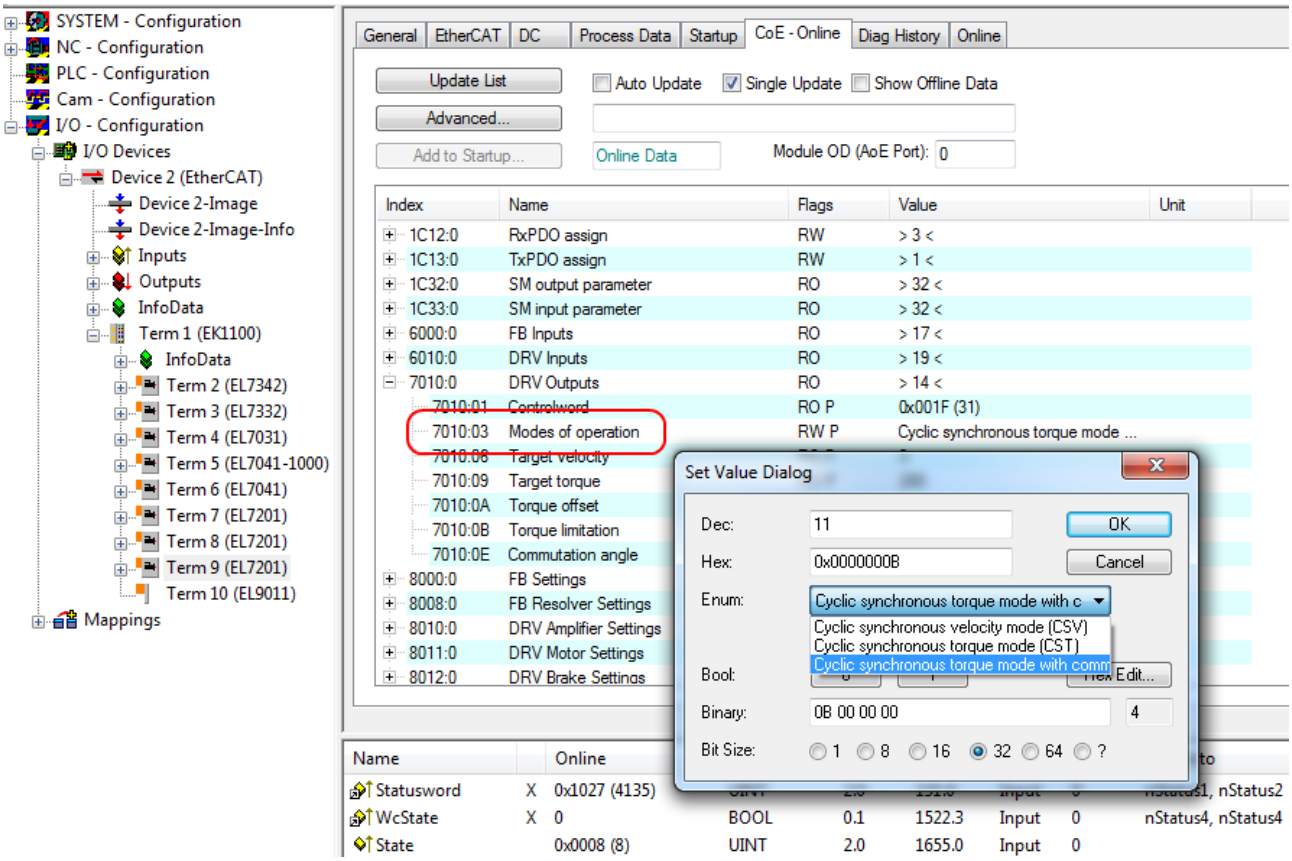


Fig. 158: Selection of the mode of operation

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

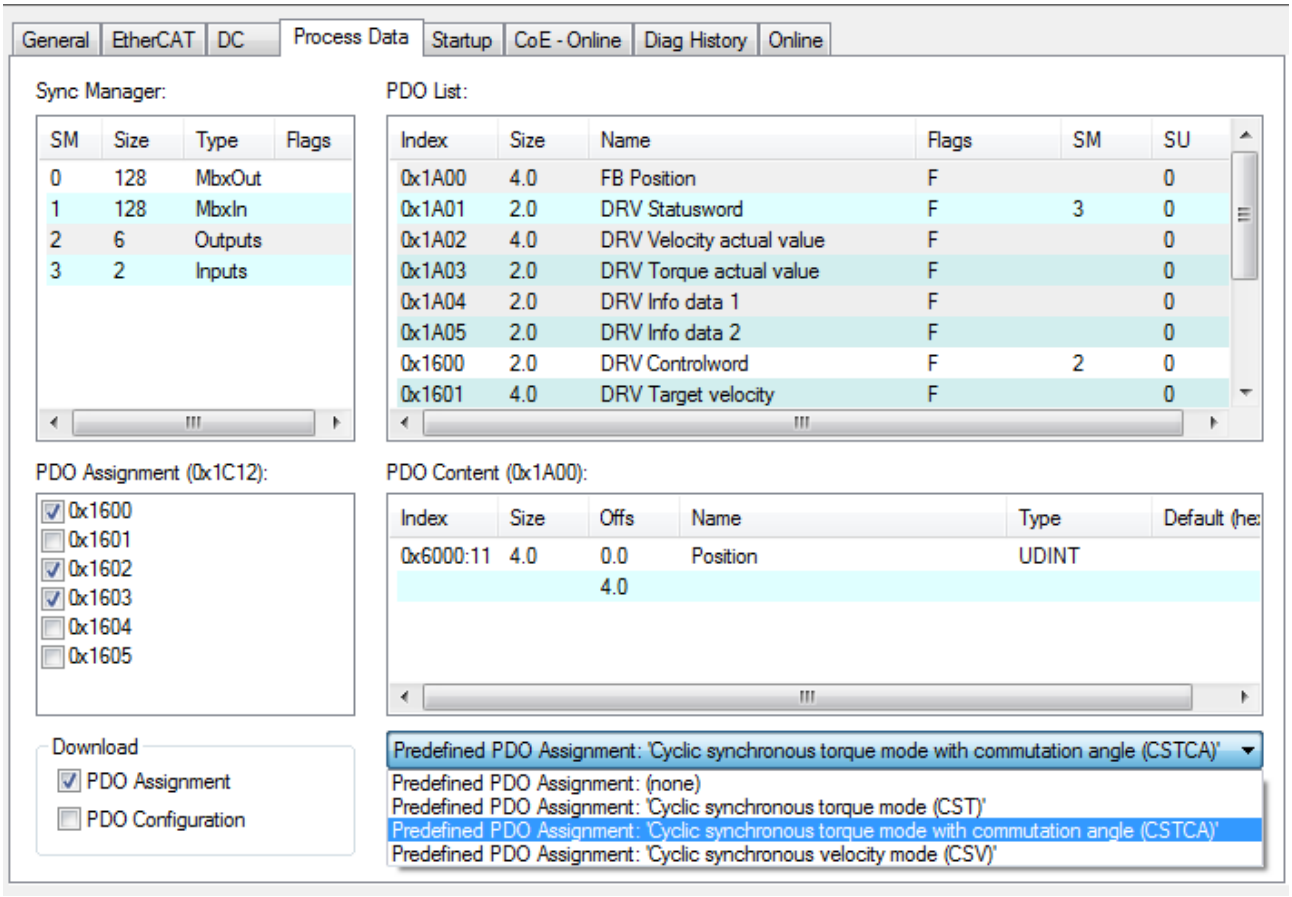


Fig. 159: Selecting a predefined PDO assignment

- 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.

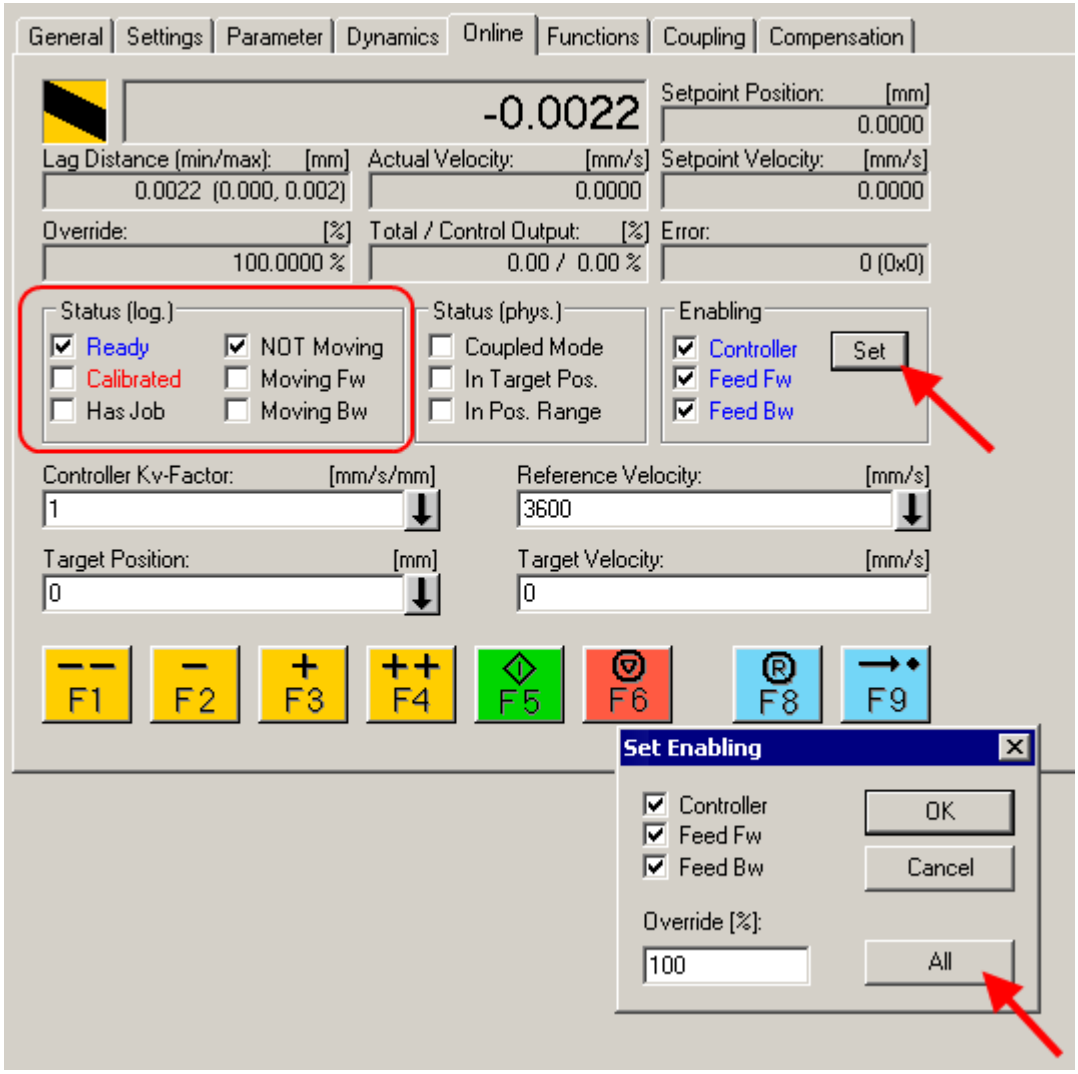


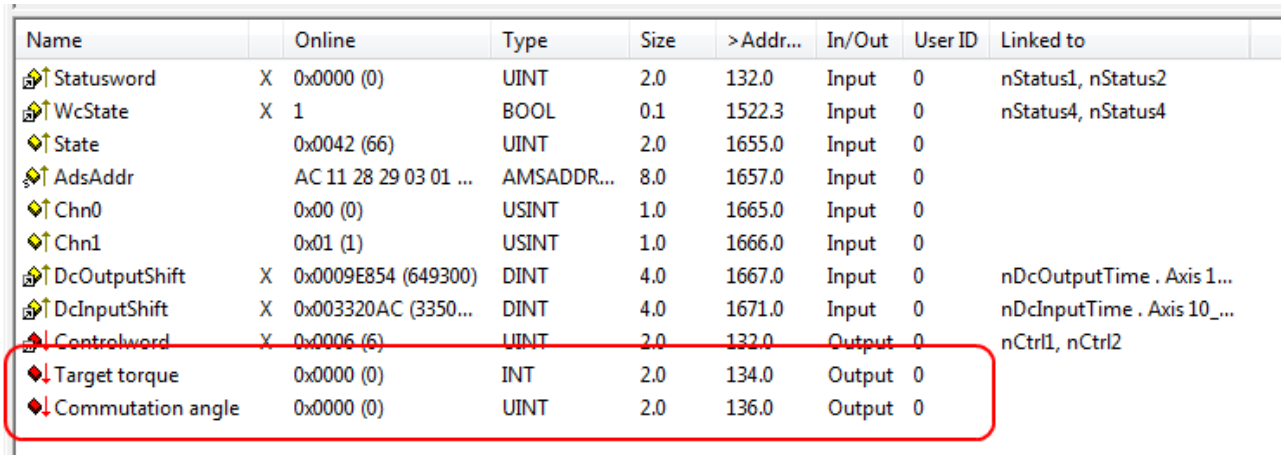
Fig. 160: Set enables

- 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 [▶ 109].

- 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 [▶ 168] (*rated current*).

$$M = \frac{\textit{Torque actual value}}{1000} \cdot \frac{\textit{rated current}}{\sqrt{2}} \cdot \textit{torque constant (datasheet motor)}$$

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. 161: Specification of torque and commutation angle

6.3.5 CSP

CSP - cyclic synchronous position (position control)

In the CSP operating mode the EL72x1-0010 operates in the cyclic position interface. A defined position can be set via the *Target position* variable.

Note	**Minimum cycle time** The cycle time in CSP mode should be at least 250 µs.

Step-by-Step

- Add the terminal to the configuration as described in the chapter TwinCAT configuration settings [▶ 52] - manual or - Online scan [▶ 58].
- Link the terminal with the NC as described in the chapter Integration in the NC configuration [▶ 85].
- Configure the motor with the help of the automatic configuration [▶ 112] using the Drive Manager [▶ 88] or import the motor XML file into the Start-up directory as described in the chapter Settings in the CoE [▶ 93].
- Set the mode of operation in the CoE directory to *Cyclic synchronous position mode (CSP), Fig. Selection of the mode of operation*.

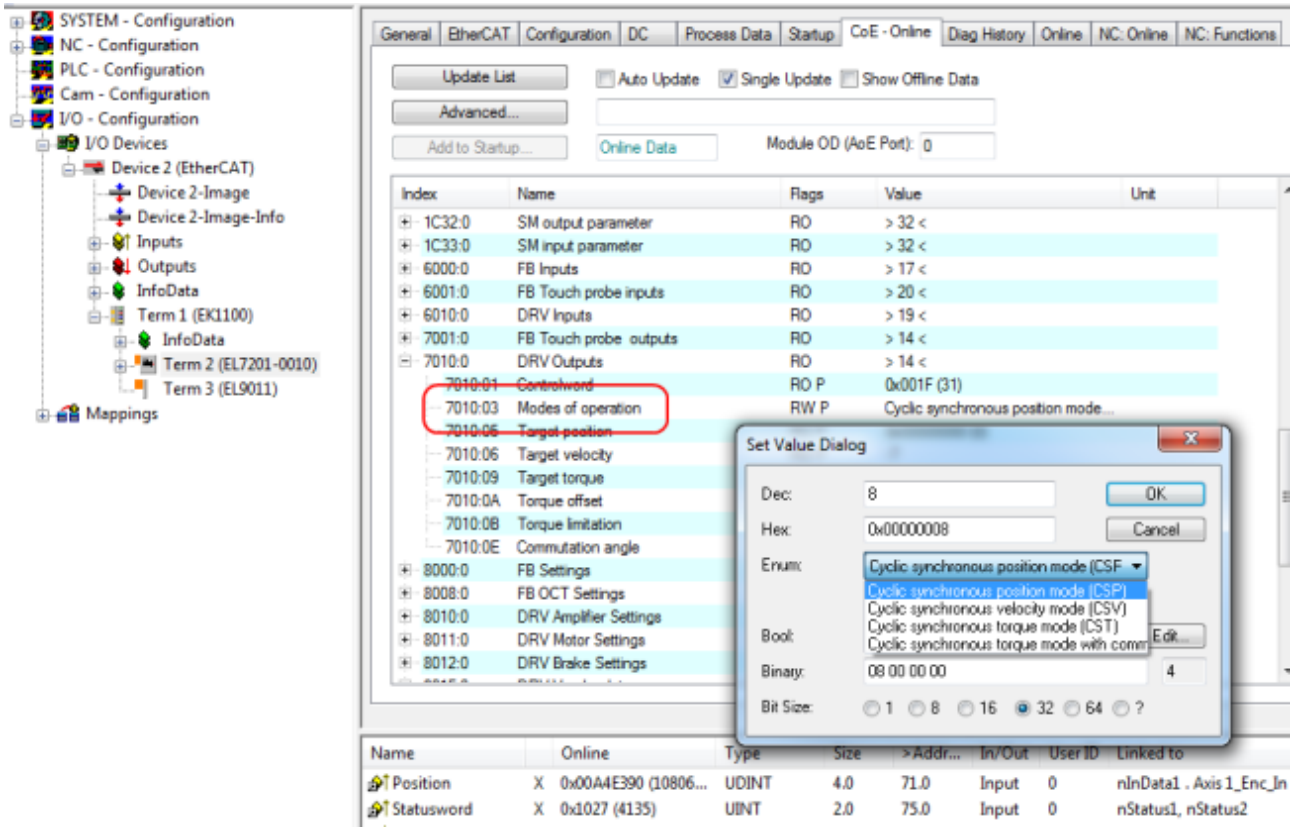


Fig. 162: Selection of the mode of operation

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

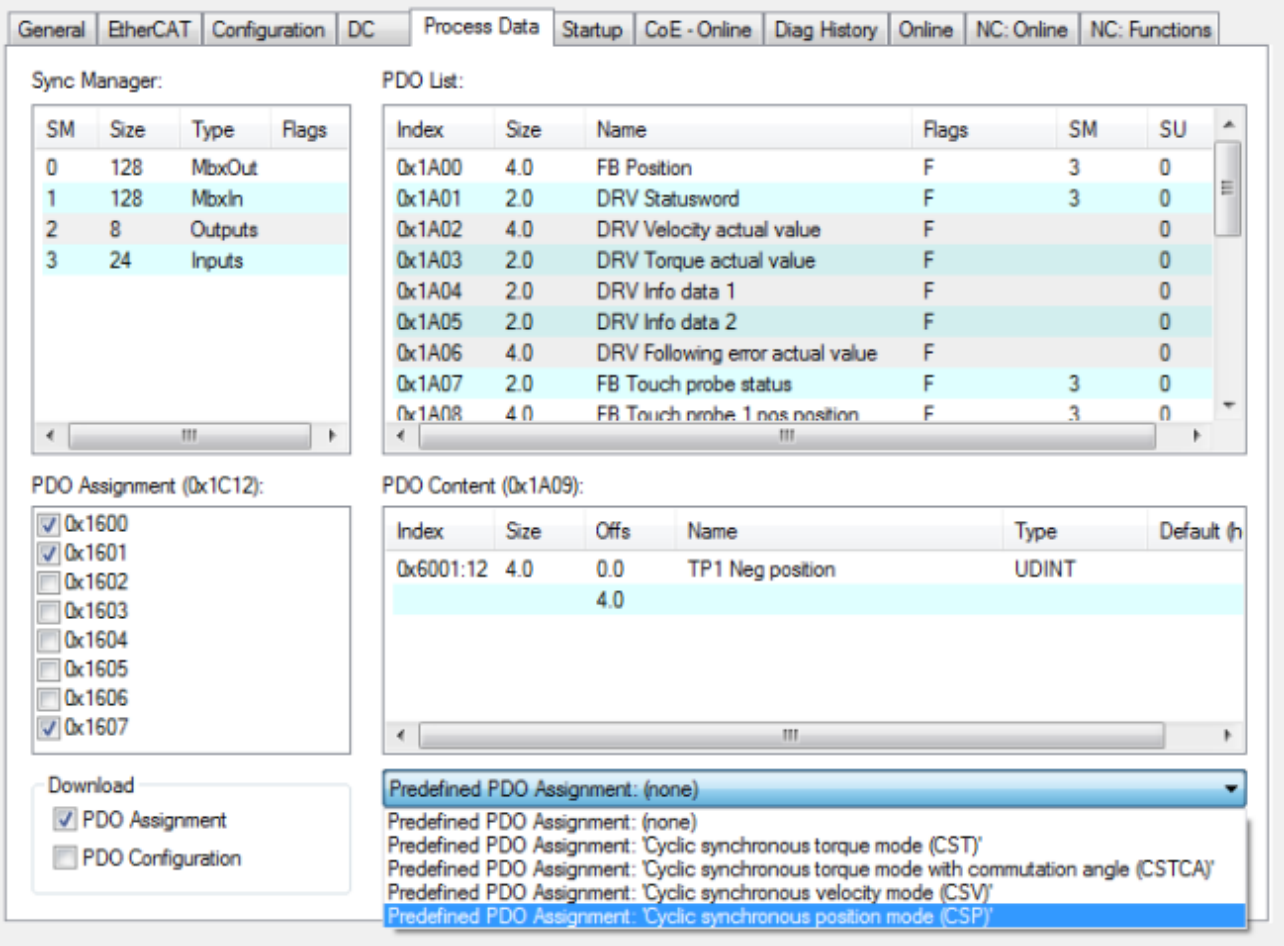


Fig. 163: Selecting a predefined PDO assignment

- 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.

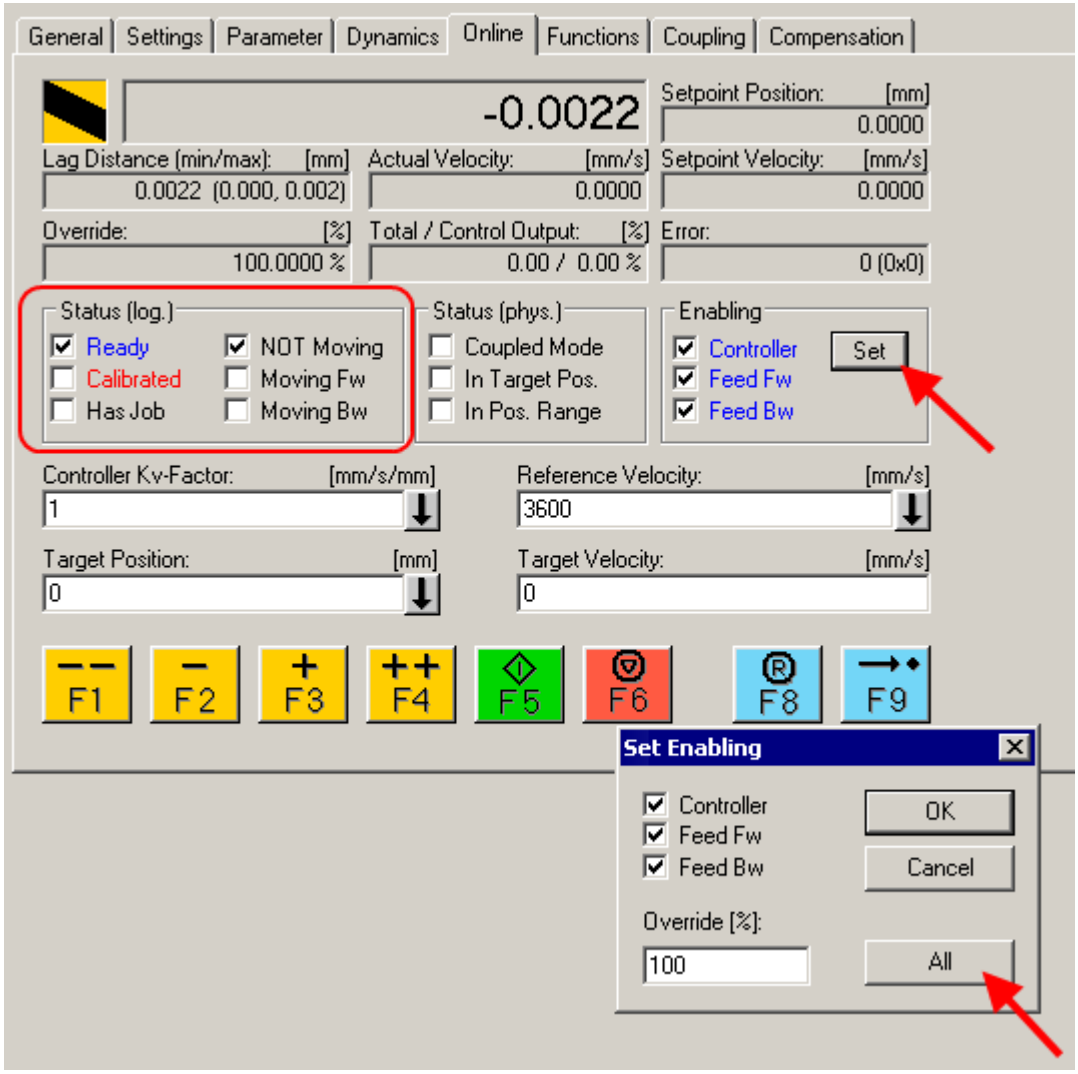


Fig. 164: Set enables

◦ 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 [▶ 109].

- 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 [▶ 100] 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. 165: 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 [▶ 166] MDP742 / Index 0x6065 [▶ 152] 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*).

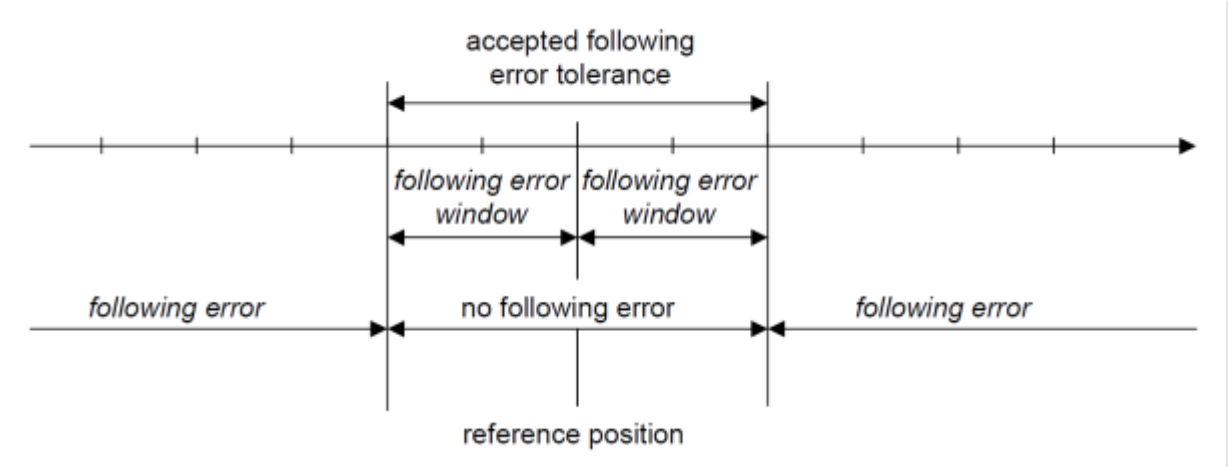


Fig. 166: Following error window

- The time (in ms) allowed for a following error exceedance can be set with the *Following error time out* (Index 0x8010:51 [▶ 166] MDP742 / Index 0x6066 [▶ 152] 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 [▶ 171] MDP742 / Index 0x60F4 [▶ 154] DS402).

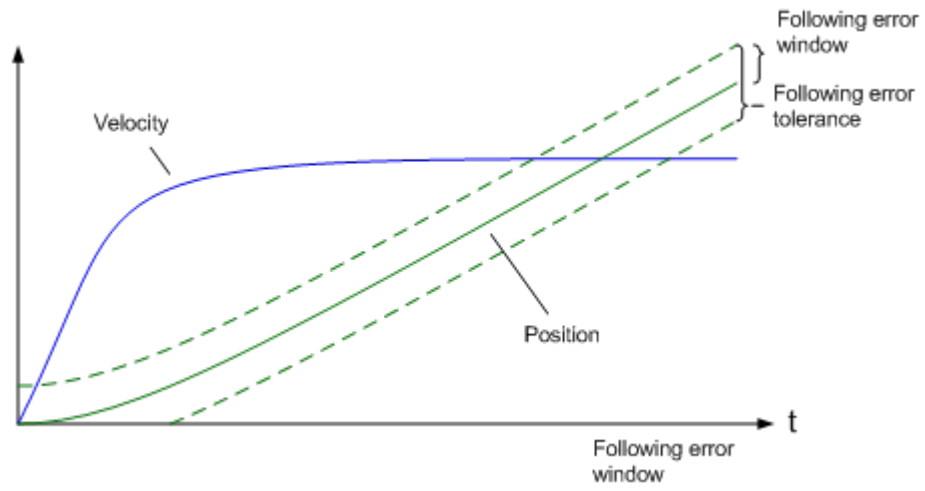


Fig. 167: Following error over time

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.

6.4 Profile MDP 742 or DS 402

The EL72x1-0010 supports the MDP 742 and DS 402 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 MDP 742 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 EL72x1-0010 is based on the CiA402 State Machine [▶ 109], which means the functional behavior is identical.
The EL72x1-0010 is delivered with the MDP 742 profile ex factory.

Changing the profile

After a profile change an EEPROM update [▶ 192] is required. The corresponding ESI description [▶ 192] 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.

6.5 MDP742 process data

Table of contents

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.).

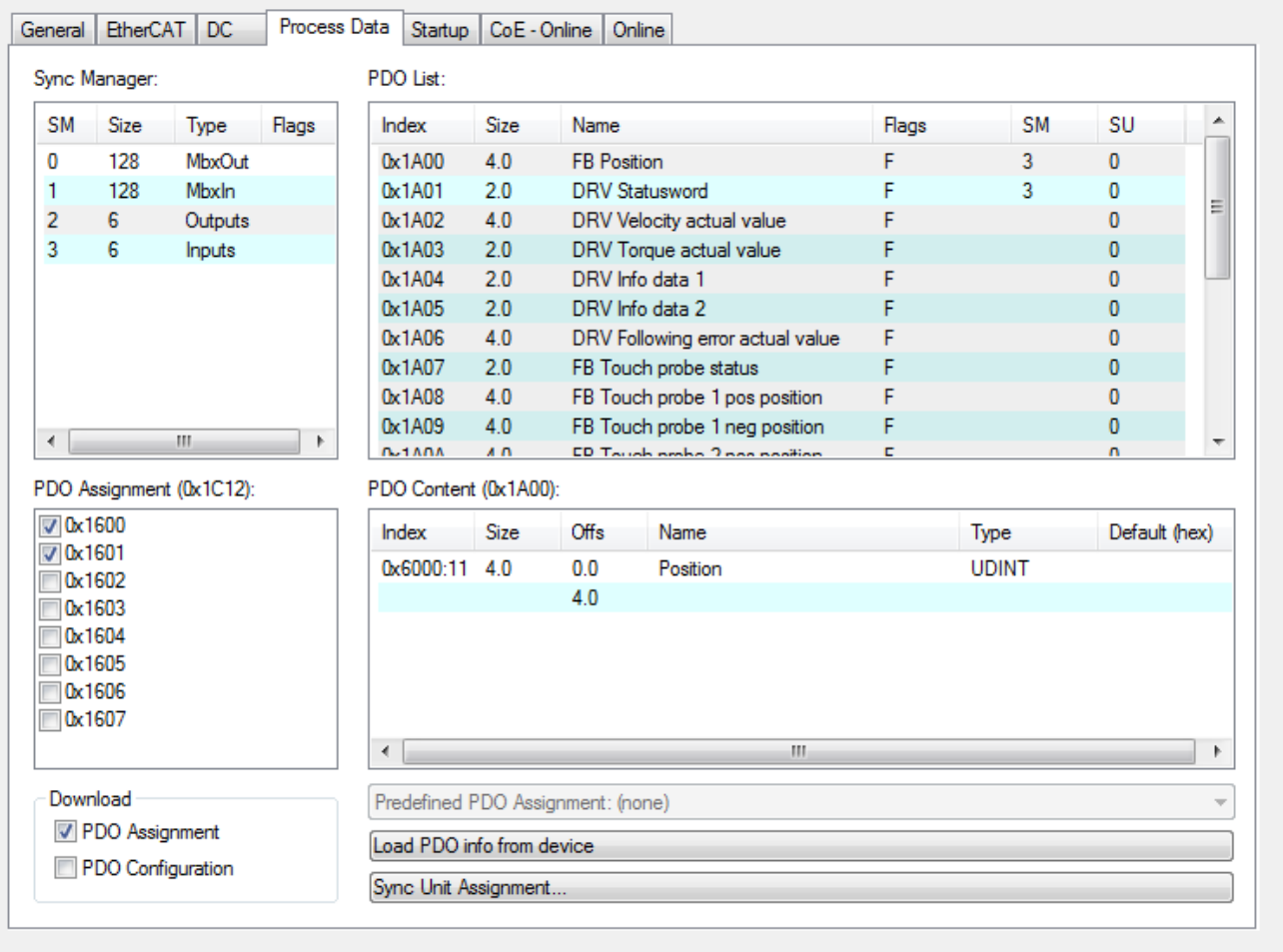


Fig. 168: Process Data tab SM2, EL72x1-0010 (default)

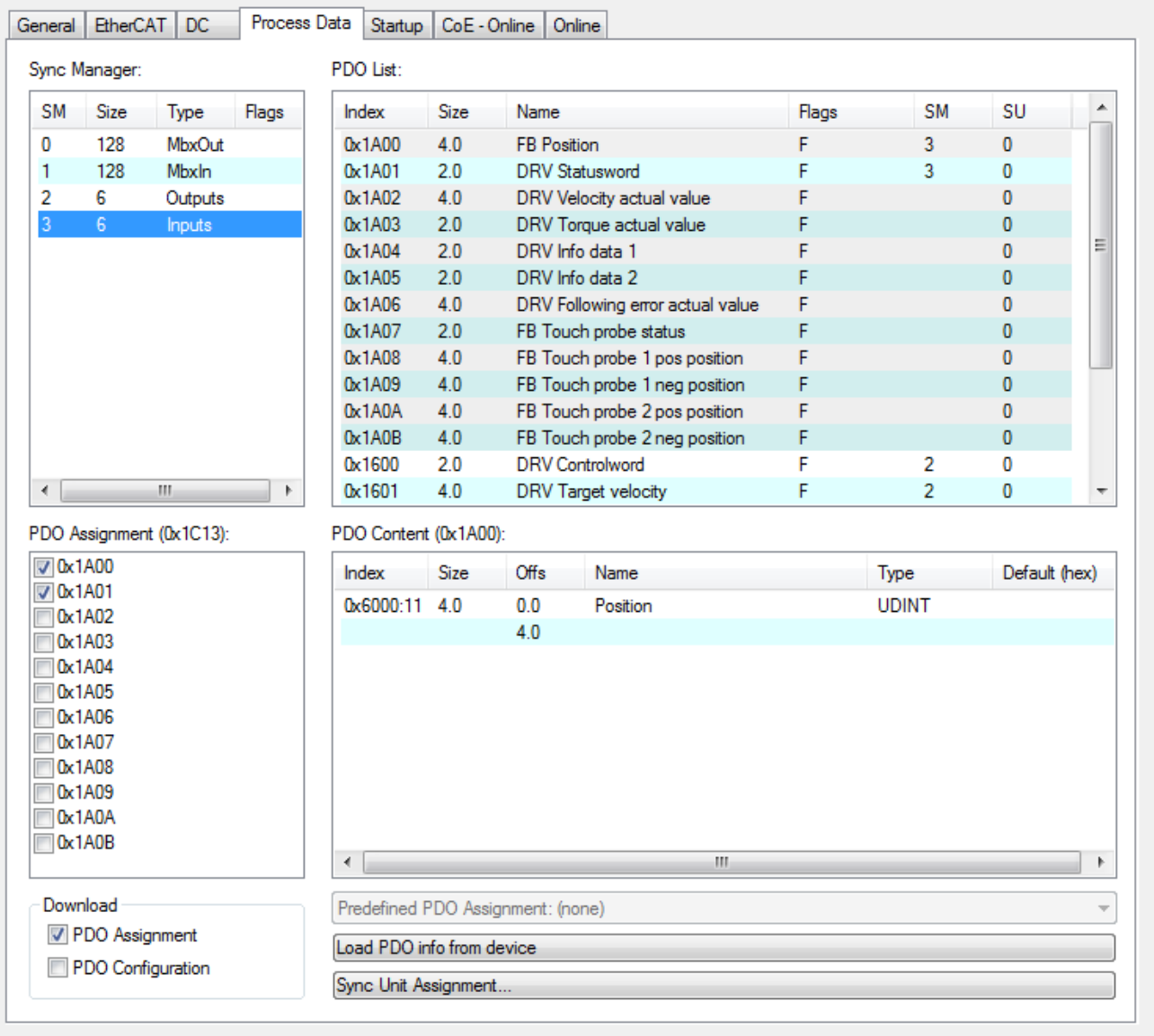


Fig. 169: Process Data tab SM3, EL72x1-0010 (default)

PDO Assignment

To configure the process data, select the required Sync Manager (SM 2 or SM 3) 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**	
0x1600 (default)	2.0	DRV Controlword	Index 0x7010:01 [▶ 173]	
0x1601(default)	4.0	DRV Target velocity	Index 0x7010:06 [▶ 173]	
0x1602	2.0	DRV Target torque	Index 0x7010:09 [▶ 173]	
0x1603	2.0	DRV Commutation angle	Index 0x7010:0E [▶ 173]	
0x1604	2.0	DRV Torque limitation	Index 0x7010:0B [▶ 173]	
0x1605	2.0	DRV Torque offset	Index 0x7010:0A [▶ 173]	
0x1606	4.0	DRV Target position	Index 0x7010:05 [▶ 173]	
0x1607	2.0	FB Touch probe control	Index 0x7001:0	
			Index 0x7001:01 [▶ 172] Index 0x7001:02 [▶ 172] Index 0x7001:03 [▶ 172] Index 0x7001:05 [▶ 172] Index 0x7001:06 [▶ 172] Index 0x7001:09 [▶ 172] Index 0x7001:0A [▶ 172] Index 0x7001:0B [▶ 172] Index 0x7001:0D [▶ 172] Index 0x7001:0E [▶ 172]	TP1 Enable TP1 Continous TP1 Trigger mode TP1 Enable pos. edge TP1 Enable neg. edge TP2 Enable TP2 Continous TP2 Trigger mode TP2 Enable pos. edge TP2 Enable neg. edge

SM3, PDO Assignment 0x1C13				
Index	**Size (byte.bit)**	**Name**	**PDO content**	
0x1A00 (default)	4.0	FB position	Index 0x6000:11 [▶ 170]	
0x1A01 (default)	2.0	DRV Statusword	Index 0x6010:01 [▶ 171]	
0x1A02	4.0	DRV Velocity actual value	Index 0x6010:07 [▶ 171]	
0x1A03	2.0	DRV Torque actual value	Index 0x6010:08 [▶ 171]	
0x1A04	2.0	DRV Info data 1	Index 0x6010:12 [▶ 171]	
0x1A05	2.0	DRV Info data 2	Index 0x6010:13 [▶ 171]	
0x1A06	4.0	DRV Following error actual value	Index 0x6010:09 [▶ 171]	
0x1A07	2.0	FB Touch probe status	Index 0x6001:0	
			Index 0x6001:01 [▶ 171] Index 0x6001:02 [▶ 171] Index 0x6001:03 [▶ 171] Index 0x6001:08 [▶ 171] Index 0x6001:09 [▶ 171] Index 0x6001:0A [▶ 171] Index 0x6001:0B [▶ 171] Index 0x6001:10 [▶ 171]	TP1 Enable TP1 Pos. value stored TP1 Neg. value stored TP1 Input TP2 Enable TP2 Pos. value stored TP2 Neg. value stored TP2 Input
0x1A08	4.0	FB Touch probe 1 pos. position	Index 0x6001:11 [▶ 171]	
0x1A09	4.0	FB Touch probe 1 neg. position	Index 0x6001:12 [▶ 171]	
0x1A0A	4.0	FB Touch probe 2 pos. position	Index 0x6001:13 [▶ 171]	
0x1A0B	4.0	FB Touch probe 2 neg. position	Index 0x6001:14 [▶ 171]	

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.

Three PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Cyclic synchronous velocity mode (CSV)	0x1600 [▶ 178] (DRV Controlword) 0x1601 [▶ 178] (DRV Target velocity)	0x1A00 [▶ 179] (FB Position) 0x1A01 [▶ 179] (DRV Statusword)
Cyclic synchronous torque mode (CST)	0x1600 [▶ 178] (DRV Controlword) 0x1602 [▶ 178] (DRV Target torque)	0x1A00 [▶ 179] (FB Position) 0x1A01 [▶ 179] (DRV Statusword) 0x1A03 [▶ 180] (DRV Torque actual value)
Cyclic synchronous torque mode with commutation angel (CSTCA)	0x1600 [▶ 178] (DRV Controlword) 0x1602 [▶ 178] (DRV Target torque) 0x1603 [▶ 178] (DRV Commutation angle)	0x1A01 [▶ 179] (DRV Statusword)
Cyclic synchronous position mode (CSP)	0x1600 [▶ 178] (DRV Controlword) 0x1606 [▶ 179] (DRV Target position)	0x1A00 [▶ 179](FB Position) 0x1A01 [▶ 179](DRV Statusword)

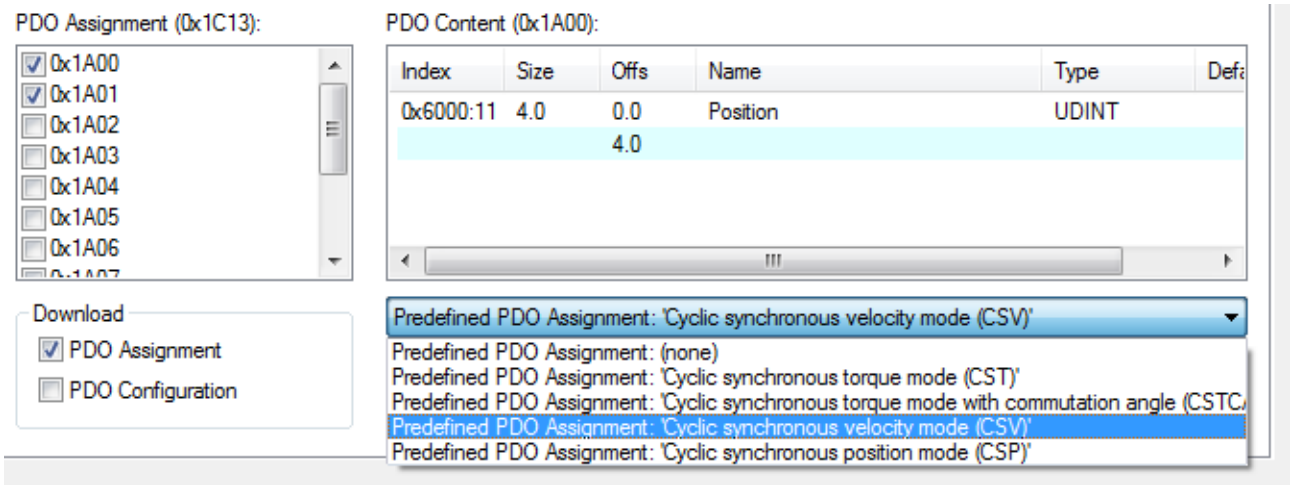


Fig. 170: Process data tab - Predefined PDO Assignment, EL72x1-0010

6.6 DS402 process data

Table of contents

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)).

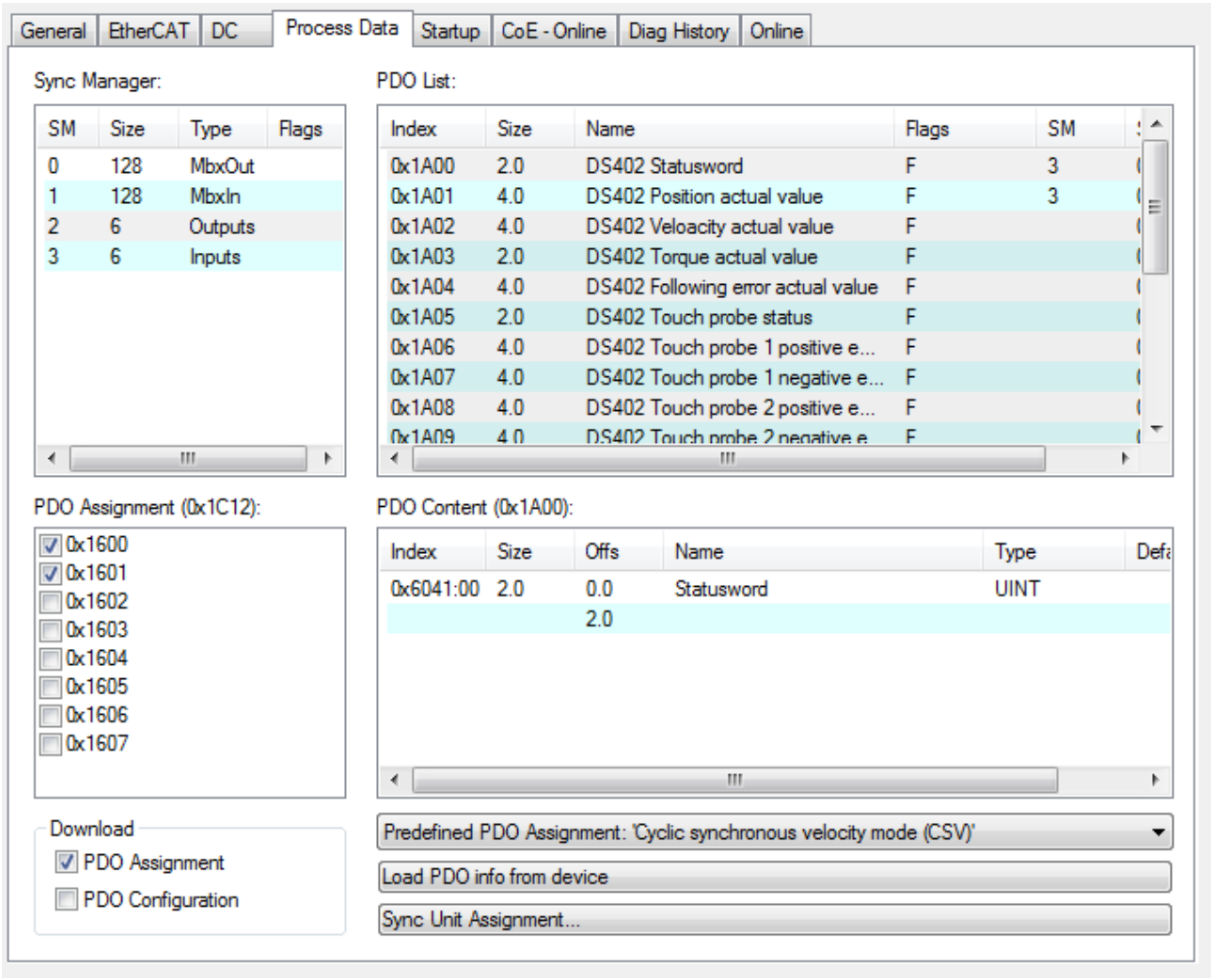


Fig. 171: Process Data tab SM2, EL72x1-0010 (default)

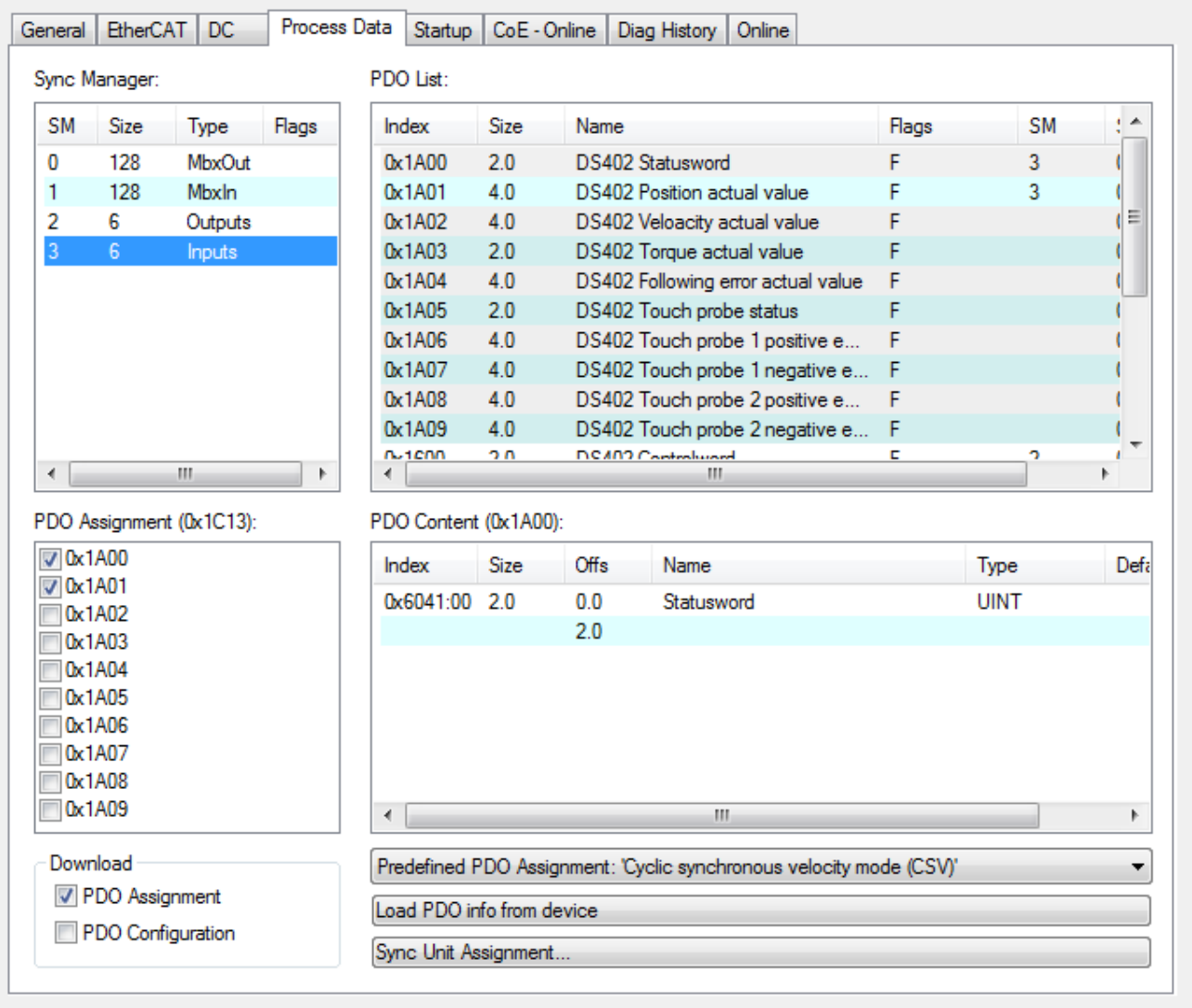


Fig. 172: Process Data tab SM3, EL72x1-0010 (default)

PDO Assignment

In order to configure the process data, select the desired Sync Manager (SM 2 & 3 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**
0x1600 (default)	2.0	DS402 Controlword	Index 0x6040 [▶ 151]
0x1601 (default)	4.0	DS402 Target velocity	Index 0x60FF [▶ 154]
0x1602	2.0	DS402 Target torque	Index 0x6071 [▶ 152]
0x1603	2.0	DS402 Commutation angle	Index 0x60EA [▶ 154]
0x1604	2.0	DS402 Torque limitation	Index 0x6072 [▶ 152]
0x1605	2.0	DS402 Torque offset	Index 0x2001:11 [▶ 150]
0x1606	4.0	DS402 Target position	Index 0x607A [▶ 153]
0x1607	2.0	DS402 FB Touch probe cfunction	Index 0x60B8 [▶ 153]
			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

SM3, PDO Assignment 0x1C13			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A00 (default)	2.0	DS402 Statusword	Index 0x6041 [▶ 151]
0x1A01 (default)	4.0	DS402 Position actual value	Index 0x6064 [▶ 151]
0x1A02	4.0	DS402 Velocity actual value	Index 0x606C [▶ 152]
0x1A03	2.0	DS402 Torque actual value	Index 0x6077 [▶ 153]
0x1A04	4.0	DS402 Following error actual value	Index 0x60F4 [▶ 154]
0x1A05	2.0	DS402 Touch probe status	Index 0x60B9 [▶ 153]
			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 pos. position	Index 0x60BA [▶ 153]
0x1A07	4.0	DS402 Touch probe 1 neg. position	Index 0x60BB [▶ 153]
0x1A08	4.0	DS402 Touch probe 2 pos. position	Index 0x60BC [▶ 154]
0x1A09	4.0	DS402 Touch probe 2 neg. position	Index 0x60BD [▶ 154]

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.

Three PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Cyclic synchronous velocity mode (CSV)	0x1600 [▶ 159] (DS402 Controlword) 0x1601 [▶ 159] (DS402 Target velocity)	0x1A00 [▶ 160] (DS402 Statusword) 0x1A01 [▶ 160] (DS402 Position actual value)
Cyclic synchronous torque mode (CST)	0x1600 [▶ 159] (DS402 Controlword) 0x1602 [▶ 159] (DS402 Target torque)	0x1A00 [▶ 160] (DS402 Statusword) 0x1A01 [▶ 160] (DS402 Position actual value) 0x1A03 [▶ 160] (DS402 Torque actual value)
Cyclic synchronous torque mode with commutation angel (CSTCA)	0x1600 [▶ 159] (DS402 Controlword) 0x1602 [▶ 159] (DS402 Target torque) 0x1603 [▶ 159] (DS402 Commutation angle)	0x1A00 [▶ 160] (DS402 Statusword)
Cyclic synchronous position mode (CSP)	0x1600 [▶ 159] (DS402 Controlword) 0x1606 [▶ 160] (DS402 Target position)	0x1A00 [▶ 160] (DS402 Statusword) 0x1A01 [▶ 160] (DS402 Position actual value)

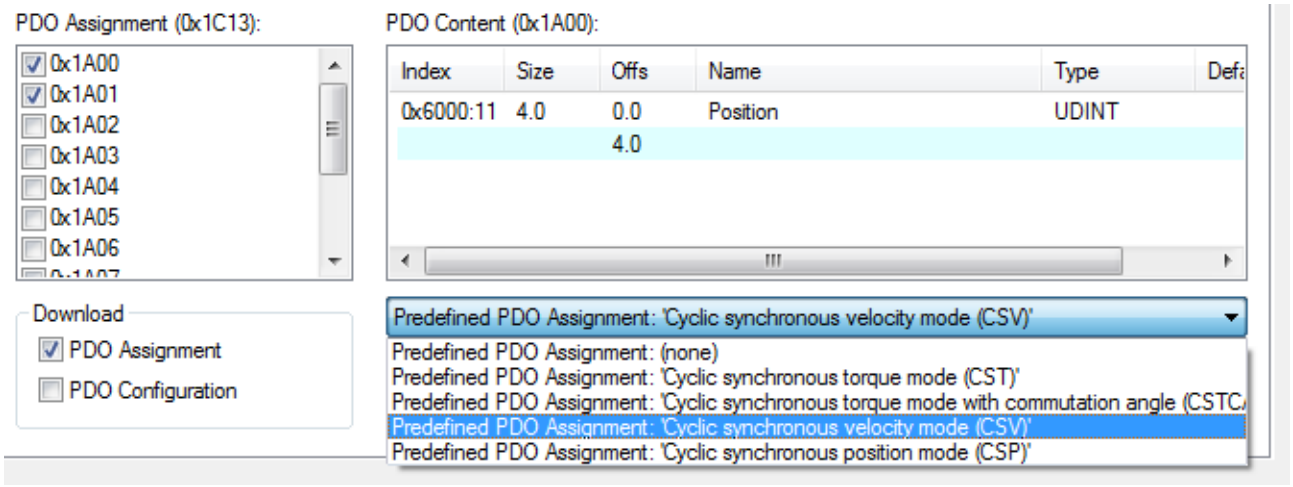


Fig. 173: Process data tab - Predefined PDO Assignment, EL72x1-0010

7 EL72x1-0010-DS402 - Object description and parameterization

Note	**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.

Note	**Parameterization via the CoE list (CAN over EtherCAT)** The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information [▶ 23] when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - use “CoE reload” for resetting changes

Attention	**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 Configuration data

Index 2002 Amplifier Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2002:0	Amplifier Settings	Maximum subindex	UINT8	RO	0x49 (73_{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 [▶ 112])	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 [▶ 112])	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 320 Hz 640 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 2059 FB OCT Nameplate

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 This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	UINT16	RW	0x0028 (40_{dec})

*) see index 2059 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	Manual override (release)	Manual release of the motor holding brake	BOOLEAN	RW	0x00 (0_{dec})
2004:11*	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 [▶ 112])	UINT16	RW	0x0000 (0_{dec})
2004:12*	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 [▶ 112])	UINT16	RW	0x0000 (0_{dec})
2004:13	Emergency application timeout	Time the amplifier waits until the speed reaches the window at which the holding brake is triggered	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 [▶ 112])	UINT16	RW	0x0000 (0_{dec})

*) see index 2059 FB OCT Nameplate

Index 2010 Feedback Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
2010:0	Feedback Settings	Maximum subindex	UINT8	RO	0x13 (19_{dec})
2010:01	Invert feedback direction	Inverting the count direction	BOOLEAN	RW	0x00 (0_{dec})
2010:11	Device type	3: OCT (not changeable)	UINT32	RW	0x00000003 (3_{dec})
2010:12	Single-turn bits	Number of single- and multi-turn-bits [▶ 96]	UINT8	RW	0x14 (20_{dec})
2010:13	Multi-turn bits		UINT8	RW	0x0C (12_{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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	BOOLEAN	RW	0x00 (0_{dec})

7.2 Configuration data (vendor-specific)

Index 2020 Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
2020:0	Vendor data	Maximum subindex	UINT8	RO	0x14 (20_{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 overcur-rent thresh-old	Threshold value for short-circuit detection **Unit**: mA	UINT32	RW	0x00002EE0 (12000_{dec})

7.3 Command object

Index FB00 Command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	Com-mand	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	Re-sponse	reserved	OCTET-STRING[4]	RO	{0}

7.4 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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot torque\ constant\ (datasheet\ motor)$	INT16	RO	0x0000 (0_{dec})

Index 6040 Controlword

Index (hex)	Name	Meaning	Data type	Flags	Default
6040:0	Control-word	DS402 Controlword [▶ 109] 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	Status-word	DS402 Statusword [▶ 109] 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 [▶ 154]) Bit 11: Internal limit active Bit 12: (Target value ignored) Bit 13 - 15: reserved	UINT16	RO	0x0000 (0_{dec})

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 [▶ 100]	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 [▶ 100] 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	UINT1616	RO	0x0000 (0_{dec})

Index 606C Velocity actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
606C:0	Velocity actual value	This object shall provide the actual velocity value	INT3232	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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot\ torque\ constant\ (datasheet\ motor)$	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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot\ torque\ constant\ (datasheet\ motor)$	UINT16	RW	0x7FFF (32767_{dec})

Index 6075 Motor rated current

Index (hex)	Name	Meaning	Data type	Flags	Default
6075:0	Motor rated current	Rated motor 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* Formula:I: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot torque\ constant\ (datasheet\ motor)$	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 [▶ 100]	INT32	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 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 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	UINT1616	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 [▶ 100]	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 [▶ 100]	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 [▶ 100]	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 [▶ 100]	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 $10^{(\text{interpolation time index})}$ (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 1-31: 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 [▶ 100]	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})

7.5 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 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 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	0x1F (31_{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})

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 2018 [▶ 149]).

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})

Index (hex)	Name	Meaning	Data type	Flags	Default
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})
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	

7.6 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	EL72x1-0011

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 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 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 [▶ 163])	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 [▶ 164], 0x1C33:06 [▶ 163], and 0x1C33:09 [▶ 164] 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 [▶ 163]	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 [▶ 163] or 0x1C33:08 [▶ 164])	UINT16	RO	0x4808 (18440_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 163]	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 [▶ 163]	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 [▶ 163]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 163]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 163]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 163]	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 EL72x1-0010-MDP742 - Object description and parameterization

Note	**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.

Note	**Parameterization via the CoE list (CAN over EtherCAT)** The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information [▶ 23] when using/manipulating the CoE parameters: - Keep a **startup list** if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - use “CoE reload” for resetting changes

Attention	**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.

8.1 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})

8.2 Configuration data

Index 8000 FB Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	FB Settings	Maximum subindex	UINT8	RO	0x13 (19_{dec})
8000:01	Invert feedback direction	Inverting the count direction	BOOLEAN	RW	0x00 (0_{dec})
8000:11	Device type	3: OCT (not changeable)	UINT32	RW	0x00000003 (3_{dec})
8000:12	Singleturn bits	Number of single- and multi-turn-bits [▶ 96]	UINT8	RW	0x14 (20_{dec})
8000:13	Multiturn bits		UINT8	RW	0x0C (12_{dec})

Index 8008 FB OCT Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8008:0	FB OCT Settings	Maximum subindex	UINT8	RO	0x00 (0_{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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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	0x42 (66_{dec})
8010:01	Enable TxPDOToggle	Show TxPDO toggle in status word (bit 10)	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 type plates [▶ 112])	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 type plates [▶ 112])	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 position controller **Unit**: (rad/s) / rad	UINT32	RW	0x0000000A (10_{dec})
8010:19	Nominal DC link voltage	Nominal 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	Velocity limitation **Unit**: rpm	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**: rpm	UINT16	RW	0x0000 (0_{dec})

*) see index 9009 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. Permitted values:		UINT8	RW	0x01 (1_{dec})
		1	Torque current (filtered 1ms)			
		2	DC link voltage (mV)			
		4	PCB temperature (0.1 °C)			
		5	Errors			
		6	Warnings			
8010:3A	Select info data 2	Selection "Info data 2" Optional display of additional information in the cyclic process data. Permitted values:		UINT8	RW	0x01 (1_{dec})
		1	Torque current (filtered 1ms)			
		2	DC link voltage (mV)			
		4	PCB temperature (0.1 °C)			
		5	Errors			
		6	Warnings			
8010:41	Low-pass filter frequency	Low-pass filter frequency **Unit**: Hz The following values can be set: 0 Hz = off 320 Hz 640 Hz		UINT16	RW	0x0140 (320_{dec})
8010:42	Halt ramp deceleration	Halt ramp deceleration **Unit**: 0.1 rad / s²		UINT32	RW	0x0000F570 (62832_{dec})
8010:50	Following error window	Following error monitor: Following error window **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 100] 0xFFFFFFFF (-1_{dec}) = following error monitor off Any other value = following error monitor on		UINT32	RW	0xFFFFFFFF (-1)
8010:51	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	RW	0x0000

*) see index 9009 FB OCT Nameplate

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 This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 112])	UINT32	RW	0x00001770 (6000_{dec})
8011:12*	Rated current	Rated current **Unit**: mA This value is affected by automatic scanning. (see Automatic scanning of the electronic type plates [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	UINT32	RW	0x00040000 (262144_{dec})

*) see index 9009 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 [▶ 112])	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 [▶ 112])	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 [▶ 112])	UINT16	RW	0x0028 (40_{dec})

*) see index 9009 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	Manual override (release)	Manual release of the motor holding 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 [▶ 112])	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 [▶ 112])	UINT16	RW	0x0000 (0_{dec})
8012:13	Emergency application timeout	Time the amplifier waits until the speed reaches the window at which the holding brake is triggered	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 [▶ 112])	UINT16	RW	0x0000 (0_{dec})

*) see index 9009 FB OCT Nameplate

8.3 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	0x14 (20_{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	Tated 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})

8.4 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 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}

8.5 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})
6000:11	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})
6001:09	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})
6001:11	TP1 pos position	Positive value of Touchprobe 1 **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 100]	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 [▶ 100]	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 [▶ 100]	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 [▶ 100]	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})
6010:07	Velocity actual value	Display of the current velocity value **Unit:** see Index 9010:14 [▶ 176]	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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot torque\ constant\ (datasheet\ motor)$	INT16	RO	0x0000 (0_{dec})
6010:09	Following error actual value	Following error **Unit**: the given value must be multiplied by the corresponding scaling factor [▶ 100]	INT32	RO	0x00000000 (0_{dec})
6010:12	Info data 1	Synchronous information (selection via subindex 8010:39 [▶ 166])	UINT16	RO	0x0000 (0_{dec})
6010:13	Info data 2	Synchronous information (selection via subindex 8010:3A [▶ 166])	UINT16	RO	0x0000 (0_{dec})

8.6 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	Control-word	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 opera-tion	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 [▶ 100]	INT32	RW	0x00000000 (0_{dec})
7010:06	Target velocity	Configured target velocity The velocity scaling can be found in object 0x9010:14 [▶ 176] (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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot$ *torque constant (datasheet motor)*	INT16	RO	0x0000 (0_{dec})
7010:0A	Torque offset	Torque value offset The value is specified in 1000th of the *rated current* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot$ *torque constant (datasheet motor)*	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* Formula: $M = \frac{Torque\ actual\ value}{1000} \cdot \frac{rated\ current}{\sqrt{2}} \cdot$ *torque constant (datasheet motor)*	UINT16	RW	0x7FFF (32767_{dec})
7010:0E	Commu-tation angle	Commutation angle (for CSTCA mode) **Unit:** 360° / 2^{16}	UINT16	RO	0x0000 (0_{dec})

8.7 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	0x1F (31_{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})

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 8001 [▶ 165]).

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})

Index (hex)	Name	Meaning	Data type	Flags	Default
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})
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	0x14 (20_{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})

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})

8.8 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	EL72x1-0010

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 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 0x0A (Torque offset))	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 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 0x01 (Statusword))	UINT32	RO	0x6010:09, 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	0x08 (8_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x07 (Velocity actual value))	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 (5 bits align)	UINT32	RO	0x0000:00, 5
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x09 (TP2 Enable))	UINT32	RO	0x6001:09, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0A (TP2 pos value stored))	UINT32	RO	0x6001:0A, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0B (TP2 neg value stored))	UINT32	RO	0x6001:0B, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5

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 0x6010 (DRV Inputs), entry 0x08 (Torque actual value))	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 0x6010 (DRV Inputs), entry 0x12 (Info data 1))	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 0x6010 (DRV Inputs), entry 0x13 (Info data 2))	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 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 [▶ 183])	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 [▶ 183], 0x1C32:05 [▶ 183], 0x1C32:06 [▶ 183], 0x1C32:09 [▶ 183], 0x1C33:03 [▶ 184], 0x1C33:06 [▶ 183], and 0x1C33:09 [▶ 184] 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 [▶ 183]	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 [▶ 183] or 0x1C33:08 [▶ 184])	UINT16	RO	0x4808 (18440_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 183]	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 [▶ 183]	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 [▶ 183]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 183]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 183]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 183]	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	0x0002 (2_{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	0x02 (2_{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})

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}

9 Error correction

9.1 Diagnose - Diag Messages

Table of contents

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:

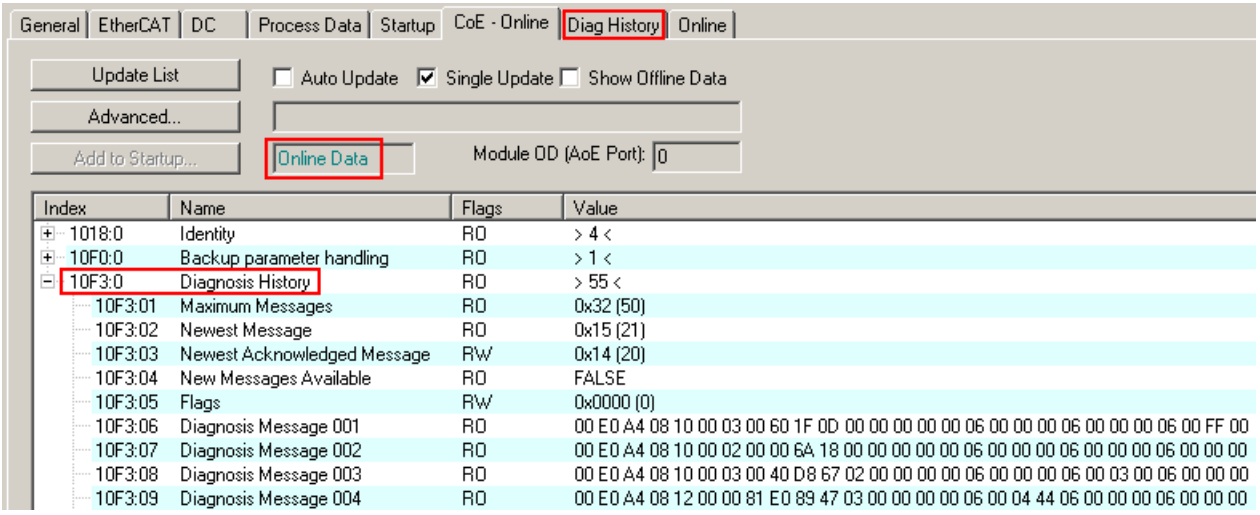


Fig. 174: DiagMessages in the CoE

The subindex of the latest *DiagMessage* can be read under x10F3:02.

Note	**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.

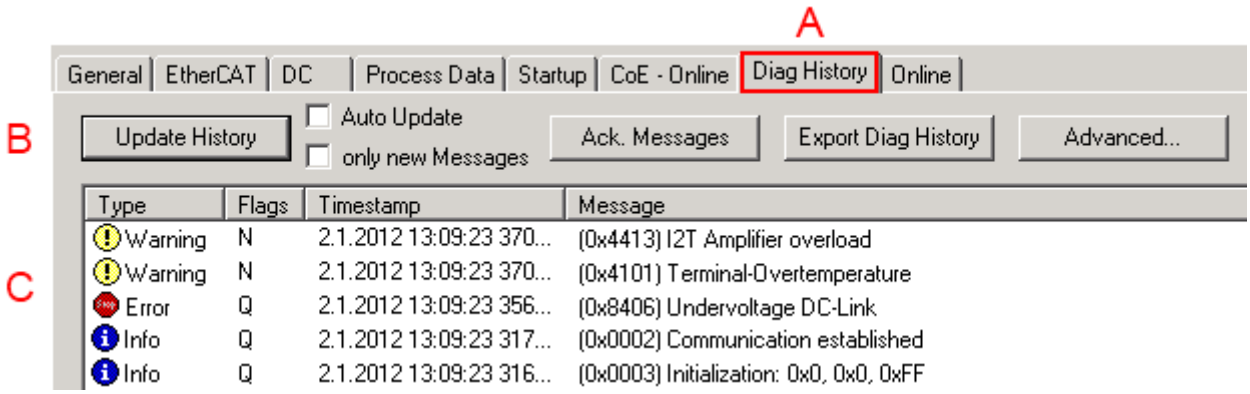


Fig. 175: Implementation of the DiagMessage system in the TwinCAT System Manager

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.

Interpretation

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 1: Info 4: Warning 8: Error	1: General 2: Communication 3: Encoder 4: Drive 5: Inputs	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Overview of text IDs

Text ID	Type	Place	Text (English only)	Cause
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Communication 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.
0x1100	Information	General		Detection of the mode of operation ended
0x1135	Information	General		Cycle time OK
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 - StartInputhandler
0x1303	Information	Encoder		Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully complete, channel: %X
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command, channel: %X
0x1400	Information	Drive		Drive is calibrated
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold
0x4300	Warning	Encoder		Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4400	Warning	Drive		Drive is not calibrated
0x4401	Warning	Drive		Start type is not supported
0x4402	Warning	Drive		Command rejected
0x4405	Warning	Drive		Modulo sub-type invalid
0x4410	Warning	Drive		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 - 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
0x4417	Warning	Drive	Motor-Overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
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 aborded	
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8100	Error	General		Error bit set in the status word
0x8101	Error	General		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

Text ID	Type	Place	Text (English only)	Cause
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
0x8140	Error	General	Sync Error	Real-time violation
0x8143	Error	General	Jitter too big	Jitter limit violation
0x8200	Error	Communication		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
0x82FF	Error	Communication		Boot mode not activated
0x8300	Error	Encoder		Error while setting the position
0x8301	Error	Encoder		Encoder increments not configured
0x8302	Error	Encoder	Feedback-Error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder supply error	Encoder power supply unit error
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error, channel: %X
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported, channel: %X
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Delay time, tolerance limit exceeded, 0x%X, channel: %X
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Delay time, maximum value exceeded, 0x%X, channel: %X
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error, channel: %X
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature %X could not be read, channel: %X
0x8400	Error	Drive		Drive incorrectly configured
0x8401	Error	Drive		Limitation of the calibration velocity
0x8402	Error	Drive		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		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
0x8415	Error	Drive		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

10 Appendix

10.1 Firmware compatibility

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

Note

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

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

EL7201-0010			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00 - 02	01	EL7201-0010-0019	2013/10
	02		2013/10
02 - 05	03	EL7201-0010-0020	2014/02
	04	EL7201-0010-0021	2014/02
	05	EL7201-0010-0022	2014/04
	06	EL7201-0010-0023	2014/05
	07		2014/07
	08	EL7201-0010-0024	2015/03
	09		2015/06
06*	10*		2015/06

EL7211-0010			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00 - 02	01	EL7211-0010-0019	2013/10
	02		2013/10
02 - 05	03	EL7211-0010-0020	2014/02
	04	EL7211-0010-0021	2014/02
	05	EL7211-0010-0022	2014/04
	06	EL7211-0010-0023	2014/05
	07		2014/07
	08	EL7211-0010-0024	2015/03
	09		2015/06
06*	10*		2015/06

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

10.2 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

10.3 Firmware Update EL/ES/EM/EPxxxx

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

Storage locations

An EtherCAT slave stores operating data in up to 3 locations:

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

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

The TwinCAT System Manager offers mechanisms for programming all 3 parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

 **Attention**	**Risk of damage to the device!** Note the following when downloading new device files • Firmware downloads to an EtherCAT device must not be interrupted • Flawless EtherCAT communication must be ensured. CRC errors or Lost Frames must be avoided. • The power supply must adequately dimensioned. The signal level must meet the specification. In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

Device description ESI file/XML

 **Attention**	**Notice 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:

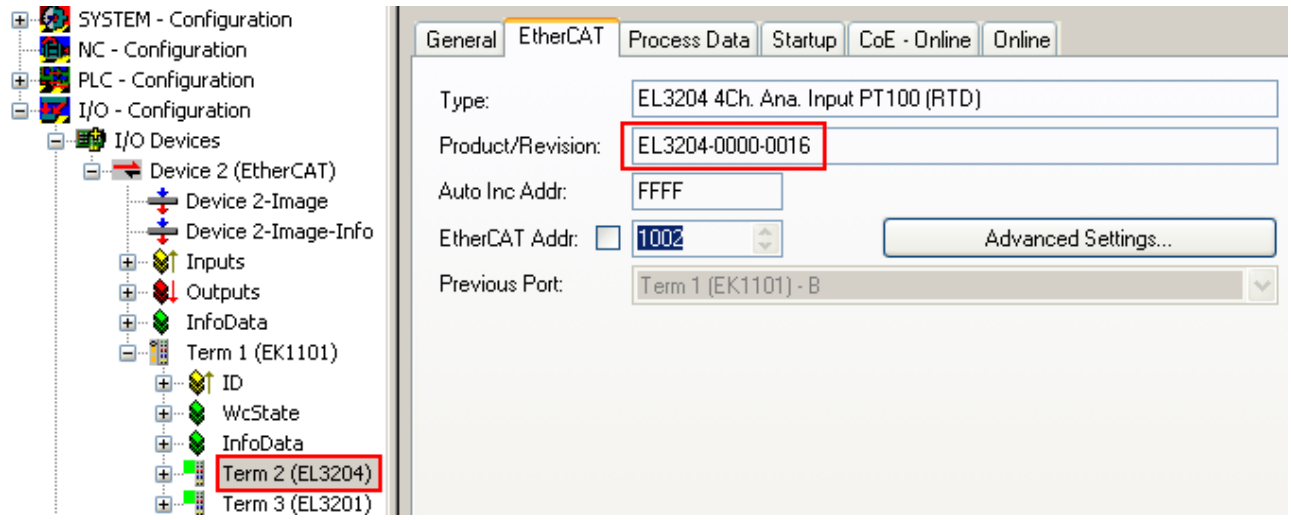


Fig. 176: Device identifier consisting of name EL3204-0000 and revision -0016

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). Generally the configured revision must be equal or lower than the version used in the terminal network.

For further information please refer to the EtherCAT System Documentation.

Note	**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:

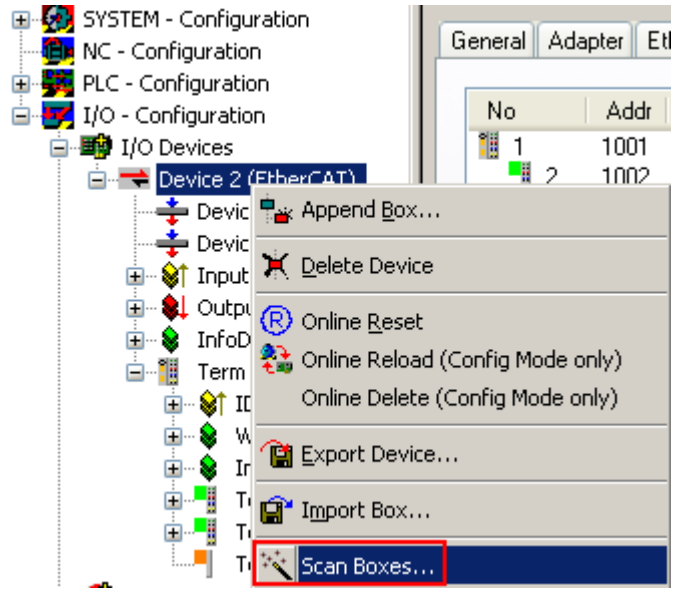


Fig. 177: Scan the subordinate field by right-clicking on the EtherCAT device in Config/FreeRun mode

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

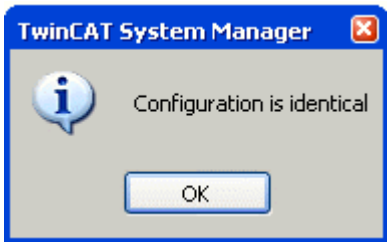


Fig. 178: Configuration is identical

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

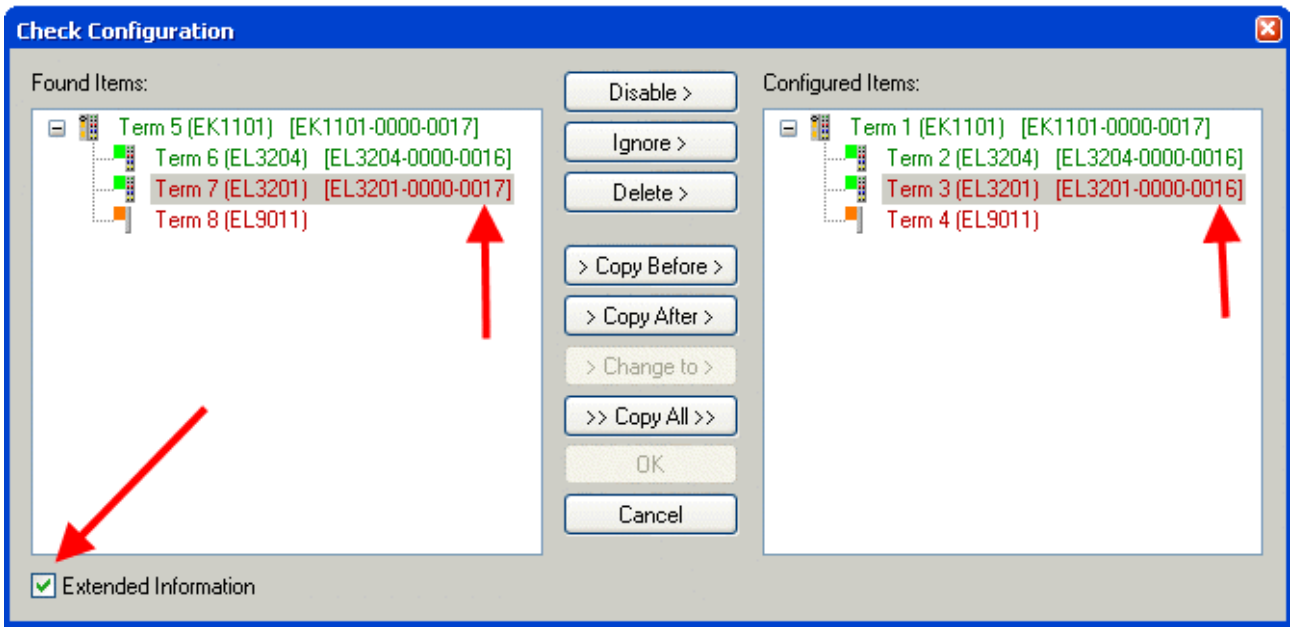


Fig. 179: Change dialog

In the example shown in Fig. *"Change dialog"*. an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** had been configured. In this case it makes sense to adapt the configuration with the *Copy Before* button. The *Extended Information* checkbox must be set in order to have the revision displayed.

Changing the ESI slave identifier

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

- The EtherCAT communication with the slave must be flawless.
- The state of the slave is irrelevant.
- Right-click on the slave in the online display to bring up the *EEPROM Update* dialog, Fig. *"EEPROM Update"*.

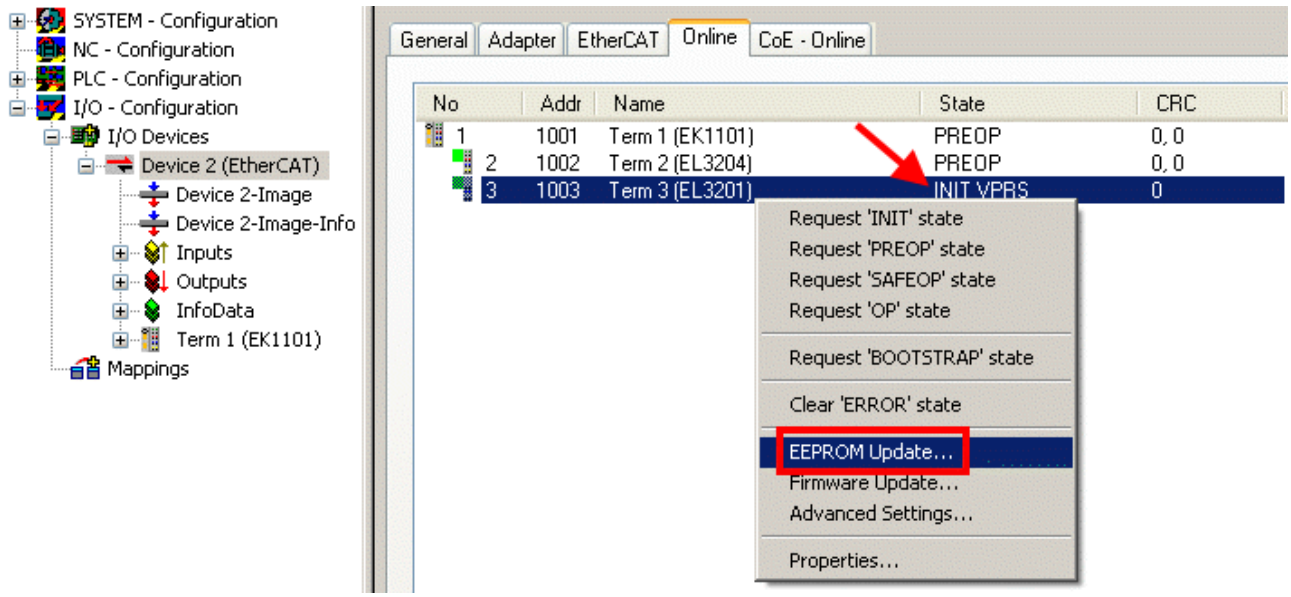


Fig. 180: EEPROM Update

Select the new ESI description in the following dialog, see Fig. *"Selecting the new ESI"*.
The *ShowHiddenDevices* checkbox also shows older, usually hidden slave versions.

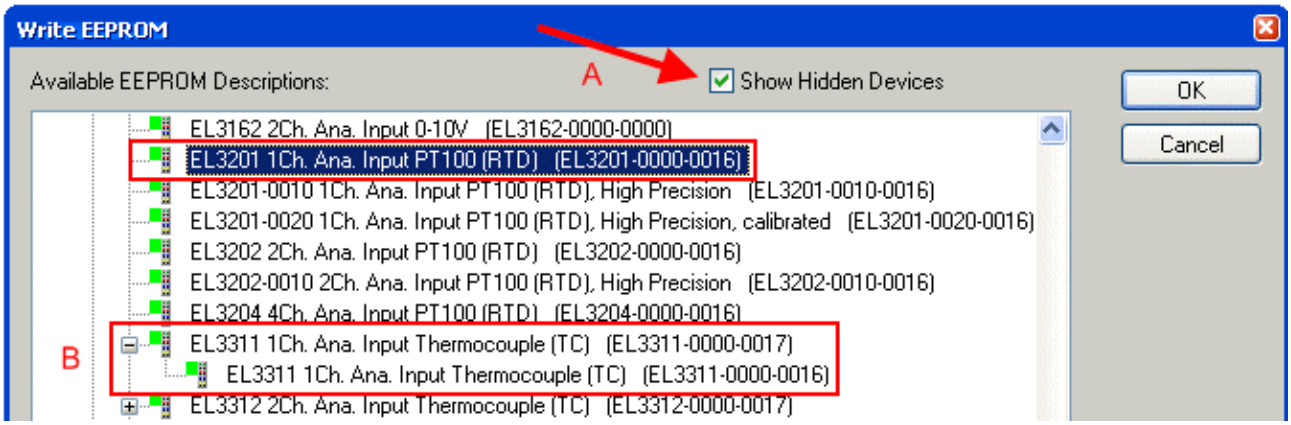


Fig. 181: Selecting the new ESI

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

Note	**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.

Determining the firmware version

Determining the version on laser inscription

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

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

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

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

Determining the version via the System Manager

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

Note	**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 does supported it. 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.

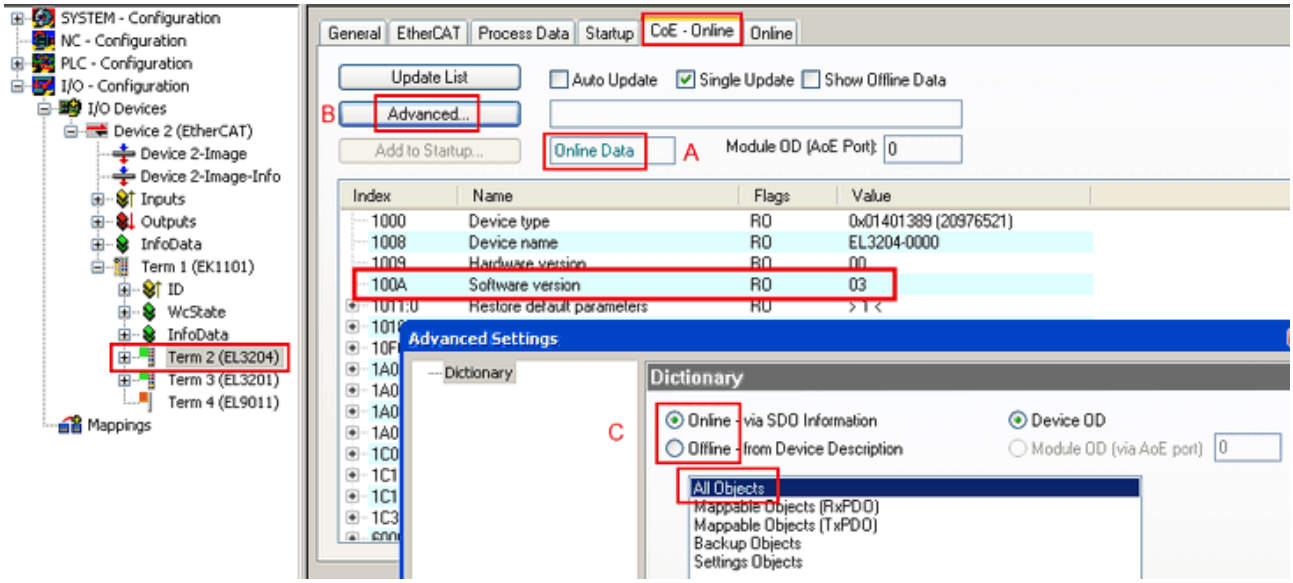


Fig. 182: Display of EL3204 firmware version

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*.

Updating controller firmware *.efw

Note	**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.

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

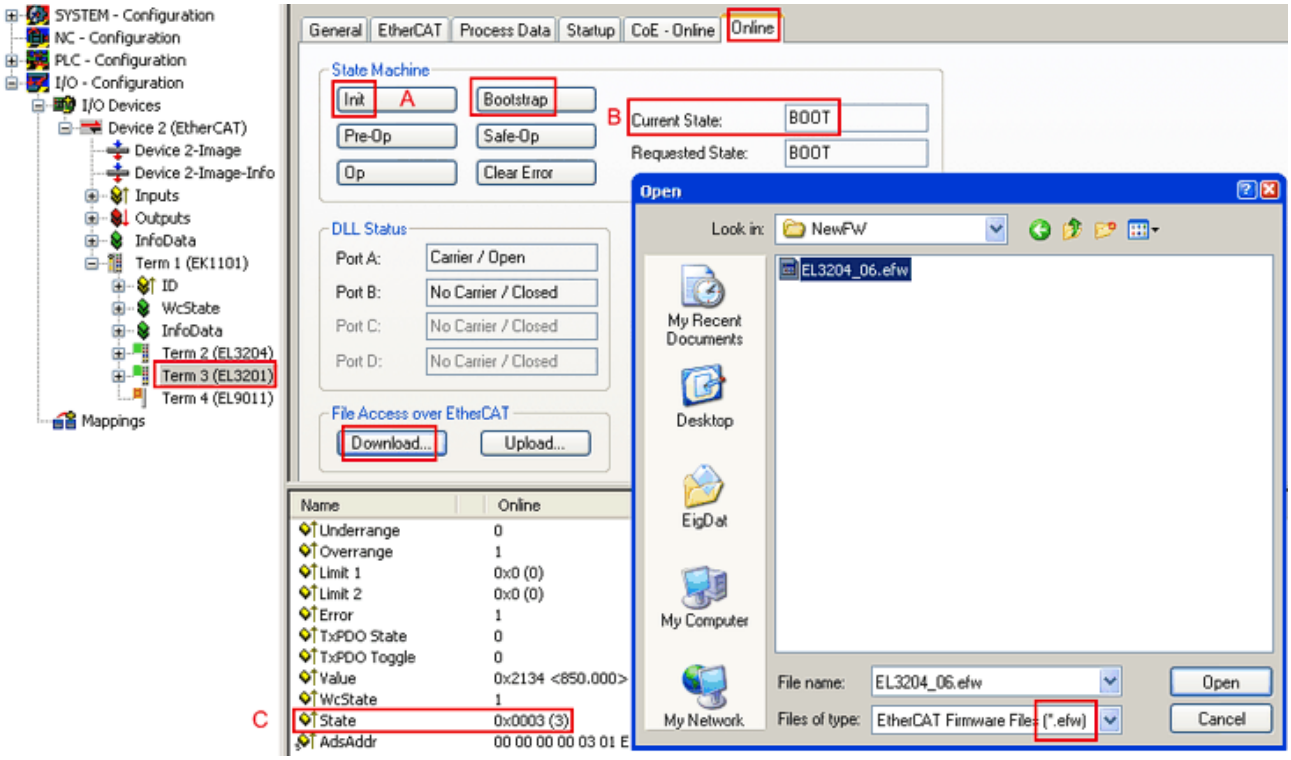


Fig. 183: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support.

- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new *efw file
- After the download switch to INIT, then OP
- Switch off the slave briefly

FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

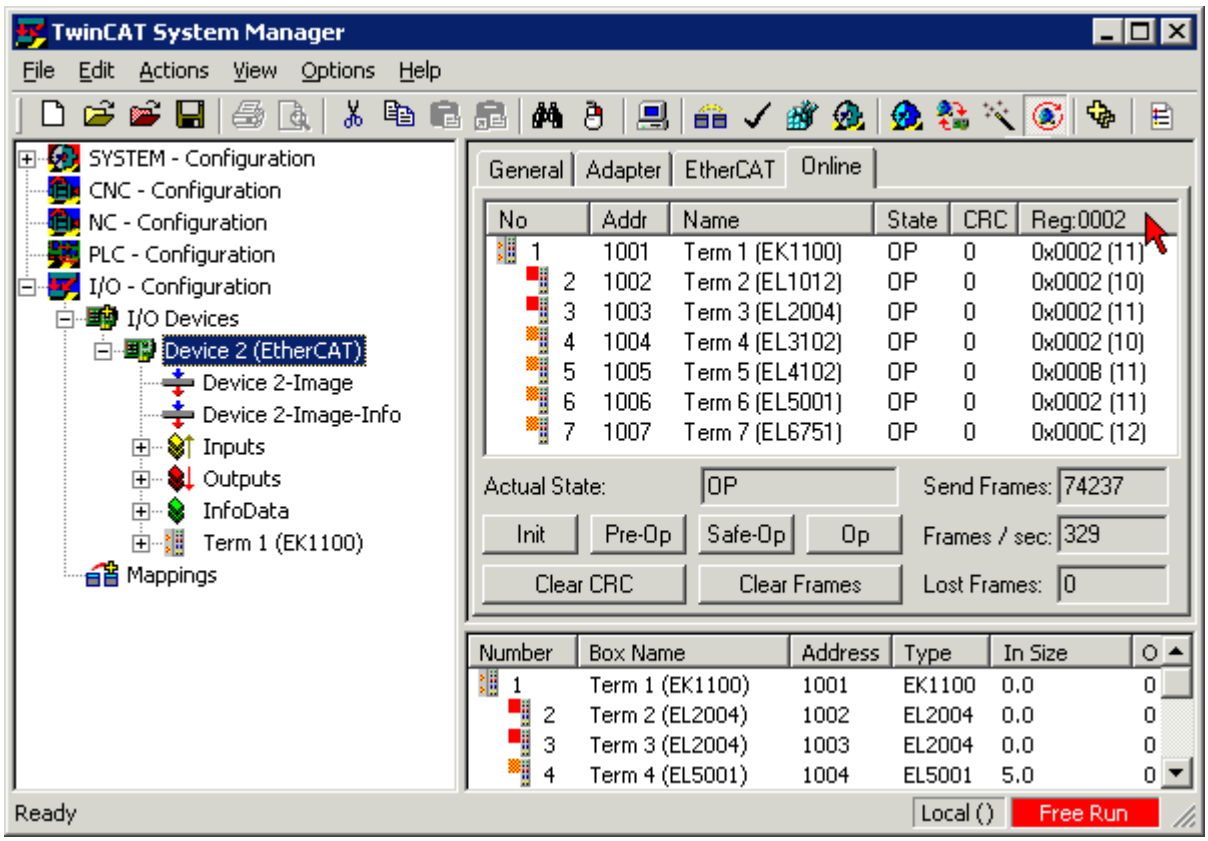


Fig. 184: FPGA firmware version definition

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

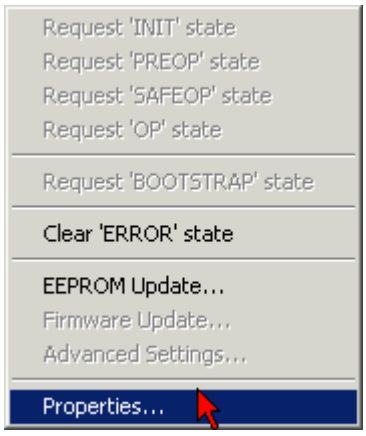


Fig. 185: Context menu Properties

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.

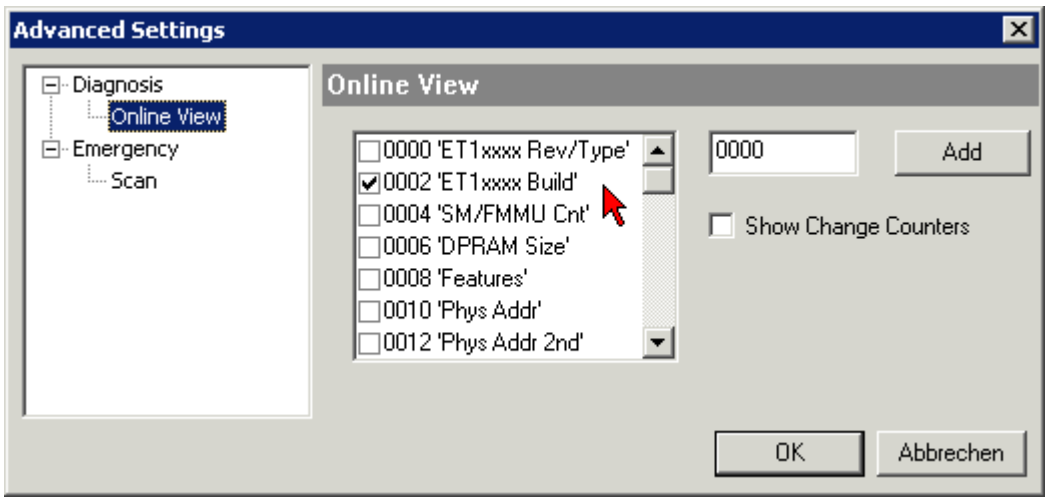


Fig. 186: Dialog Advanced Settings

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

In the TwinCAT System Manager select the terminal whose FPGA firmware you want to update (in this example terminal 5: EL5001) and
click on *Advanced Settings* in the *EtherCAT* tab.

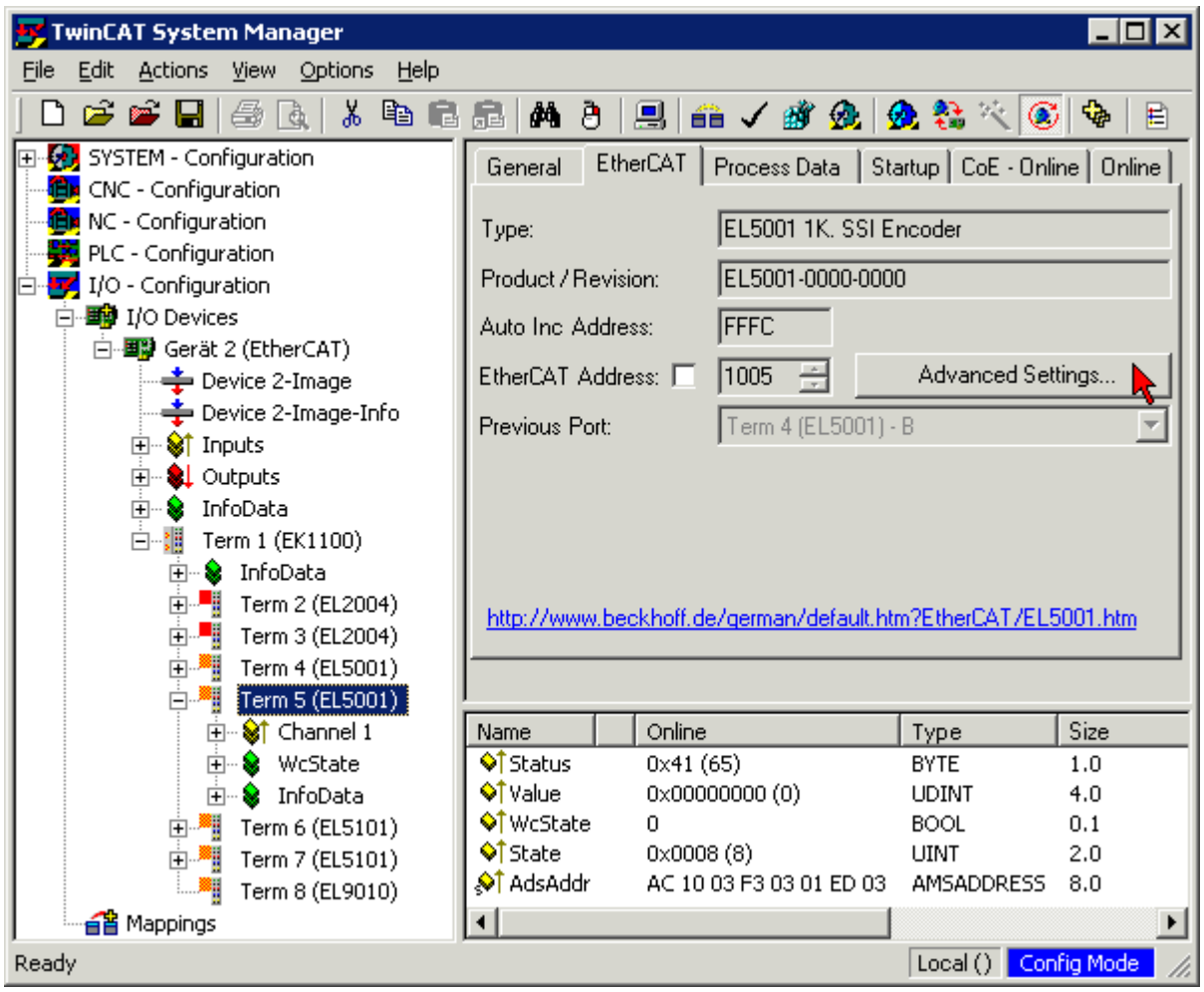


Fig. 187: Select dialog Advanced Settings

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

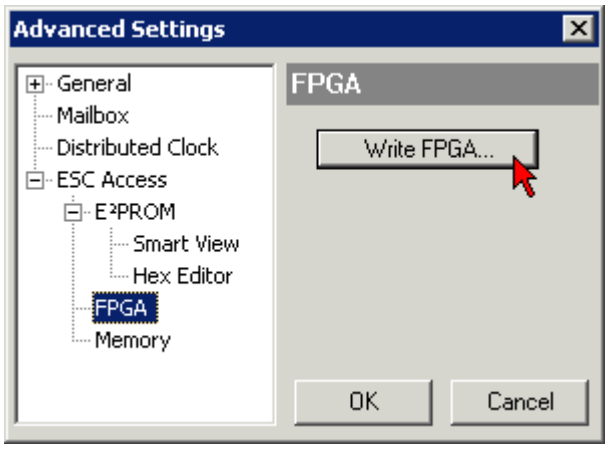


Fig. 188: Select dialog FPGA

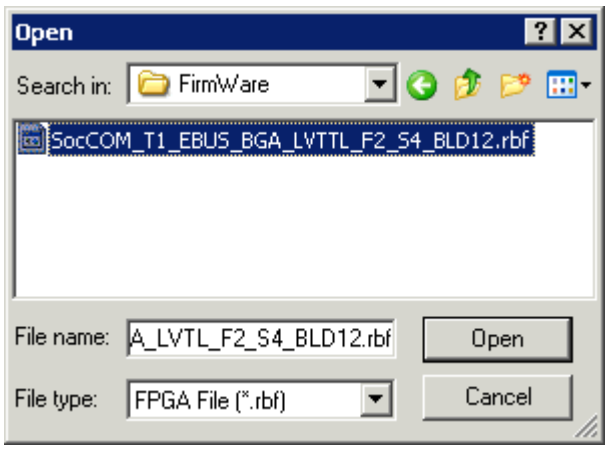


Fig. 189: Write FPGA

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

Attention	**Risk of damage to the device!** A firmware download to an EtherCAT device must never be interrupted! If this process is cancelled, the supply voltage switched off or the Ethernet connection interrupted, the EtherCAT device can only be recommissioned by the manufacturer!

In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.

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.

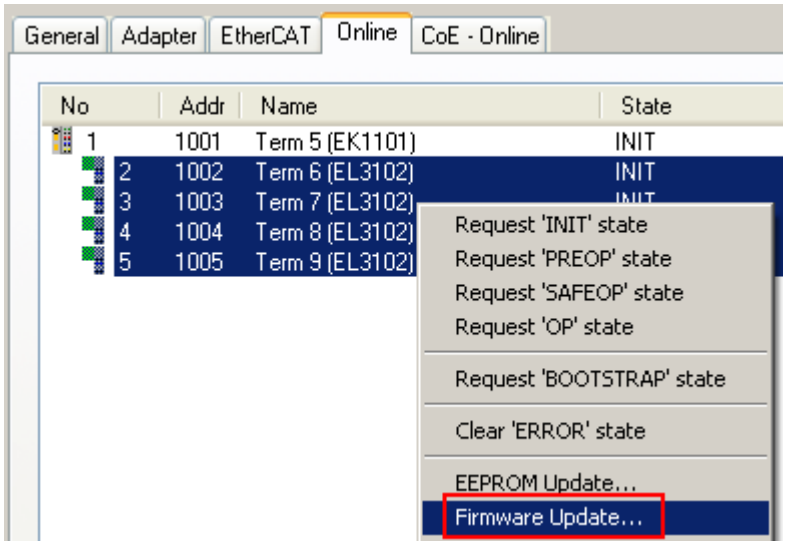


Fig. 190: Multiple selection and firmware update

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

10.4 Restoring the delivery state

Restoring the delivery state To restore the delivery state for backup objects in ELxxxx terminals, the CoE object "Restore default parameters", SubIndex 001 can be selected in the TwinCAT System Manager (Config mode) (see Fig. *“Selecting the ‘Restore default parameters’ PDO”*)

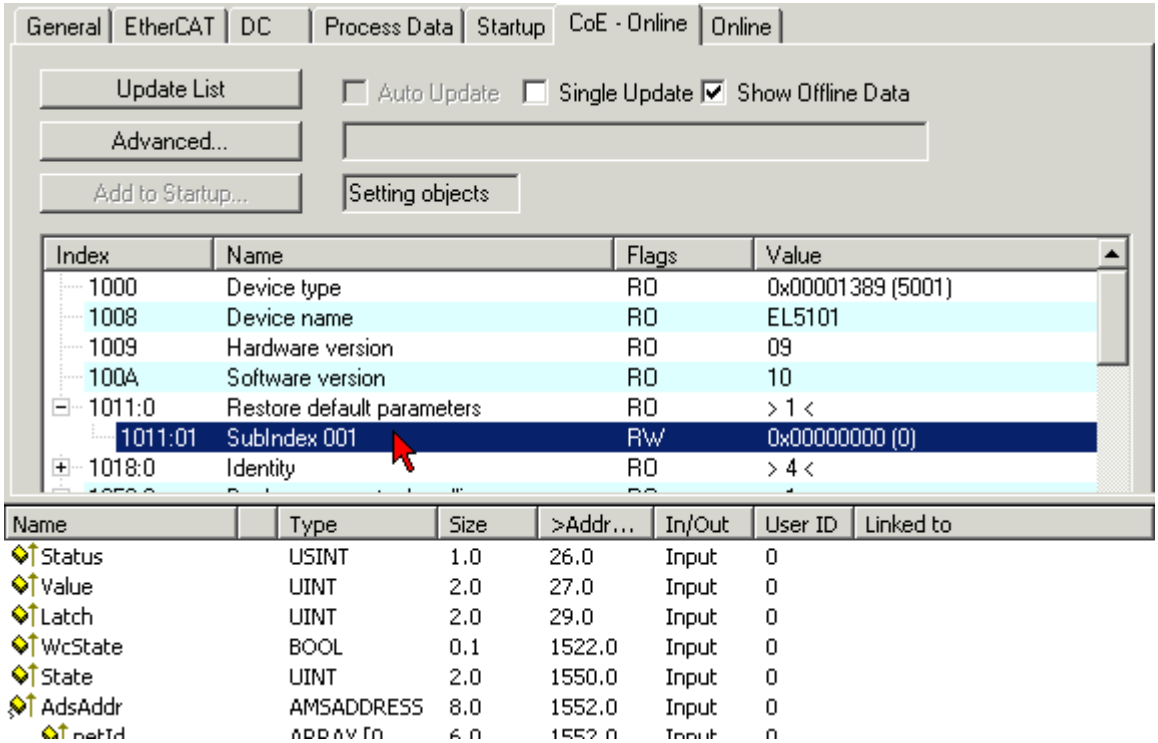


Fig. 191: Selecting the "Restore default parameters" PDO

Double-click on SubIndex 001 to enter the Set Value dialog. Enter the value **1684107116** in field "Dec" or the value **0x64616F6C** in field "Hex" and confirm with *OK* (Fig. *“Entering a restore value in the Set Value dialog”*).
All backup objects are reset to the delivery state.

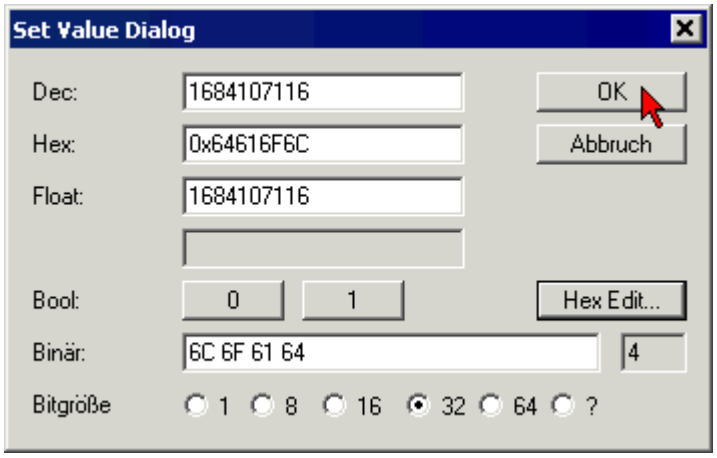


Fig. 192: Entering a restore value in the Set Value dialog

Note	**Alternative restore value** In some older terminals the backup objects can be switched with an alternative restore value:Decimal value: "1819238756", Hexadecimal value: "0x6C6F6164"An incorrect entry for the restore value has no effect.

10.5 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

Beckhoff Support

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

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

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

Beckhoff Service

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

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

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

List of illustration