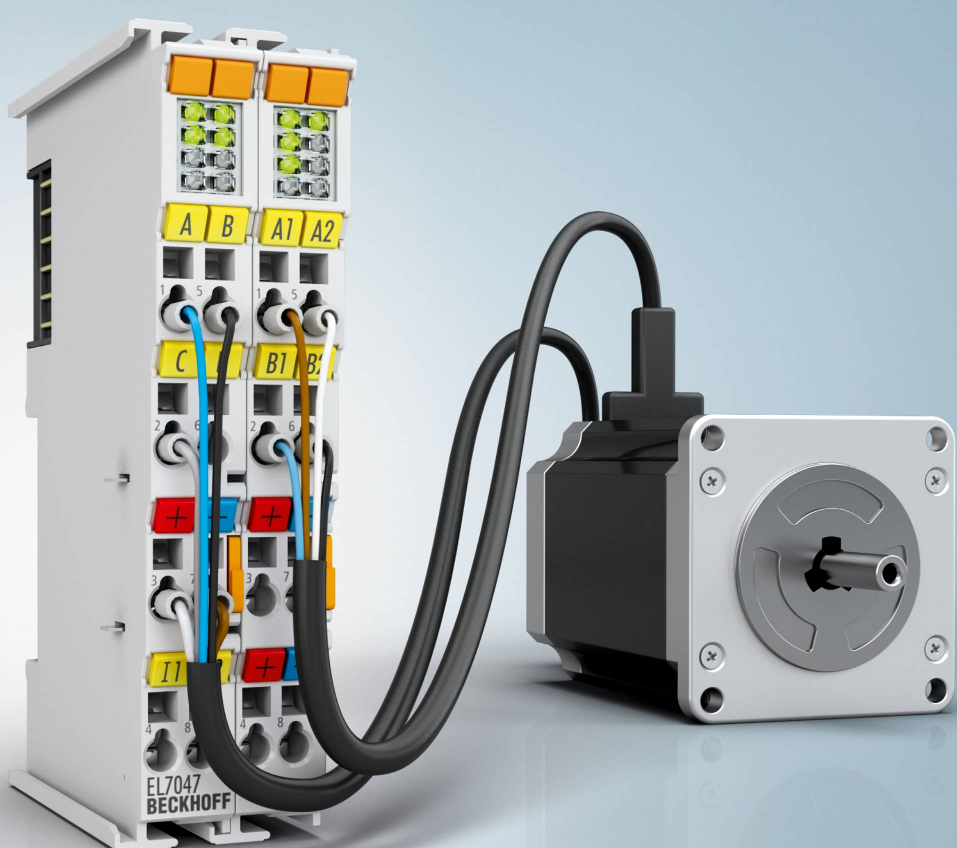
BECKHOFF New Automation Technology

Documentation | EN

EL70x7

Stepper Motor Terminals, vector control



18.09.2020 | Version: 1.9

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER

Serious risk of injury!

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

⚠ WARNING

Risk of injury!

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

⚠ CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
1.9	• Update chapter "EL7047 – Introduction" • Update chapter "EL7047 – Technical data" • Update chapter "Technology" • Update structure
1.8	• Note for fuse protection of the supply voltage added • Update revision status • Update structure
1.7	• Update revision status • Update structure
1.6	• Update chapter „Foreword" • Update revision status • Update structure
1.5	• Update chapter „Technical data" • Update chapter „Installation" • Update revision status • Update structure
1.4	• Update chapter „Technical data" • Update chapter „PDO assignment" • Update chapter „Prededined PDO Assignment" • Update revision status • Update structure
1.3	• Update chapter „Technical data" • Update revision status • Update structure
1.2	• Update chapter „Technical data" • Update chapter „Commissioning" • Update chapter „Diagnosis" • Update revision status • Update structure
1.1	• Update Technical data
1.0	• Minor corrections • Layout adaption • 1st public issue
0.4	• Minor corrections • Addenda EL7037
0.3	• Minor corrections
0.2	• Minor corrections
0.1	• Preliminary documentation

1.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

Notes

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

Identification number

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

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

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

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

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

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

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

Syntax: D ww yy x y z u

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

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

Unique serial number/ID, ID number

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

See also the further documentation in the area

- IP67: EtherCAT Box
- Safety: TwinSafe
- Terminals with factory calibration certificate and other measuring terminals

Examples of markings

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

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

Fig. 3: CU2016 switch with serial/ batch number

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

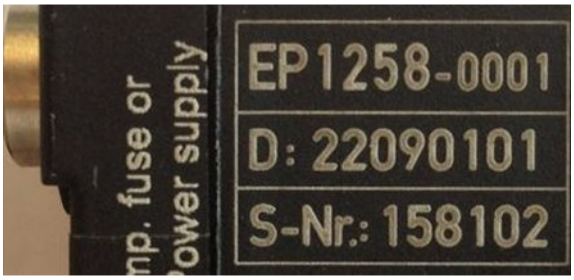


Fig. 5: EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

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

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

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

1.4.1 Beckhoff Identification Code (BIC)

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

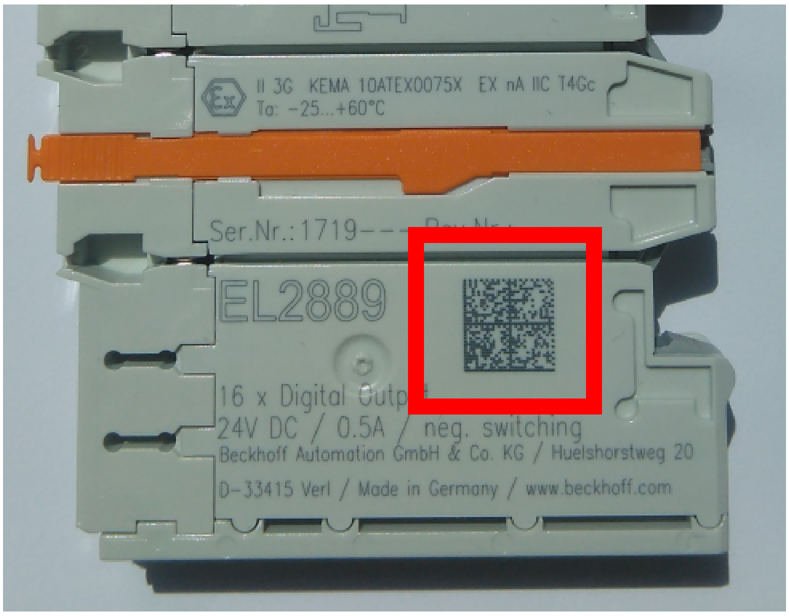


Fig. 9: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it. The data under positions 1 to 4 are always available.

The following information is contained:

Item no.	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	S	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

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

Structure of the BIC

Example of composite information from item 1 to 4 and 6. The data identifiers are marked in red for better display:

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

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

2 Product overview

2.1 EL7037

2.1.1 EL7037 - Introduction

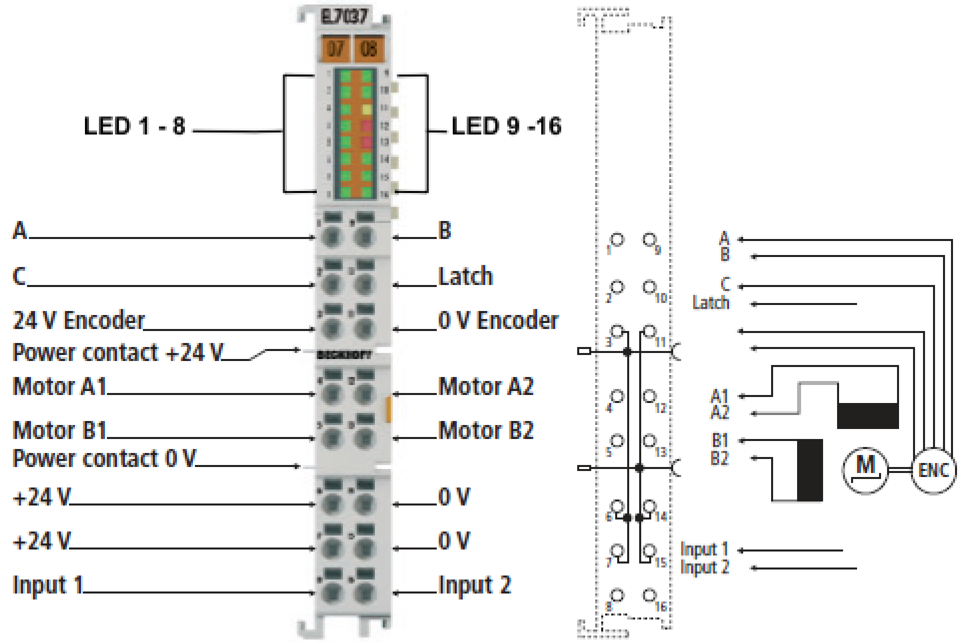


Fig. 10: *EL7037*

Stepper motor terminal, 24 V DC, 1,5 A, vector control

The The EL7037 EtherCAT Terminal is intended for stepper motors with low performance range. The PWM output stages cover a wide range of voltages and currents. Together with two inputs for limit switches, they are located in the EtherCAT Terminal.

The EL7037 can be adjusted to the motor and the application by changing just a few parameters. Stepper motors from the AS10xx series can be operated with vector control. This control technique offers various benefits, such as better dynamics and lower power consumption.

Together with a stepper motor, the EL7037 represents an inexpensive compact drive.

Quick links

Connection instructions

• Section "Installation and wiring",
- LEDs and pin assignment [▶ 56]
- Connection examples [▶ 58]

Commissioning instructions

• Section "Commissioning",
- Installation under TwinCAT [▶ 92]
- Integration into the NC configuration [▶ 139]
- Basic principles: "Positioning interface" [▶ 172]

Configuration instructions

• Section "Commissioning",
- Configuring the main parameters - Settings in the CoE register [▸ 144]
- Configuring the main parameters - NC settings

• Section "Configuration with the TwinCAT System Manager",
- Object description and parameterization [▸ 187]

Application example

• Section "Commissioning",
- Application example [▸ 152]

2.1.2 EL7037 - Technical data

Technical data	EL7037
Number of outputs	1 stepper motor, 2 phases
Number of digital inputs	2 limit position, 4 for an encoder system
Number of digital outputs	1 configurable for brake (0.5 A)
Supply voltage	24 V DC (-15 %/+20 %)
Output current without fan cartridge ZB8610	1.5 A (overload- and short-circuit-proof)
Output current with fan cartridge ZB8610	3.0 A (overload- and short-circuit-proof)
Operating modes	Standard mode (velocity direct / position controller) Field-oriented control (extended velocity mode / extended position mode) Sensorless operation Travel distance control (positioning interface)
Maximum step frequency	1000, 2000, 4000, 8000 or 16000 full steps/s (configurable)
Step pattern	up to 64-fold micro stepping (automatic switching, speed-dependent)
Current controller frequency	approx. 30 kHz
Encoder pulse frequency	maximum 400,000 increments/s (4-fold evaluation)
Input signal voltage "0"	-3 V … 2 V
Input signal voltage "1"	3.7 V … 28 V
Input current	typ. 5 mA
Diagnostics LED	Warning strand A and B, error strand A and B, power, enable
Resolution	approx. 5,000 positions in typical applications (per revolution)
Power supply	via the E-bus, encoder/driver stage: via the power contacts, motor: via terminal contacts
Current consumption from the E-bus	typ. 100 mA
Electrical isolation	500 V (E-bus/signal voltage)
Support NoCoEStorage [▸ 35]	yes
Configuration	no address setting required Configuration via TwinCAT System Manager
Weight	approx. 60 g
Permissible ambient temperature range during operation	0□ ... +55□
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 (connected width: 12 mm)
Installation	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-27, see Installation instructions for enhanced mechanical load capacity [▸ 50]
EMC immunity/emission	according 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	IP 20
Installation position	without fan cartridge ZB8610: standard installing position with fan cartridge ZB8610: standard installing position, other installing positions (example 1 & 2) see notice [▸ 47]
Approval	CE cULus [▸ 55]

2.2 EL7047

2.2.1 EL7047 - Introduction

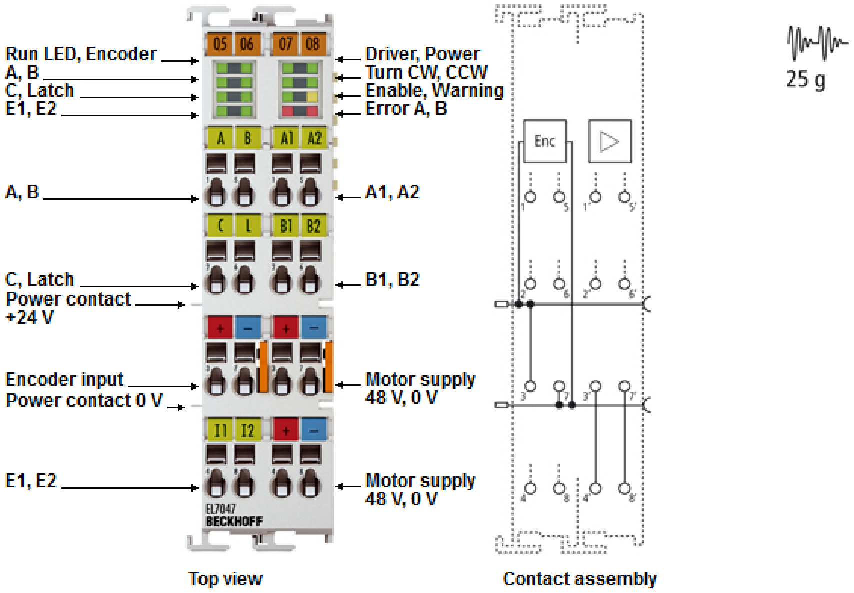


Fig. 11: *EL7047*

Stepper motor terminal, 48 V_{DC}, 5 A, vector control

The EL7047 EtherCAT Terminal is intended for stepper motors with medium performance range. The PWM output stages cover a wide range of voltages and currents. Together with two inputs for limit switches, they are located in the EtherCAT Terminal.

The EL7047 can be adjusted to the motor and the application by changing just a few parameters. 64-fold micro-stepping ensures particularly quiet and precise motor operation.
Field-oriented control can be selected for AS1xxx series stepper motors from Beckhoff Automation. This offers a number of advantages, such as a better dynamics and lower power consumption.

Together with a stepper motor and an encoder, the EL7047 represents an inexpensive small servo axis.

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

Quick links

Connection instructions

• Section "Installation and wiring",
- LEDs and pin assignment [▸ 61]
- Connection examples [▸ 63]

Commissioning instructions

• Section "Commissioning",
- Installation under TwinCAT [▸ 92]
- Integration into the NC configuration [▸ 139]
- Basic principles: "Positioning interface" [▸ 172]

Configuration instructions

• Section "Commissioning",
- Configuring the main parameters - Settings in the CoE register [▸ 144]
- Configuring the main parameters - NC settings

• Section "Configuration with the TwinCAT System Manager",
- Object description and parameterisation [▸ 213]

Application example

• Section "Commissioning",
- Application example [▸ 152]

2.2.2 EL7047 - Technical data

Technical data	EL7047
Number of outputs	1 stepper motor, 2 phases
Number of digital inputs	2 limit position, 4 for an encoder system
Number of digital outputs	1 configurable for brake (0.5 A)
Supply voltage	8 … 48 V_{DC}
Output current without fan cartridge ZB8610	5 A (overload- and short-circuit-proof)
Output current with fan cartridge ZB8610	6.5 A (overload- and short-circuit-proof)
Operating modes	Standard mode (velocity direct / position controller) Field-oriented control (extended velocity mode / extended position mode) Sensorless operation Travel distance control (positioning interface)
Maximum step frequency	1000, 2000, 4000, 8000 or 16000 full steps/s (configurable)
Step pattern	up to 64-fold micro stepping (automatic switching, speed-dependent)
Current controller frequency	approx. 30 kHz
Encoder pulse frequency	maximum 400,000 increments/s (4-fold evaluation)
Input signal voltage "0"	-3 V … 2 V
Input signal voltage "1"	3.7 V … 28 V
Input current	typ. 5 mA
Diagnostics LED	Warning strand A and B, error strand A and B, power, enable
Resolution	approx. 5,000 positions in typical applications (per revolution)
Power supply	via the E-bus, encoder/driver stage: via the power contacts, motor: via terminal contacts
Current consumption from the E-bus	typ. 140 mA
Electrical isolation	500 V (E-bus/signal voltage)
Support NoCoEStorage [▸ 35]	yes
Configuration	no address setting required Configuration via TwinCAT System Manager
Weight	approx. 105 g
Permissible ambient temperature range during operation	0□ ... +55□
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 27 mm x 100 mm x 70 mm (connected width: 24 mm)
Installation	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6/EN 60068-2-27, see Installation instructions for enhanced mechanical load capacity [▸ 50]
EMC immunity/emission	according 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	IP 20
Installation position	without fan cartridge ZB8610: standard installing position with fan cartridge ZB8610: standard installing position, other installing positions (example 1 & 2) see notice [▸ 47]
Approval	CE cULus [▸ 55]

2.3 Technology

The EL70x7 stepper motor terminal integrates a compact Motion Control solution for stepper motors in a very compact design.

The user can control stepper motors in the low to medium performance range. With an output current of up to 5 A, the EL7047 can achieve a considerable torque of e.g. 5 Nm at a standard stepper motor. The supply voltage of up to 48 V_{DC} allows high speeds with good torque and thus high mechanical performance. The stepper motor and an incremental encoder can be connected directly to the EL70x7.

The stepper motor terminal provides three basic modes of operation.
In standard mode [▸ 24] all unipolar and bipolar stepper motors that comply with the specifications of the corresponding EL70x7 can be controlled. Two currents with sine/cosine curve are provided. The current is clocked with 64 kHz and resolved with up to 64-fold microstepping to achieve a smooth current.

Extended mode [▸ 25] is based on field-oriented control. This mode can only be used for stepper motors from Beckhoff. The current is not only provided, but controlled in a comprehensive manner. Typical stepper motor problems such as pronounced resonance are therefore finally a thing of the past. Furthermore, the current is adjusted depending on the load, thereby enabling considerable energy savings and lower thermal loads at the stepper motor.

In sensorless mode [▸ 27] stepper motors from Beckhoff can be controlled load-dependent without a feedback system.

Realisation of more demanding positioning tasks

More demanding positioning tasks can be realised via the TwinCAT automation software from Beckhoff. Like other axes, the stepper motor terminals are integrated via the TwinCAT System Manager and can be used like standard servo axes. Special stepper motor features, such as speed reduction in the event of large following errors, are automatically taken into account via the *stepper motor axis* option. The effort for changing from a servomotor to a stepper motor - and back - is no greater than changing from one fieldbus to another one under TwinCAT.

The output stages of the stepper motor terminals have an overload protection in the form of an overtemperature warning and switch-off. Together with short circuit detection, diagnostic data are accessible in the process image of the controller. In addition, this status is displayed by the Bus Terminal LEDs, along with other information. The output stage is switched on via an Enable-Bit. The motor current can be set and reduced via a parameter value.

Optimum adaptation to the motor and the implementation of energy-saving features require minimum programming effort. Since all data are set in the form of parameters in the CoE register, it is easily possible to replace an EtherCAT Terminal or store certain parameters for transfer to the next project. It is therefore no longer necessary to transfer certain potentiometer settings or to document DIP switch settings.

2.3.1 Stepper motor

Stepper motors are electric motors and are comparable with synchronous motors. The rotor is designed as a permanent magnet, while the stator consists of a coil package. The frequency of the stator rotary field is always in a fixed ratio relative to the rotor speed. In contrast to synchronous motors, stepper motors have a large number of pole pairs. In a minimum control configuration, the stepper motor is moved from pole to pole, or from step to step.

Stepper motors have been around for many years. They are robust, easy to control, and provide high torque. In many applications, the step counting facility saves expensive feedback systems. Even with the increasingly widespread use of synchronous servomotors, stepper motors are by no means "getting long in the tooth". They are considered to represent mature technology and continue to be developed further in order to reduce costs and physical size, increase torque and improve reliability. For a standard stepper motor with 200 full steps, the best possible positioning accuracy is approx. 1.8°.

Today, the most widely used type in industry is the hybrid stepper motor type. In this type of motor the rotor consists of a toothed iron core with one or a few permanent magnets in the rotor core. The rotor is designed such that the polarity of successive teeth is inverse. This enables the production of motors with a high number of steps, which is essential for positioning accuracy, combined with a relatively high torque. The electrical behaviour of such a hybrid stepper motor is comparable with a multipole synchronous servomotor. However, thanks to the synchronous toothing of stator and rotor, hybrid stepper motors offer a significantly higher cogging torque.

Hybrid stepper motors with two or more phases are available on the market. Since the terminals described here are designed for two-phase motors, the description focuses on the two-phase type, with the phases referred as A and B in this documentation.

The development of the EL70x7 EtherCAT Terminals for the Beckhoff EtherCAT Terminal system opens up new fields of application. The use of microstepping, the latest semiconductor technology and field-oriented control (only with Beckhoff motors) offers many advantages:

- smoother operation
- avoidance of resonance
- reduced energy consumption
- lower thermal load on the motor
- minimum electromagnetic emissions
- long cable lengths
- simpler handling
- reduced size of the power electronics
- simple integration into higher-level systems
- integrated feedback system

Stepper motor parameters

• Mechanical system

Irrespective of the drive and the stepper motor itself, the configuration of the mechanism attached to the motor shaft has significant influence on the achievable control quality.

Natural resonances, load resonances, gear backlash (loose) and static friction have negative affect on the controllability of the drive system. This often requires "softer" controller parameterisation, which in turn leads to a higher position lag in the system. Sliding friction can result in reduced efficiency (due to increased energy demand), but on the other hand it can have a positive effect on the control stability, due to its dampening effect.

As a general rule, the "stiffer" the mechanics of a drive system, the easier it is to control, which is beneficial for achieving a small position lag in the drive system.

• Speed

Stepper motors have low maximum speed, which is usually specified as a maximum step frequency.

• Number of phases

Motors with 2 to 5 phases are common. The EL70x7 EtherCAT Terminals support 2-phase motors. 4-phase motors are basically 2-phase motors with separate winding ends. They can be connected directly to the EtherCAT Terminal.

• Torque

Refers to the maximum motor torque at different speeds. This parameter is usually represented by a characteristic curve. Stepper motors have comparatively high torque in the lower speed range. In many applications, this enables them to be used directly without gearing. Compared with other motors, stepper motors can quite easily provide a holding moment of the same order of magnitude as the torque.

• Cogging torque

In many cases the stepper motors design results in high cogging torque, which can lead to relatively strong natural resonance in a motor- and load-dependent speed range. In relation to the cogging torque, increased inertia often leads to a less strong resonance and smoother operation.

- **Mass moment of inertia**

In standard mode, the key parameter of the mechanical system is the mass moment of inertia J_{Σ}. It is essentially composed of the mass moment of inertia of the stepper motor rotor J_M and the mass moment of inertia of the connected load J_L. The friction moment J_{fric} and the moment of inertia of the encoder J_{Enc} can be neglected in a first approximation.

$J_{\Sigma} \approx J_M + J_L$

The ratio between the load torque and the motor torque is defined by the constant k_J.

$k_J \approx J_L / J_M$

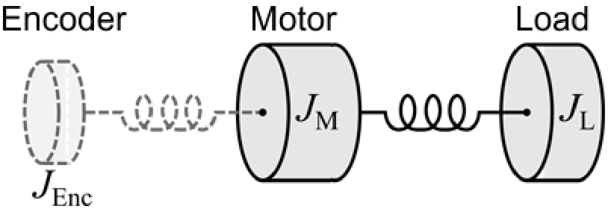


Fig. 12: *Simplified representation of the mass moments of inertia*

As a first approximation, the coupling of the individual masses over the rotor shaft can be modelled as two-mass oscillator. The resonance frequency between the motor and the encoder lies in a relatively high frequency range, which is usually not relevant for stepper motor drives and is suppressed within the drive by low-pass filtering. The resonance frequency between the motor and the load is frequently in the range between 20 and 500 Hz. It is therefore often in the operating range of the drive control. Design measures to reduce the influence of the load resonance include a small load ratio k_J and a rigid coupling of the motor shaft to the connected load.

- **Resonance**

At certain speeds, stepper motors run less smoothly. This phenomenon is particularly pronounced when the motor runs without coupled load, in which case it may even stop (in standard mode). This is caused by resonance. A distinction can roughly be made between

- resonances in the lower frequency range up to approx. 250Hz; and
- resonances in the medium to upper frequency range.

Resonances in the medium to upper frequency range essentially result from electrical parameters such as inductance of the motor winding and supply line capacity. They can be controlled relatively easily through high pulsing of the control system.

Resonances in the lower range essentially result from the mechanical motor parameters. Apart from their impact on smooth running, such resonances can lead to significant loss of torque, or even loss of step of the motor, and are therefore particularly undesirable.
In principle, the stepper motor represents an oscillatory system (comparable to a mass/spring system), consisting of the moving rotor with a moment of inertia and a magnetic field that creates a restoring force that acts on the rotor. Moving and releasing the rotor creates a damped oscillation. If the control frequency corresponds to the resonance frequency, the oscillation is amplified, so that in the worst case the rotor will no longer follow the steps, but oscillate between two positions.
The EL70x7 EtherCAT Terminals prevent this effect thanks to their field-oriented control (Extended Operation Modes) for all Beckhoff stepper motors.

- **Torque constant**

In the Extended Operation Modes the torque constant k_T is used as an additional parameter for the mechanical controlled system. It indicates the ratio between the torque-forming motor current and the active torque at the shaft. However, since the field-oriented operating mode is not common for stepper motors, the torque constant is usually not listed in the motor data sheet.

Electrical system

• Nominal voltage, supply voltage and winding resistance

Under steady-state conditions, the rated current at the rated voltage depends on the winding resistance. This voltage should not be confused with the supply voltage of the power output stage in the EtherCAT Terminal. The EL70x7 applies a controlled current to the motor winding. If the supply voltage falls below the nominal voltage, the power output stage can no longer apply the full current, resulting in a loss of torque. It is desirable to aim for systems with small winding resistance and high supply voltage in order to limit warming and achieve high torque at high speeds.

• Induced countervoltage

Like servomotors, hybrid stepper motors induce a voltage u_i [Vs/rad] in the stator winding of the motor, which is proportional to the speed. It is also referred to as Back Electromotive Force (BEMF). In conjunction with the DC link voltage (motor voltage), the induced countervoltage determines the physically achievable maximum speed of the motor.

The ratio of the magnitude of the induced countervoltage and the motor speed varies depending on the design and is described via the voltage constant k_e.

$u_i = k_e \cdot \omega_m$

The motor parameter k_e [mV/(rad/s)] is required for step loss recognition without encoder and for sensorless control.

For stepper motors where the voltage constant is not specified in the data sheet, it can be relatively easily determined using a digital multimeter. To this end the motor to be measured must be operated (within the rated speed range) by an auxiliary motor via a coupling with constant speed. The motor phases of the motor to be measured must be open (not connected to the terminal or shorted). The multimeter can then be used to determine the RMS value of the induced countervoltage, and therefore the voltage constant, at one of the two open motor phases (A or B).

• Step angle

The step angle indicates the angle travelled during each step. Typical values are 3.6°, 1.8° and 0.9°. This corresponds to 100, 200 and 400 steps per motor revolution. Together with the downstream transmission ratio, this value is a measure for the positioning accuracy. For technical reasons, the step angle cannot be reduced below a certain value. Positioning accuracy can only be improved further by mechanical means (transmission). An elegant solution for increasing the positioning accuracy is the microstepping function offered by the EL70x7. It enables up to 64 intermediate steps. The smaller "artificial" step angle has a further positive effect: The drive can be operated at higher speed, yet with the same precision. The maximum speed is unchanged, despite the fact that the drive operates at the limit of mechanical resolution.

• Winding resistance, winding inductance

The winding inductance and winding resistance of the stepper motor stator determine the electrical motor time constant $T_e = L / R$, which is a key parameter for current controller configuration.

Specifying the stepper motor

1. Determine the required positioning accuracy and hence the step resolution. The first task is to determine the maximum resolution that can be achieved. The resolution can be increased via mechanical gear reduction devices such as spindles, gearing or toothed racks. The 64-fold microstepping of the stepper motor terminals also has to be taken into account.
2. Determine mass m and moment of inertia (J) of all parts to be moved
3. Calculate the acceleration resulting from the temporal requirements of the moved mass.
4. Calculate the forces from mass, moment of inertia, and the respective accelerations.

5. Convert the forces and velocities to the rotor axis, taking account of efficiencies, moments of friction and mechanical parameters such as gear ratio. It is often best to start the calculation from the last component, usually the load. Each further element transfers a force and velocity and leads to further forces or torques due to friction. During positioning, the sum of all forces and torques acts on the motor shaft. The result is a velocity/torque curve that the motor has to provide.
6. Using the characteristic torque curve, select a motor that meets these minimum requirements. The moment of inertia of the motor has to be added to the complete drive. Verify your selection. In order to provide an adequate safety margin, the torque should be oversized by 20% to 30%. The optimisation is different if the acceleration is mainly required for the rotor inertia. In this case, the motor should be as small as possible.
7. Test the motor under actual application conditions: Monitor the housing temperatures during continuous operation. If the test results do not confirm the calculations, check the assumed parameters and boundary conditions. It is important to also check side effects such as resonance, mechanical play, settings for the maximum operation frequency and the ramp slope.
8. Different measures are available for optimising the performance of the drive: using lighter materials or hollow instead of solid body, reducing mechanical mass. The control system can also have significant influence on the behaviour of the drive. The Bus Terminal enables operation with different supply voltages. The characteristic torque curve can be extended by increasing the voltage. In this case, a current increase factor can supply a higher torque at the crucial moment, while a general reduction of the current can significantly reduce the motor temperature. For specific applications, it may be advisable to use a specially adapted motor winding.

2.3.2 Standard mode

Stepper motors were originally operated with very simple output stages, which were only able to switch the voltage of the motor phases separately (nowadays current control takes place via PWM with pulse-width modulation as standard). Initially the motor phases there were controlled individually in turn. A switching sequence in the positive direction of rotation corresponds to the switching sequence (+A, +B, -A, -B). Sequential switching results in rather irregular operation in this mode. In order to make the operation smoother, so-called microstepping was introduced later, in which the four set voltages were extended by intermediate values (e.g. from a stored sine table). These days, microstepping based on 64 steps is commonly used.

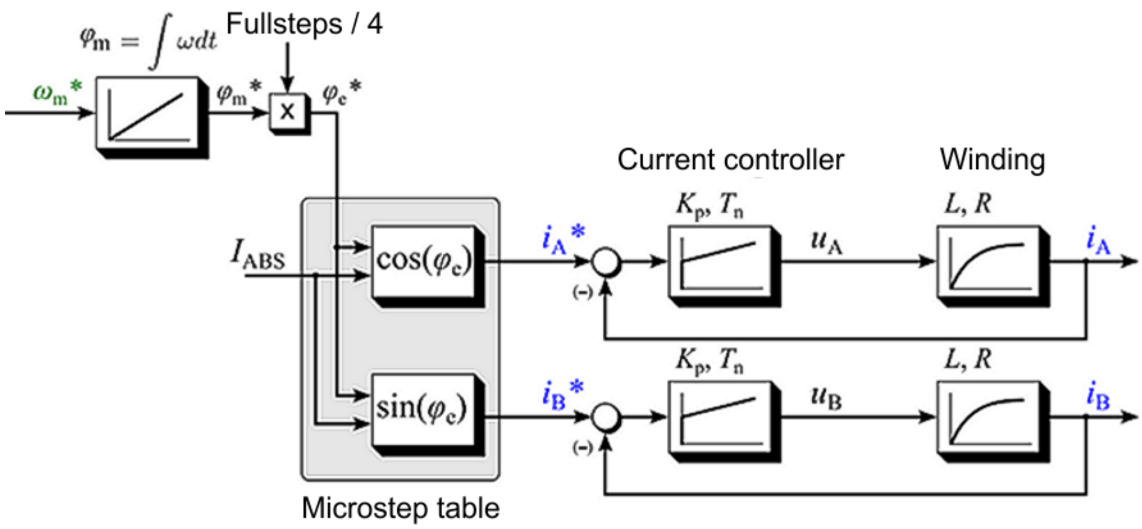


Fig. 13: *Control structure of a standard stepper motor drive*

Neglecting the sampling resulting from the microstepping, the motor current I as function of the electrical angle φe and of the magnitude of the motor current I_{ABS} (when using a current controller) can be described as follows:

$I(\varphi_e) = I_A + jI_B = I_{ABS} cos(\varphi_e) + jI_{ABS} sin(\varphi_e)$

Represented by magnitude and angle:

$I(\varphi_e) = I_{ABS} \cdot e^{j\varphi e}$

It follows that a rotation of the electrical angle φe is equivalent to four full steps. (A stepper motor with 200 full steps therefore has 50 pole pairs).

The shaft aligns itself if a constant current is set with no load at the motor shaft. Within a pole pairs the shaft points in the direction of the active stator field.

If an external load is applied to the motor shaft, the shaft is turned out of the field direction, resulting in a load angle (also referred to as angular displacement) (relative to an electric rotation of the angle φe). The load angle depends on the design of the stepper motor itself, the motor current and the torque acting on the shaft. The relationship is non-linear!

If the load angle exceeds a motor-dependent maximum value (i.e. if the maximum machine torque under these boundary conditions is exceeded), the load torque can no longer be maintained by the motor. If the shaft is turned further out of the rotary field, it "tips", resulting in one or more step losses. The "tip angle" may vary between motor types. Often, it lies between around 45° and 65°.

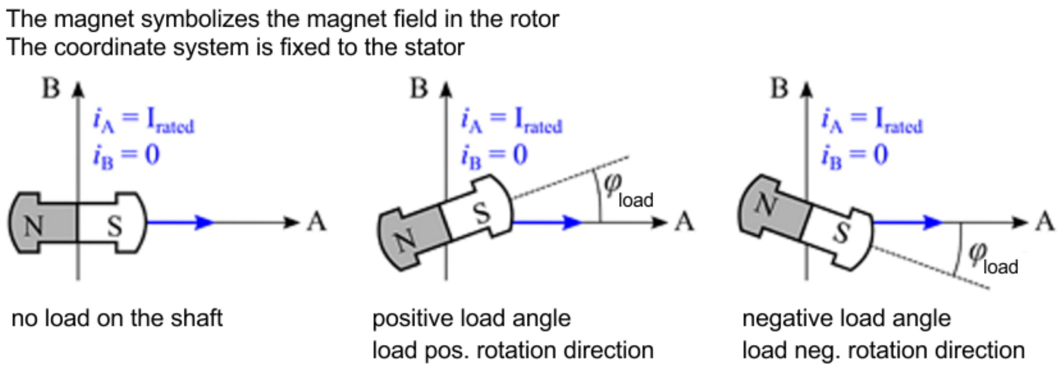


Fig. 14: *Behaviour of the rotor under load*

The load angle is of interest for the user, because it allows conclusions about the load on the shaft. It is measured by evaluating the induced countervoltage* and can be used to optimise the drive system.

2.3.3 Field-oriented control

In the *Extended Operation Modes* the stepper motor is operated like a servomotor, based on the principle of field-oriented control.

Function

The operating behaviour of the motor corresponds to that of a traditional DC motor, with commutation via a mechanical commutator. With a constant exciter field, the torque of the DC machine is directly proportional to the stator current and can be directly influenced by it. The exciter field is generated, depending on the machine type, by permanent magnets or, with a separately excited DC machine, for example, via a separate excitation winding.

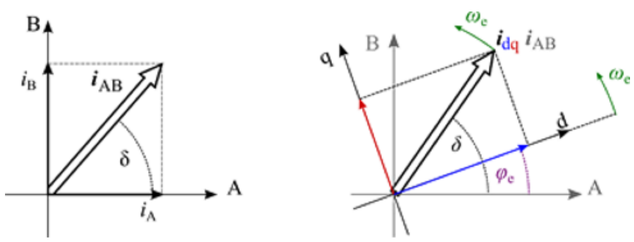


Fig. 15: *Coordinate transformation of field-oriented control*

For servomotors and also hybrid stepper motors, initially there is no direct link between the phase currents and the torque. Field and torque are decoupled mathematically via Park's transformation. Two current components, "d" for "direct" in field direction and "q" for "quadrature" in torque-forming direction, are calculated from the phase currents. Via the torque-forming current component i_q, the torque of the machine can now be regulated directly, like for a DC machine.

A prerequisite is that the rotor position is available with sufficiently high accuracy. For a stepper motor the encoder resolution should be at least 4000 increments per mechanical revolution, in order to achieve adequate positioning accuracy. The minimum encoder resolution also depends on the number of full steps and can be calculated approximately as follows.

$$ENCRESmin\left[\frac{inc}{360°}\right] \geq fullsteps \cdot 12 \geq 4000\left[\frac{inc}{360°}\right]$$

Fig. 16: *Calculation of the resolution*

Commutation determination for Extended Operation Modes

Because the absolute actual position is not available for incremental encoders, on system start-up there is no direct reference to the rotor position, which is required for field-oriented operation. Therefore, the reference between the actual position and the rotor position must be generated at start-up via a commutation determination process. During this process the rotor is moved forward and back several times up to two full steps.

Commutation determination

- The maximum current should be set just below the rated motor current.
- During commutation determination the rotor shaft should not be subject to an external torque. If this condition is not met, the Extended Operation Modes cannot be used.

Control structure

The drive control structure is a cascade control structure with a position control loop and a lower-level speed and current control loop. If a speed setpoint is specified, the external position control loop can be omitted.

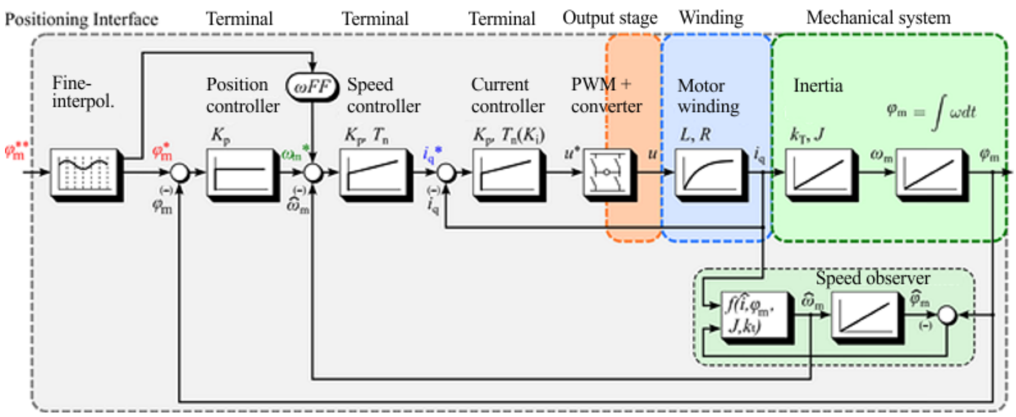


Fig. 17: *Cascade control structure with field-oriented control (Extended Operating modes)*

Motor dependency

Due to the fact that the control is strongly dependent on the motor parameters, the controller parameters and motor behaviour itself, field-oriented control is limited to Beckhoff motors. This mode is not supported for motors from other manufacturers.

Main advantages compared with standard mode

- Low current consumption (almost full load-dependence)
- High efficiency
- Consistent dynamics compared with standard mode
- Step losses are inherently avoided

Requirement

- Encoder with sufficiently high resolution required (minimum 4000 [INC/360°])

- Slightly higher parametrisation effort required (speed controller)
- Commutation determination at startup (due to incremental encoder)
- Only possible with stepper motors from Beckhoff Automation (AS10xx)

2.3.4 Sensorless operation

Because the default operation of a stepper motor with a constant load-independent current is not energy-efficient and leads to a permanently high thermal load, efforts are made to reduce this load.

Function

By analyzing the speed-proportional induced countervoltage, it is possible to control the stator current depending on the load with the aid of a machine model (without sensor/encoder), thereby significantly increasing the efficiency.

Since this operating mode requires a minimum amplitude of the magnitude of the induced countervoltage, sensorless control only works in the medium and upper speed range. In the lower speed range the motor is operated in standard mode. The changeover to sensorless operation take place via a programmable, motor-dependent switching speed. The switching speed is usually in the range between half and three revolutions per second (crossover velocity 1).
When sensorless control is activated, the transient phenomenon results in a slight mechanical jerk of the shaft, which is proportional to the load acting on the shaft.

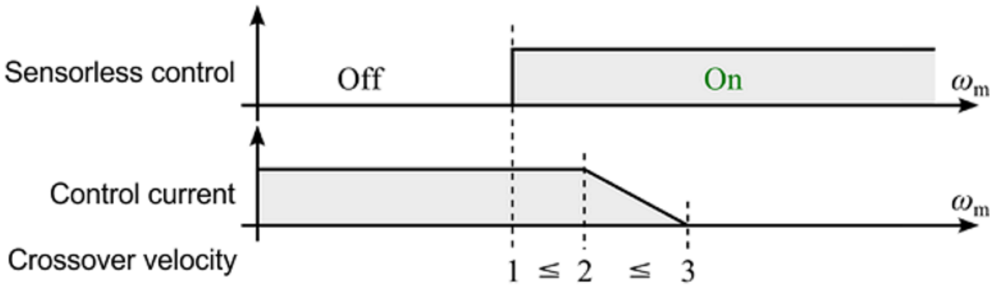


Fig. 18: *Influence of the crossover velocity thresholds (1,2,3) on sensorless control*

After switching on, the control current remains constant up to a second configurable speed and is reduced to a third parameterizable speed via a linear ramp.

A long control current ramp leads to a stronger stabilization of the transient phenomenon of the control. However, it also leads to a longer flowing constant motor current and therefore slightly higher losses.

Motor dependency

Due to the fact that the control is strongly dependent on the motor parameters, the controller parameters and motor behaviour itself, sensorless operation is limited to Beckhoff motors. This mode is not supported for motors from other manufacturers.

Parameterisation

Compared to the other operating modes, the parameterisation effort is relatively high. However, all the required necessary parameters are pre-specified via a startup list for the respective motor types. All that is required during commissioning is an adjustment of the speed control parameters, due to the given mass inertia ratios of the connected loads in the mechanical system.

For the speed controller, in principle the same dependence on the mass moment of inertia and the torque constant applies as in the Extended Operation Modes. Thanks to the lower-level sensorless control it is, however, possible to achieve a better overall result through different parameterisation.

All parameters required for sensorless operation can be found in the table "Overview of parameter settings for individual operating modes [▶ 159]".

Summary

In this mode, above a minimum speed the motor current without encoder is controlled load-dependent. In this way it is possible to realise a particularly cost-effective drive in combination with high efficiency. The achievable dynamic performance of the drive control is slightly reduced compared to the other operating modes.

Advantages compared with standard mode

- Low current consumption (almost full load-dependence)
- High efficiency
- no encoder required

Prerequisites

- relatively high parameterisation effort required (speed controller + additional parameters)
- minimum speed required (if the speed is too low, the motor automatically switches to standard mode)
- dynamic performance somewhat lower than in standard mode
- Only possible with stepper motors from Beckhoff Automation (AS10xx)

2.4 Start-up

For commissioning:

- Install the EL70x7 as described in section Installation [▶ 40].
- Configure the EL70x7 in TwinCAT as described in section Commissioning [▶ 66].

3 Basics communication

3.1 EtherCAT basics

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

3.2 EtherCAT cabling – wire-bound

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

Cables and connectors

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

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

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

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

Recommended cables

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

E-Bus supply

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

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

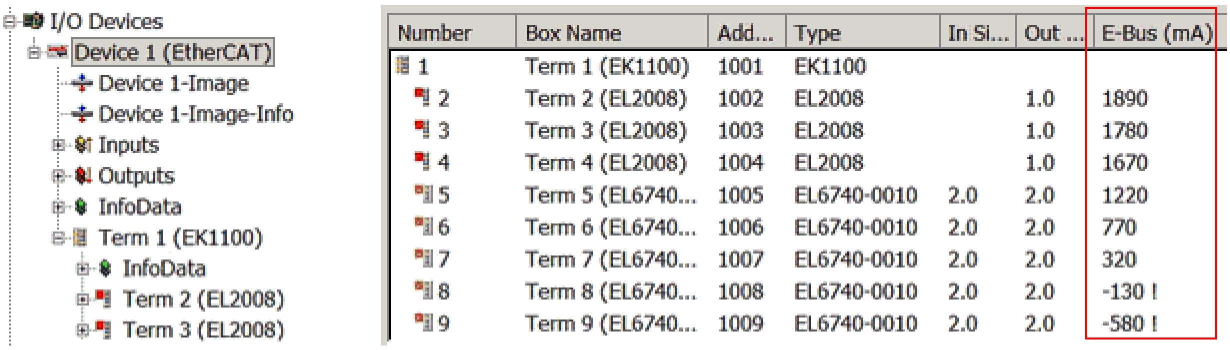


Fig. 19: System manager current calculation

NOTE

Malfunction possible!

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

3.3 General notes for setting the watchdog

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 two 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.

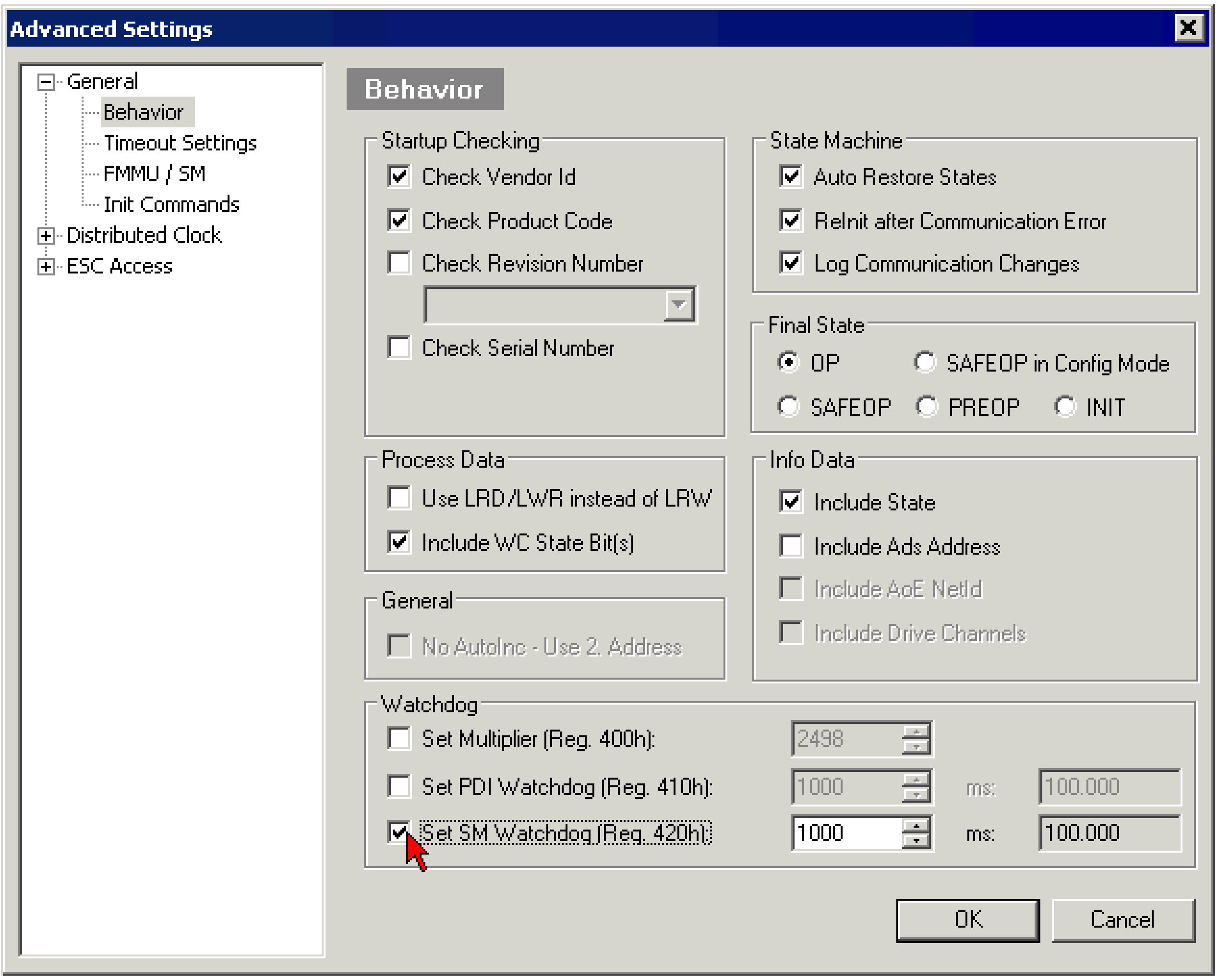


Fig. 20: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 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.

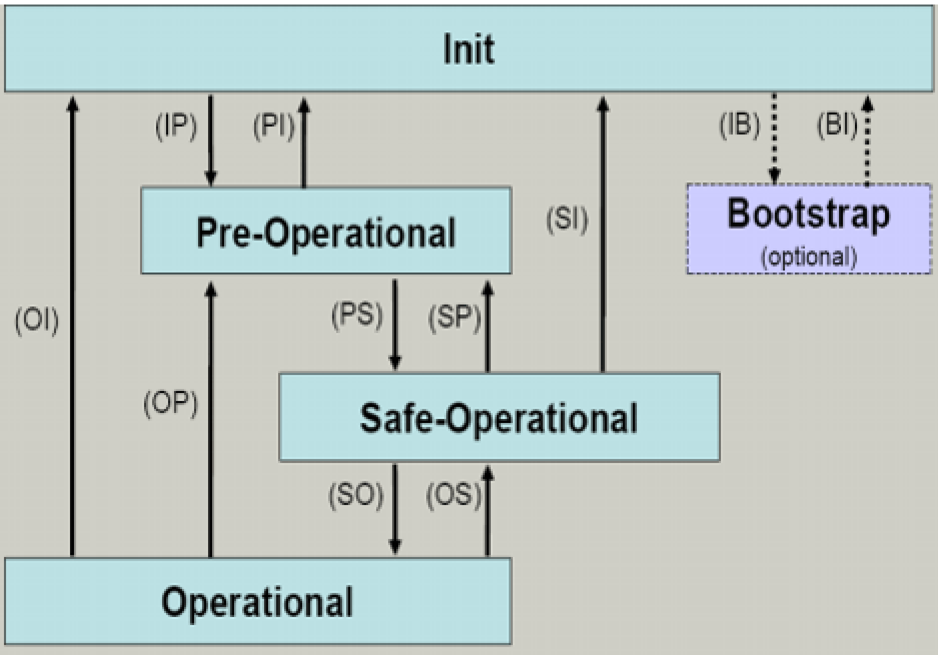


Fig. 21: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

Outputs in SAFEOP state

The default set watchdog [▸ 30] monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

Availability

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

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

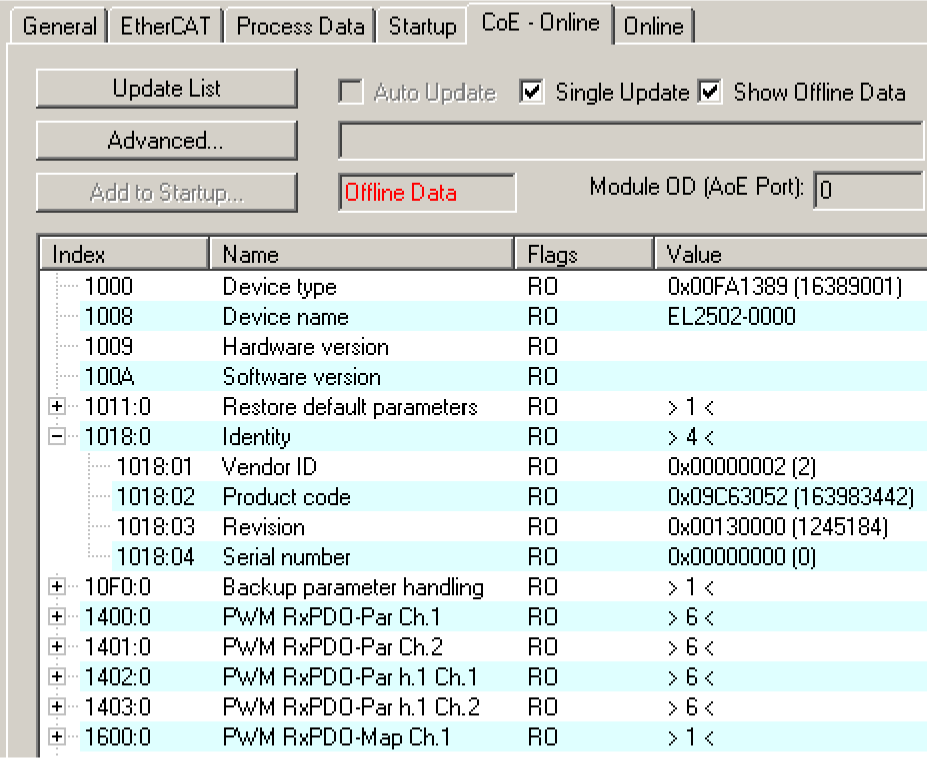


Fig. 22: “CoE Online” tab

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

Data management and function “NoCoeStorage”

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

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

Data management

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

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

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

Startup list

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

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

Recommended approach for manual modification of CoE parameters

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

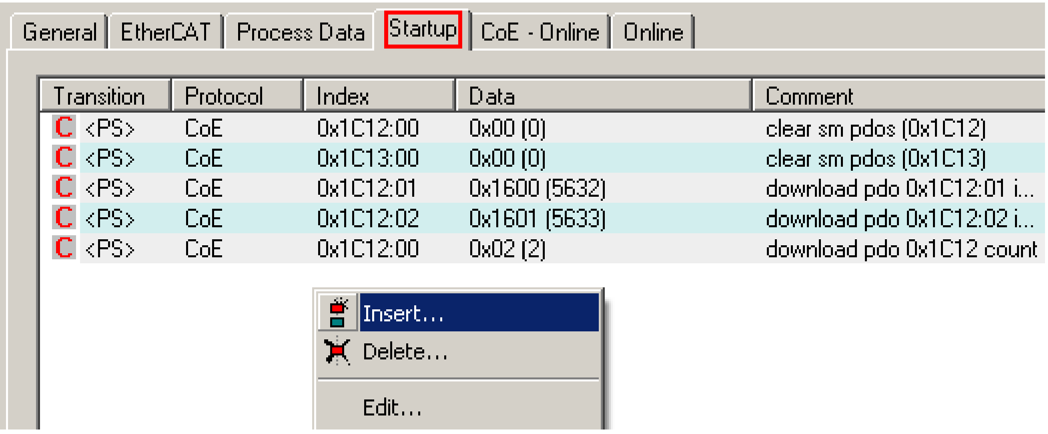


Fig. 23: Startup list in the TwinCAT System Manager

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.

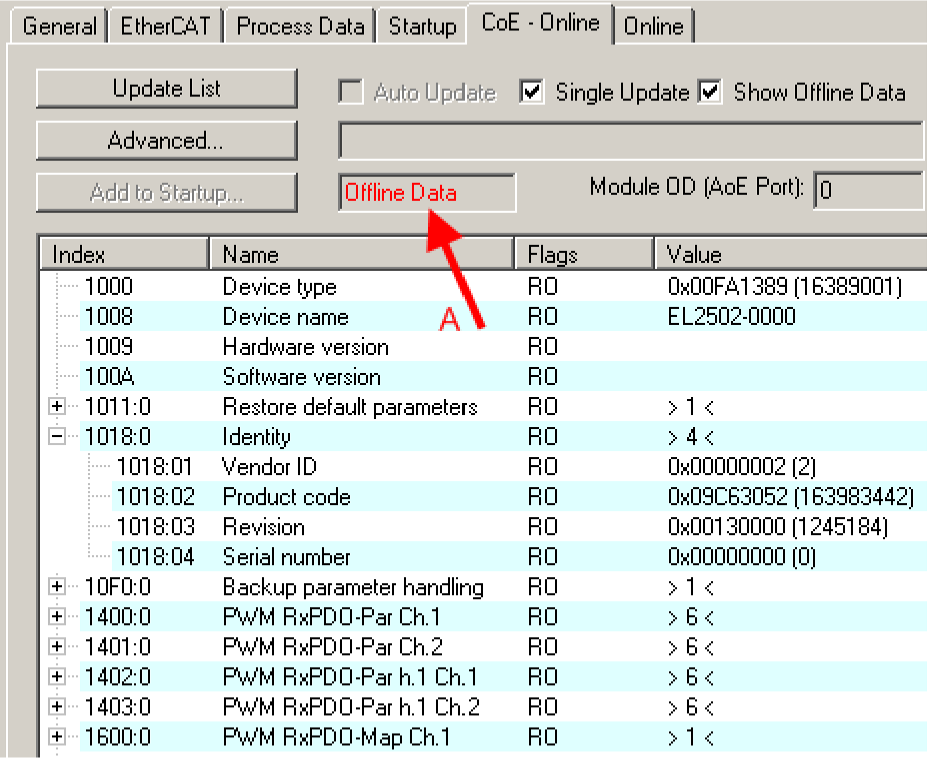


Fig. 24: Offline list

- 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.

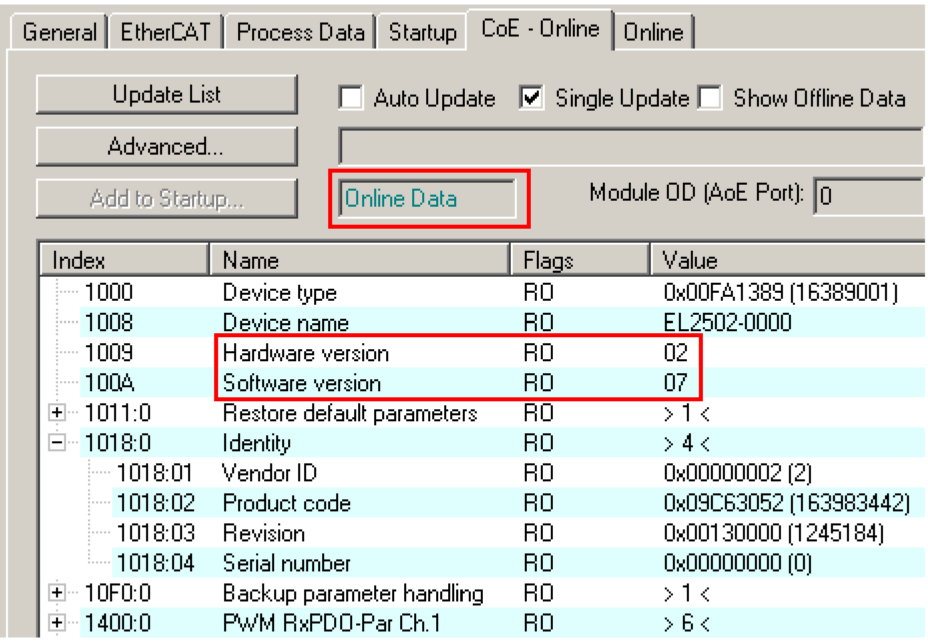


Fig. 25: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

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

3.6 Distributed Clock

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

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

For detailed information please refer to the EtherCAT system description.

4 Installation

4.1 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

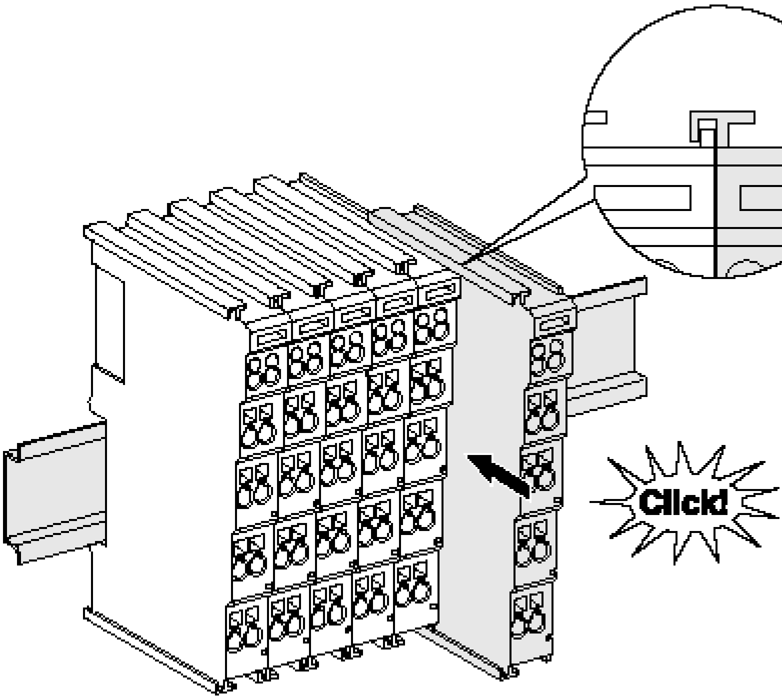


Fig. 26: Attaching on mounting rail

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.

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

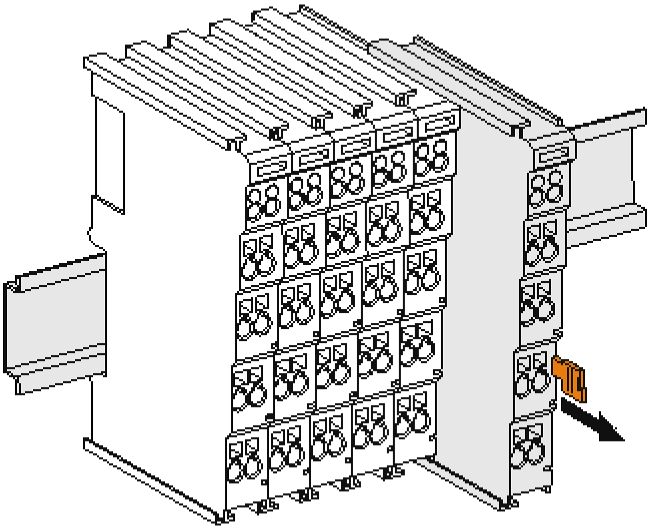


Fig. 27: Disassembling of terminal

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.

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.

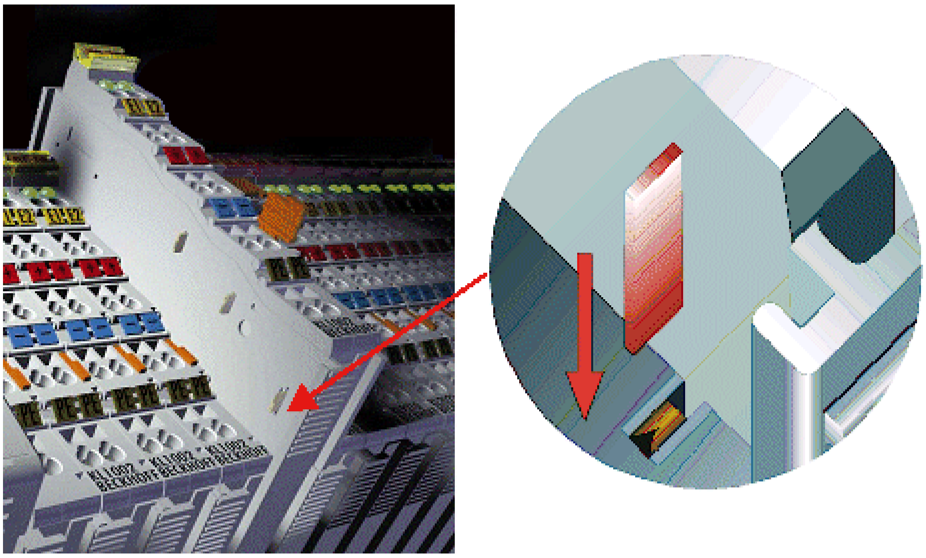


Fig. 28: Power contact on left side

NOTE

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!

4.2 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

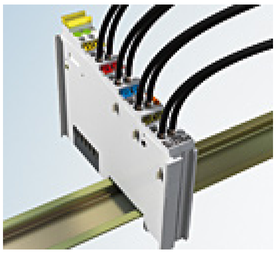


Fig. 29: Standard wiring

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

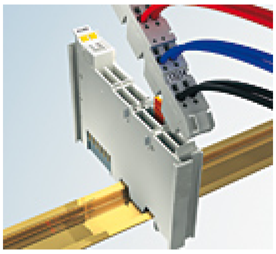


Fig. 30: Pluggable wiring

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)

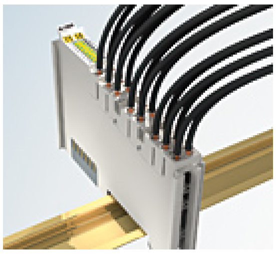


Fig. 31: High Density Terminals

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.

Wiring HD Terminals

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

Ultrasonically "bonded" (ultrasonically welded) conductors

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 [▸ 45] below!

Wiring

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

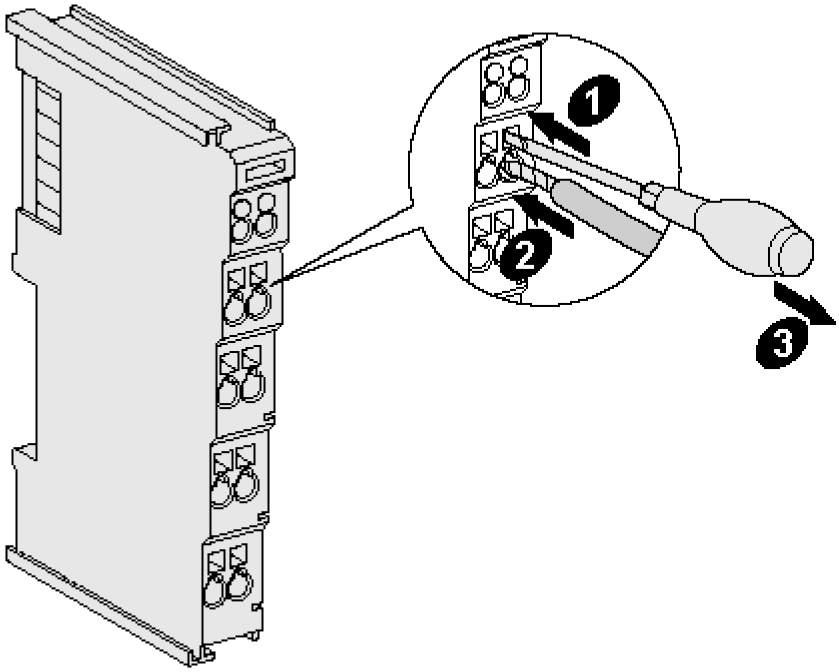


Fig. 32: Mounting a cable on a terminal connection

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 [▸ 44]!)
Wire stripping length	8 ... 9 mm

Shielding

Shielding

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

4.3 Installation position for operation with or without fan

NOTE

Constraints regarding installation position and operating temperature range

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

Prescribed installation position for operation without fan

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

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

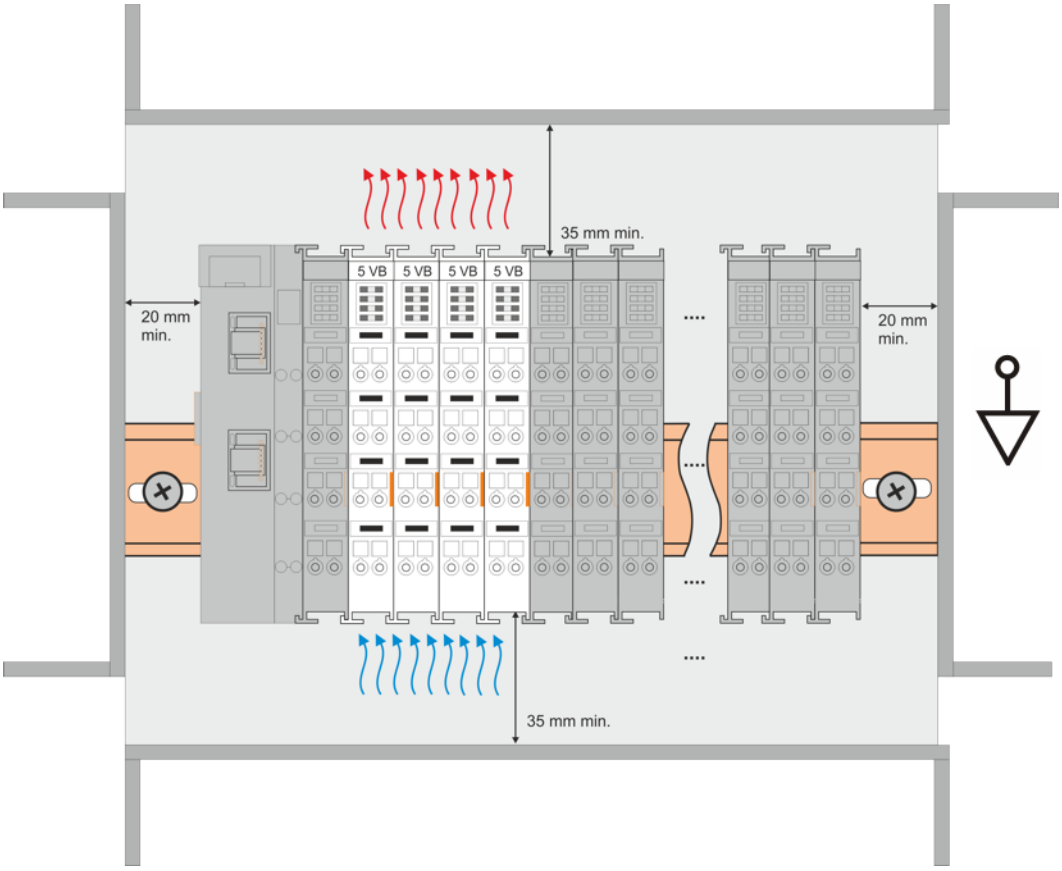


Fig. 33: *Recommended distances of installation position for operating without fan*

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

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

Standard installation position for operation with fan

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

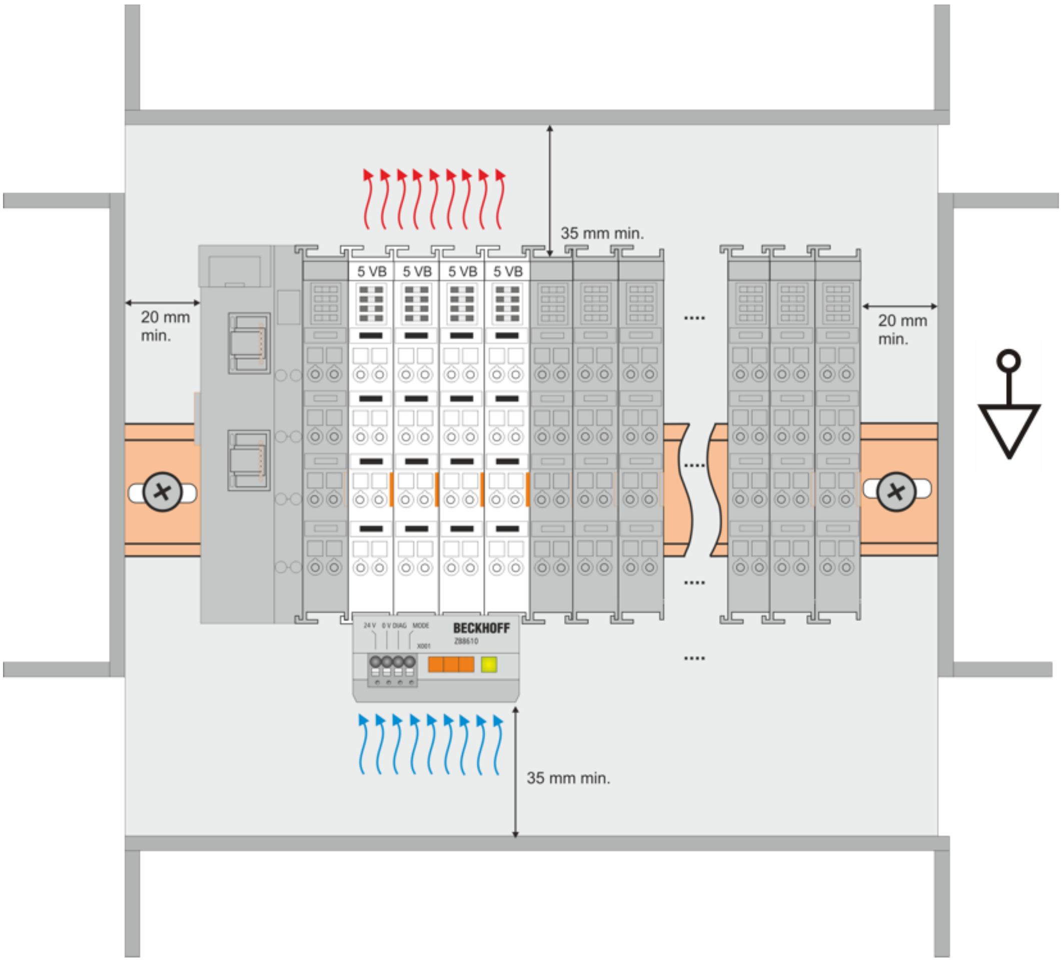


Fig. 34: *Recommended distances for installation position for operation with fan*

Other installation positions

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

See corresponding notes in the Technical Data of the terminal.

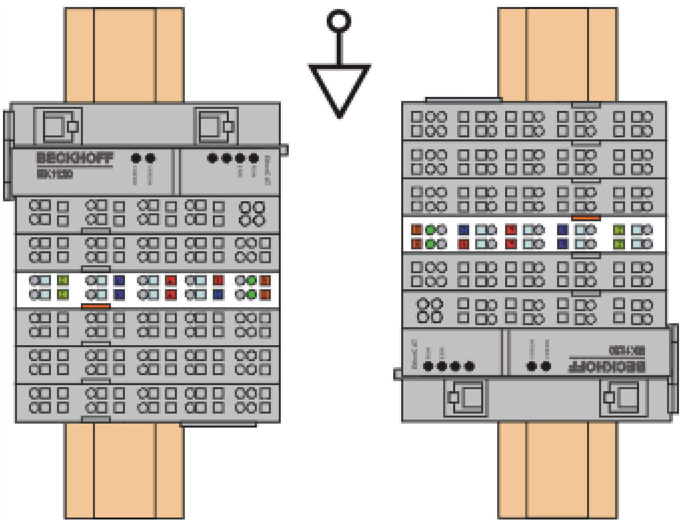


Fig. 35: *Other installation positions, example 1*

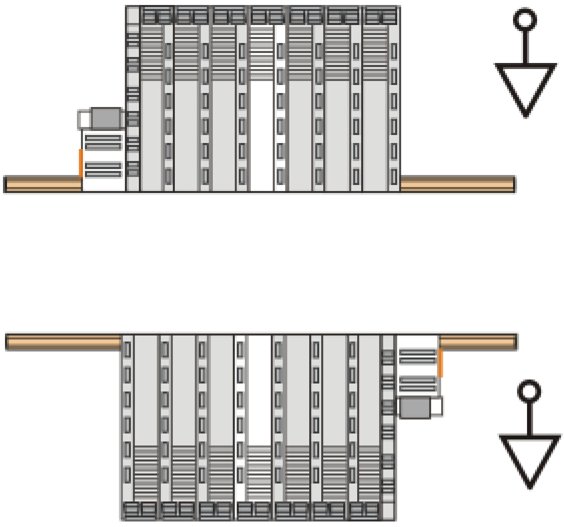


Fig. 36: *Other installation positions, example 2*

4.4 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:

- The enhanced mechanical load capacity is valid for all permissible installation positions
- Use a mounting rail according to EN 60715 TH35-15
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting 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.

4.5 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block

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

Examples for positioning of passive terminals (highlighted)

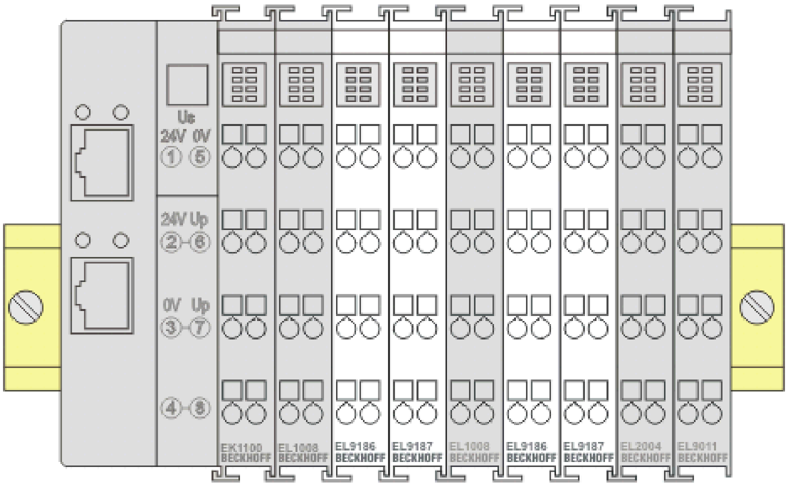


Fig. 37: Correct positioning

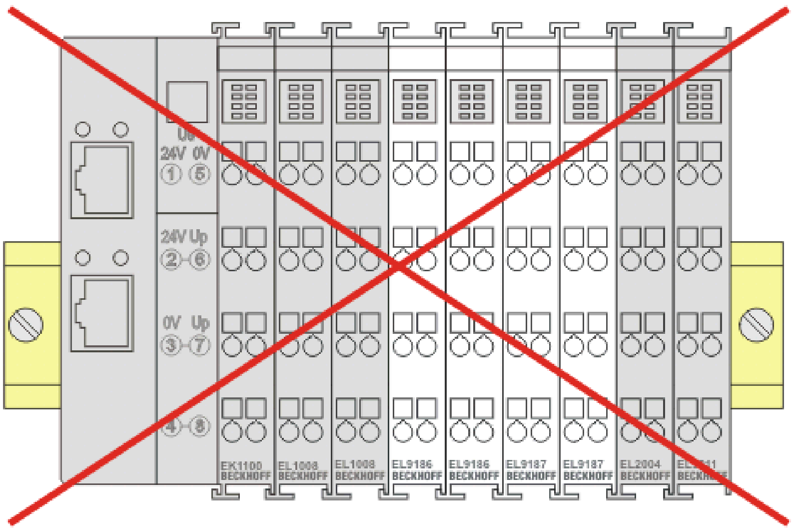


Fig. 38: Incorrect positioning

4.6 Shielding concept

Together with the shield busbar, the prefabricated cables from Beckhoff Automation offer optimum protection against electromagnetic interference.
It is highly recommended to apply the shield as close as possible to the terminal, in order to minimize operational disturbances.

Connection of the motor cable to the shield busbar

Fasten the shield busbar supports 1 to the DIN rail 2. The mounting 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.
As an alternative, a shield busbar clamp 3a can be screwed directly to the metallic rear wall of the control cabinet (fig. “shield busbar clamp”)

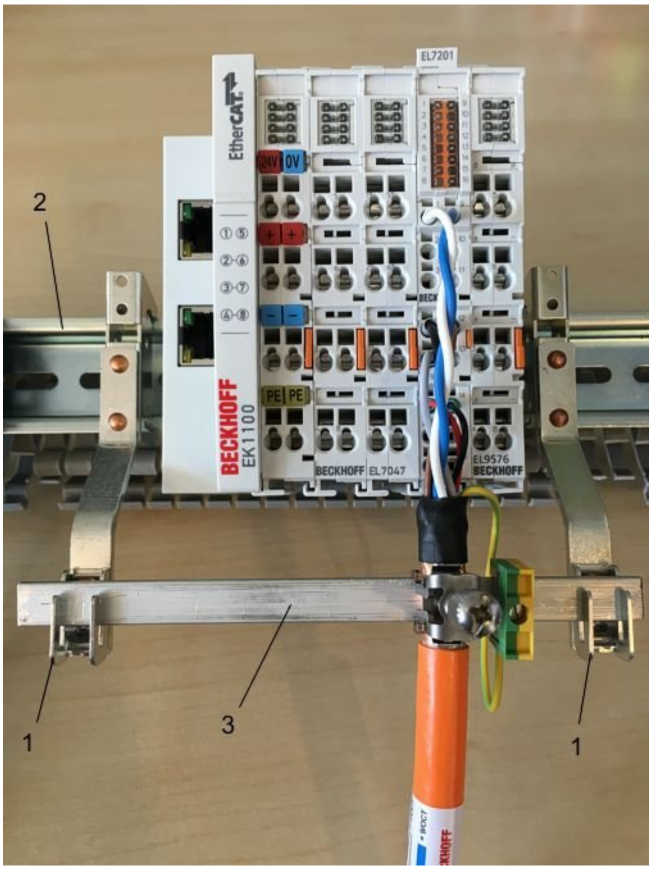


Fig. 39: Shield busbar

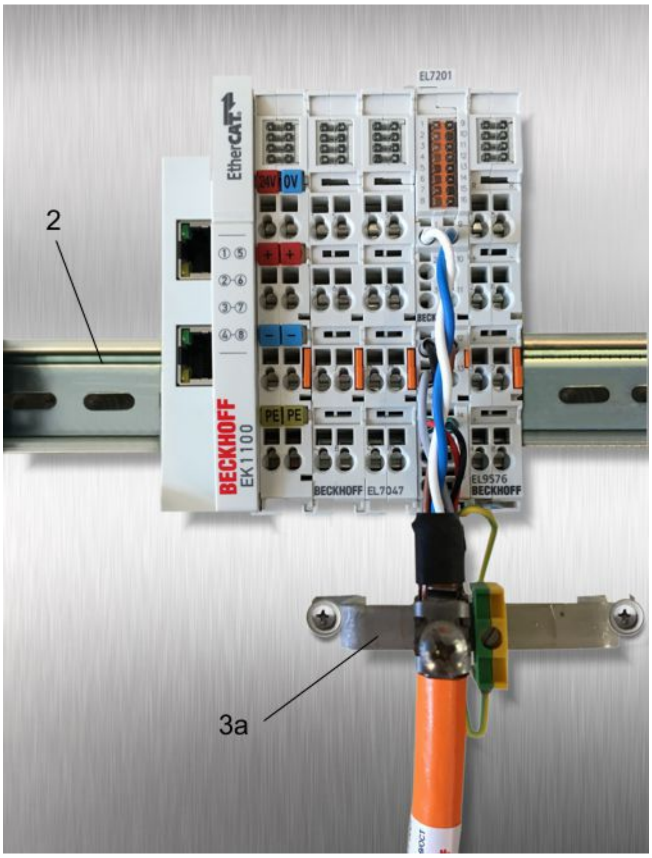


Fig. 40: Shield busbar clamp

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 or shield busbar clamp 3a. Tighten the screw 8 to the stop. Fasten the PE clamp 9 to the shield busbar 3 or shield busbar clamp 3a. Clamp the PE core 10 of the motor cable 5 under the PE clamp 9.

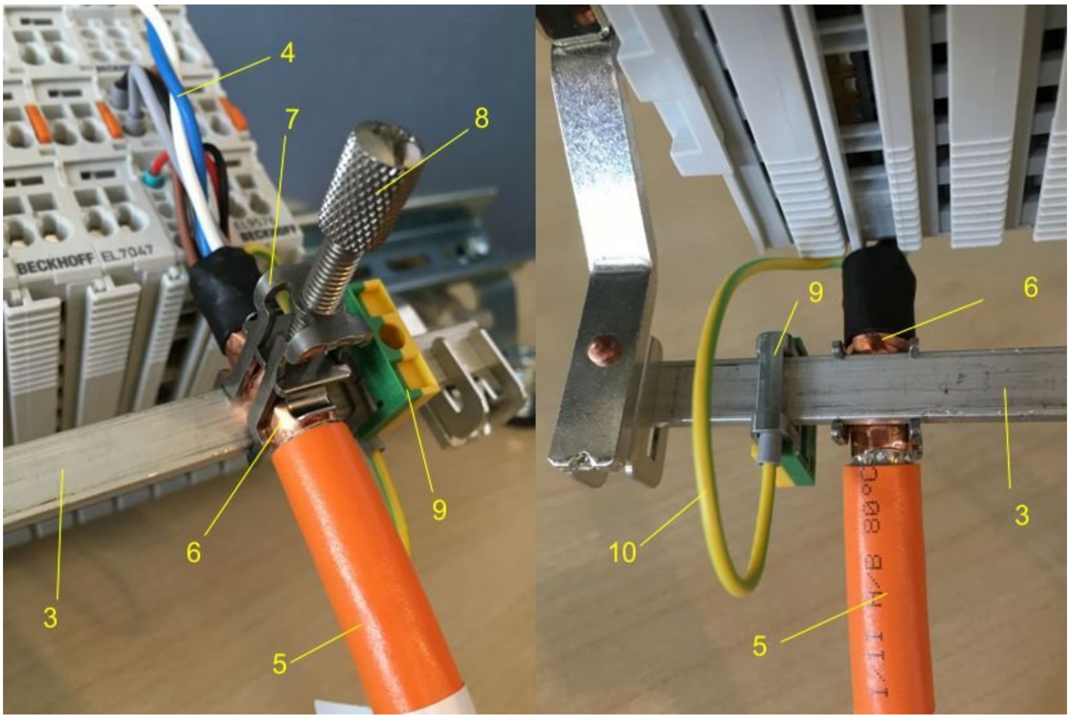


Fig. 41: Shield connection

Connection of the feedback cable to the motor

Twisting of the feedback cable cores

The feedback cable cores should be twisted, in order to avoid operational disturbances.

When screwing the feedback plug to the motor, the shield of the feedback cable is connected via the metallic plug fastener.
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 or shield busbar clamp 3a with the shield clamp 7. The motor cable and the feedback cable can be connected to the shield clamp 7 with the screw 8.

4.7 UL notice - Compact Motion

Application

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

Examination

For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Notes on motion devices

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

Basic principles

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

Application

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

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

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

4.8 EL7037

4.8.1 EL7037 - LEDs and connection

⚠ WARNING

WARNING! Risk of electric shock and damage of devices possible!

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

EL7037-0000

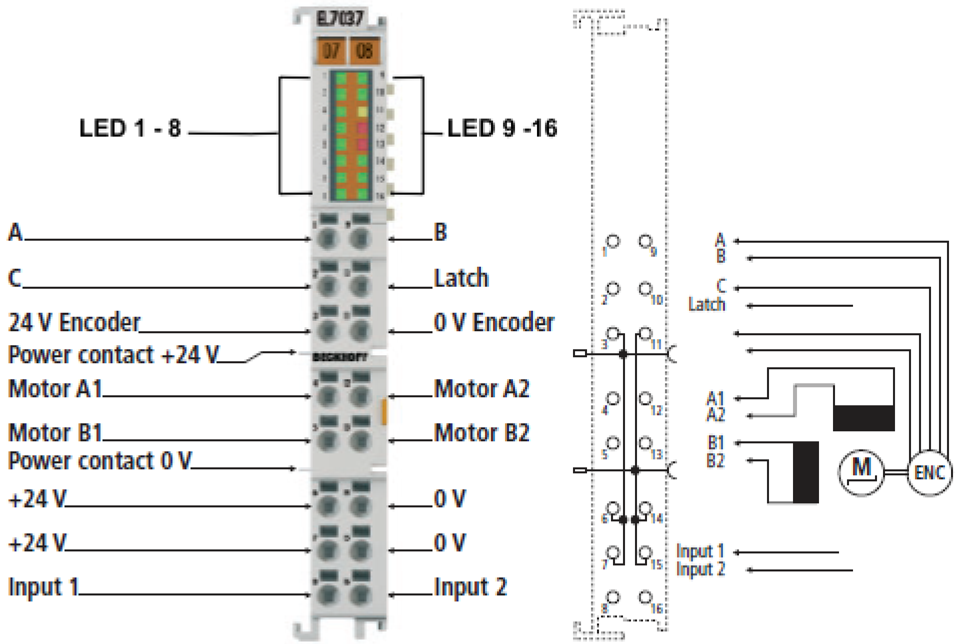


Fig. 42: *LEDs and connection EL7037*

LEDs

No.	LED	Color	Meaning	
1	RUN	green	This LED indicates the terminal's operating state:	
			off	State of the EtherCAT State Machine: **INIT** = Initialization of the terminal or **BOOTSTRAP** = Function for firmware updates of the terminal
			blinking	State of the EtherCAT State Machine: **PREOP** = Setting for mailbox communication and variant standard settings
			single flash	State of the EtherCAT State Machine: **SAFEOP** = Channel checking of the Sync Manager and the Distributed Clocks. Outputs stay in safe operation mode.
			on	State of the EtherCAT State Machine: **OP** = Normal operation mode, mailbox- and process data communication possible
2	Encoder	green	on	Encoder ready for operation
3	A	green	on	Signal at encoder input A
4	B	green	on	Signal at encoder input B
5	C	green	on	Signal at encoder input C
6	Latch	green	on	Signal at latch input
7	Turn CW	green	on	Motor is triggered clock wise
8	Input 1	green	on	Signal at digital input 1
9	Driver	green	on	Driver stage ready for operation
10	Power	green	off	The power supply voltage (24 V_{DC}) is absent or the motor control is blocked (Index 6010:02 [▸ 219] is not set))
			on	The power supply voltage (24 V_{DC}) is present
11	Warning	yellow	on	Configuration error, e.g.: • Motor power supply not connected • 80°C temperature exceeded • 100% duty cycle reached • ...
12	Error A	red	on	Configuration error of output stage A, e.g.: • 100°C temperature exceeded • short circuit • ...
13	Error B	red	on	Configuration error of output stage B, e.g.: • 100°C temperature exceeded • short circuit • ...
14	Enable	green	off	The motor control is blocked (Index 6010:02 [▸ 219] is not set) or EL7037 is not ready for operation
			on	The motor control is activated (Index 6010:02 [▸ 219] is set) or EL7037 is ready for operation
15	Turn CCW	green	on	Motor is triggered counter clock wise
16	Input 2	green	on	Signal at digital input 2

Terminal points

Terminal point	Name	Signal
1	A	Encoder input A
2	C	Encoder input C (zero input). If object 7000:01 [▶ 220] is set in the control word and a rising edge occurs at encoder input C, the current counter value is stored as a reference mark in the latch register.
3	Encoder supply +24V	Encoder supply + 24 V, internally connected with positive power contact and pin 6, 7
4	A1	Motor winding A1
5	B1	Motor winding B1
6	+24V	+24 V_{DC}, internally connected with positive power contact and pin 3, 7
7	+24V	+24 V_{DC}, internally connected with positive power contact and pin 3, 7
8	Input 1	Digital input 1 (24 V_{DC})
9	B	Encoder input B
10	Latch	Latch input. The current counter value is stored as a reference mark in the latch register, if • object 7000:02 [▶ 220] is set and a rising edge occurs at the latch input; or • object 7000:04 [▶ 220] is set and a falling edge occurs at the latch input.
11	Encoder supply 0V	Encoder supply 0 V, internally connected with negative power contact and pin 14, 15
12	A2	Motor winding A2
13	B2	Motor winding B2
14	0V	0 V_{DC}, internally connected with negative power contact and pin 11, 15
15	0V	0 V_{DC}, internally connected with negative power contact and pin 11, 14
16	Input 2	Digital input 2 (24 V_{DC}), also configurable as a digital output (0,5 A)

4.8.2 EL7037 - General connection examples

⚠ **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.

NOTE

Connect the motor strands correctly!

Connect the windings of a motor strand only to the terminal points of the same output driver of the stepper motor terminal, e.g.:
• one motor strand to terminal points A1 and A2,
• the other motor strand to terminal points B1 and B2.
Connecting a motor strand to the terminal points of different output drivers (e.g. to A1 and B1) can lead to destruction of the output drivers of stepper motor terminal!

Connection types

The EL7047 Stepper Motor terminal has bipolar output stages and can control bipolar and unipolar motors.

NOTE

Fuse protection of the supply voltage

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

Bipolar motors

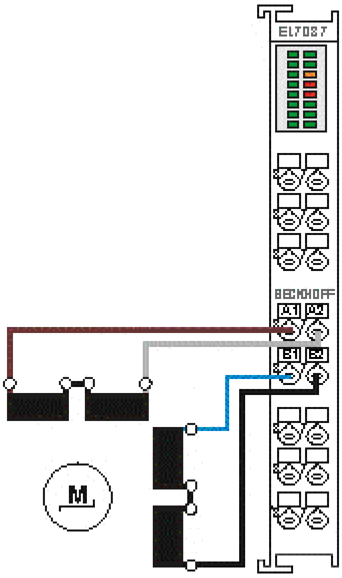


Fig. 43: *Bipolar control (serial) of a bipolar motor*

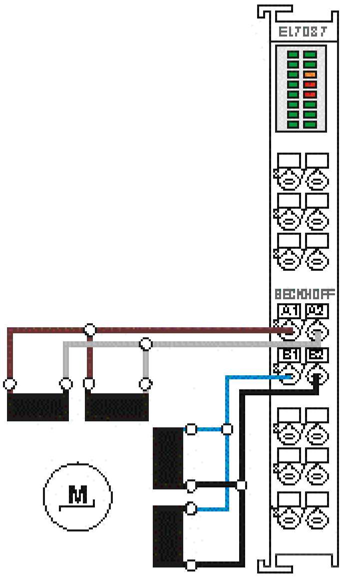


Fig. 44: *Bipolar control (parallel) of a bipolar motor*

Documentation for stepper motors from Beckhoff

These two examples show the connection of the bipolar Beckhoff motors AS1010, AS1020, AS1030, AS1050 or AS1060. Further information on stepper motors from Beckhoff can be found in the associated documentation available for download from our website at http://www.beckhoff.com.

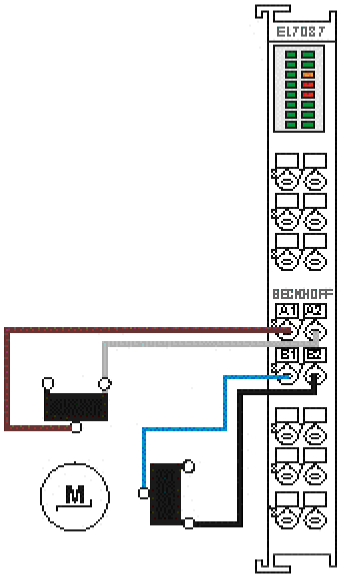


Fig. 45: *Bipolar control of a unipolar motor*

Only one half of each winding is controlled.

4.9 EL7047

4.9.1 EL7047 - LEDs and connection

⚠ WARNING

WARNING! Risk of electric shock and damage of devices possible!

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

EL7047-0000

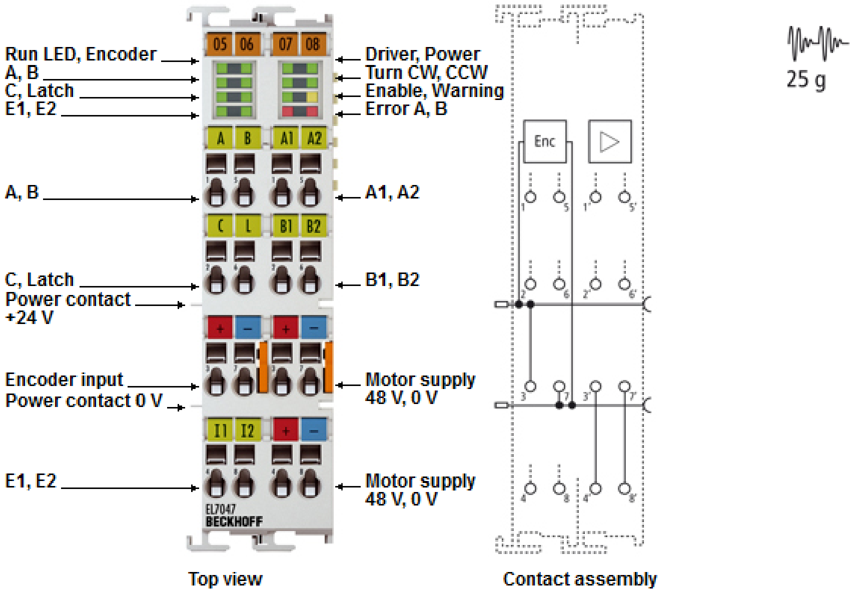


Fig. 46: *LEDs and Connection EL7047*

LEDs (left prism)

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = Initialization of the terminal or **BOOTSTRAP** = Function for firmware updates of the terminal
		blinking	State of the EtherCAT State Machine: **PREOP** = Setting for mailbox communication and variant standard settings
		single flash	State of the EtherCAT State Machine: **SAFEOP** = Channel checking of the Sync Manager and the Distributed Clocks. Outputs stay in safe operation mode.
		on	State of the EtherCAT State Machine: **OP** = Normal operation mode, mailbox- and process data communication possible
Encoder	green	on	Encoder ready for operation
A	green	on	Signal at encoder input A
B	green	on	Signal at encoder input B
C	green	on	Signal at encoder input C
Latch	green	on	Signal at latch input
Input 1	green	on	Signal at digital input 1
Input 2	green	on	Signal at digital input 2

LEDs (right prism)

LED	Color	Meaning	
Driver	green	on	Driver stage ready for operation
Power	green	off	The power supply voltage (50 V_{DC}) is absent or the motor control is blocked (Index 6010:02 [▸ 219] is not set))
		on	The power supply voltage (50 V_{DC}) is present
Turn CW	green	on	Motor is triggered clock wise
Turn CCW	green	on	Motor is triggered counter clock wise
Enable	green	off	The motor control is blocked (Index 6010:02 [▸ 219] is not set) or EL7047 is not ready for operation
		on	The motor control is activated (Index 6010:02 [▸ 219] is set) or EL7047 is ready for operation
Warning	yellow	off	No errors
		on	Configuration error, e.g.: • Motor power supply not connected • 80°C temperature exceeded • 100% duty cycle reached • ...
Error A	red	on	Configuration error of output stage A, e.g.: • 100°C temperature exceeded • short circuit • ...
Error B	red	on	Configuration error of output stage B, e.g.: • 100°C temperature exceeded • short circuit • ...

Terminal Points - Left-hand section of the housing

Terminal point	Name	Signal
1	A	Encoder input A
2	C	Encoder input C (zero input). If object 7000:01 [▸ 220] is set in the control word and a rising edge occurs at encoder input C, the current counter value is stored as a reference mark in the latch register.
3	Encoder supply +24V	Encoder supply (from positive power contact)
4	Input 1	Digital input 1 (24 V_{DC})
5	B	Encoder input B
6	Latch / Gate	Latch input. The current counter value is stored as a reference mark in the latch register, if • object 7000:02 [▸ 220] is set and a rising edge occurs at the latch input; or • object 7000:04 [▸ 220] is set and a falling edge occurs at the latch input.
7	Encoder supply 0V	Encoder supply (from negative power contact)
8	Input 2	Digital input 2 (24 V_{DC})

Terminal Points - Right-hand section of the housing

Terminal point	Name	Signal
1'	A1	Motor winding A1
2'	B1	Motor winding B1
3'	Motor supply +50V	Feeding for output stage (max. +50 V_{DC})
4'	Motor supply +50V	Feeding for output stage (max. +50 V_{DC})
5'	A2	Motor winding A2
6'	B2	Motor winding B2
7'	Motor supply 0V	Feeding for output stage (0 V_{DC})
8'	Motor supply 0V	Feeding for output stage (0 V_{DC})

4.9.2 EL7047 - General connection examples

⚠ **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.

NOTE

Connect the motor strands correctly!

Connect the windings of a motor strand only to the terminal points of the same output driver of the stepper motor terminal, e.g.:
• one motor strand to terminal points A1 and A2,
• the other motor strand to terminal points B1 and B2.
Connecting a motor strand to the terminal points of different output drivers (e.g. to A1 and B1) can lead to destruction of the output drivers of stepper motor terminal!

NOTE

Use a brake chopper terminal (EL9576) for short deceleration ramps!

Very short deceleration ramps may lead to temporarily increased feedback. In this case the terminal would report an error. In order to avoid this, a brake chopper terminal (EL9576) should be connected in parallel to the power supply for the motor so that any energy being fed back is absorbed.

NOTE

Fuse protection of the supply voltage

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

Connection types

The EL7047 Stepper Motor terminal has bipolar output stages and can control bipolar and unipolar motors.

Bipolar motors

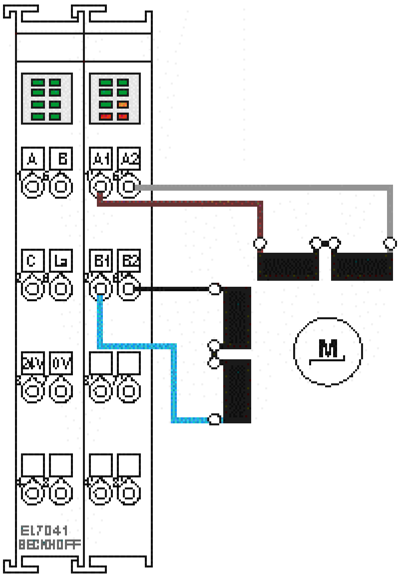


Fig. 47: *Bipolar control (serial) of a bipolar motor*

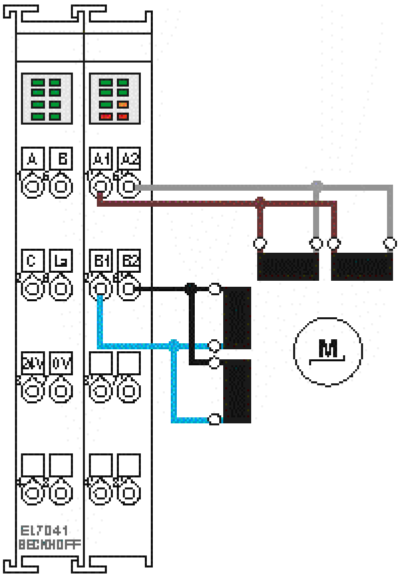


Fig. 48: *Bipolar control (parallel) of a bipolar motor*

Documentation for stepper motors from Beckhoff

These two examples show the connection of the bipolar Beckhoff motors AS1010, AS1020, AS1030, AS1050 or AS1060. Further information on stepper motors from Beckhoff can be found in the associated documentation available for download from our website at http://www.beckhoff.com.

Unipolar motors

Bipolar control of a unipolar motor

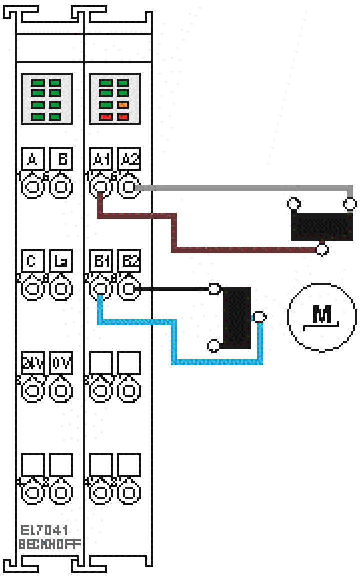


Fig. 49: *Bipolar control with only one half of each winding is controlled*

Encoder

Connecting an encoder (24 V)

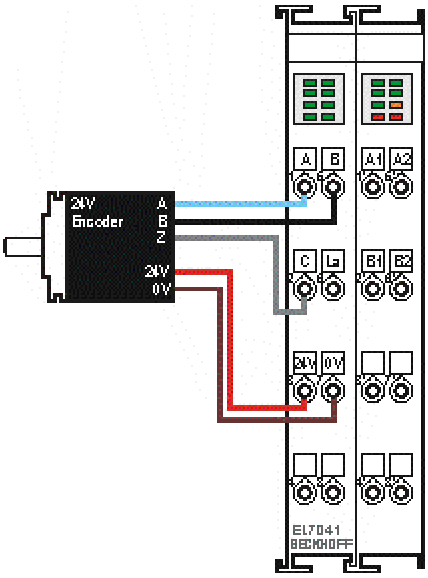


Fig. 50: *The encoder is supplied from the power contacts via terminal points 3 (+24 V) and 7 (0 V).*

5 Commissioning

5.1 TwinCAT Quick Start

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

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

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

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

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

The following relationship is envisaged from user PC to the individual control elements:

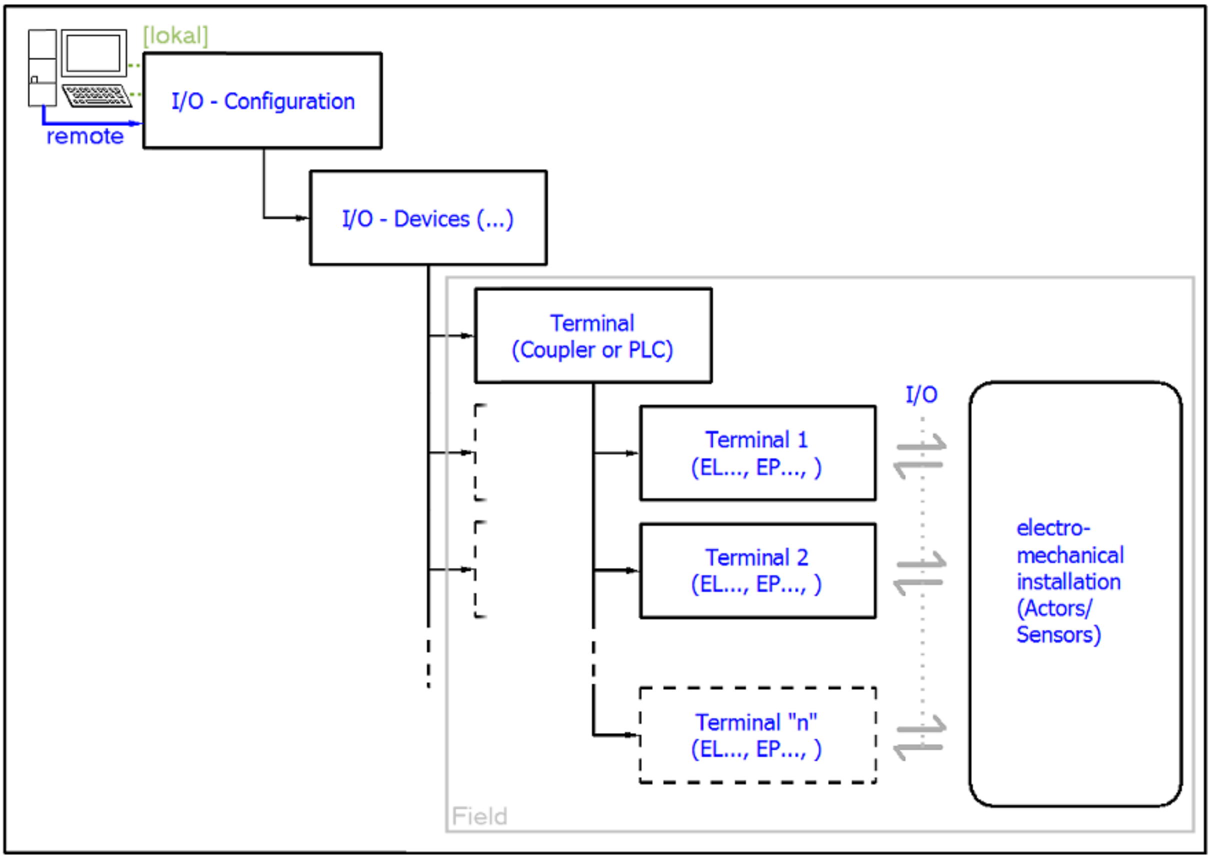


Fig. 51: Relationship between user side (commissioning) and installation

The user inserting of certain components (I/O device, terminal, box...) is the same in TwinCAT 2 and TwinCAT 3. The descriptions below relate to the online procedure.

Sample configuration (actual configuration)

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

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

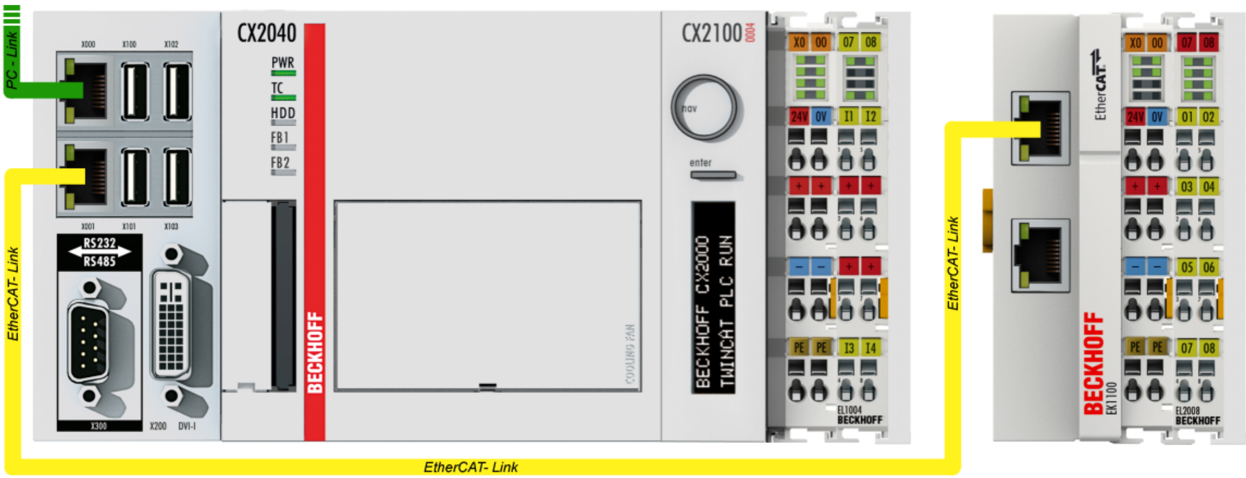


Fig. 52: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1 TwinCAT 2

Startup

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

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

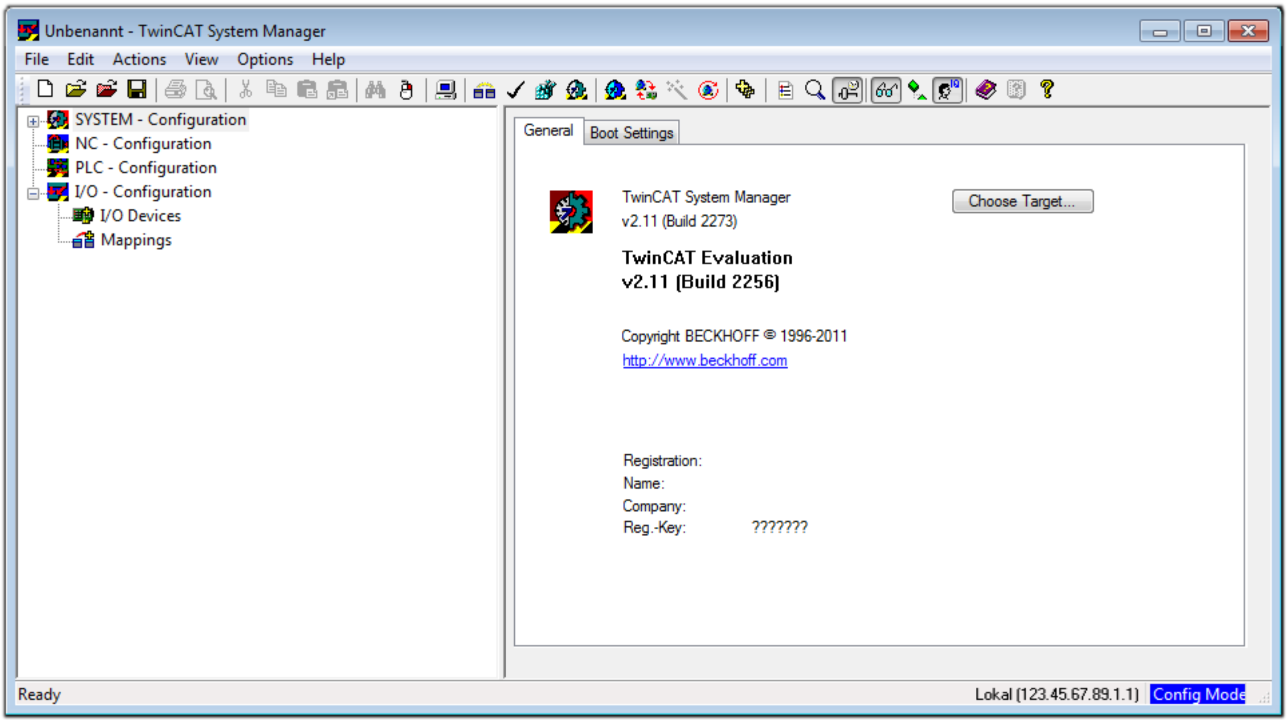


Fig. 53: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 71]".

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

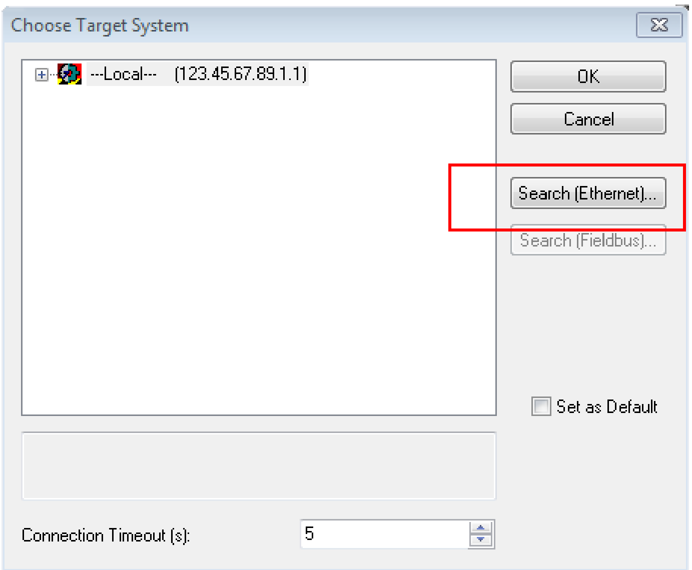


Fig. 54: Selection of the target system

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

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

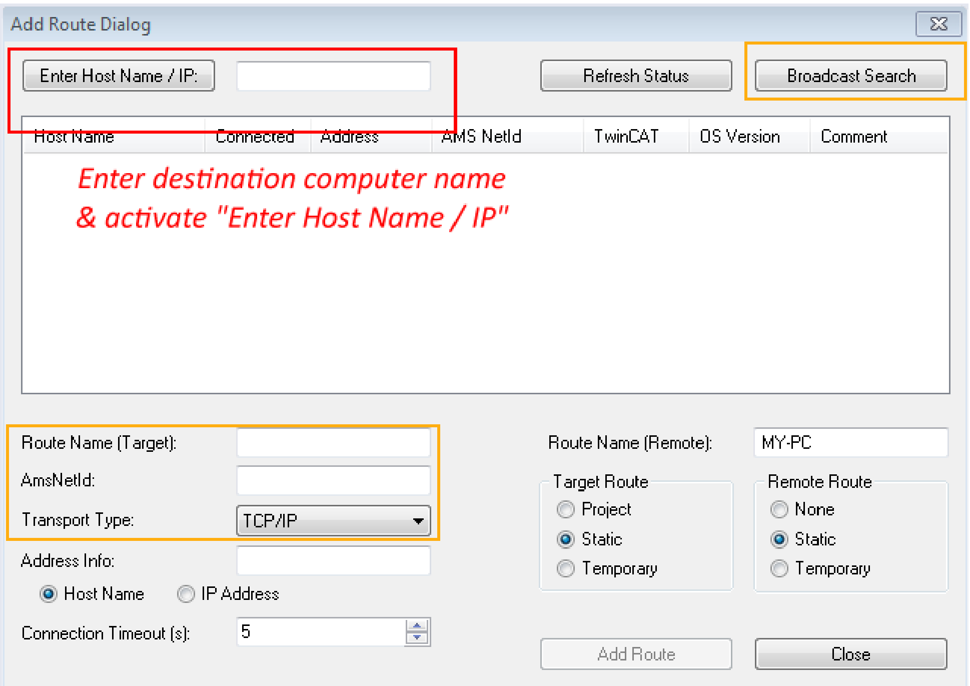
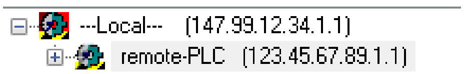


Fig. 55: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

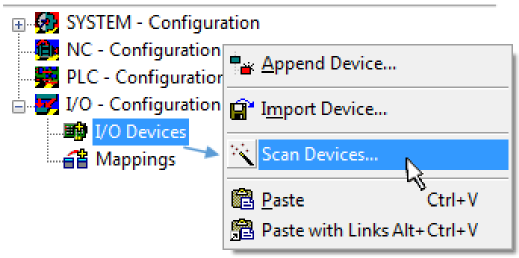


Fig. 56: Select “Scan Devices...”

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

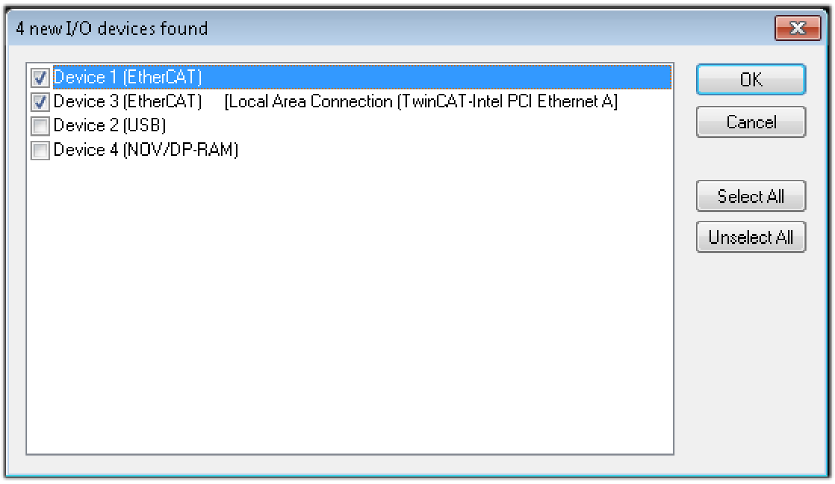


Fig. 57: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the sample configuration [▶ 67] described at the beginning of this section, the result is as follows:

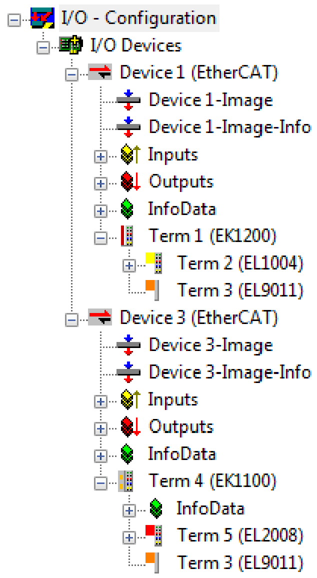


Fig. 58: Mapping of the configuration in the TwinCAT 2 System Manager

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

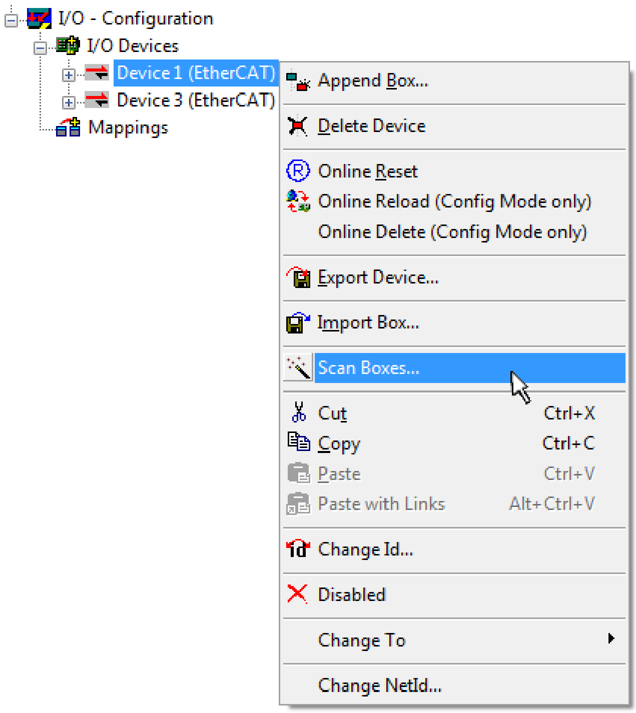


Fig. 59: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

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

The following section refers to Structured Text (ST).

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

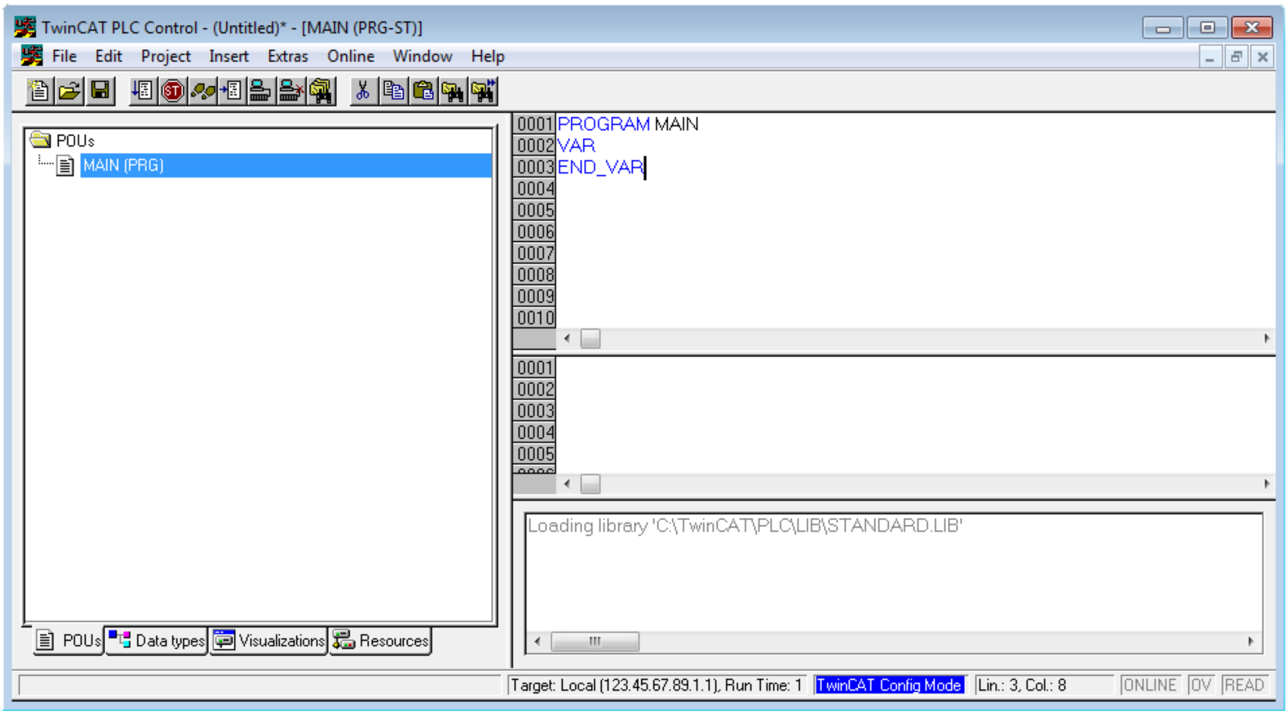


Fig. 60: TwinCAT PLC Control after startup

Sample variables and a sample program have been created and stored under the name “PLC_example.pro”:

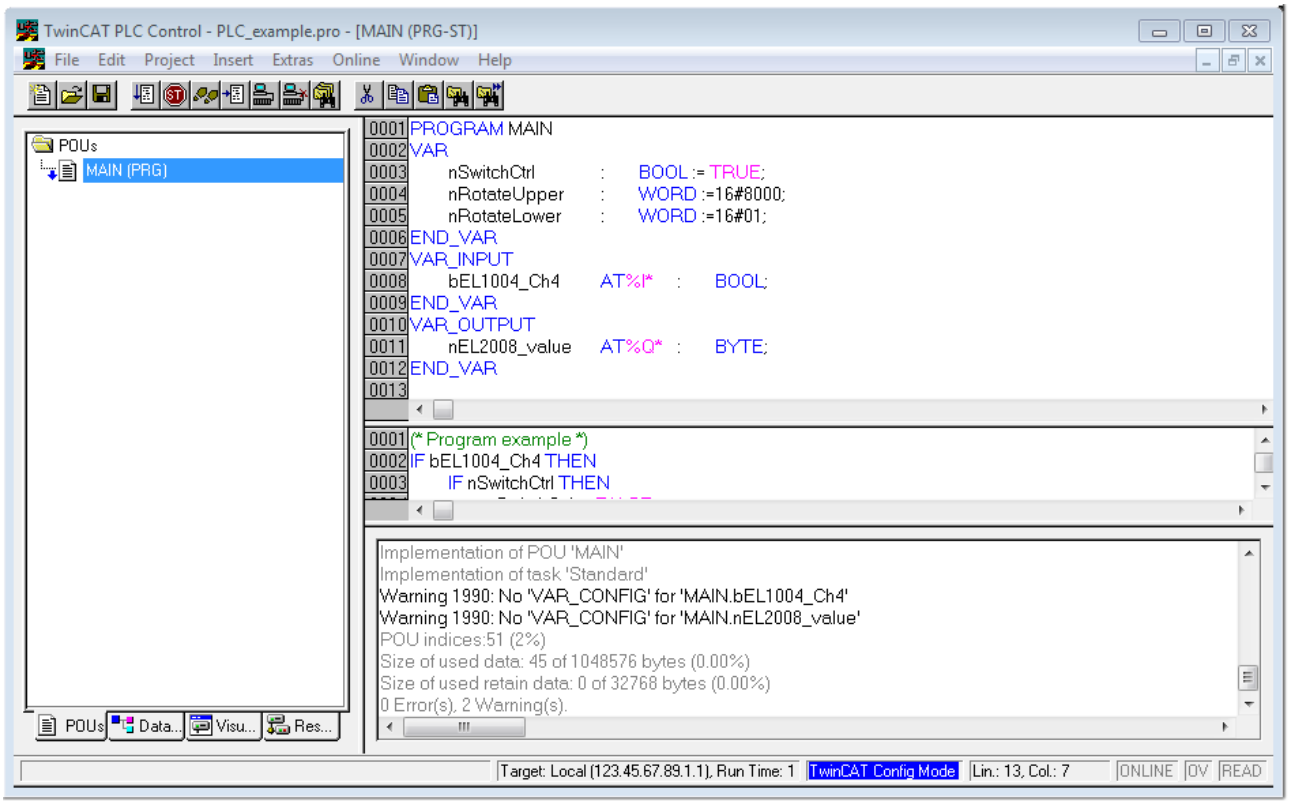


Fig. 61: Sample program with variables after a compile process (without variable integration)

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

First, integrate the TwinCAT PLC Control project in the **System Manager** via the context menu of the PLC configuration; right-click and select “Append PLC Project…”:

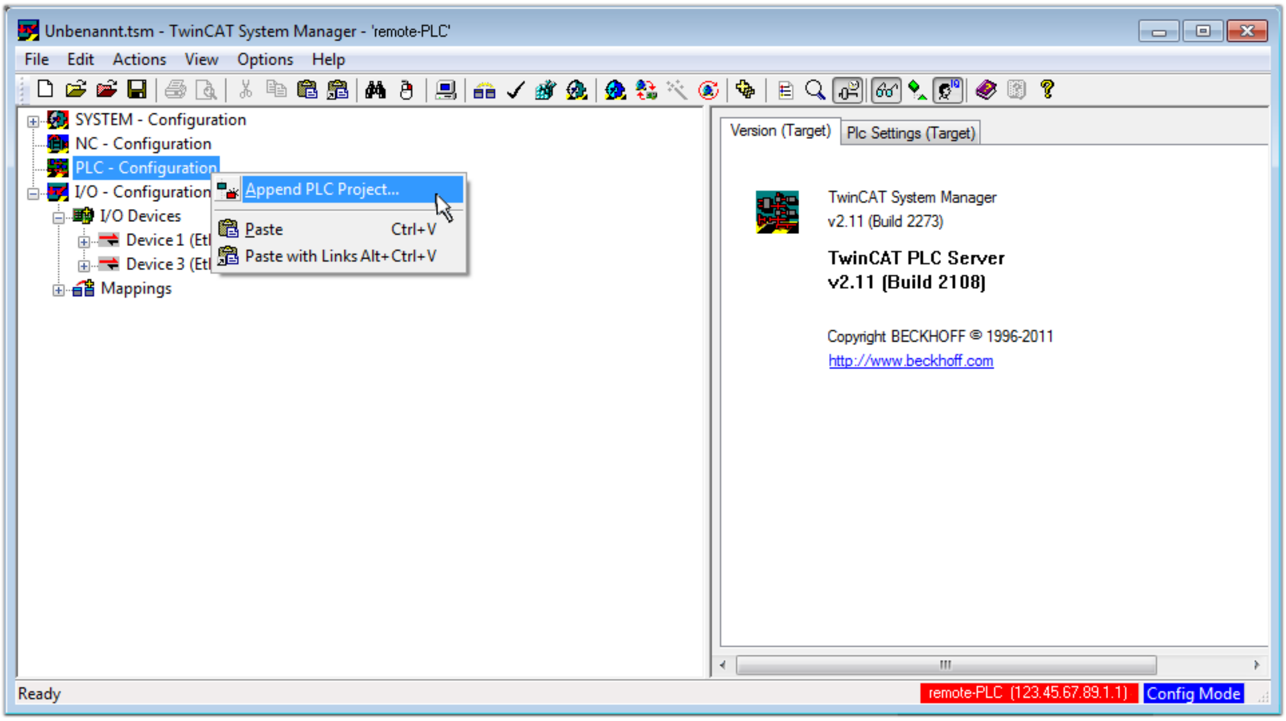


Fig. 62: Appending the TwinCAT PLC Control project

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

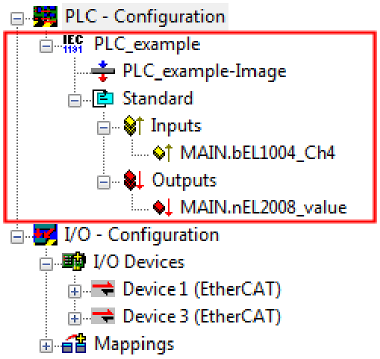


Fig. 63: PLC project integrated in the PLC configuration of the System Manager

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

Assigning variables

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

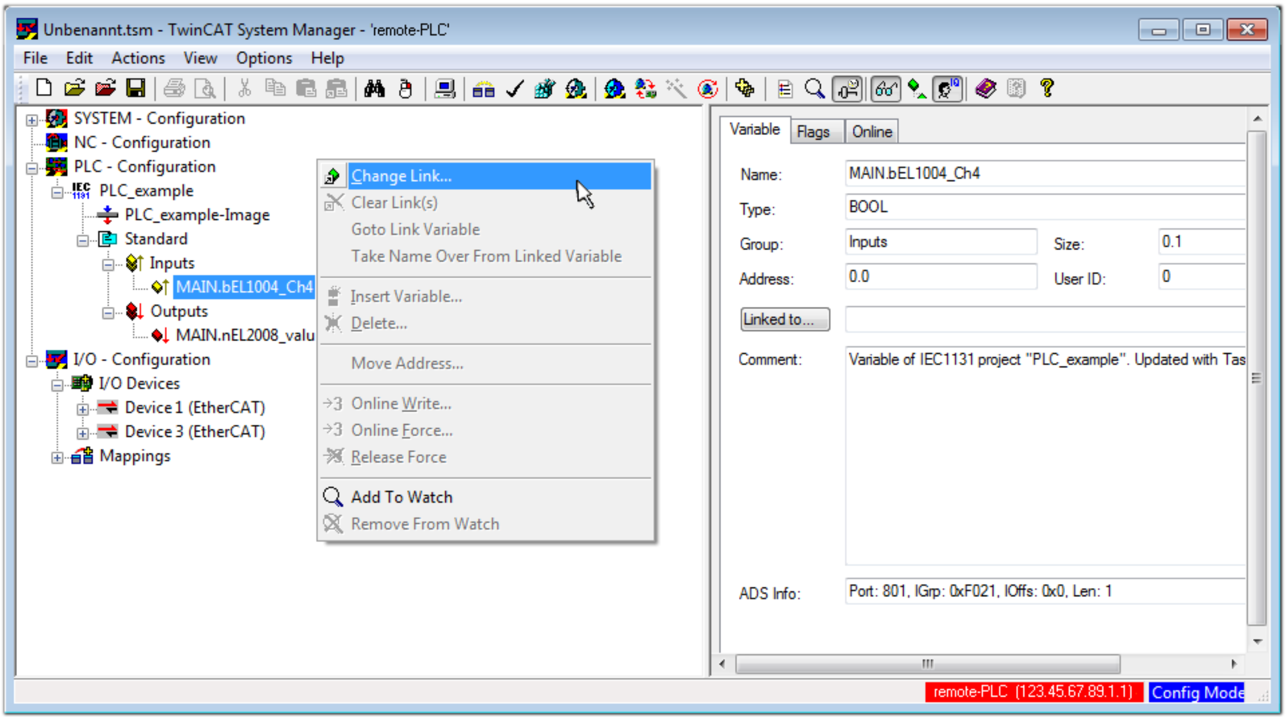


Fig. 64: Creating the links between PLC variables and process objects

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

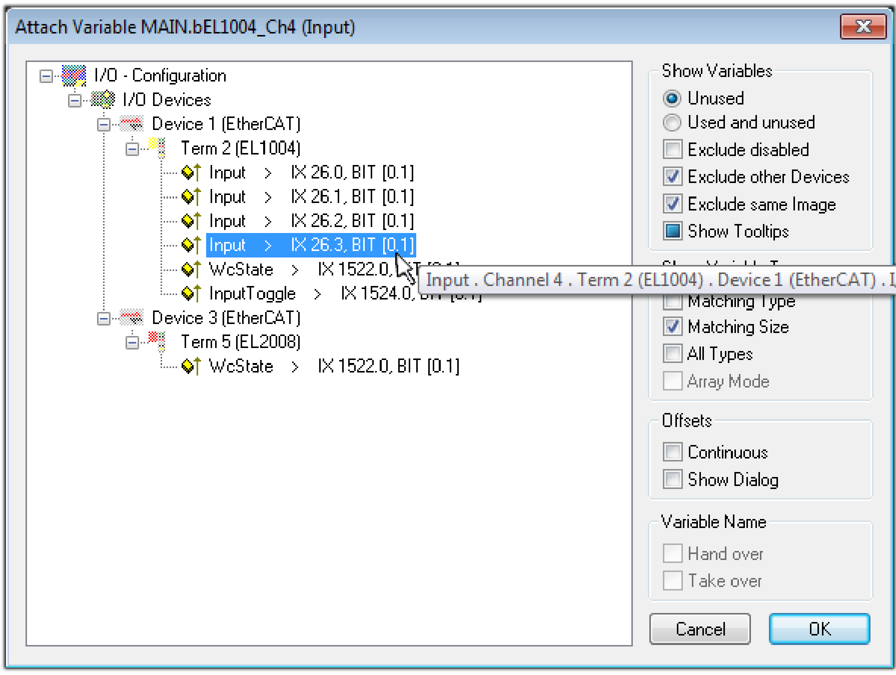


Fig. 65: Selecting PDO of type BOOL

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

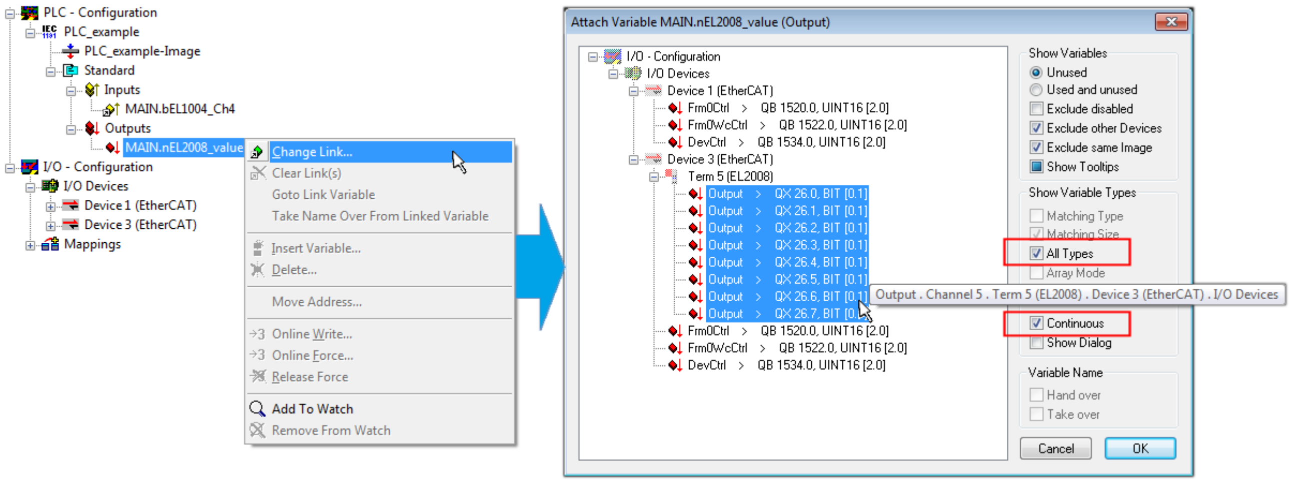


Fig. 66: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

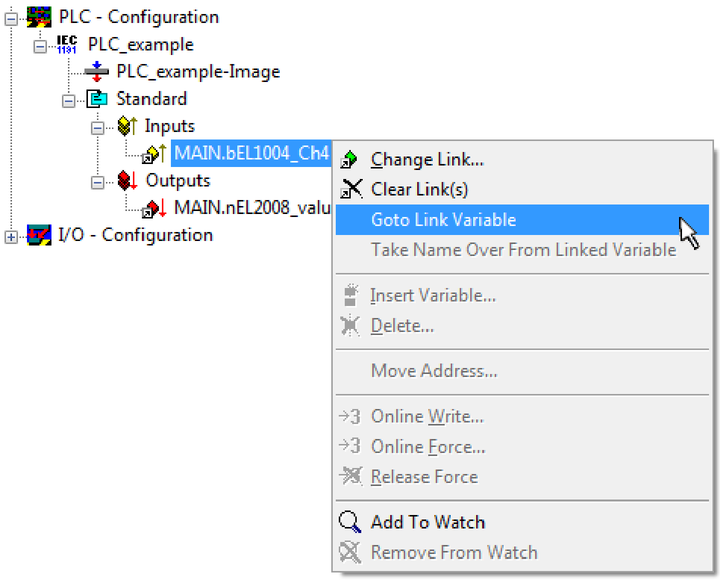
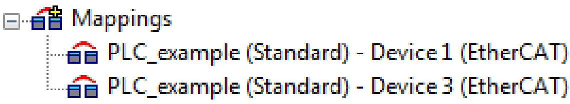


Fig. 67: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

The process of assigning variables to the PDO is completed via the menu selection “Actions” → “Generate Mappings”, key Ctrl+M or by clicking on the symbol  in the menu.

This can be visualized in the configuration:



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

Activation of the configuration

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

A few seconds later the real-time status 

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

Starting the controller

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

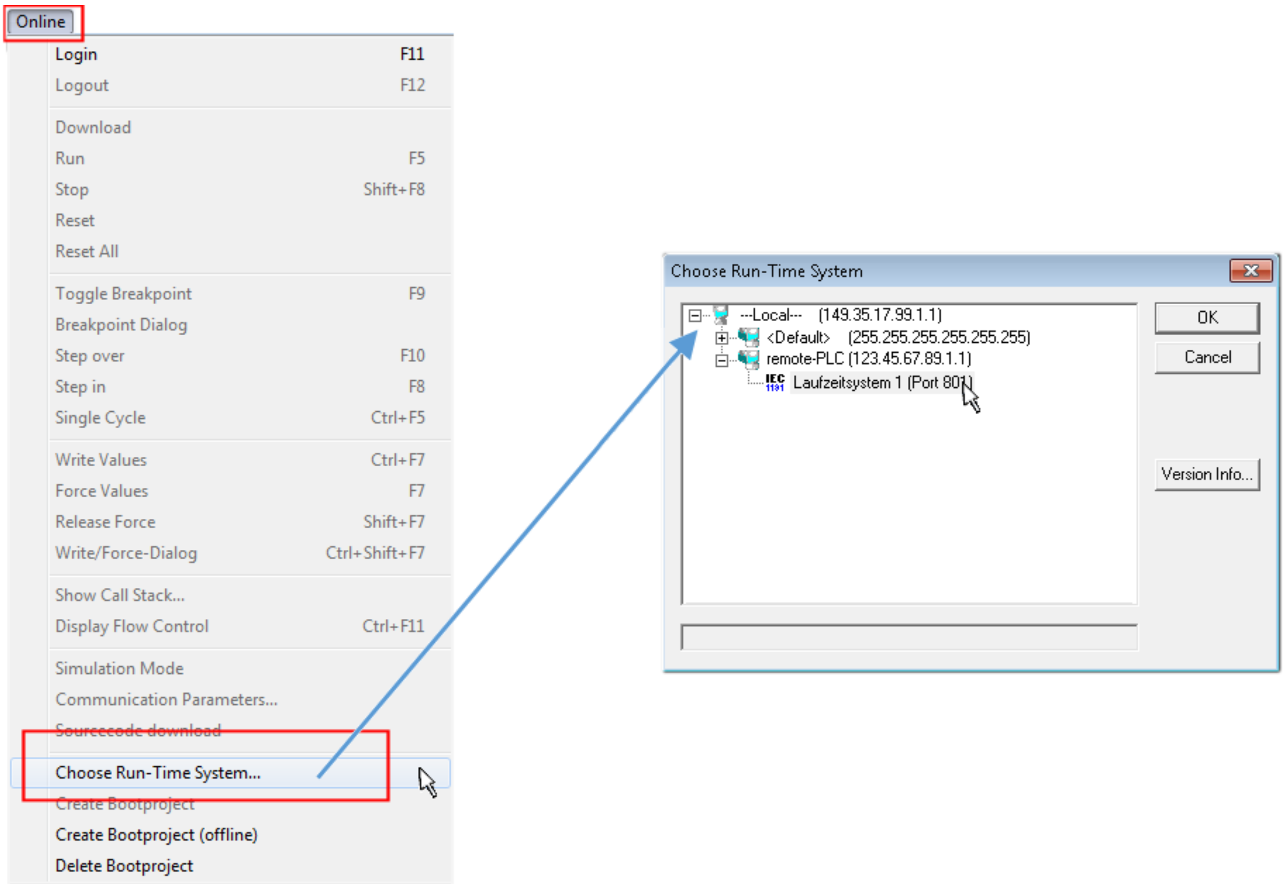
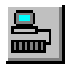


Fig. 68: Choose target system (remote)

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

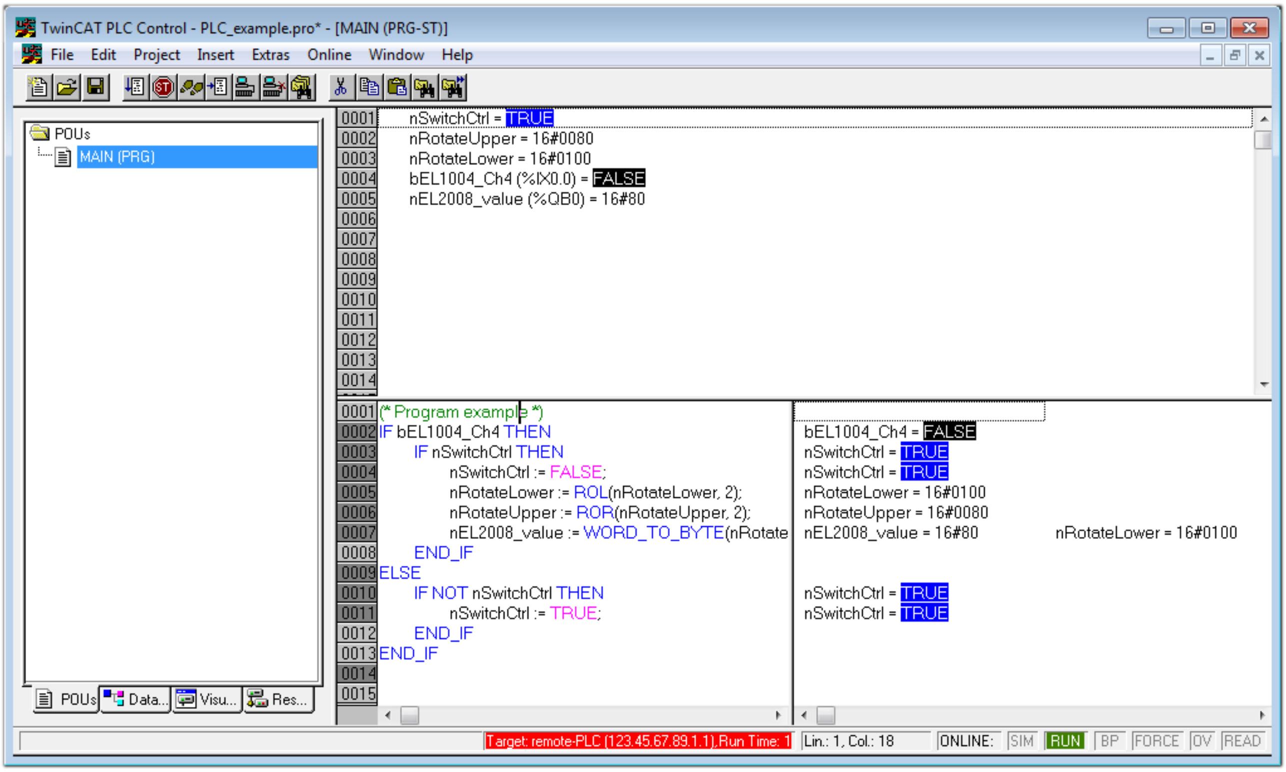


Fig. 69: PLC Control logged in, ready for program startup

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

5.1.2 TwinCAT 3

Startup

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

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

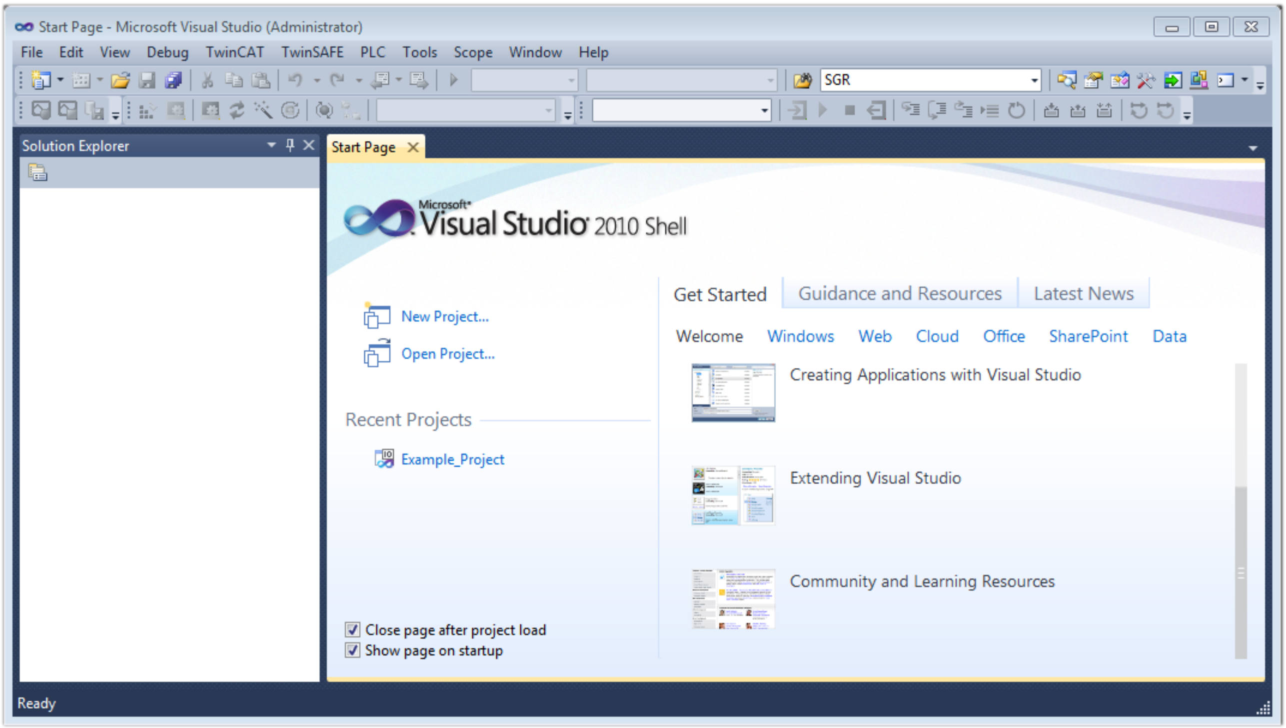


Fig. 70: Initial TwinCAT 3 user interface

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

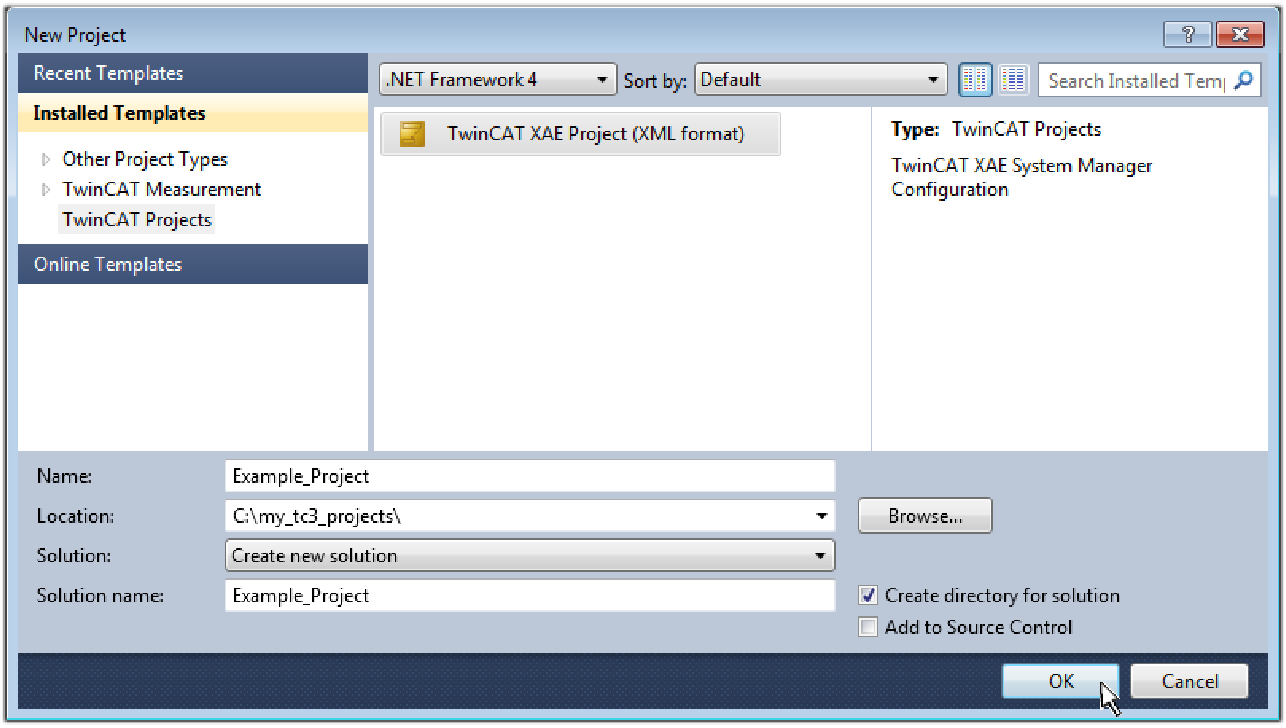


Fig. 71: Create new TwinCAT project

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

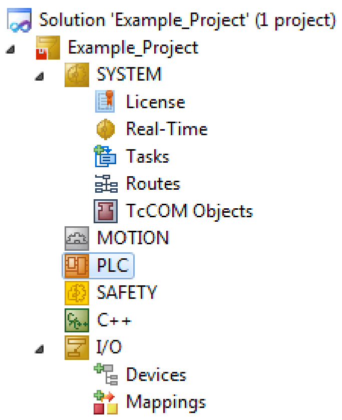
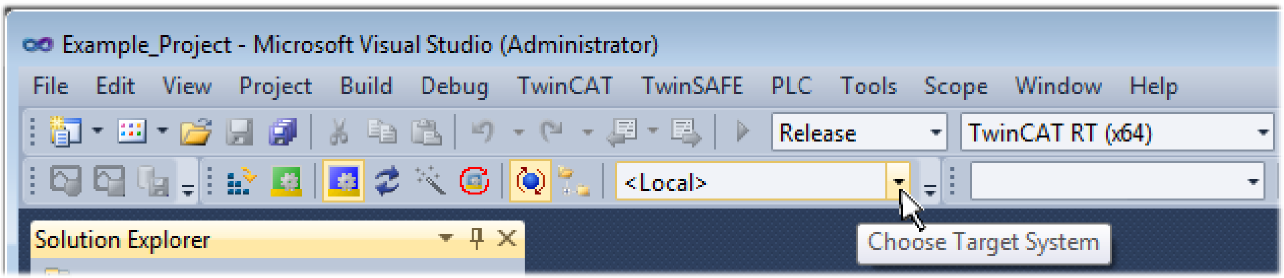


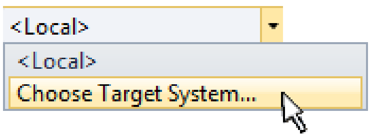
Fig. 72: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 82]".

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



expand the pull-down menu:



and open the following window:

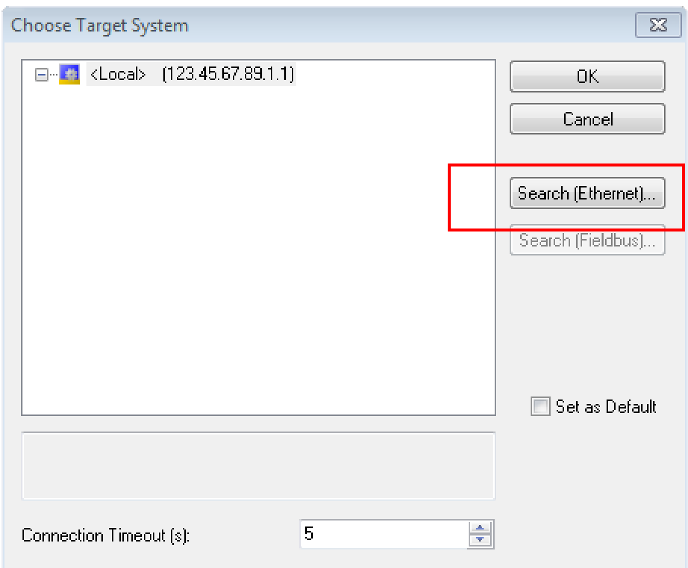


Fig. 73: Selection dialog: Choose the target system

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

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

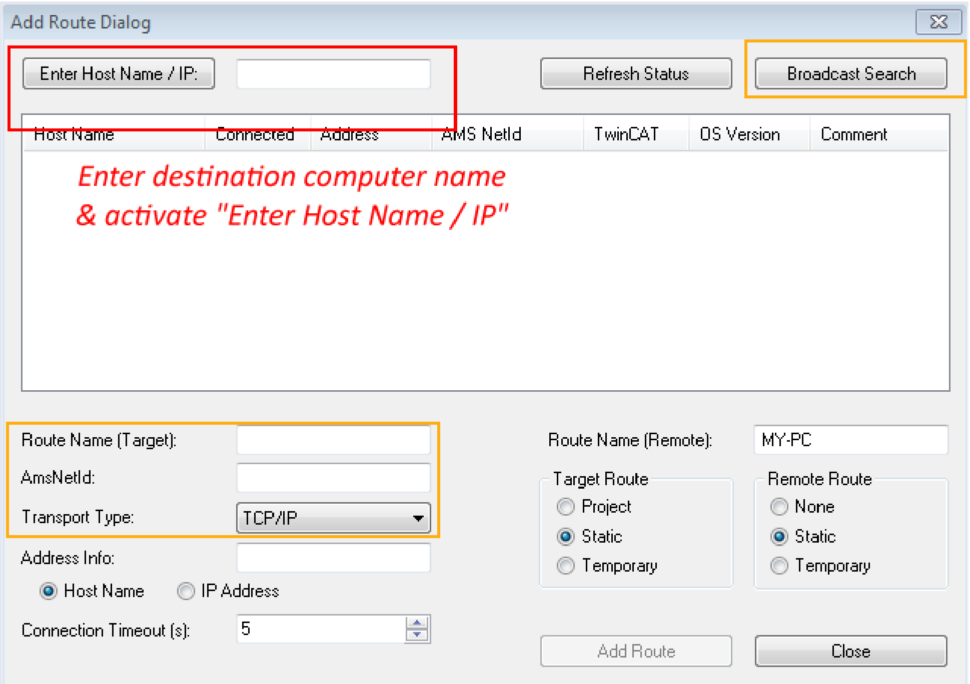
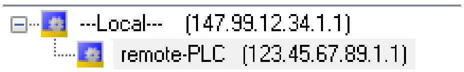


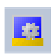
Fig. 74: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

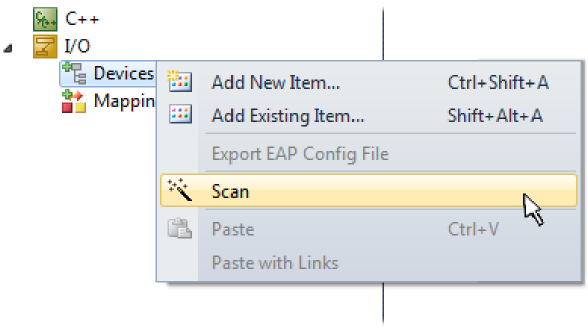


Fig. 75: Select “Scan”

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

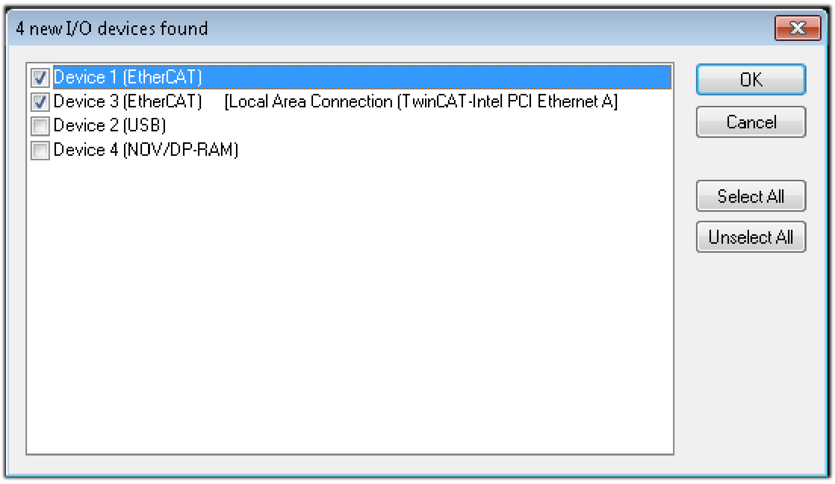


Fig. 76: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the sample configuration [▶ 67] described at the beginning of this section, the result is as follows:

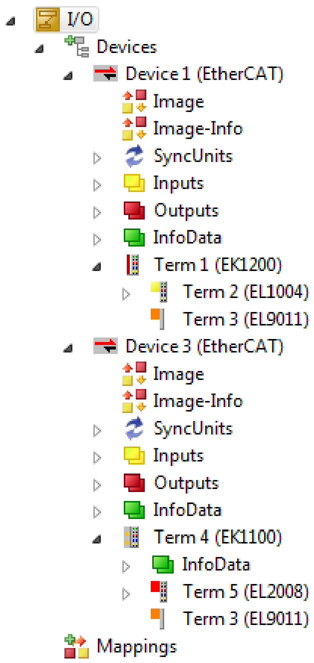


Fig. 77: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

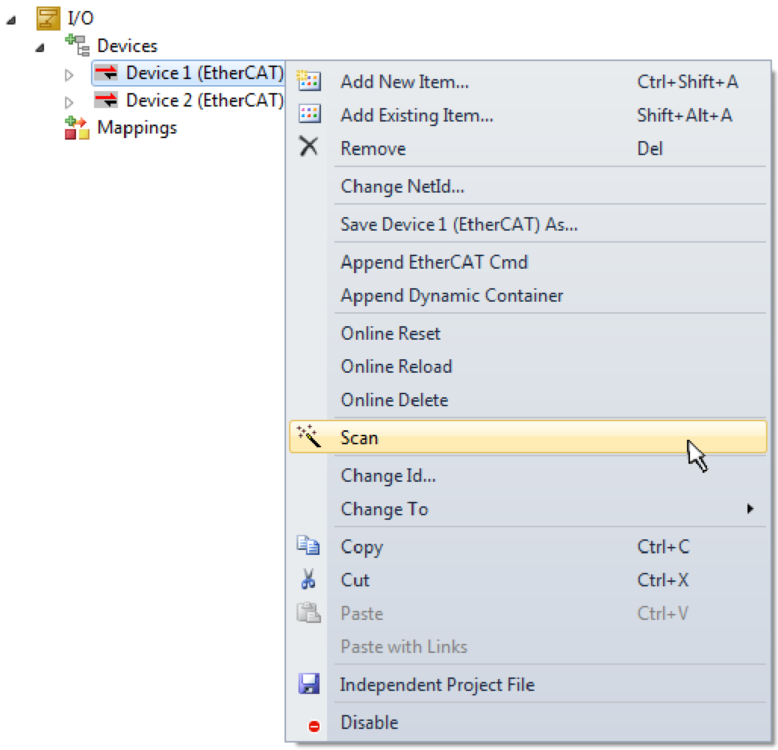


Fig. 78: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers to Structured Text (ST).

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

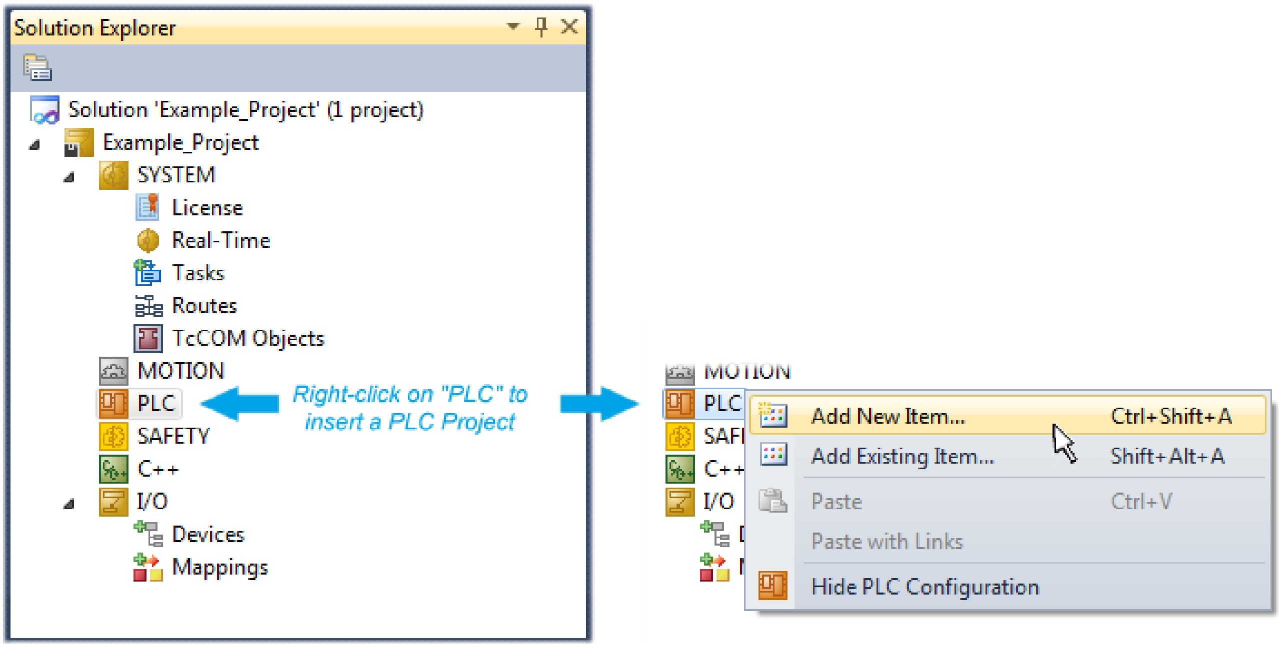


Fig. 79: Adding the programming environment in “PLC”

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

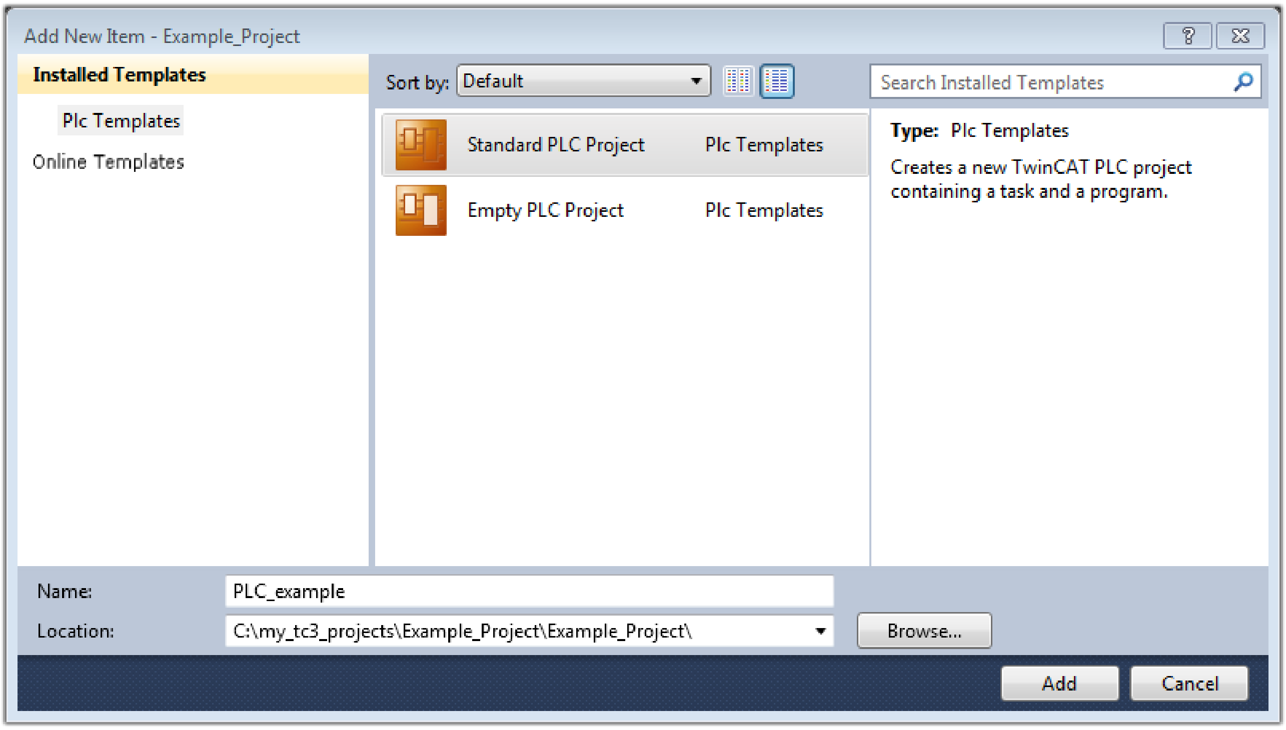


Fig. 80: Specifying the name and directory for the PLC programming environment

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

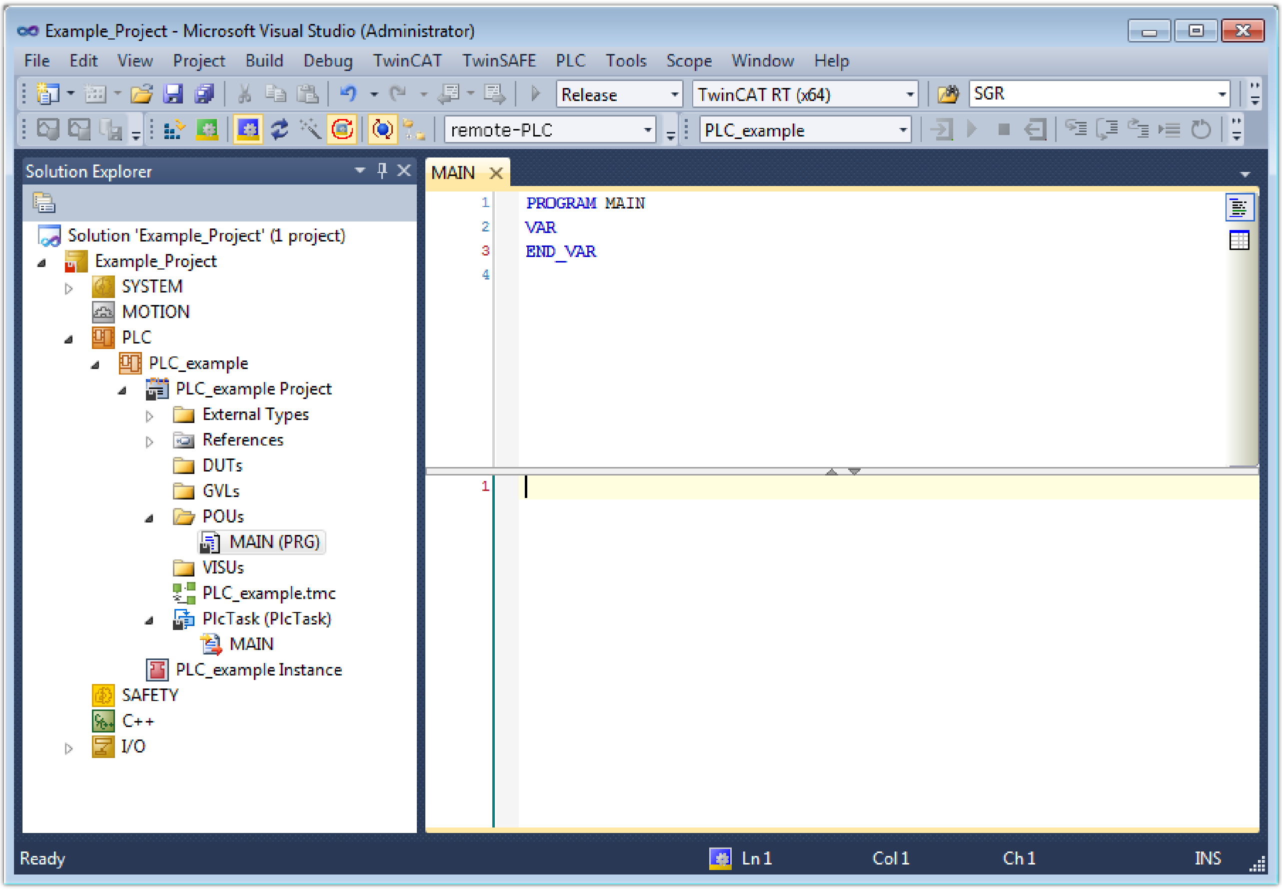


Fig. 81: Initial “Main” program of the standard PLC project

To continue, sample variables and a sample program have now been created:

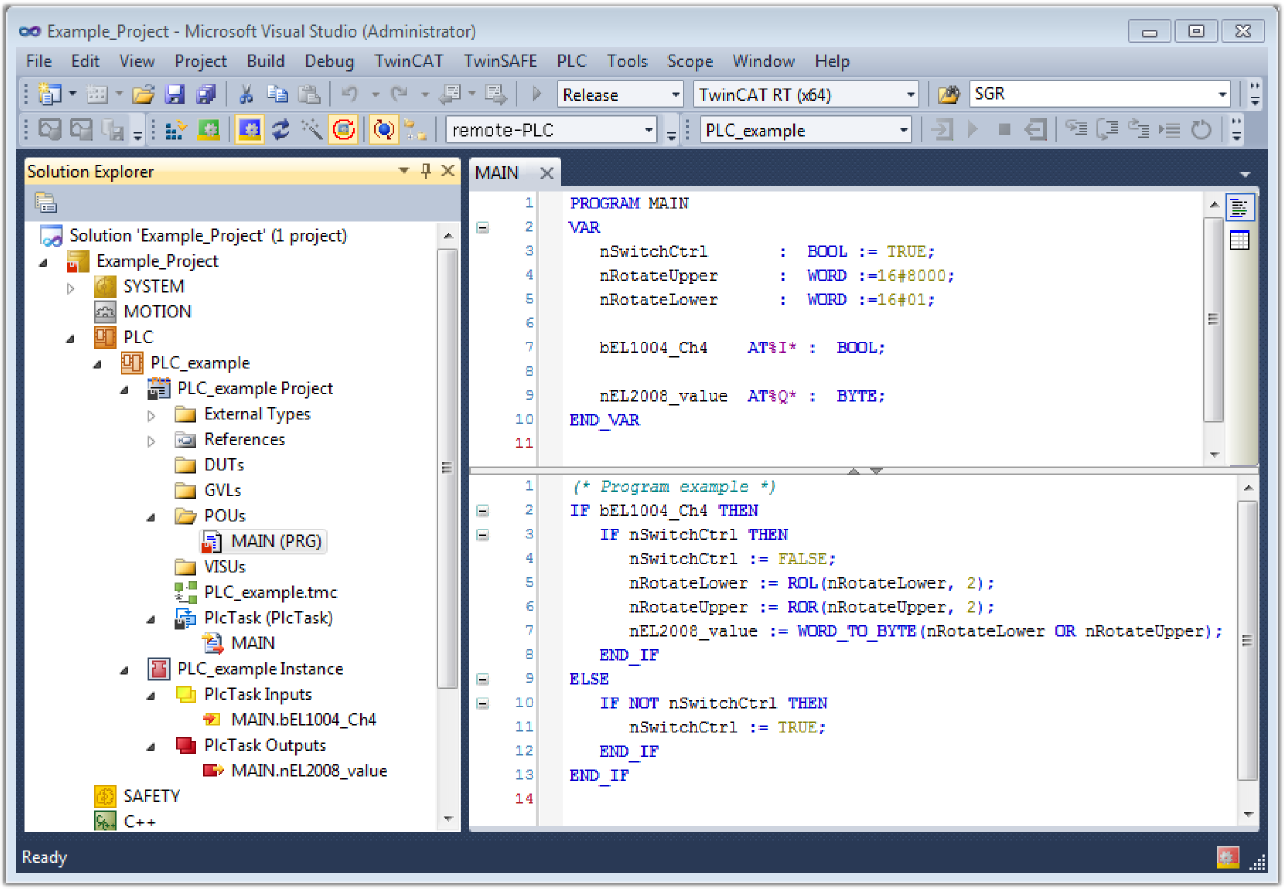


Fig. 82: Sample program with variables after a compile process (without variable integration)

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

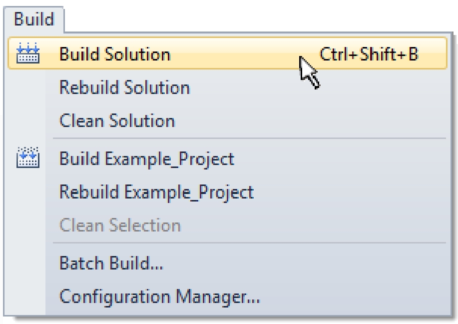
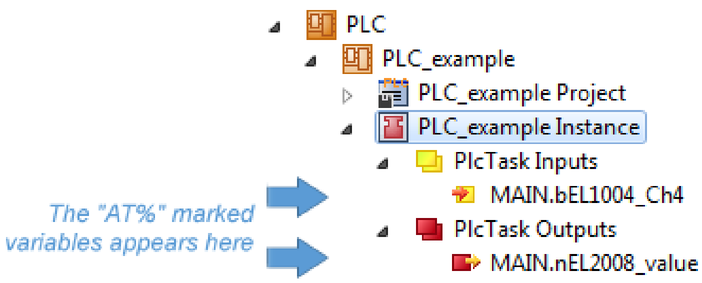


Fig. 83: Start program compilation

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



Assigning variables

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

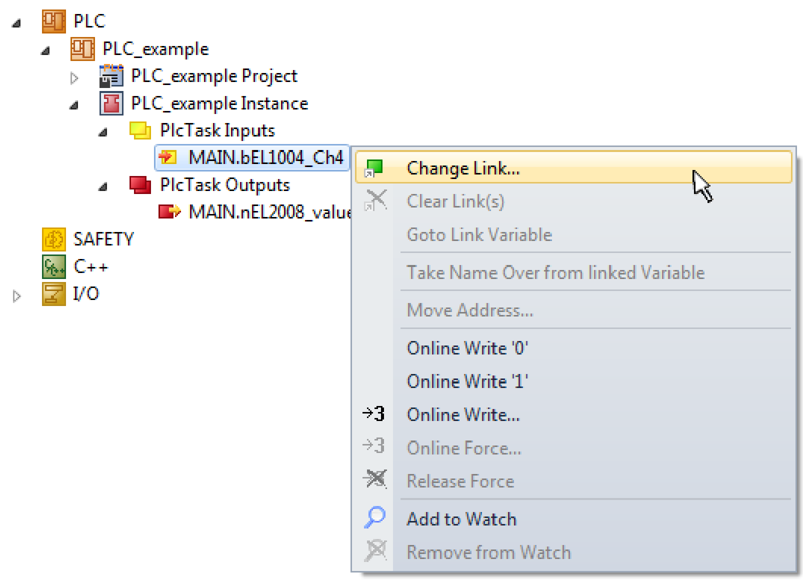


Fig. 84: Creating the links between PLC variables and process objects

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

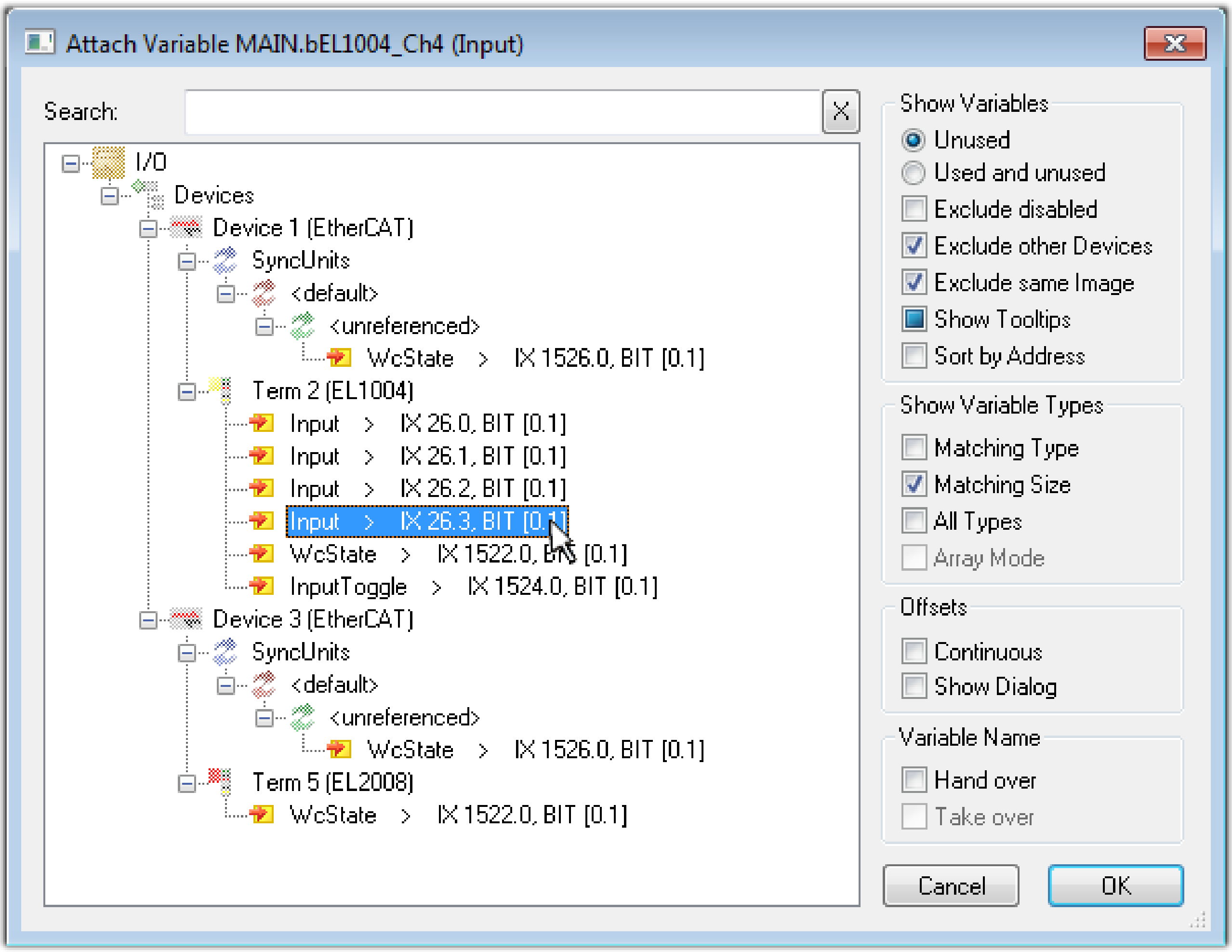


Fig. 85: Selecting PDO of type BOOL

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

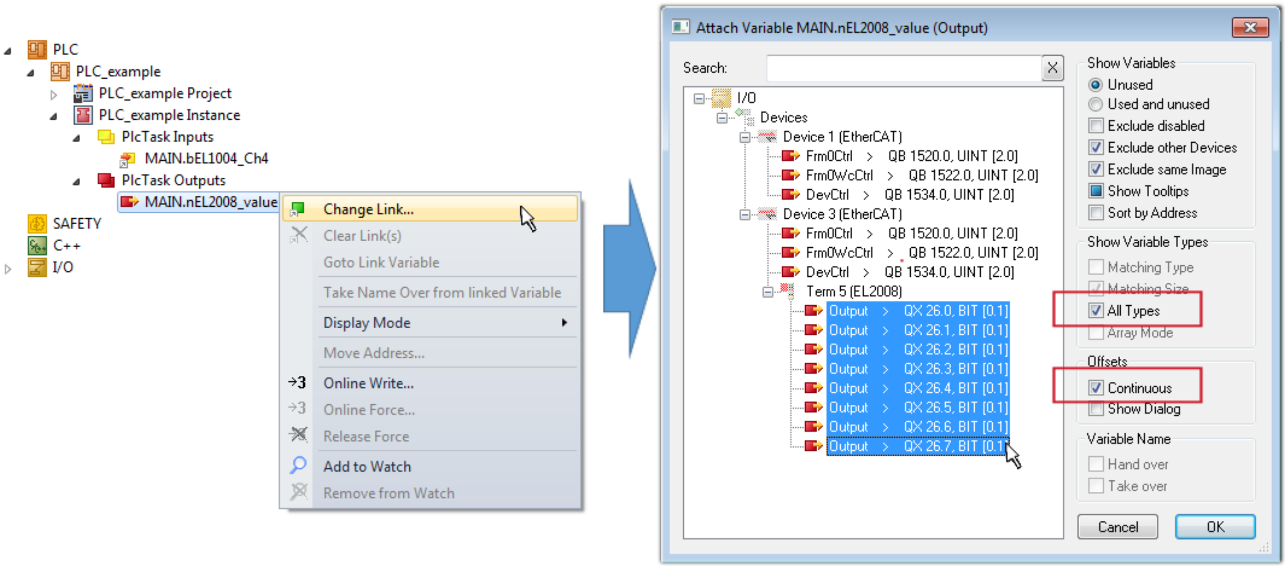


Fig. 86: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

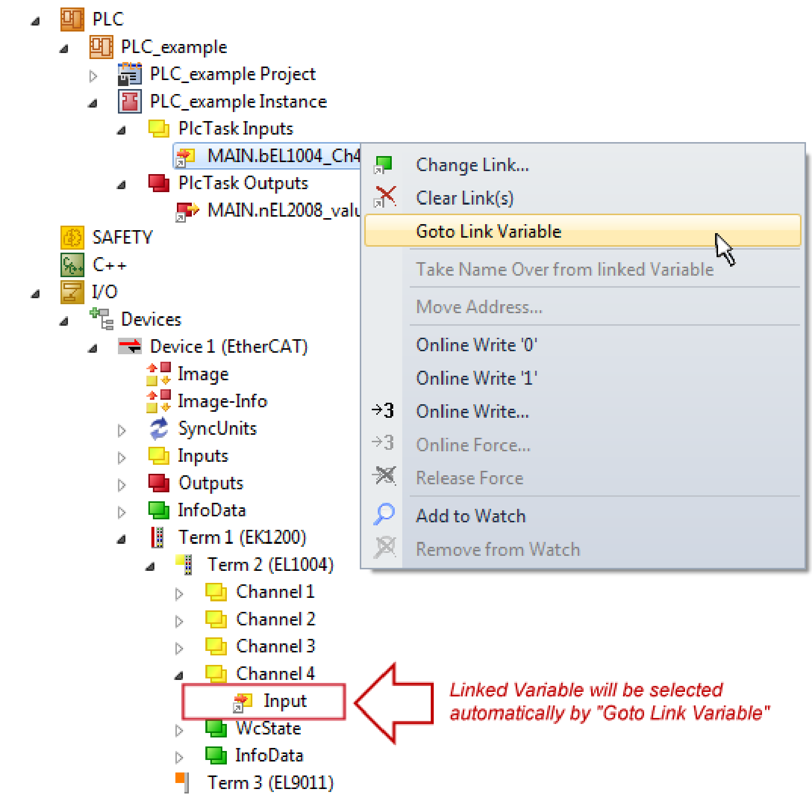


Fig. 87: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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

Note on the type of variable assignment

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

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

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

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

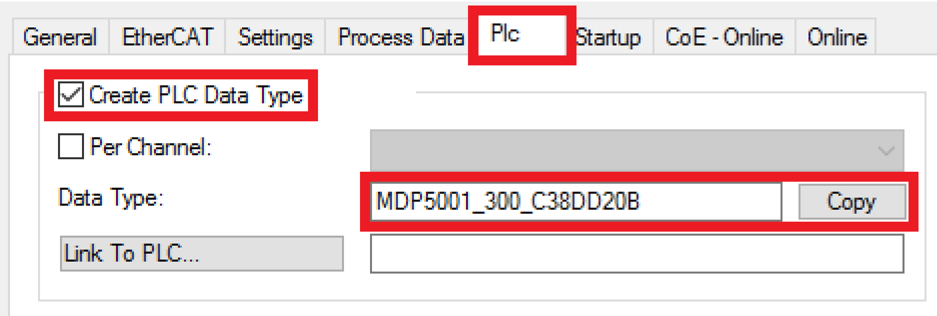


Fig. 88: Creating a PLC data type

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

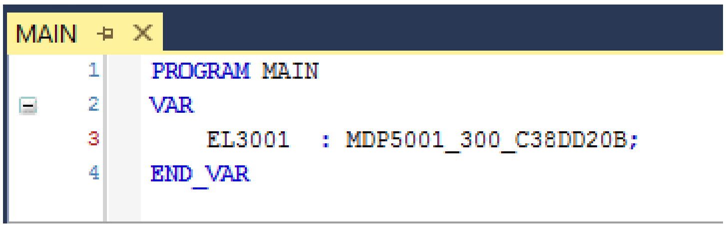


Fig. 89: Instance_of_struct

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

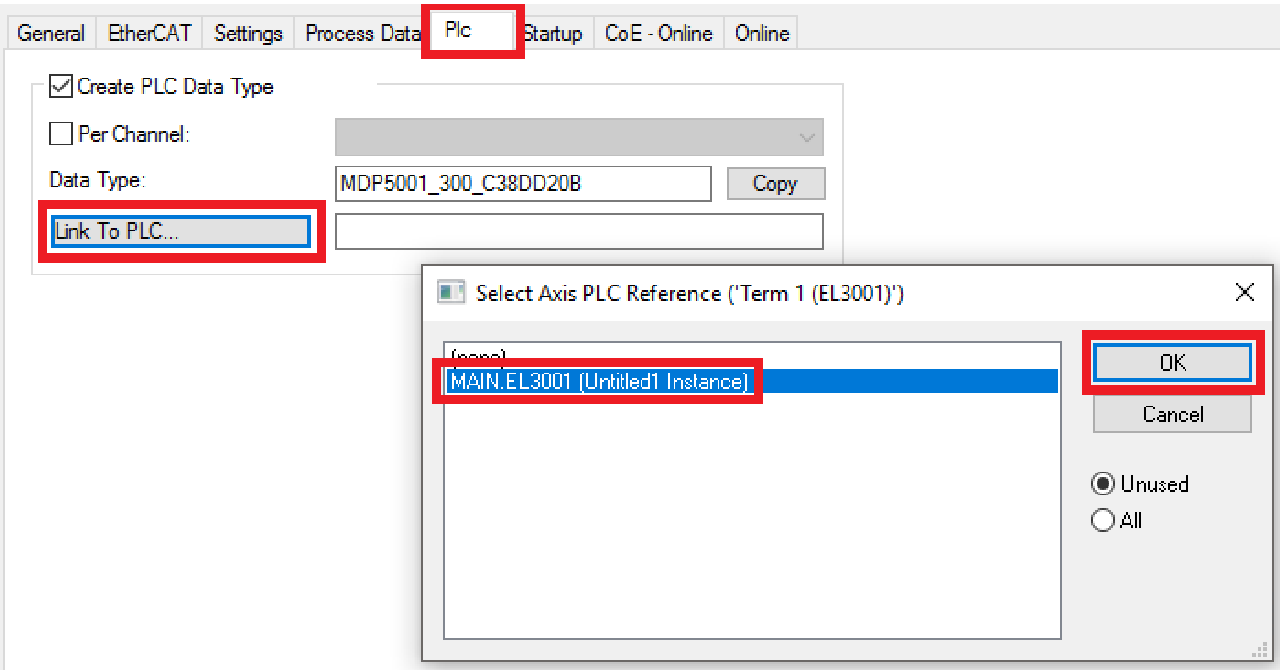


Fig. 90: Linking the structure

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

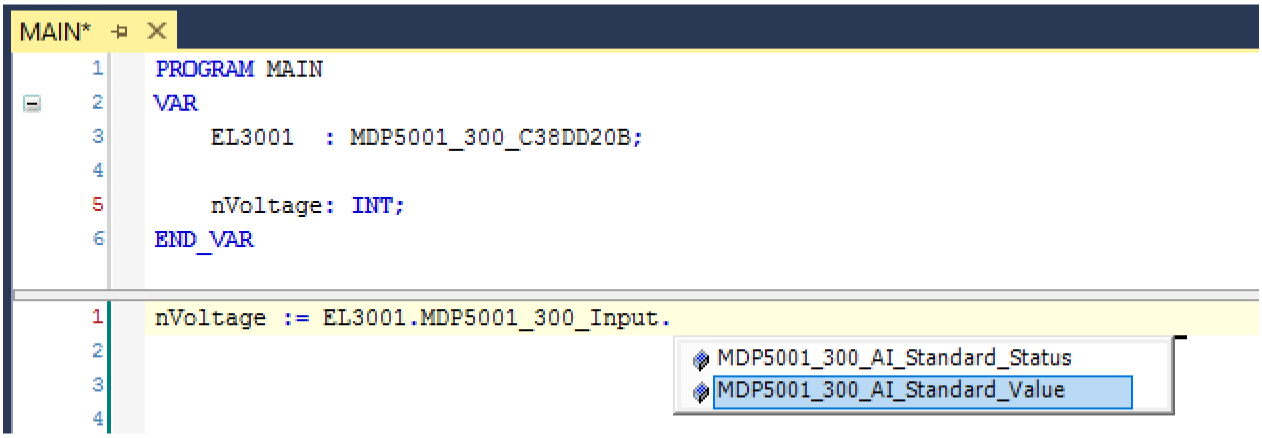
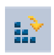
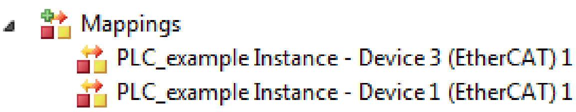


Fig. 91: Reading a variable from the structure of the process data

Activation of the configuration

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



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

Starting the controller

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

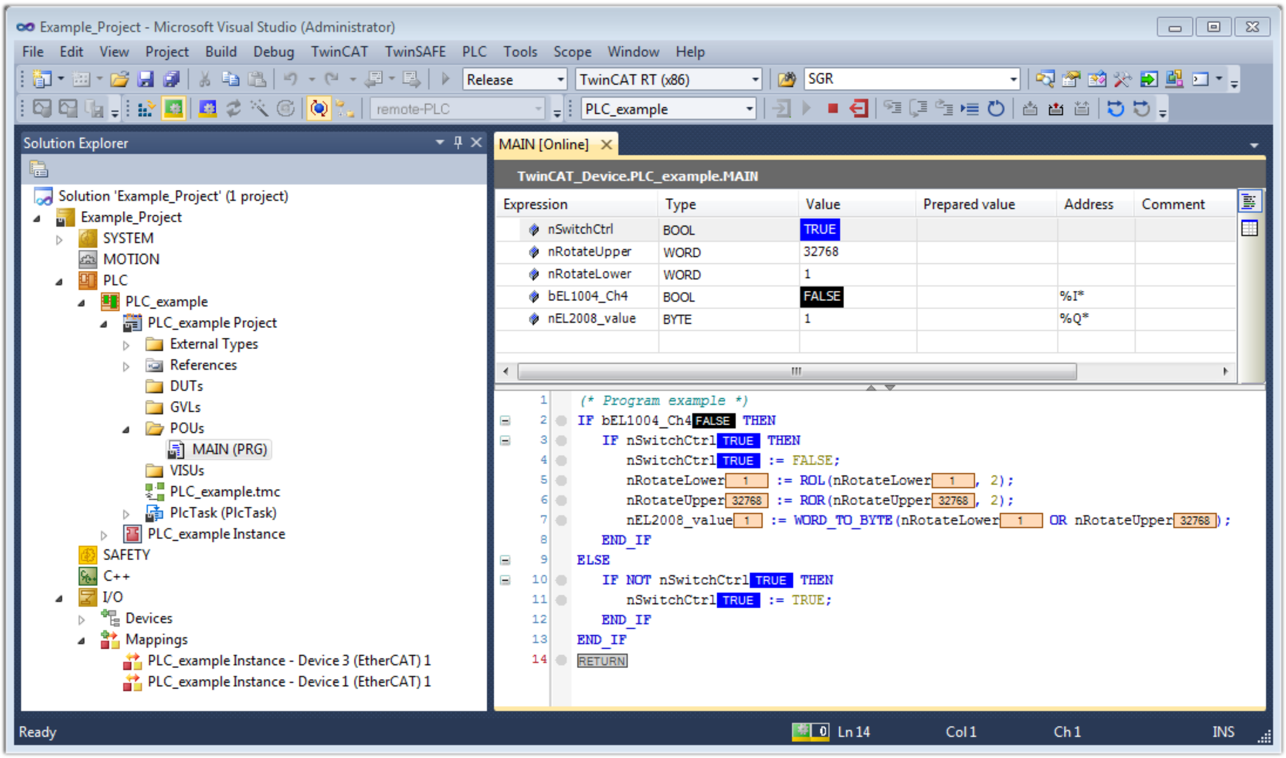


Fig. 92: TwinCAT development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
    - Connects I/O devices to tasks in a variable-oriented manner
    - Connects tasks to tasks in a variable-oriented manner
    - Supports units at the bit level
    - Supports synchronous or asynchronous relationships
    - Exchange of consistent data areas and process images
    - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)

- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways. 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.

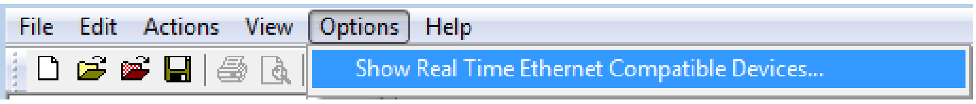


Fig. 93: System Manager “Options” (TwinCAT 2)

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

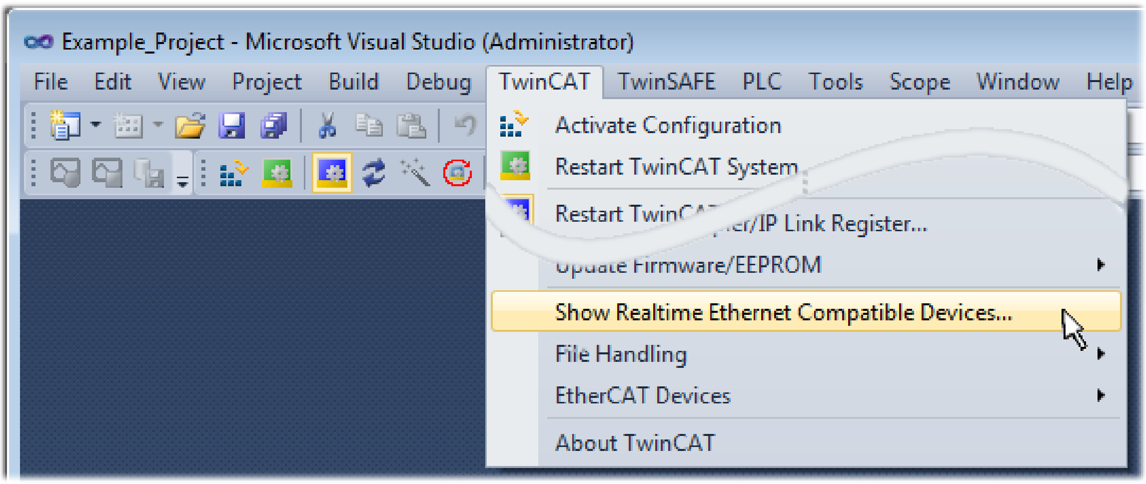


Fig. 94: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

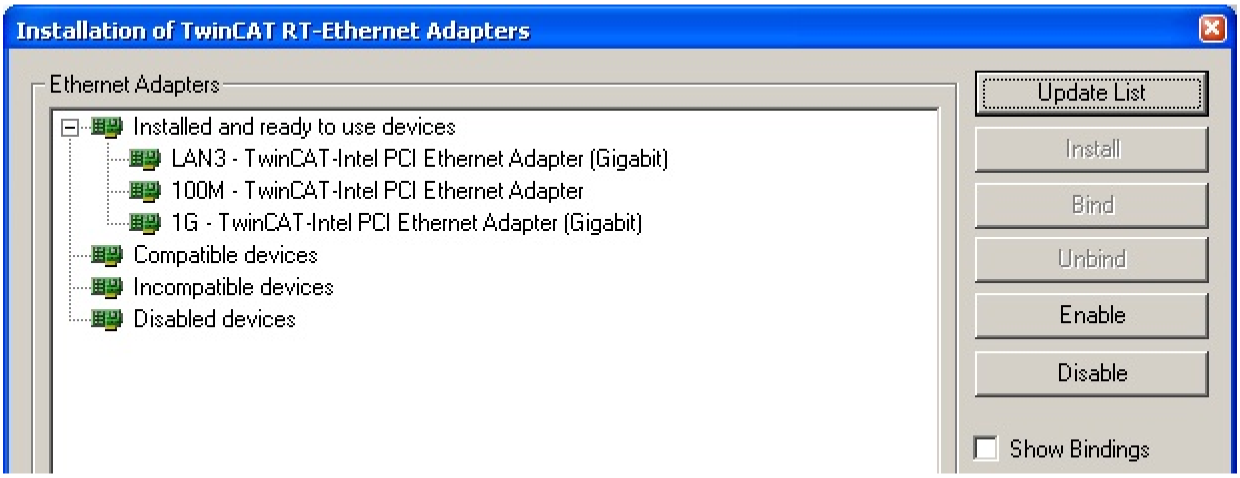


Fig. 95: Overview of network interfaces

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

A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▸ 103] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):

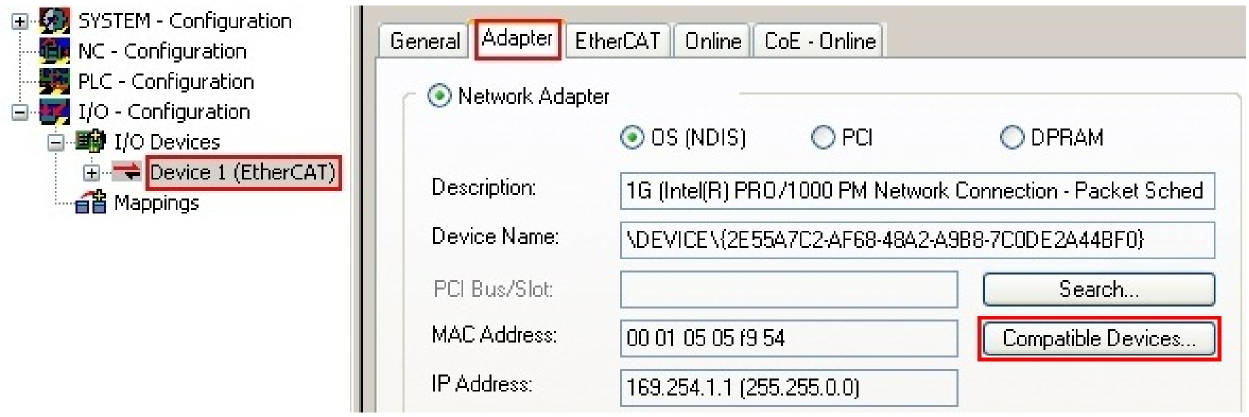
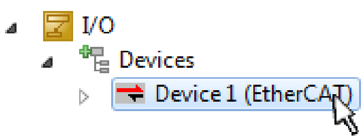


Fig. 96: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices…” of tab “Adapte””

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



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

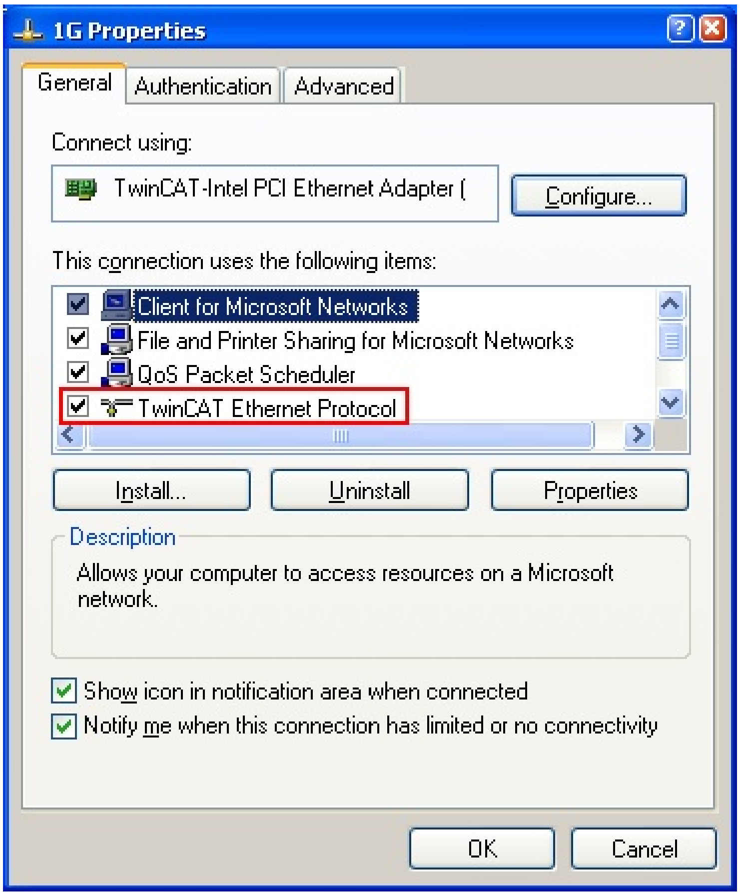


Fig. 97: Windows properties of the network interface

A correct setting of the driver could be:

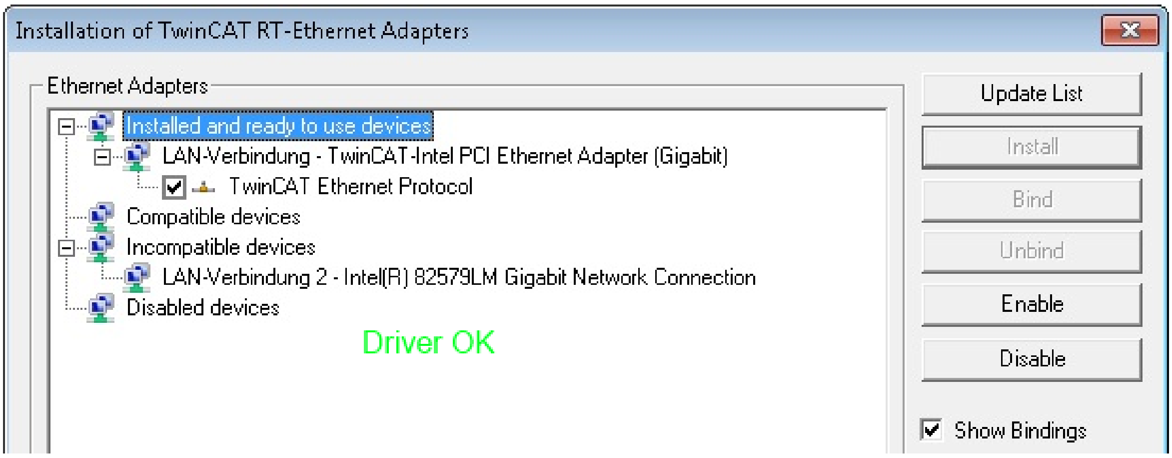


Fig. 98: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

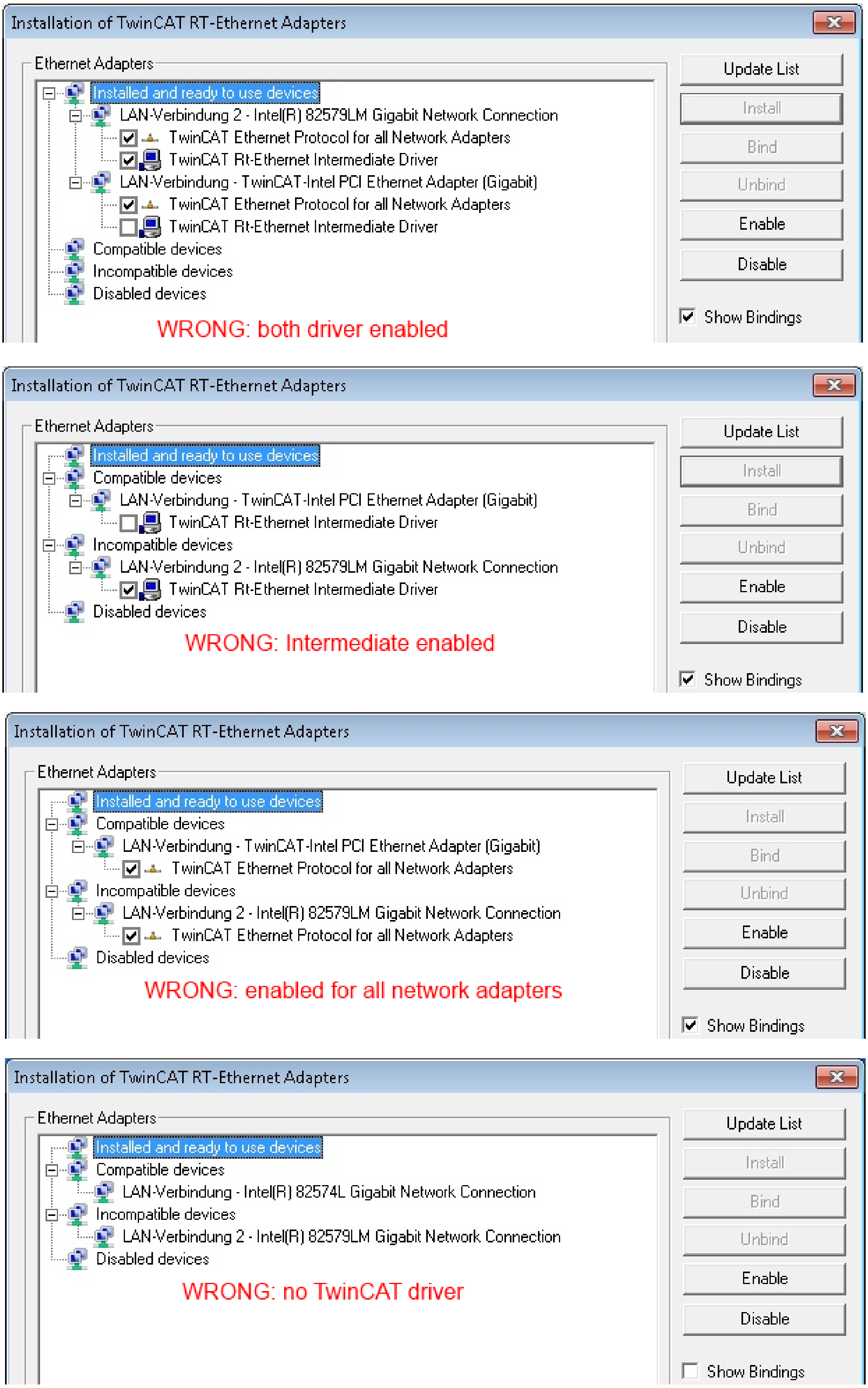


Fig. 99: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

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

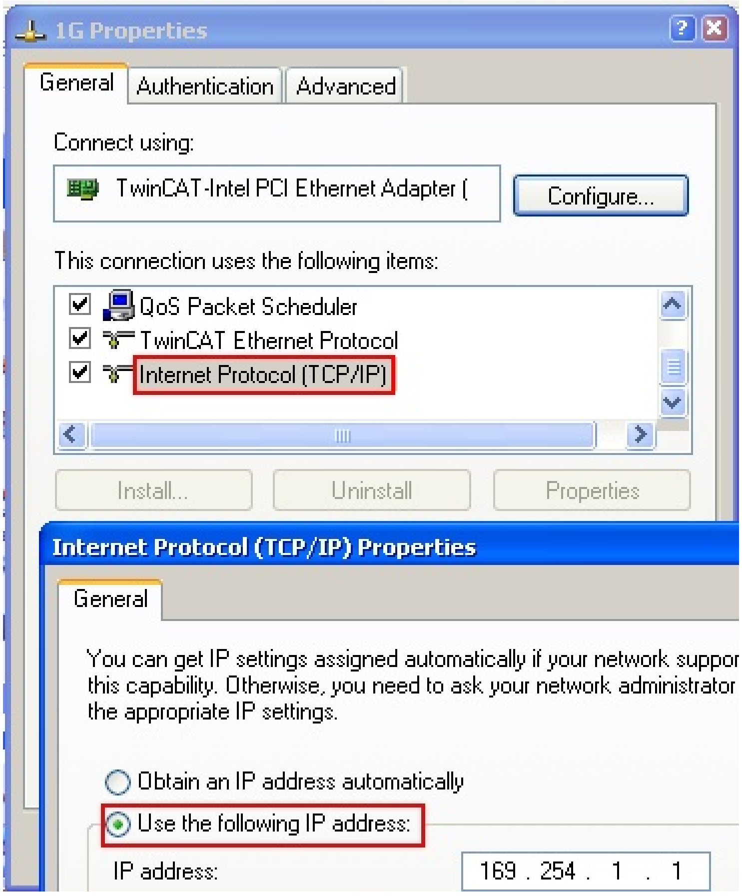


Fig. 100: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

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

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

Default settings:

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

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

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

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

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

The TwinCAT ESI Updater [▶ 102] is available for this purpose.

ESI

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

Device differentiation

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

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

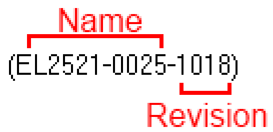


Fig. 101: Identifier structure

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.

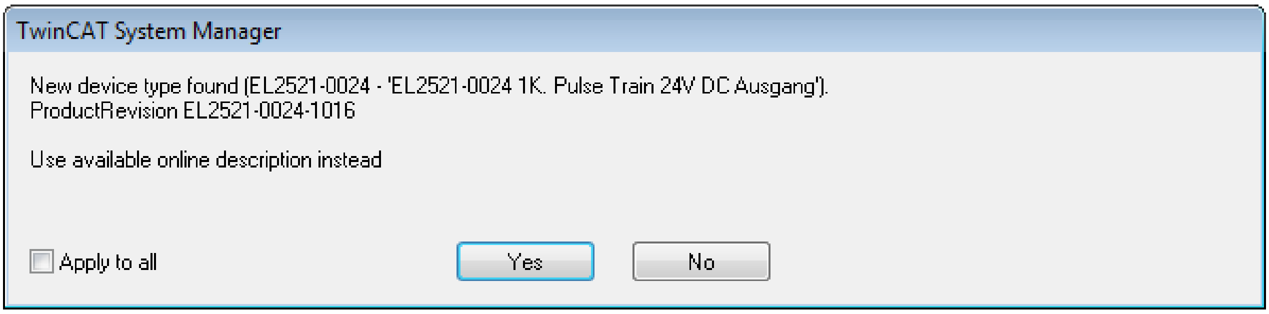


Fig. 102: OnlineDescription information window (TwinCAT 2)

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

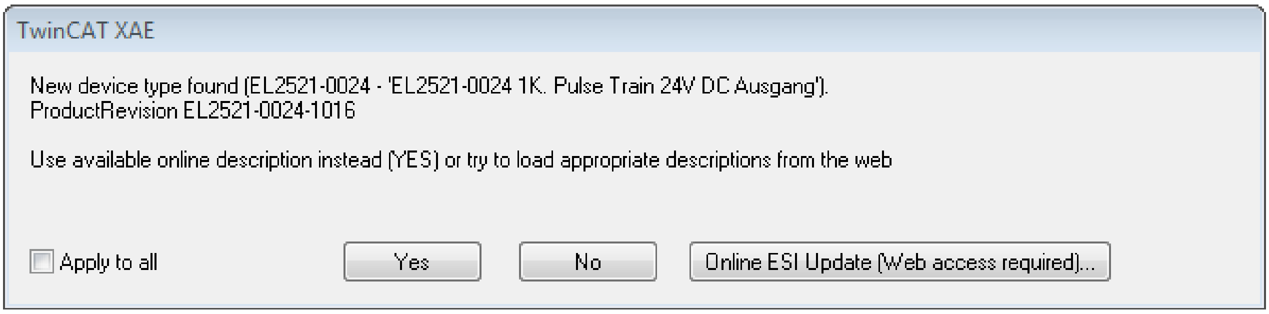


Fig. 103: Information window OnlineDescription (TwinCAT 3)

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

NOTE

Changing the "usual" configuration through a scan

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

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

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

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 103]".

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

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

Fig. 104: File OnlineDescription.xml created by the System Manager

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

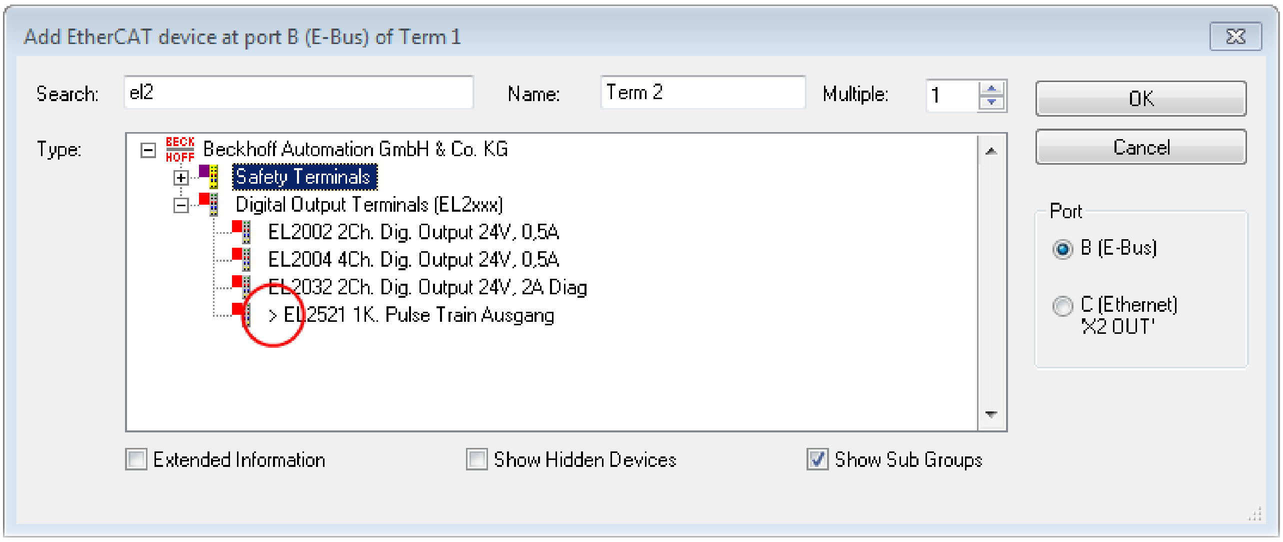


Fig. 105: Indication of an online recorded ESI of EL2521 as an example

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

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

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

OnlineDescription for TwinCAT 3.x

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

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

(Please note the language settings of the OS!)
You have to delete this file, too.

Faulty ESI file

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

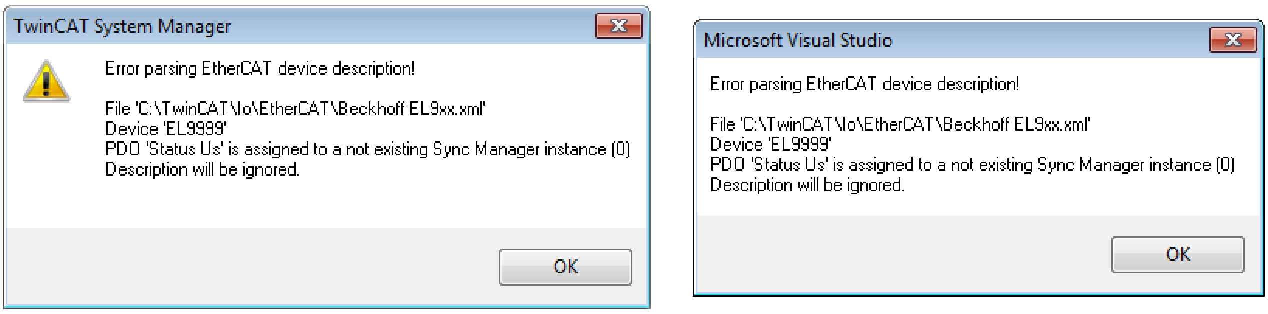


Fig. 106: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

5.2.3 TwinCAT ESI Updater

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

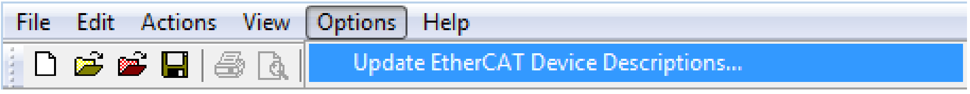


Fig. 107: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

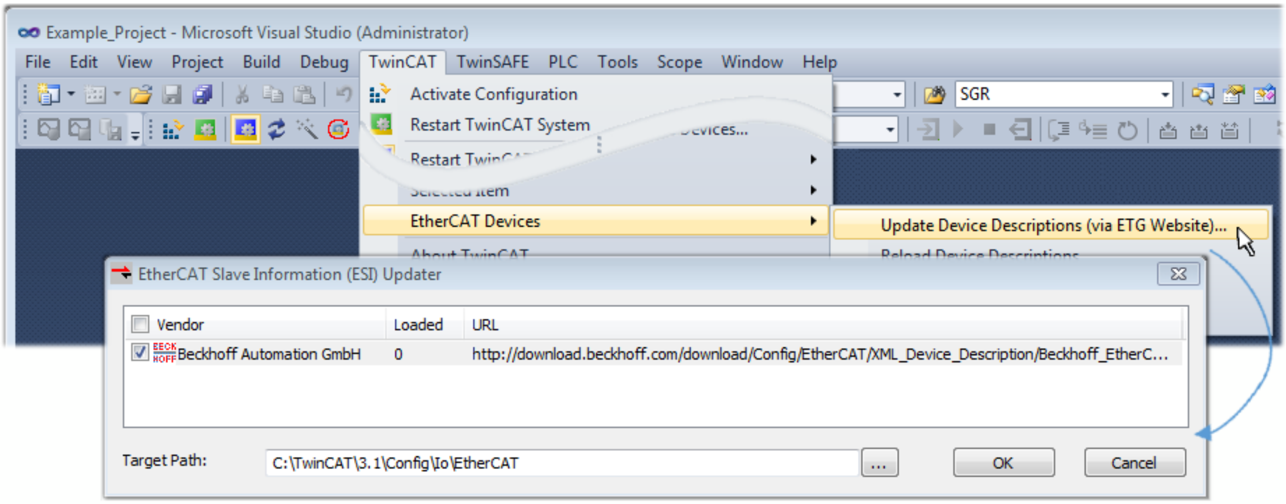


Fig. 108: Using the ESI Updater (TwinCAT 3)

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

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

5.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 98].

For preparation of a configuration:

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

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

The online scan process consists of:

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

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

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

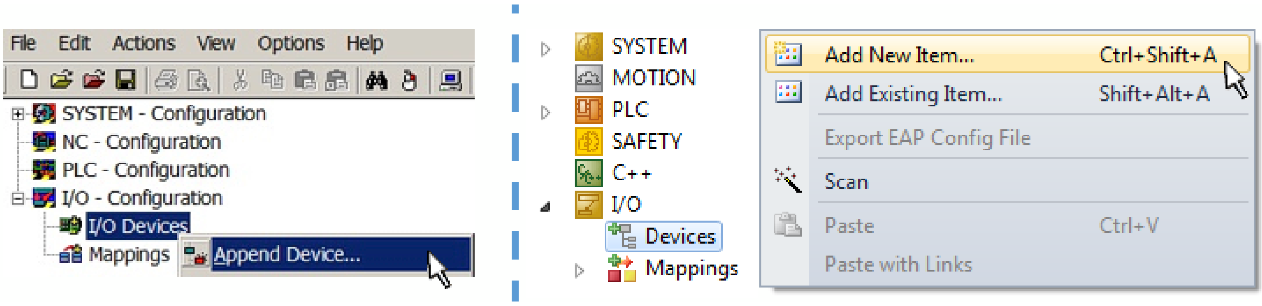


Fig. 109: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

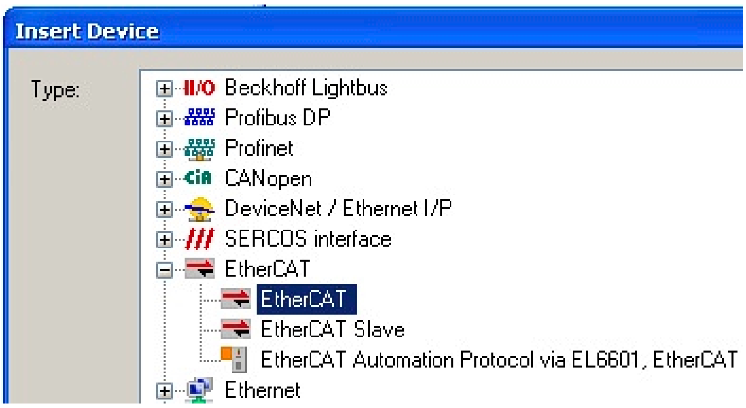


Fig. 110: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

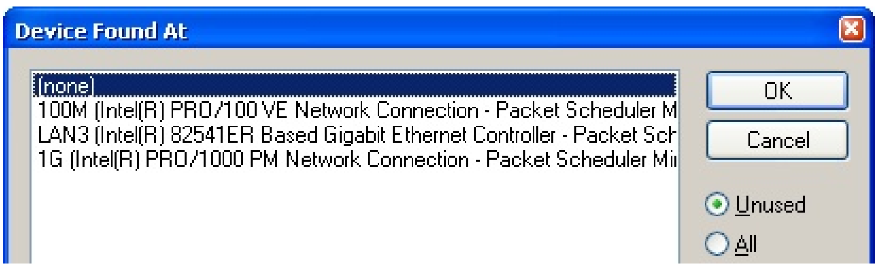


Fig. 111: Selecting the Ethernet port

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

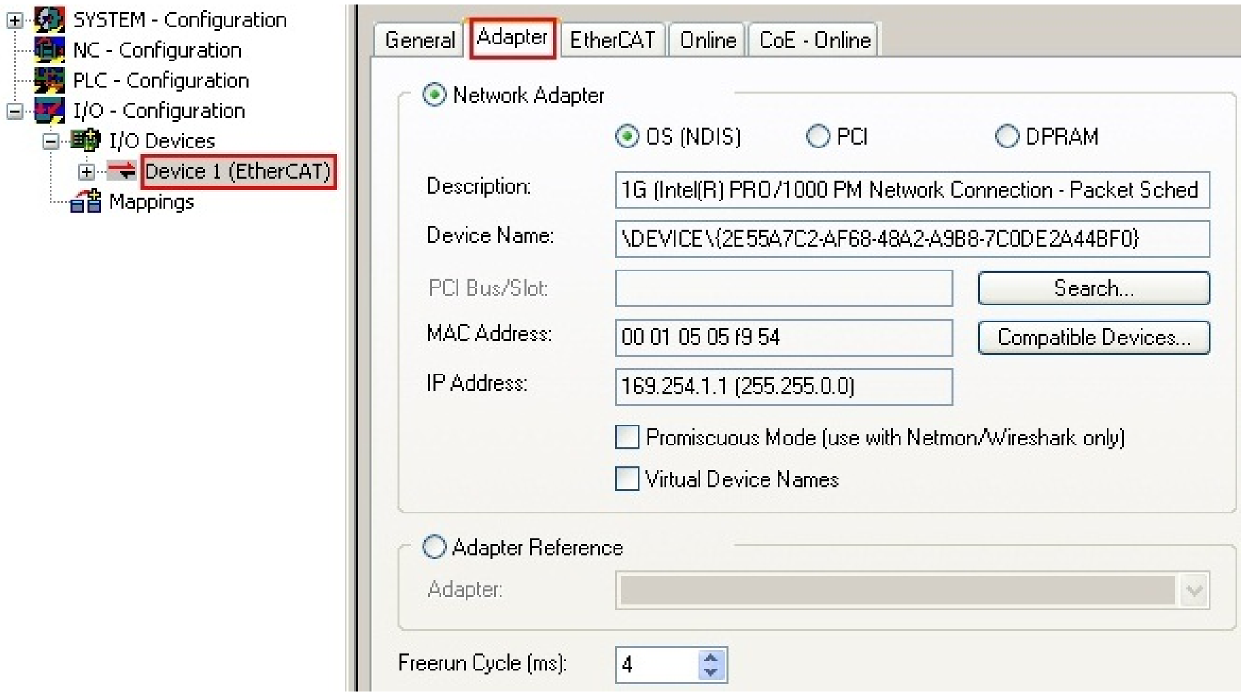
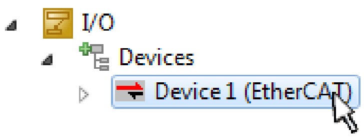


Fig. 112: EtherCAT device properties (TwinCAT 2)

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



Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 93].

Defining EtherCAT slaves

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

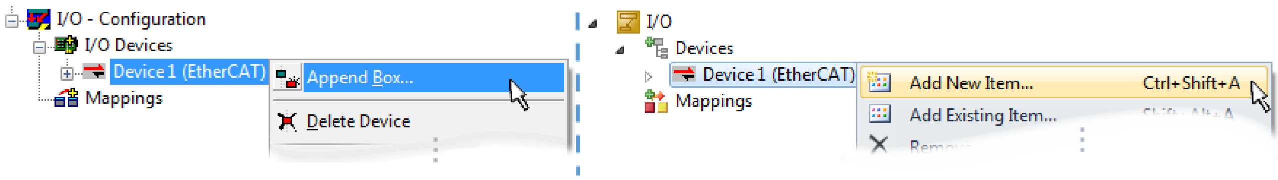


Fig. 113: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

- "Ethernet": cable-based 100BASE-TX: EK couplers, EP boxes, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, “EJ-module”: EL/ES terminals, various modular modules

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

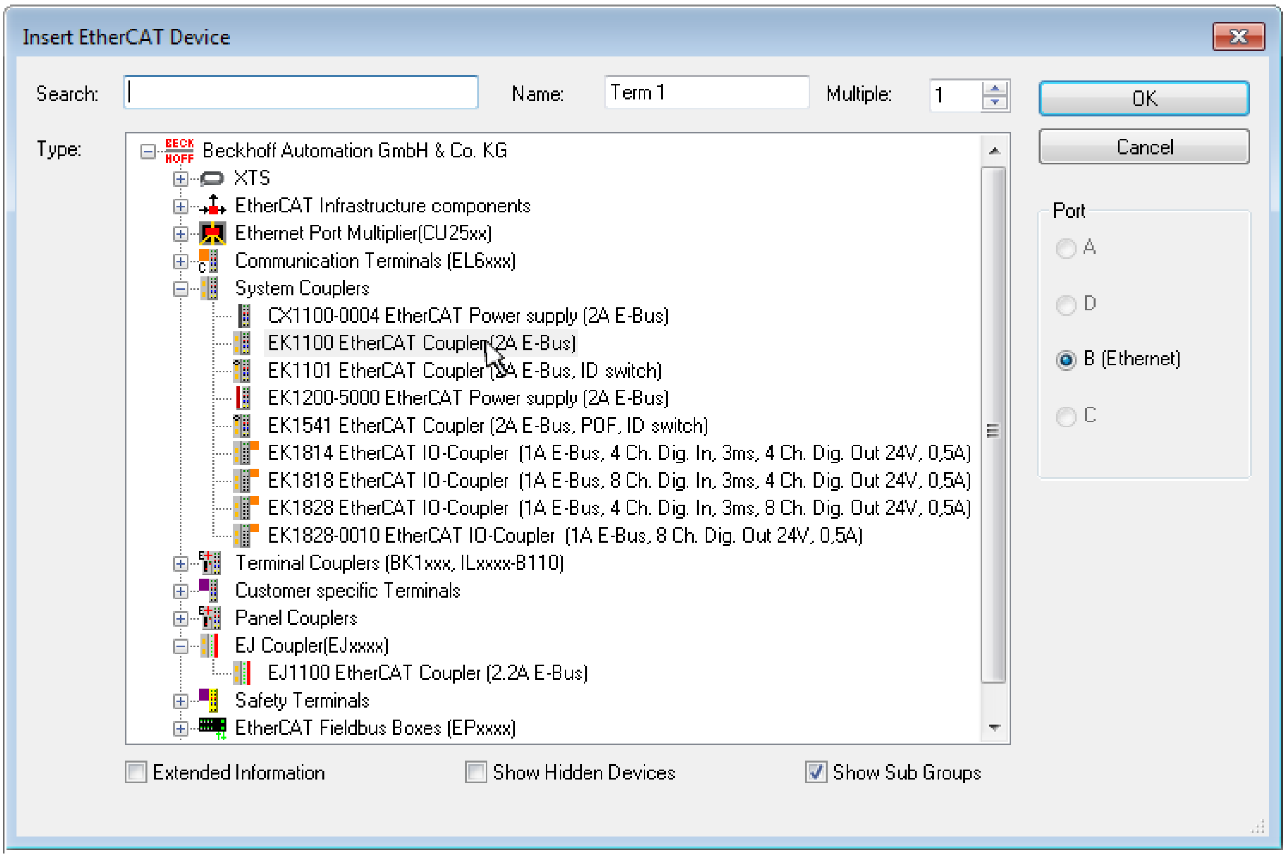


Fig. 114: Selection dialog for new EtherCAT device

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”.

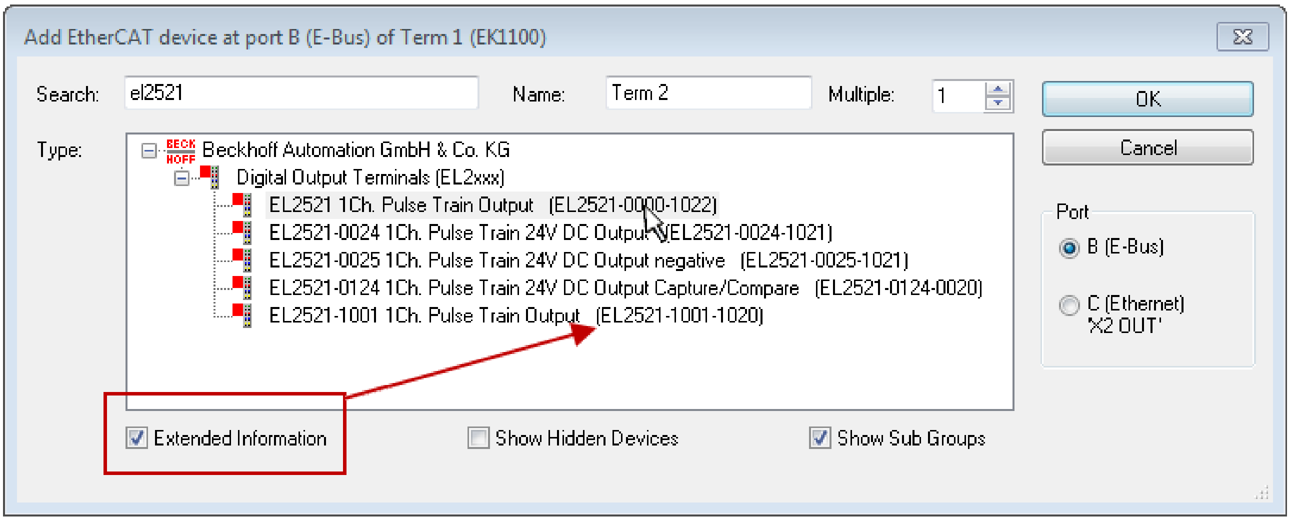


Fig. 115: Display of device revision

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”.

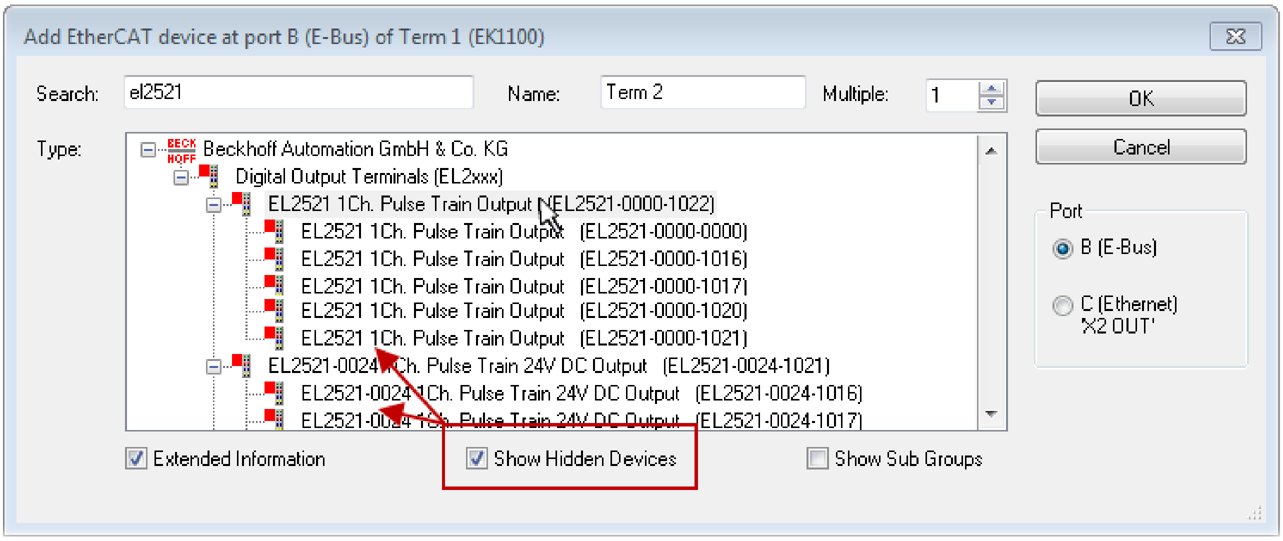


Fig. 116: Display of previous revisions

Device selection based on revision, compatibility

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

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

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

Example

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

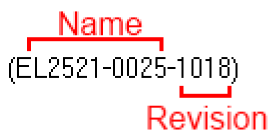


Fig. 117: Name/revision of the terminal

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

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

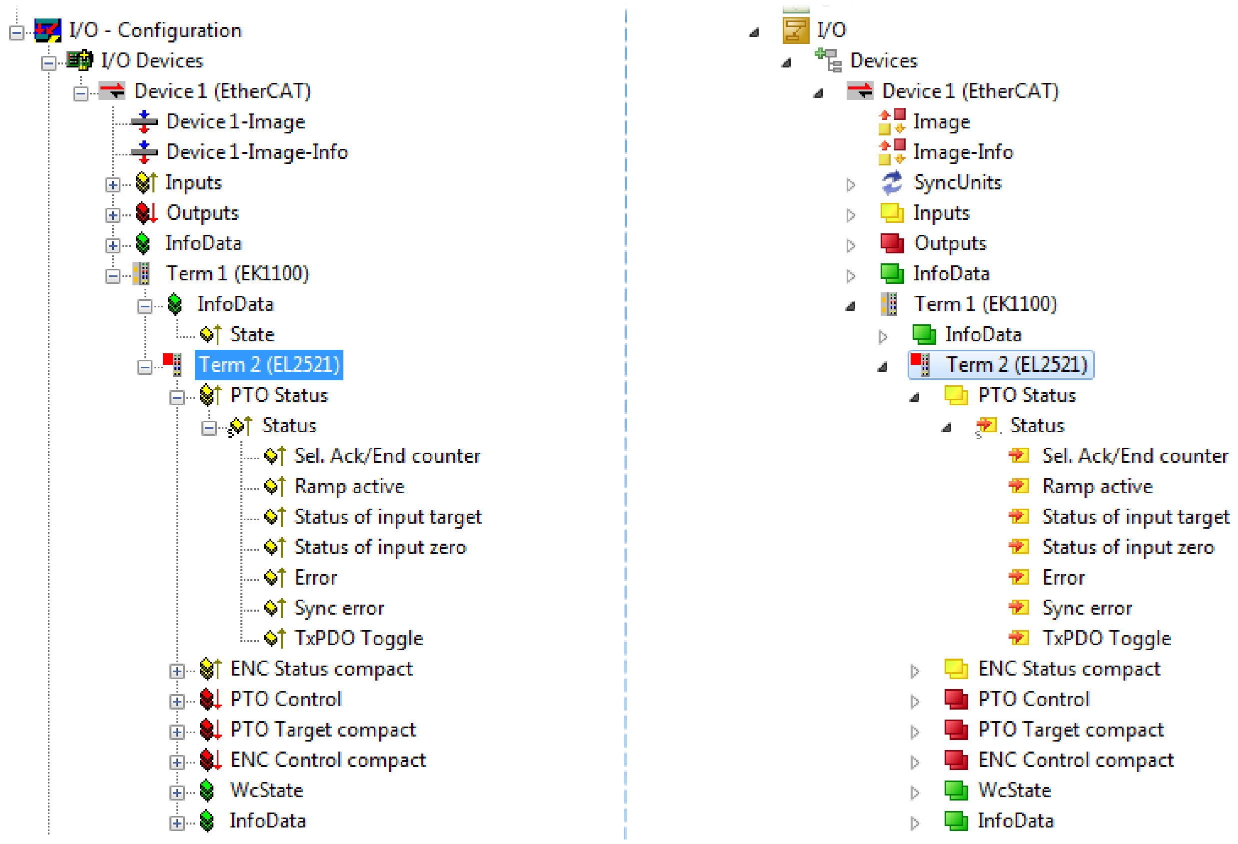


Fig. 118: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

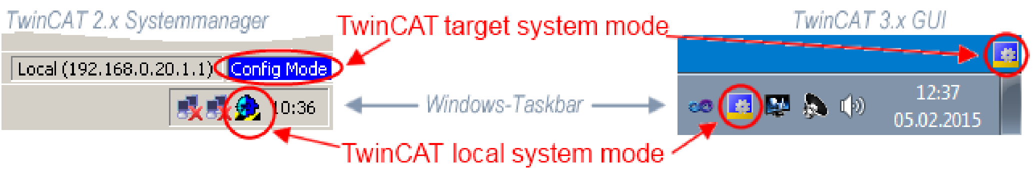


Fig. 119: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

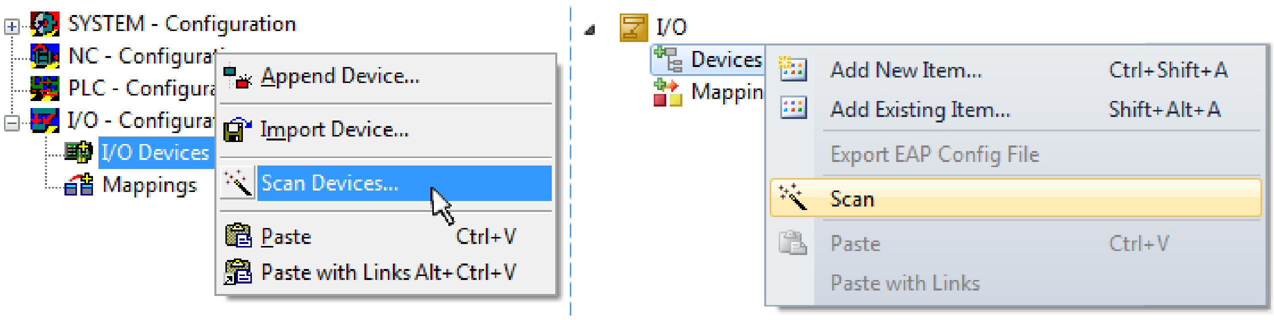


Fig. 120: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

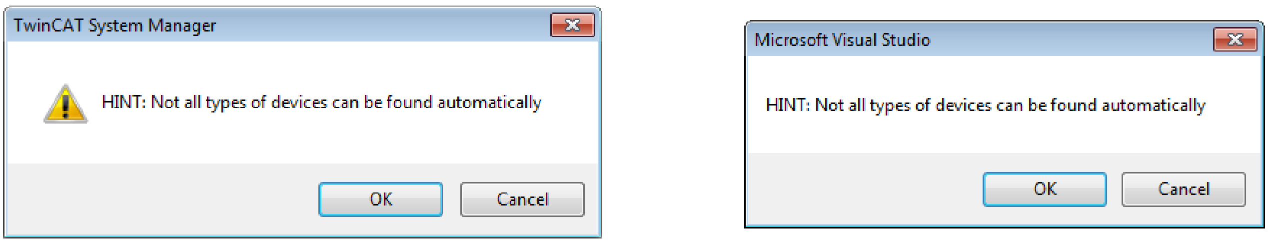


Fig. 121: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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

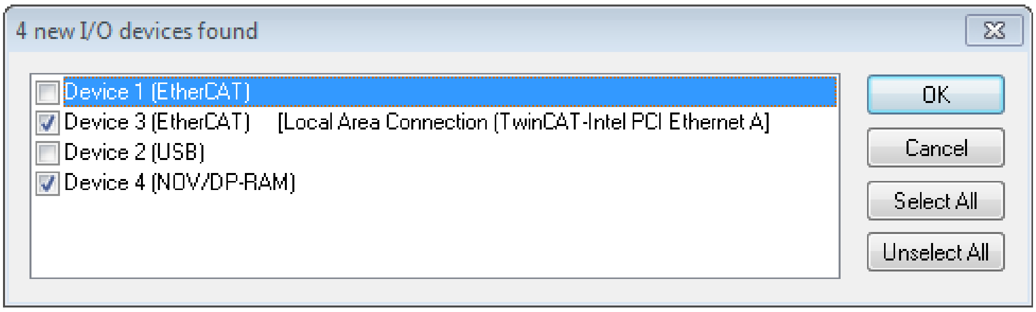


Fig. 122: Detected Ethernet devices

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

Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 93].

Detecting/Scanning the EtherCAT devices

Online scan functionality

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

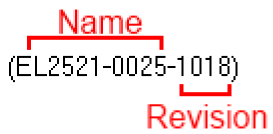


Fig. 123: Example default state

NOTE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 113] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

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

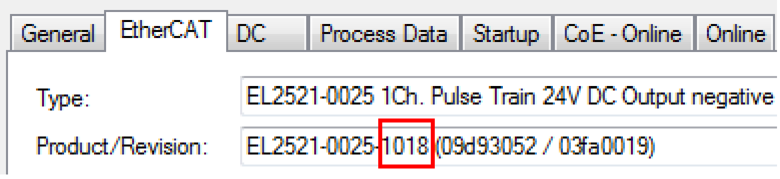


Fig. 124: Installing EthetCAT terminal with revision -1018

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

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

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

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

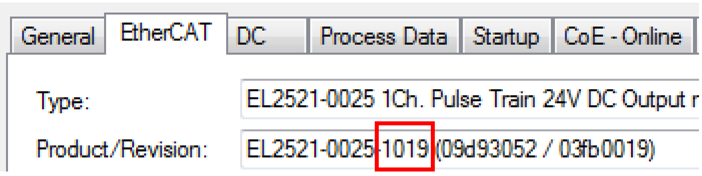


Fig. 125: Detection of EtherCAT terminal with revision -1019

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

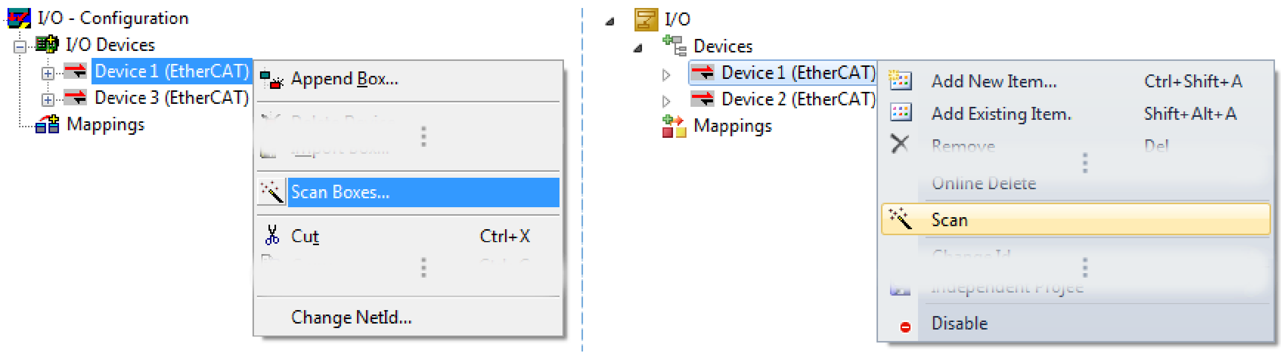


Fig. 127: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

Fig. 128: *Scan progressexemplary by TwinCAT 2*

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

Fig. 129: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

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

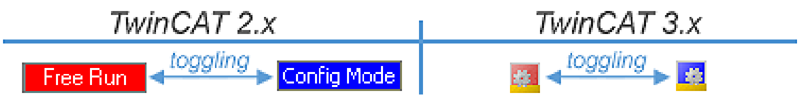


Fig. 130: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

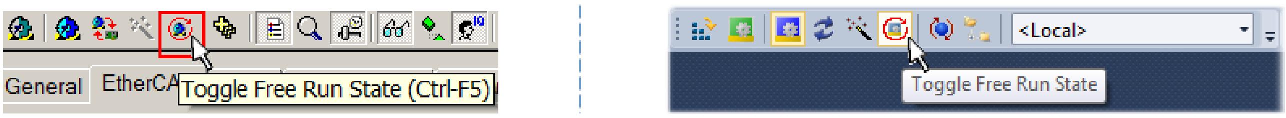


Fig. 131: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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

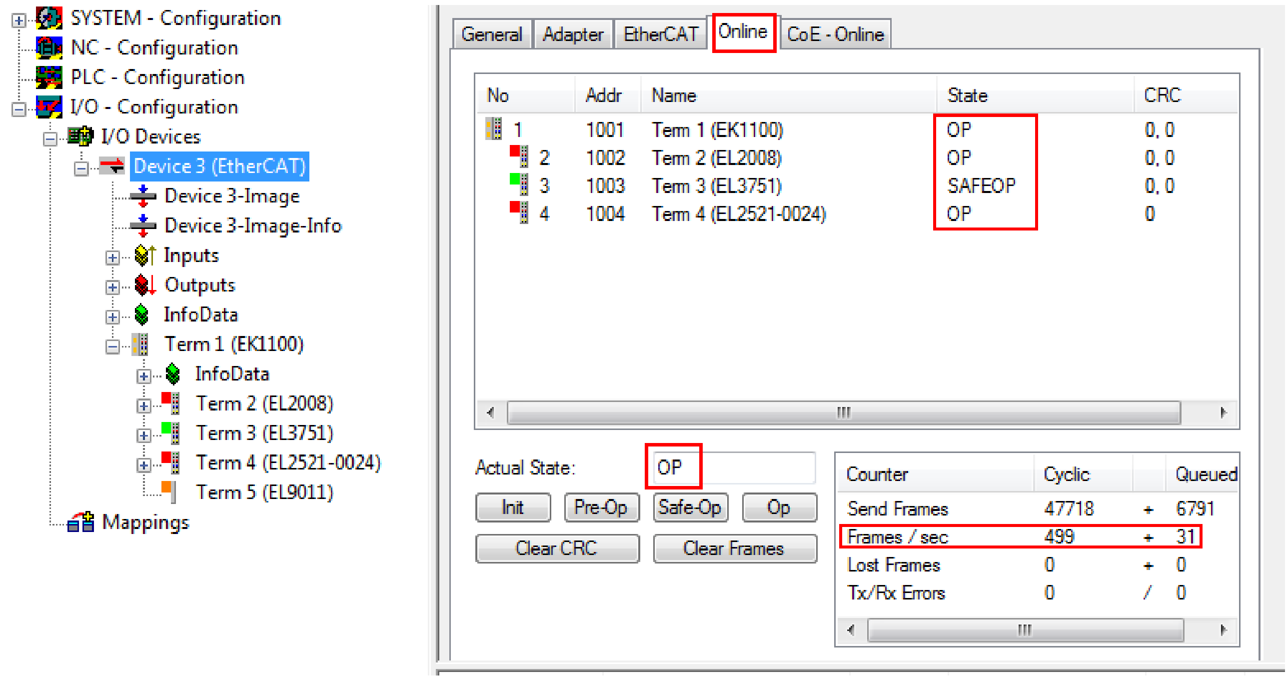


Fig. 132: Online display example

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 [▶ 103].

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.

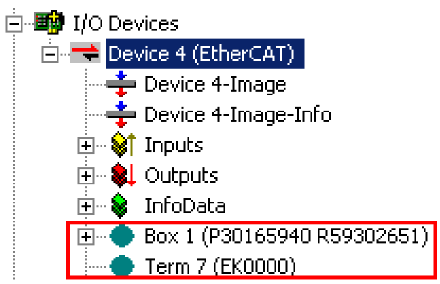


Fig. 133: Faulty identification

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

Scan over existing Configuration

NOTE

Change of the configuration after comparison

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

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

Fig. 134: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

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

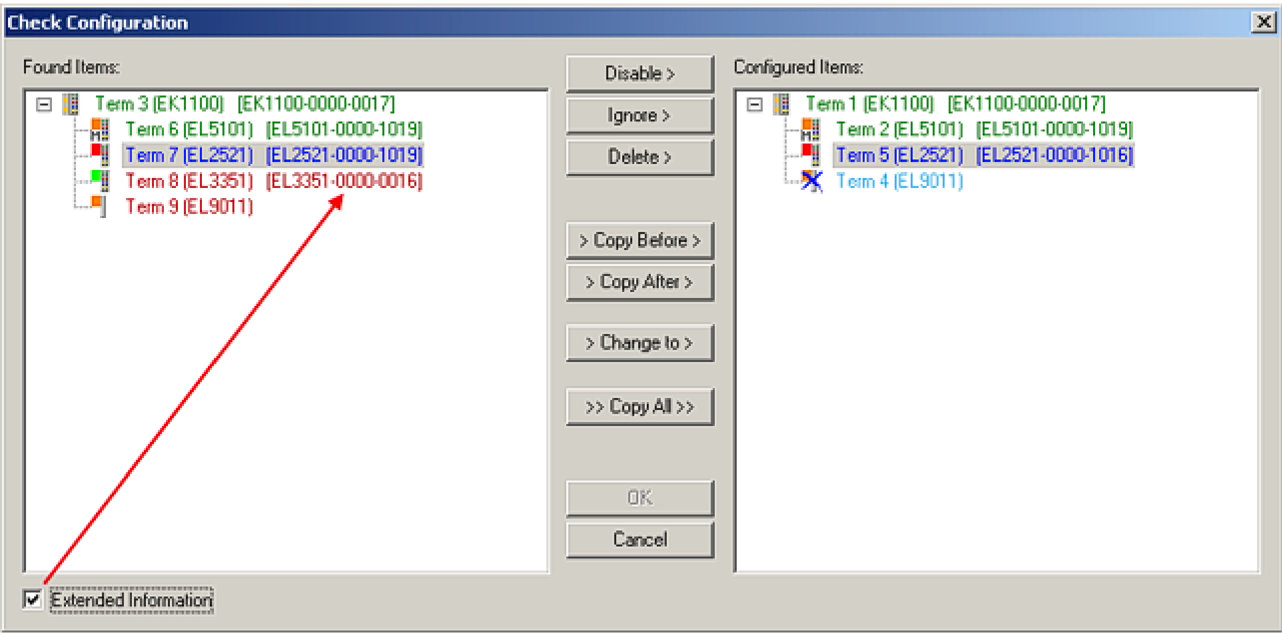


Fig. 135: Correction dialog

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

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored (“Ignore” button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

Device selection based on revision, compatibility

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

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

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

Example

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

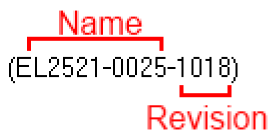


Fig. 136: Name/revision of the terminal

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

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

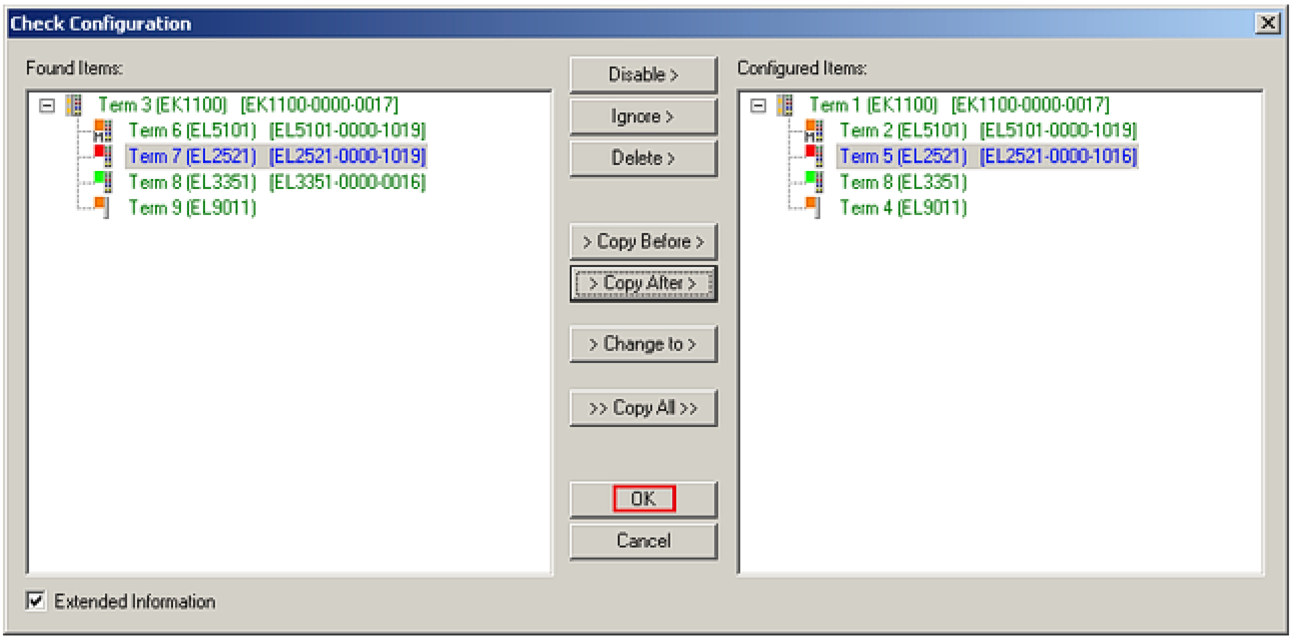


Fig. 137: Correction dialog with modifications

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

Change to Compatible Type

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

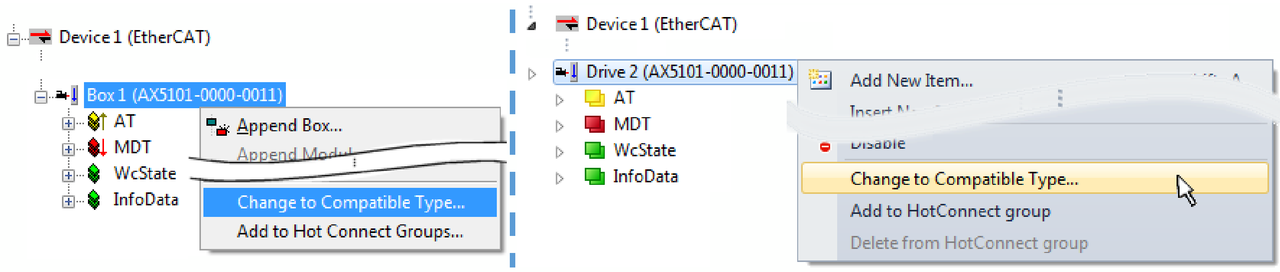


Fig. 138: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

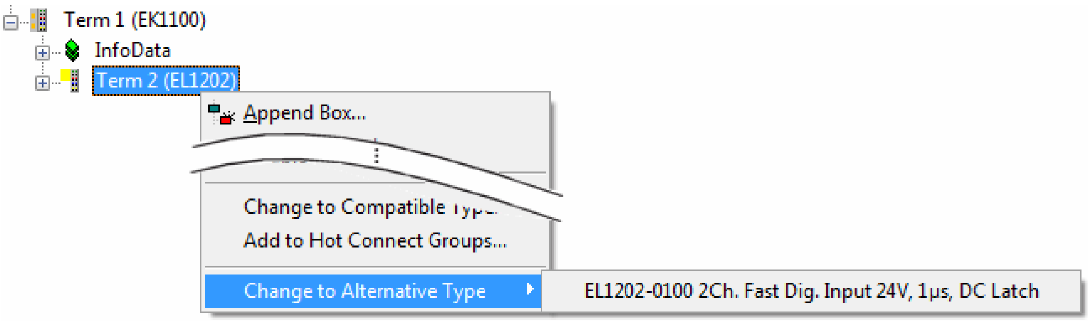


Fig. 139: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.7 EtherCAT subscriber configuration

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

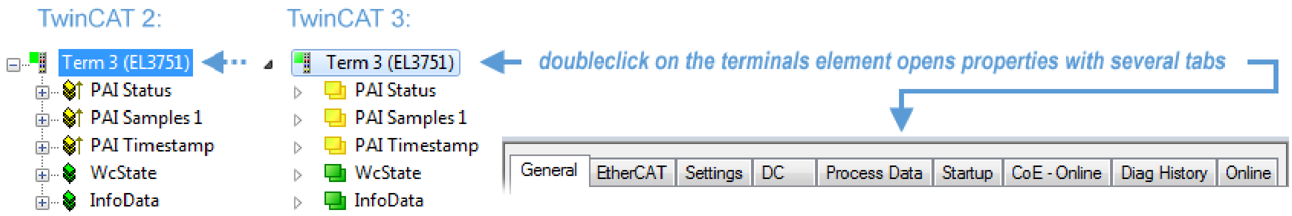


Fig. 140: Branch element as terminal EL3751

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

“General” tab

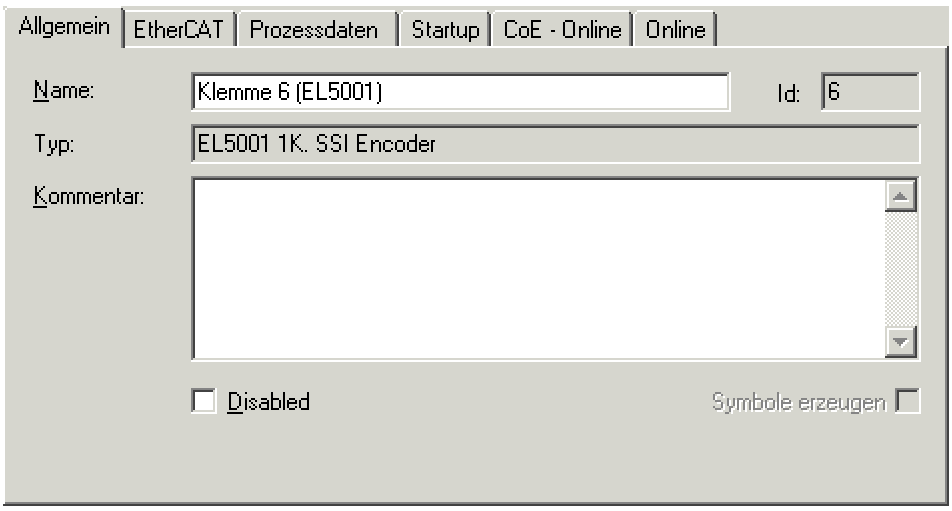


Fig. 141: “General” tab

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

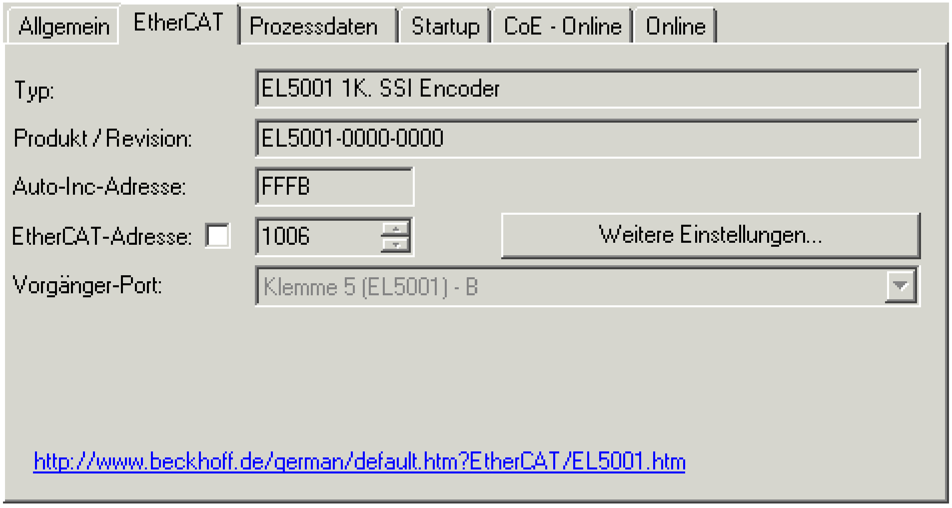


Fig. 142: “EtherCAT” tab

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

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

“Process Data” tab

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

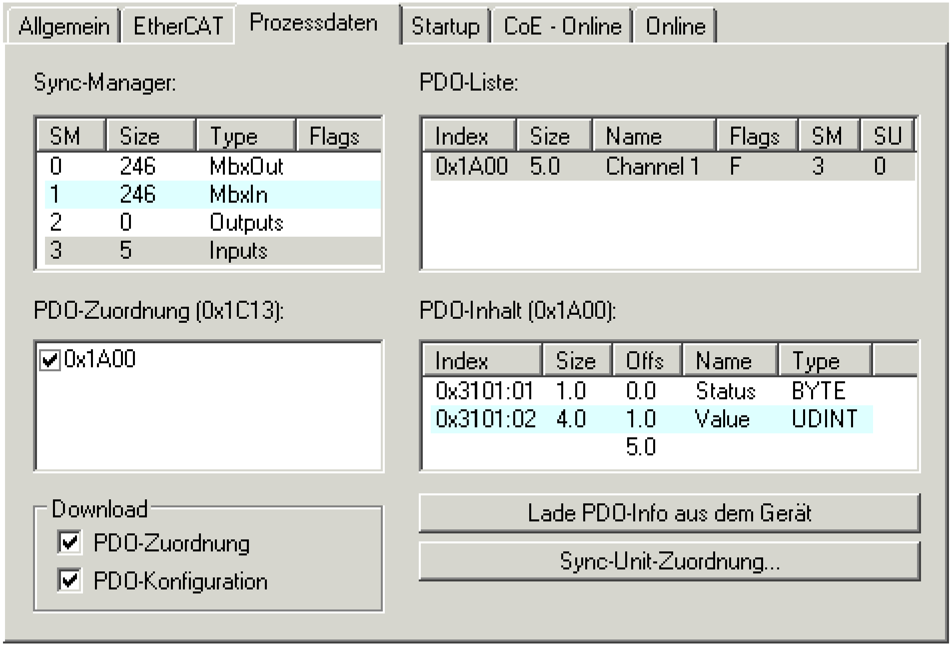


Fig. 143: “Process Data” tab

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

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

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

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

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

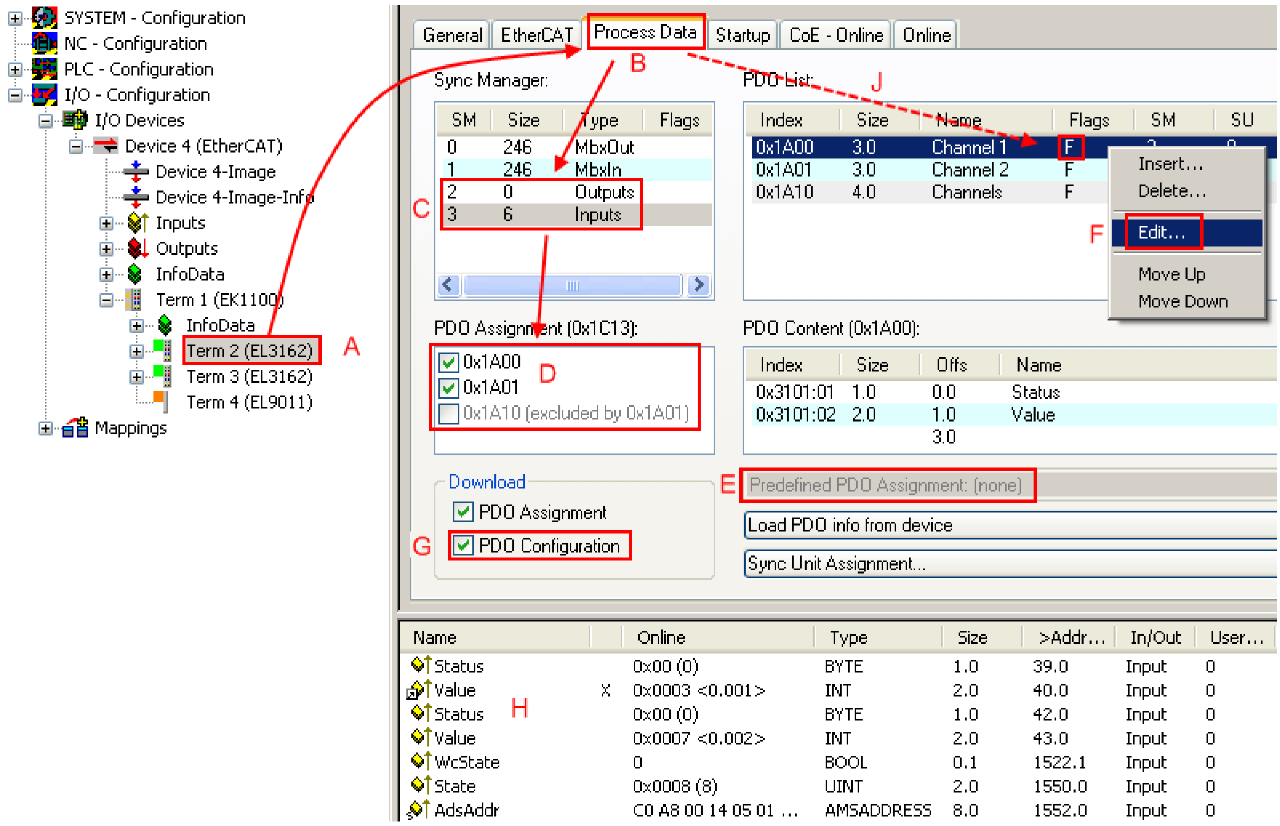


Fig. 144: Configuring the process data

Manual modification of the process data

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

A detailed description [▸ 124] can be found at the end of this section.

“Startup” tab

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

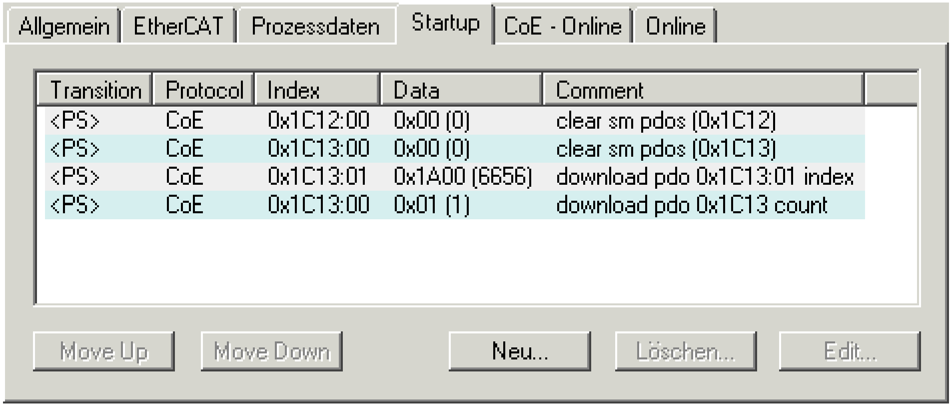


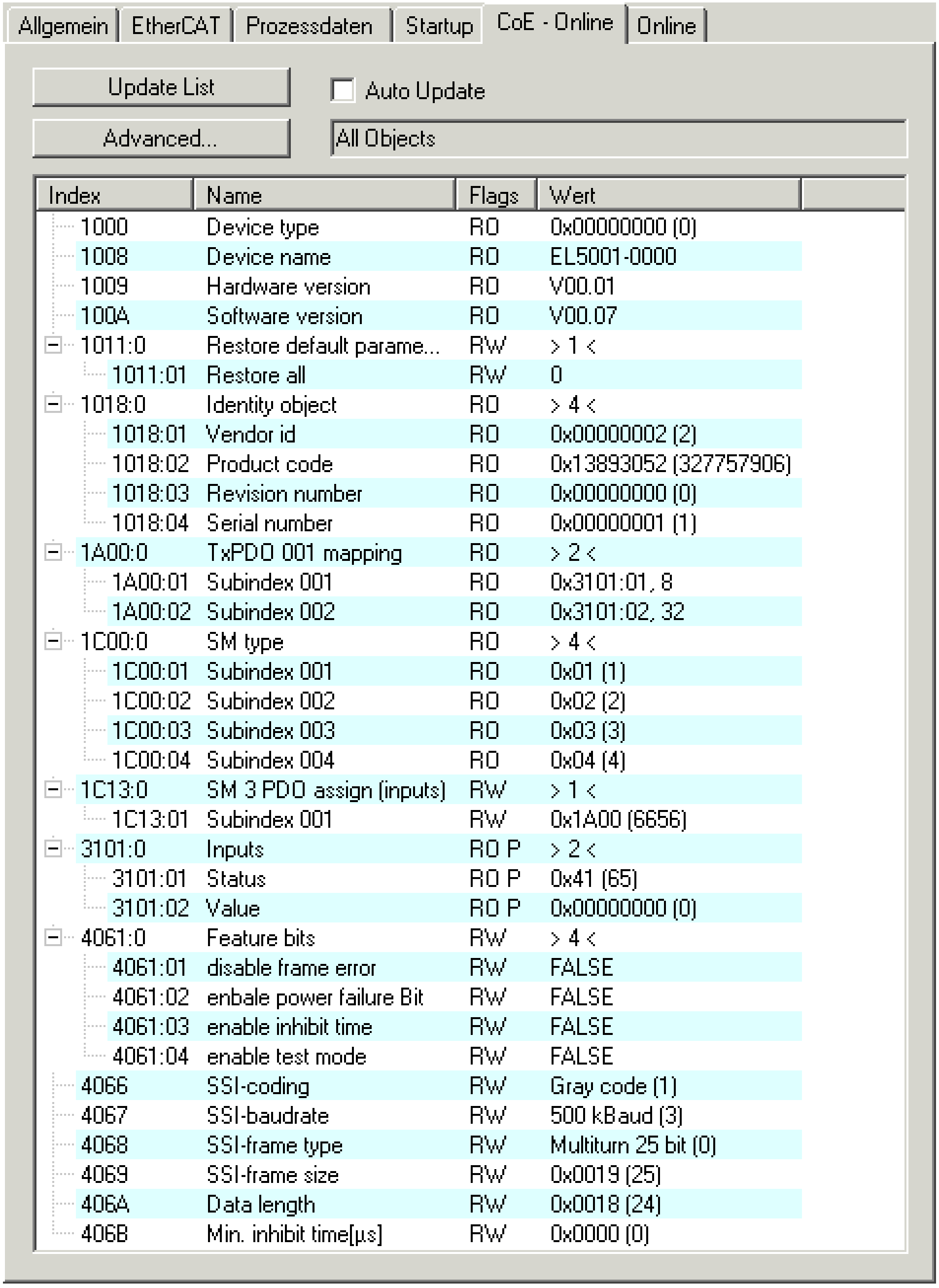
Fig. 145: “Startup” tab

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.



Index	Name	Flags	Wert
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.07
1011:0	Restore default parame...	RW	> 1 <
1011:01	Restore all	RW	0
1018:0	Identity object	RO	> 4 <
1018:01	Vendor id	RO	0x00000002 (2)
1018:02	Product code	RO	0x13893052 (327757906)
1018:03	Revision number	RO	0x00000000 (0)
1018:04	Serial number	RO	0x00000001 (1)
1A00:0	TxPDO 001 mapping	RO	> 2 <
1A00:01	Subindex 001	RO	0x3101:01, 8
1A00:02	Subindex 002	RO	0x3101:02, 32
1C00:0	SM type	RO	> 4 <
1C00:01	Subindex 001	RO	0x01 (1)
1C00:02	Subindex 002	RO	0x02 (2)
1C00:03	Subindex 003	RO	0x03 (3)
1C00:04	Subindex 004	RO	0x04 (4)
1C13:0	SM 3 PDO assign (inputs)	RW	> 1 <
1C13:01	Subindex 001	RW	0x1A00 (6656)
3101:0	Inputs	RO P	> 2 <
3101:01	Status	RO P	0x41 (65)
3101:02	Value	RO P	0x00000000 (0)
4061:0	Feature bits	RW	> 4 <
4061:01	disable frame error	RW	FALSE
4061:02	enbale power failure Bit	RW	FALSE
4061:03	enable inhibit time	RW	FALSE
4061:04	enable test mode	RW	FALSE
4066	SSI-coding	RW	Gray code (1)
4067	SSI-baudrate	RW	500 kBaud (3)
4068	SSI-frame type	RW	Multiturn 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[µs]	RW	0x0000 (0)

Fig. 146: “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.

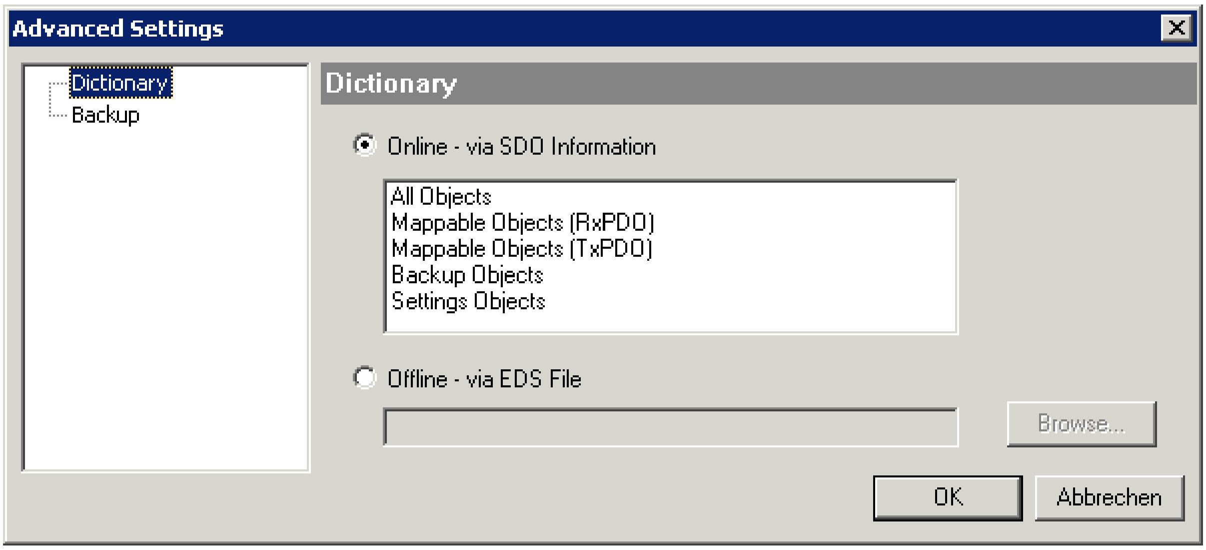


Fig. 147: Dialog “Advanced settings”

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

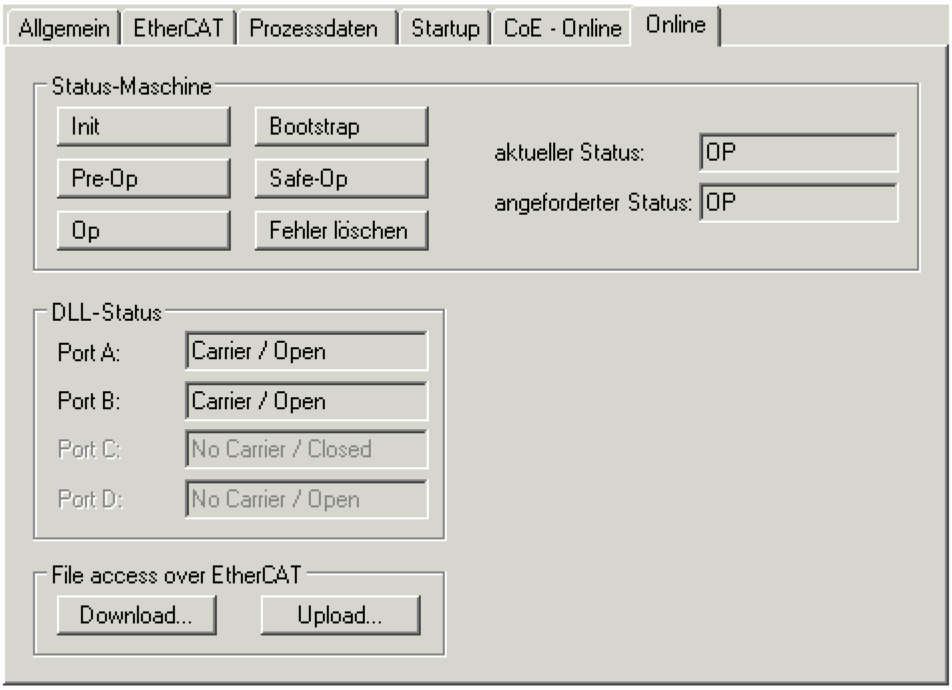


Fig. 148: “Online” tab

State Machine

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

DLL Status

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

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

File Access over EtherCAT

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

"DC" tab (Distributed Clocks)

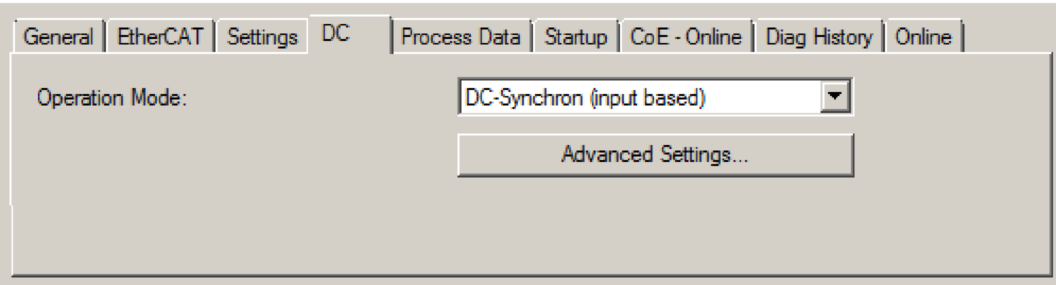


Fig. 149: "DC" tab (Distributed Clocks)

Operation Mode Options (optional):

- FreeRun
- SM-Synchron
- DC-Synchron (Input based)
- DC-Synchron

Advanced Settings… Advanced settings for readjustment of the real time determinant TwinCAT-clock

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

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

5.2.7.1 Download revision

Download revision in Start-up list

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

The object 0xF081:01 (Download revision) describes the revision of the terminal / module, e.g. 0x00**18**000*A* for EL7201-00*10*-00**24**, and is necessary to ensure compatibility.

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

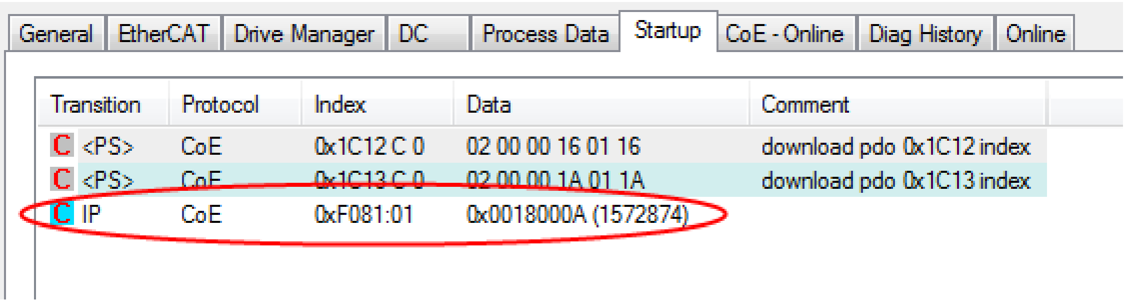


Fig. 150: Download revision in Start-up list

5.2.7.2 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).
If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).
SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

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

PDO Assignment

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

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

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

Activation of PDO assignment

✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▸ 122]),

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

(button for TwinCAT 2 or button for TwinCAT 3)

PDO list

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

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

PDO Content

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

Download

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

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 119] tab.

PDO Configuration

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

5.3 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.

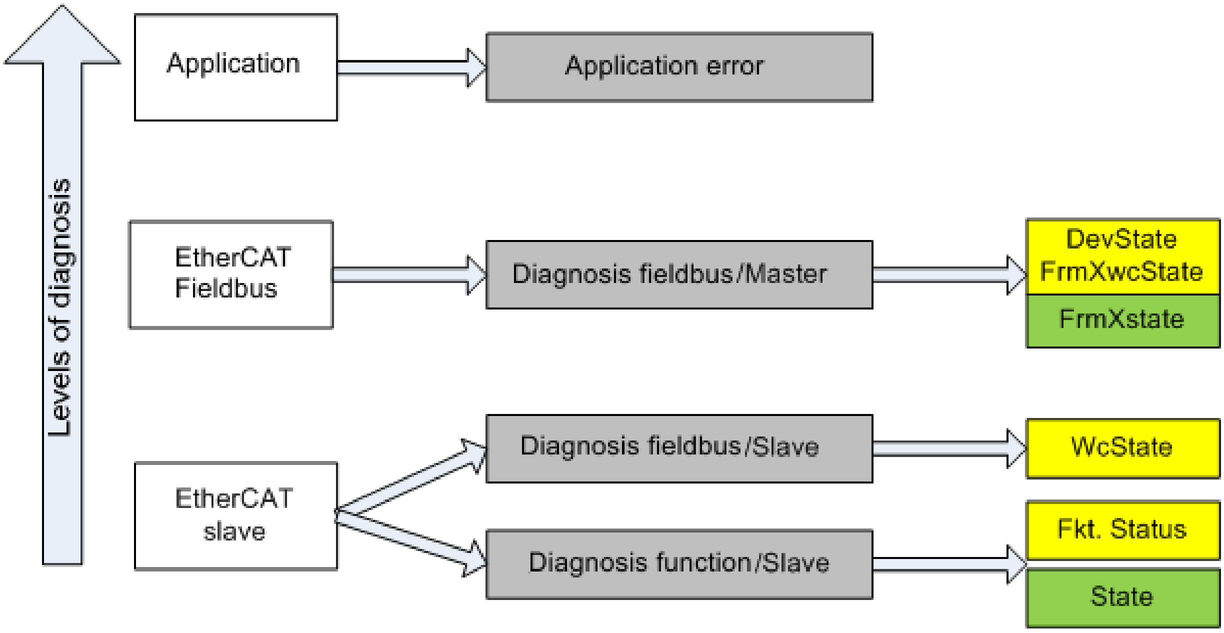


Fig. 151: Selection of the diagnostic information of an EtherCAT Slave

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.

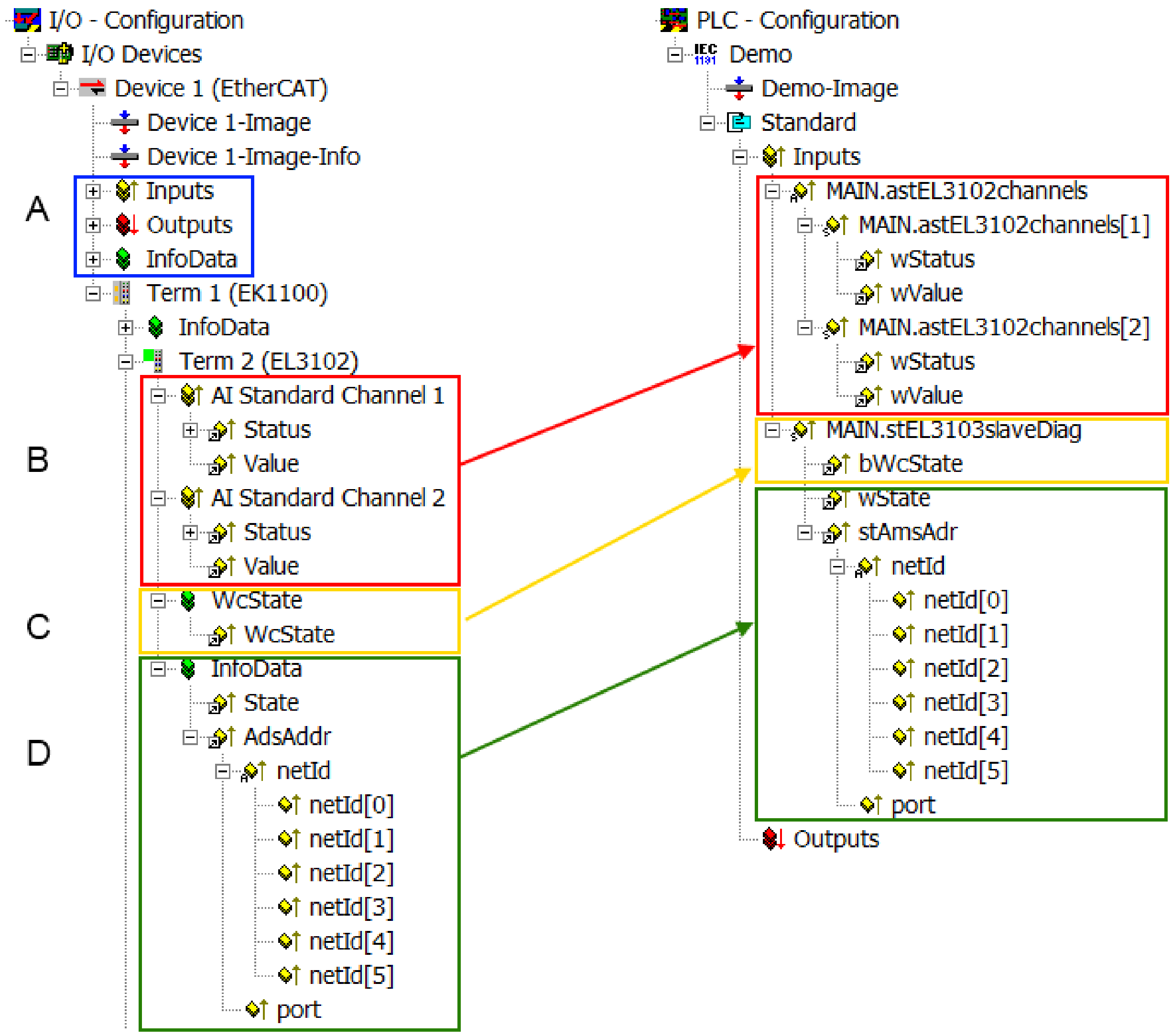


Fig. 152: Basic EtherCAT Slave Diagnosis in the PLC

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.
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.

NOTE

Diagnostic information

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

CoE Parameter Directory

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

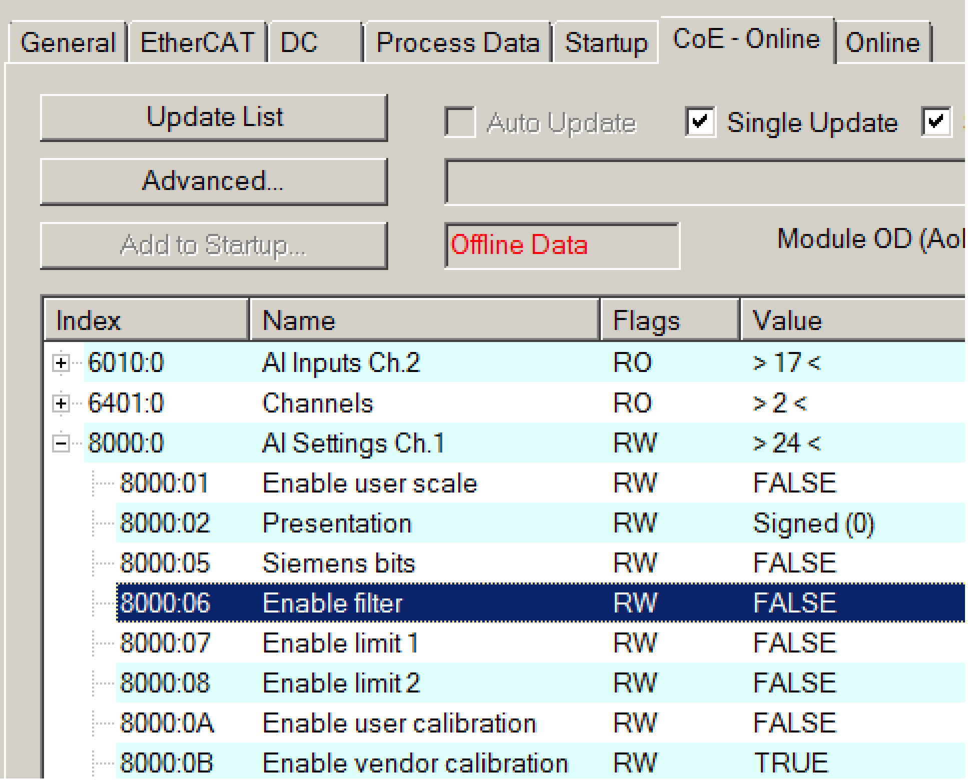


Fig. 153: EL3102, CoE directory

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.

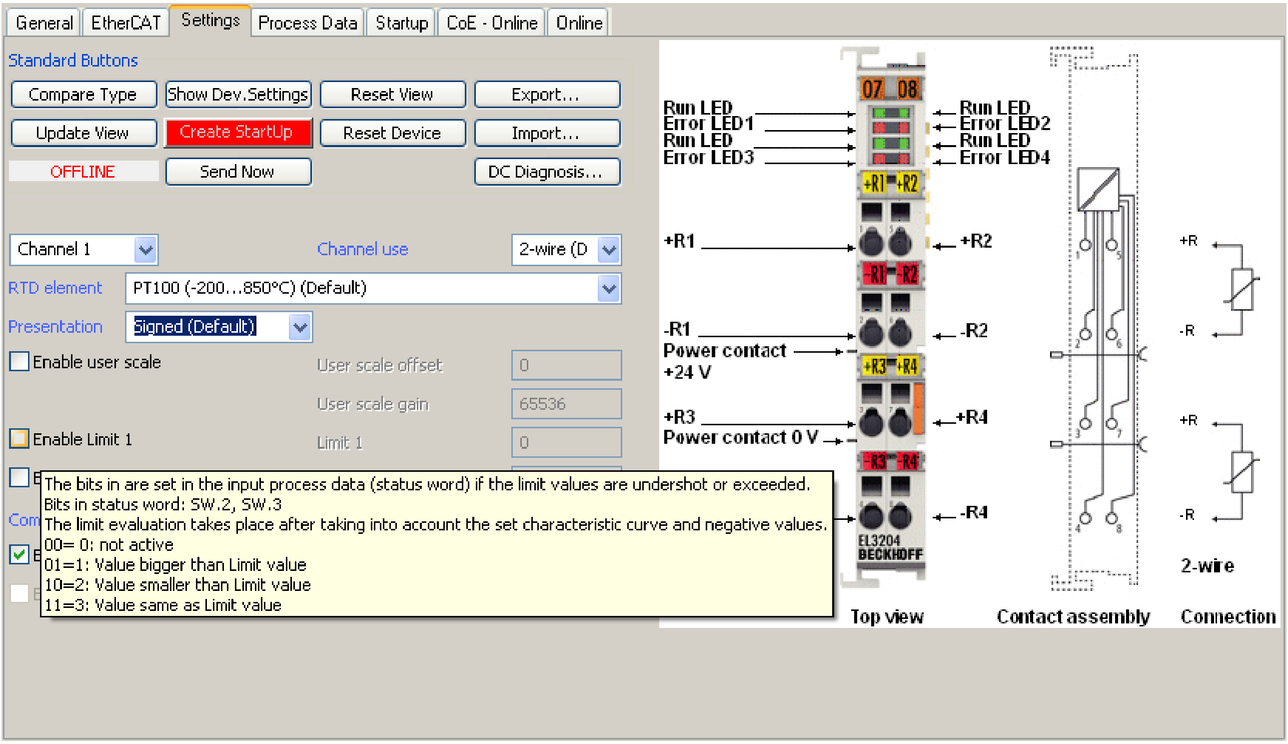


Fig. 154: Example of commissioning aid for a EL3204

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 [▶ 32]" 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.

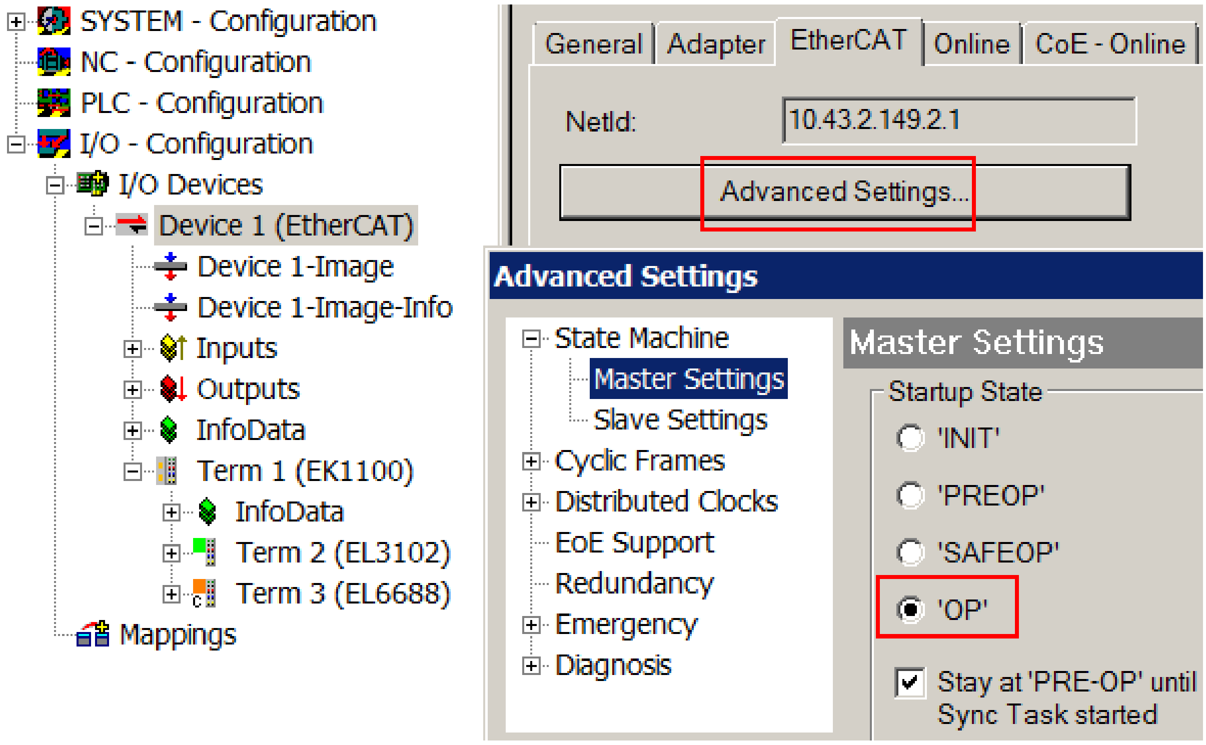


Fig. 155: Default behaviour of the System Manager

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

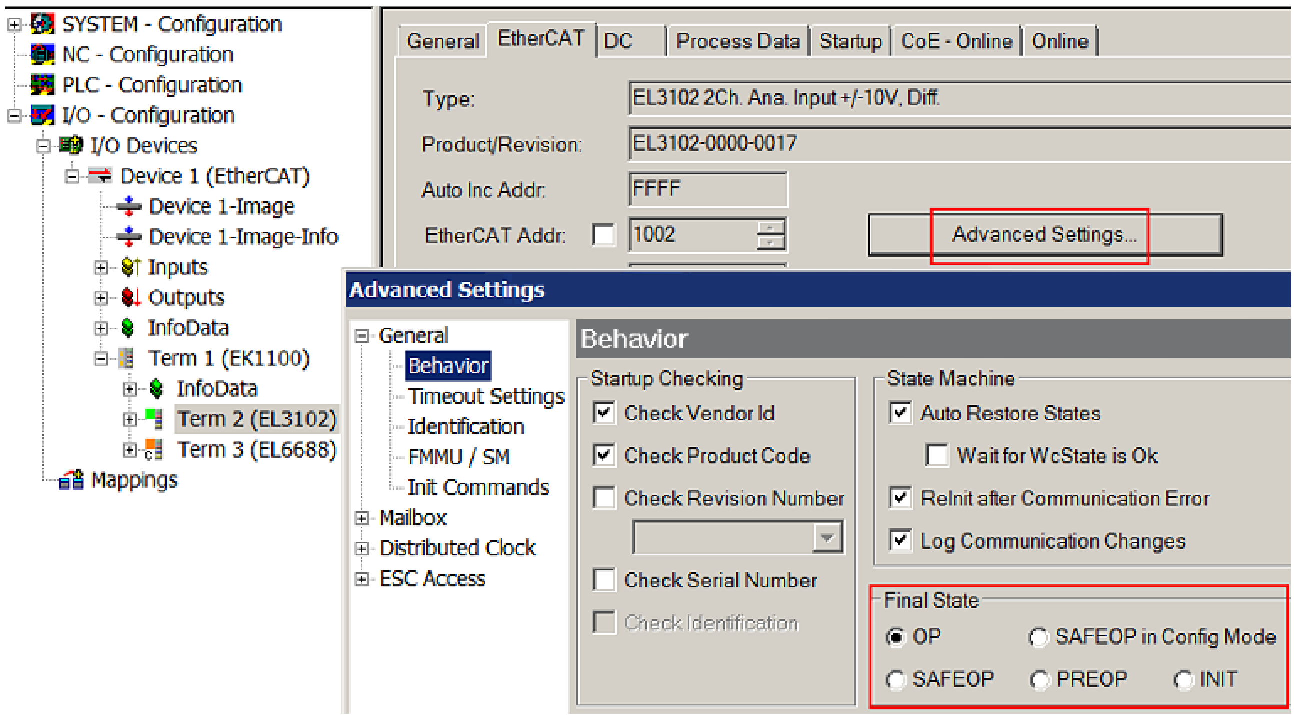


Fig. 156: Default target state in the Slave

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.

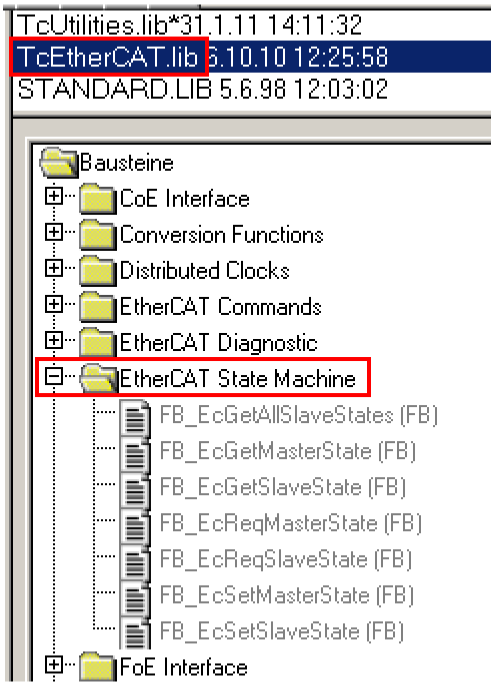
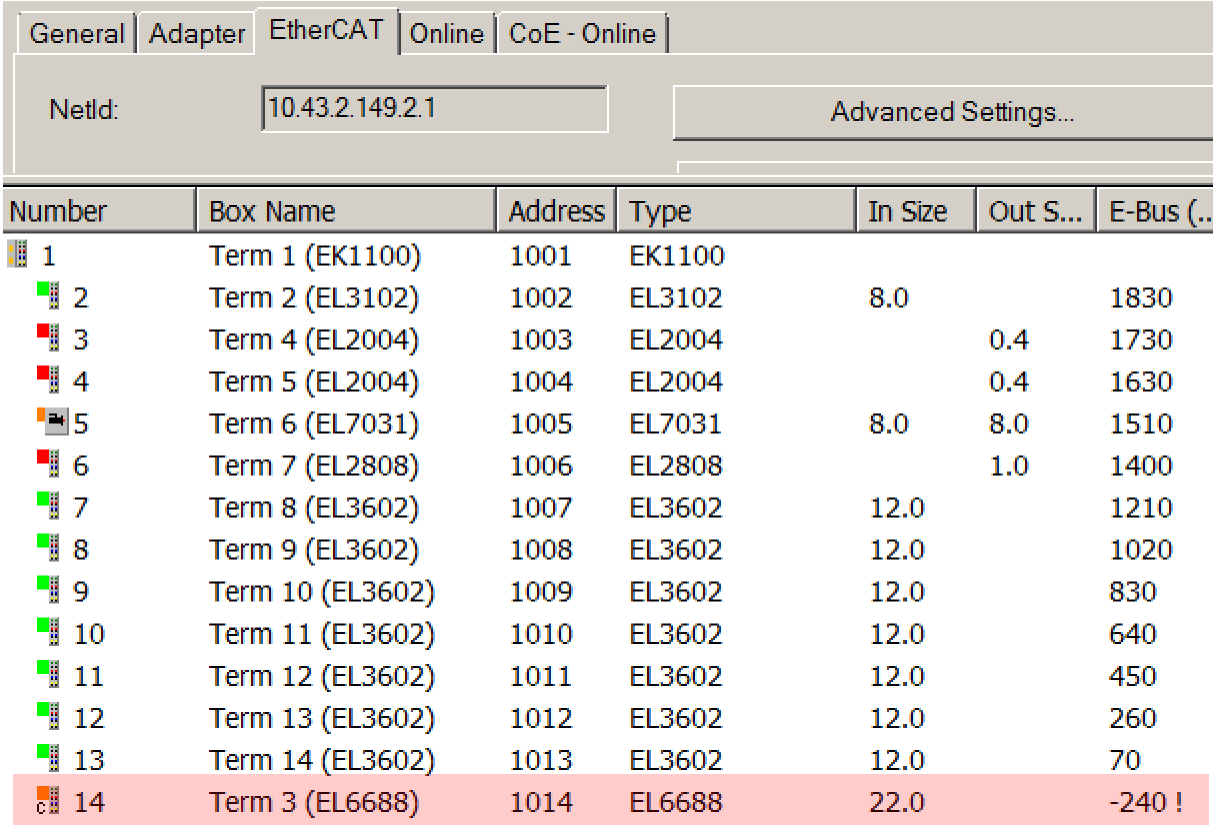


Fig. 157: PLC function blocks

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. 158: 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:

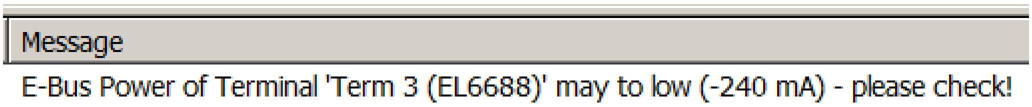


Fig. 159: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

5.4 Start-up and parameter configuration

5.4.1 Process data

Sync Manager (SM)

Sync Manager (SM) The scope of the offered process data can be changed via the "Process data" tab (see Fig. *"Tab Process data SM2, EL70xx (default), Process data tab SM3, EL70xx (default)"*).

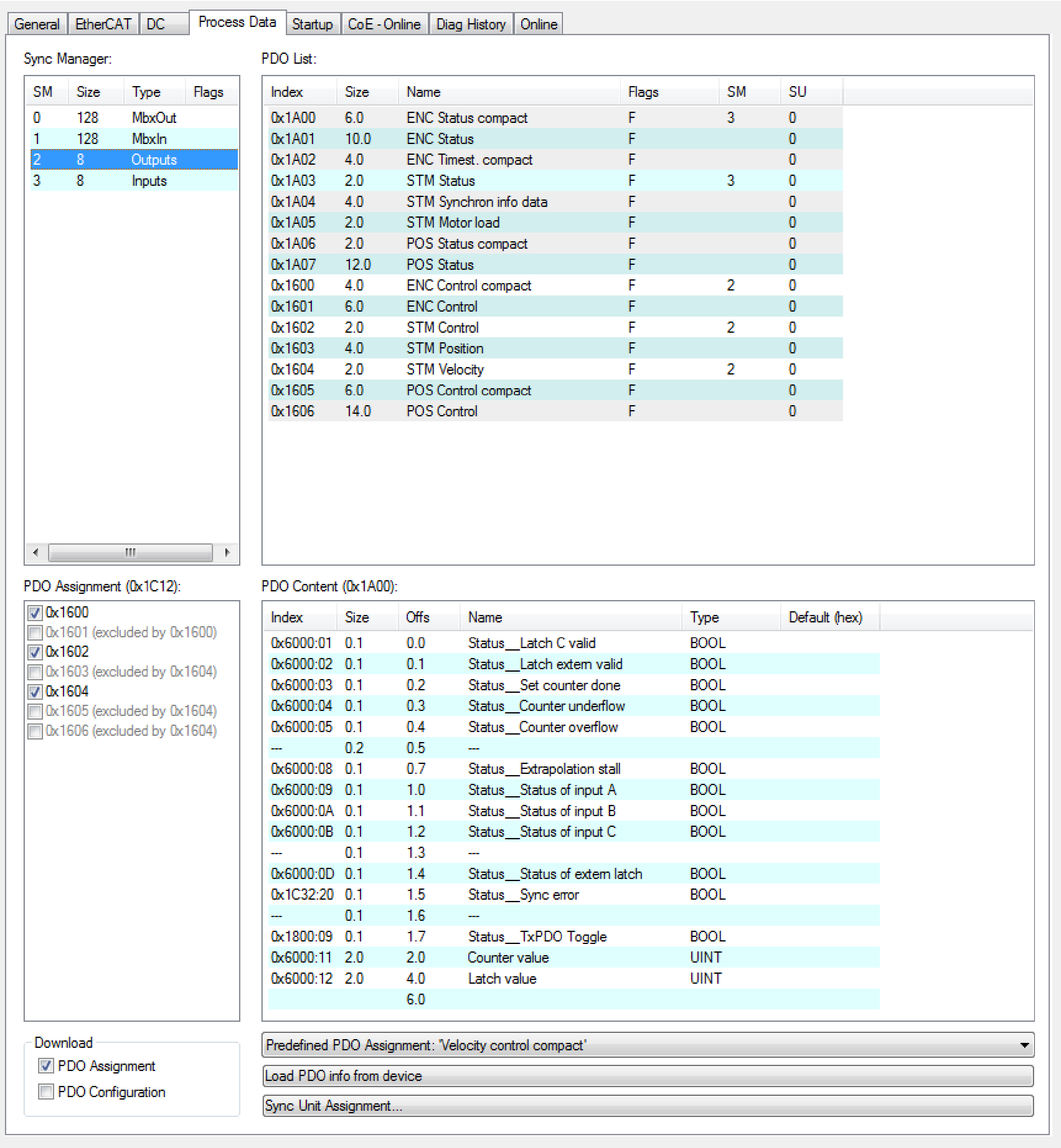


Fig. 160: *Process Data tab SM2, EL70xx (default)*

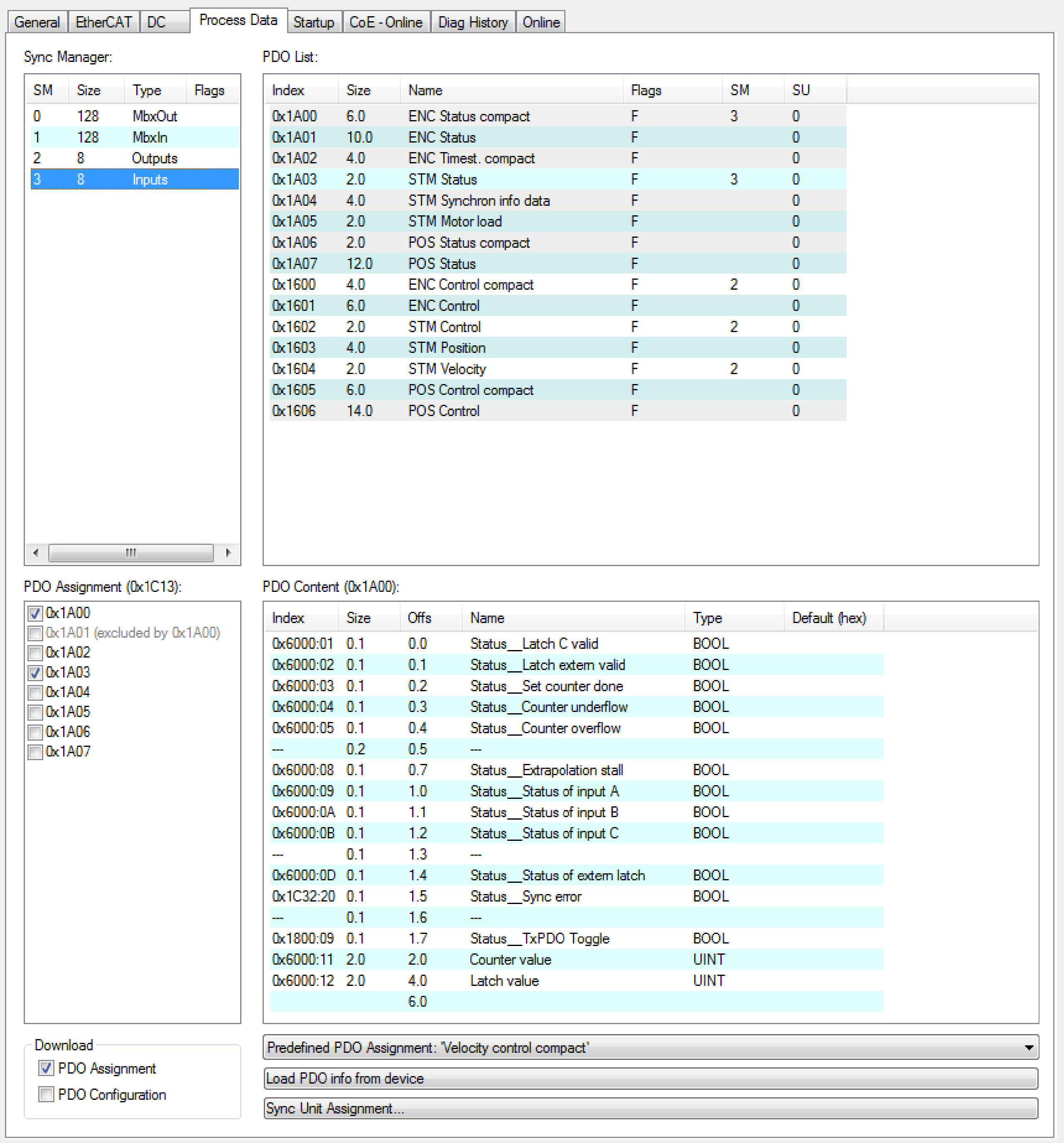


Fig. 161: *Process Data tab SM3, EL70xx (default)*

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	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO content**
0x1600 (default)	0x1601	4.0	ENC Control compact	Index 0x7000:01 [▶ 220] - Enable Latch C Index 0x7000:02 [▶ 220] - Enable Latch extern on positive edge Index 0x7000:03 [▶ 220] - Set counter Index 0x7000:04 [▶ 220] - Enable Latch extern on negative edge Index 0x7000:11 [▶ 220] - Set counter value (16-bit)
0x1601	0x1600	6.0	ENC Control	Index 0x7000:01 [▶ 220] - Enable Latch C Index 0x7000:02 [▶ 220] - Enable Latch extern on positive edge Index 0x7000:03 [▶ 220] - Set counter Index 0x7000:04 [▶ 220] - Enable Latch extern on negative edge Index 0x7000:11 [▶ 220] - Set counter value (32-bit)
0x1602 (default)	-	2.0	STM Control	Index 0x7010:01 [▶ 220] - Enable Index 0x7010:02 [▶ 220] - Reset Index 0x7010:03 [▶ 220] - Reduce torque Index 0x7010:0C [▶ 220] - Digital Output 1
0x1603	0x1604 0x1605 0x1606	4.0	STM Position	Index 0x7010:11 [▶ 220] - Position
0x1604 (default)	0x1603 0x1605 0x1606	2.0	STM Velocity	Index 0x7010:21 [▶ 220] - Velocity
0x1605	0x1603 0x1604 0x1606	6.0	POS Control compact	Index 0x7020:01 [▶ 221] - Execute Index 0x7020:02 [▶ 221] - Emergency stop Index 0x7020:11 [▶ 221] - Target position
0x1606	0x1603 0x1604 0x1605	14.0	POS Control	Index 0x7020:01 [▶ 221] - Execute Index 0x7020:02 [▶ 221] - Emergency stop Index 0x7020:11 [▶ 221] - Target position Index 0x7020:21 [▶ 221] - Velocity Index 0x7020:22 [▶ 221] - Start type Index 0x7020:23 [▶ 221] - Acceleration Index 0x7020:24 [▶ 221] - Deceleration
0x1607	0x1603 0x1604 0x1605	14.0	POS Control 2	Index 0x7021:03 [▶ 196] – Enable auto start Index 0x7021:03 [▶ 196] – Target position Index 0x7021:21 [▶ 196] – Velocity Index 0x7021:22 [▶ 196] – Start type Index 0x7021:23 [▶ 196] – Acceleration Index 0x7021:24 [▶ 196] – Deceleration

SM3, PDO Assignment 0x1C13				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO content**
0x1A00 (default)	0x1A01	6.0	ENC Status compact	Index 0x6000:01 [▸ 219] - Latch C valid Index 0x6000:02 [▸ 219] - Latch extern valid Index 0x6000:03 [▸ 219] - Set counter done Index 0x6000:04 [▸ 219] - Counter underflow Index 0x6000:05 [▸ 219] - Counter overflow Index 0x6000:08 [▸ 219] - Extrapolation stall Index 0x6000:09 [▸ 219] - Status of input A Index 0x6000:0A [▸ 219] - Status of input B Index 0x6000:0B [▸ 219] - Status of input C Index 0x6000:0D [▸ 219] - Status of extern latch Index 0x6000:0E [▸ 219] - Sync error Index 0x6000:10 [▸ 219] - TxPDO Toggle Index 0x6000:11 [▸ 219] - Counter value (16-Bit) Index 0x6000:12 [▸ 219] - Latch value (16-Bit)
0x1A01	0x1A00	10.0	ENC Status	Index 0x6000:01 [▸ 219] - Latch C valid Index 0x6000:02 [▸ 219] - Latch extern valid Index 0x6000:03 [▸ 219] - Set counter done Index 0x6000:04 [▸ 219] - Counter underflow Index 0x6000:05 [▸ 219] - Counter overflow Index 0x6000:08 [▸ 219] - Extrapolation stall Index 0x6000:09 [▸ 219] - Status of input A Index 0x6000:0A [▸ 219] - Status of input B Index 0x6000:0B [▸ 219] - Status of input C Index 0x6000:0D [▸ 219] - Status of extern latch Index 0x6000:0E [▸ 219] - Sync error Index 0x6000:10 [▸ 219] - TxPDO Toggle Index 0x6000:11 [▸ 219] - Counter value (32-Bit) Index 0x6000:12 [▸ 219] - Latch value (32-Bit)
0x1A02	-	4.0	ENC Timest. compact	Index 0x6000:16 [▸ 219] - Timestamp
0x1A03 (default)	-	2.0	STM Status	Index 0x6010:01 [▸ 219] - Ready to enable Index 0x6010:02 [▸ 219] - Ready Index 0x6010:03 [▸ 219] - Warning Index 0x6010:04 [▸ 219] - Error Index 0x6010:05 [▸ 219] - Moving positive Index 0x6010:06 [▸ 219] - Moving negative Index 0x6010:07 [▸ 219] - Torque reduced Index 0x6010:08 [▸ 219] - Motor stall Index 0x6010:0C [▸ 219] - Digital input 1 Index 0x6010:0D [▸ 219] - Digital input 2 Index 0x6000:0E [▸ 219] - Sync error Index 0x6000:10 [▸ 219] - TxPDO Toggle
0x1A04	-	4.0	STM Synchron info data	Index 0x6010:11 [▸ 219] - Info data 1 Index 0x6010:12 [▸ 219] - Info data 2
0x1A05	-	2.0	STM Motor load	Index 0x6010:13 [▸ 219] - Motor load

SM3, PDO Assignment 0x1C13				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO content**
0x1A06	0x1A07	2.0	POS Status compact	Index 0x6020:01 [▸ 220] - Busy Index 0x6020:02 [▸ 220] - in-Target Index 0x6020:03 [▸ 220] - Warning Index 0x6020:04 [▸ 220] - Error Index 0x6020:05 [▸ 220] - Calibrated Index 0x6020:06 [▸ 220] - Accelerate Index 0x6020:07 [▸ 220] - Decelerate
0x1A07	0x1A06	12.0	POS Status	Index 0x6020:01 [▸ 220] - Busy Index 0x6020:02 [▸ 220] - in-Target Index 0x6020:03 [▸ 220] - Warning Index 0x6020:04 [▸ 220] - Error Index 0x6020:05 [▸ 220] - Calibrated Index 0x6020:06 [▸ 220] - Accelerate Index 0x6020:07 [▸ 220] - Decelerate Index 0x6020:11 [▸ 220] - Actual position Index 0x6020:21 [▸ 220] - Actual velocity Index 0x6020:22 [▸ 220] - Actual drive time
0x1A08	-	4.0	STM Internal position	Index 0x6010:14 [▸ 219] - Internal position
0x1A09	-	4.0	STM External position	Index 0x6010:15 [▸ 219] – External position
0x1A0A	-	4.0	POS Actual position lag	Index 0x6020:23 [▸ 220] – Actual position lag

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.

The following PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Velocity control compact	0x1600 0x1602 0x1604	0x1A00 0x1A03
Velocity control compact with info data	0x1600 0x1602 0x1604	0x1A00 0x1A03 0x1A04
Velocity control	0x1601 0x1602 0x1604	0x1A01 0x1A03
Position control	0x1601 0x1602 0x1603	0x1A01 0x1A03
Positioning interface compact	0x1601 0x1602 0x1605	0x1A01 0x1A03 0x1A06
Positioning interface	0x1601 0x1602 0x1606	0x1A01 0x1A03 0x1A07
Positioning interface with info data	0x1601 0x1602 0x1606	0x1A01 0x1A03 0x1A04 0x1A07
Positioning interface (Auto start)	0x1601 0x1602 0x1606 0x1607	0x1A01 0x1A03 0x1A06
Positioning interface (Auto start) with info data	0x1601 0x1602 0x1606 0x1607	0x1A01 0x1A03 0x1A04 0x1A06

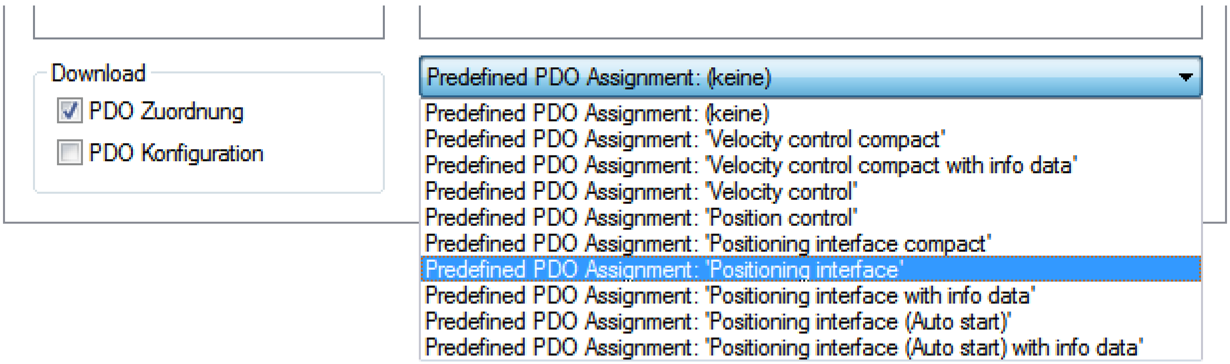


Fig. 162: *Process data tab - Predefined PDO Assignment, EL70x7*

5.4.2 Integration into the NC configuration

(Master: TwinCAT 2.11 R3)

Installation of the latest 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.

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 [▶ 93]").

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.

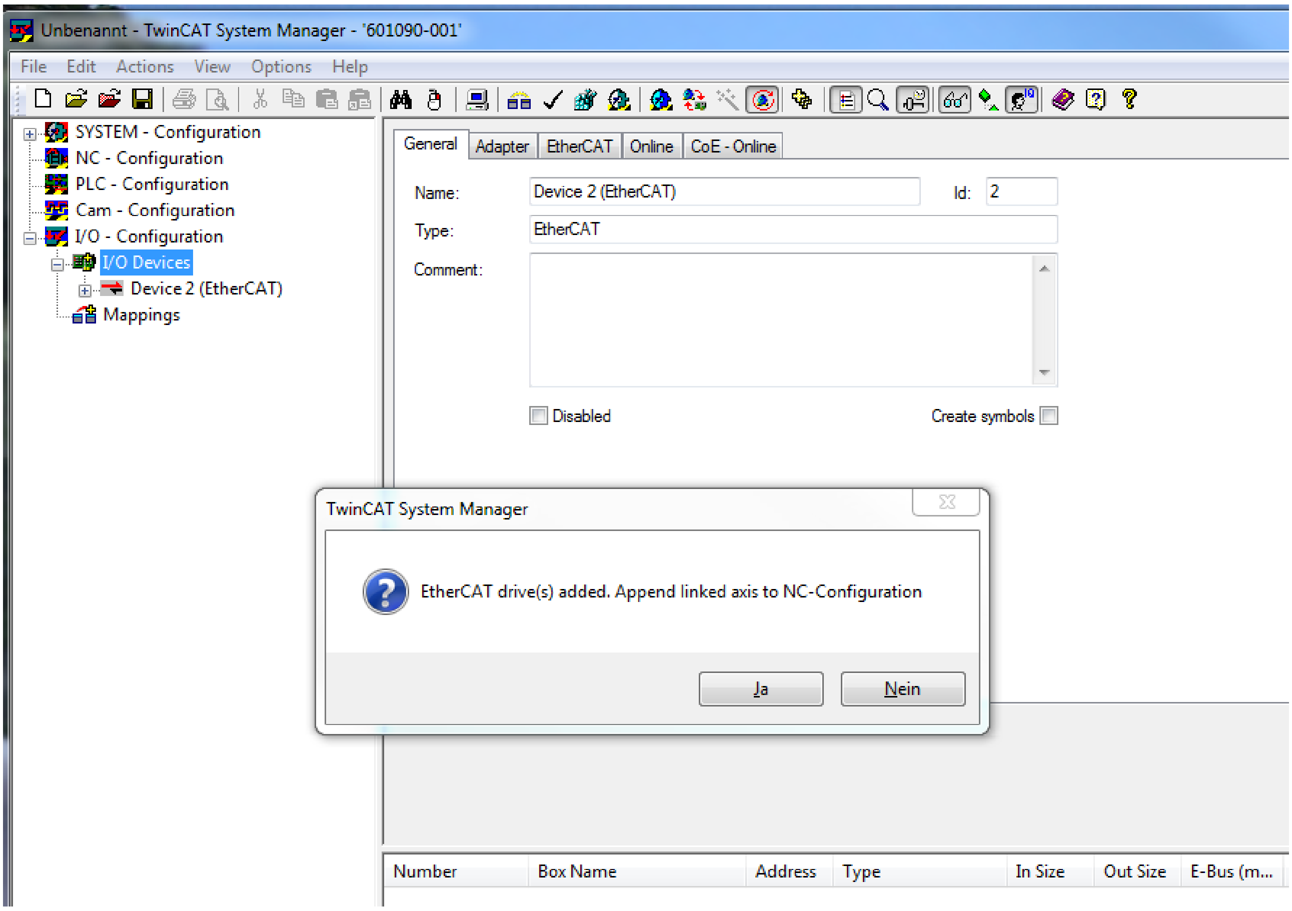


Fig. 163: *Axis detected*

- 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 [▶ 144]".
  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.

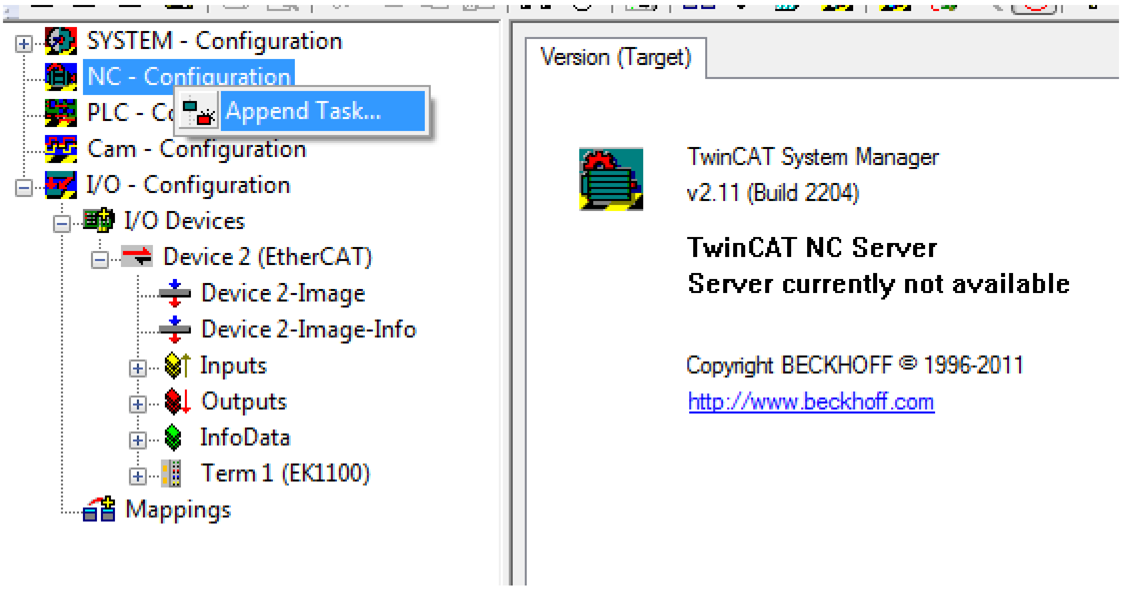


Fig. 164: *Adding a new task*

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

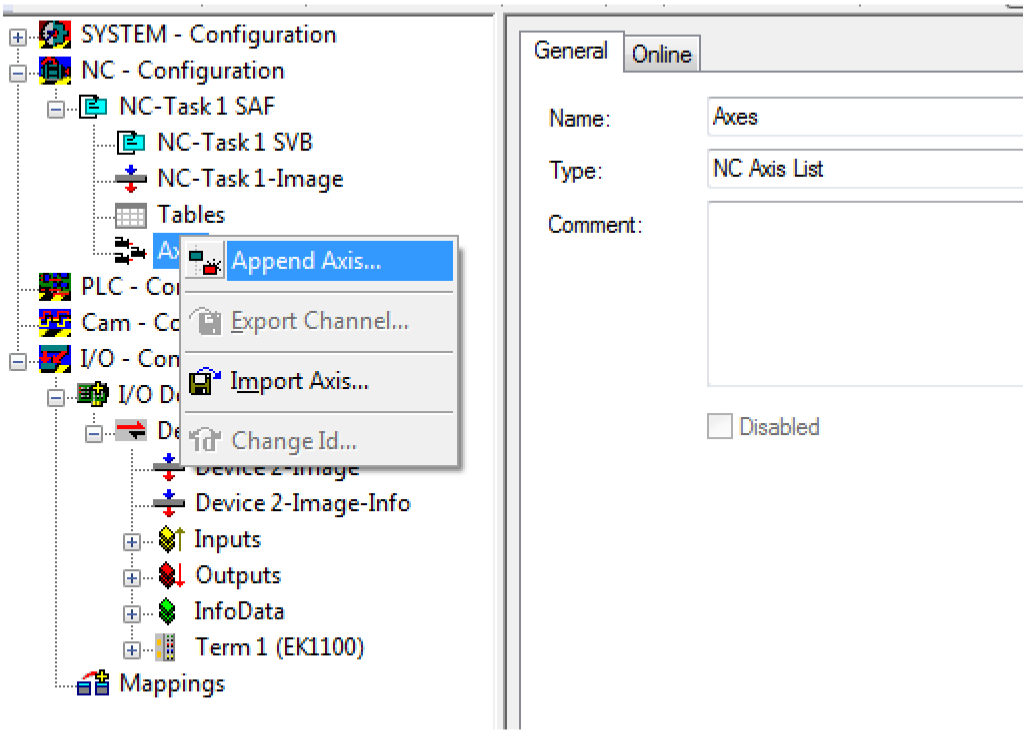


Fig. 165: *Adding a new axis*

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

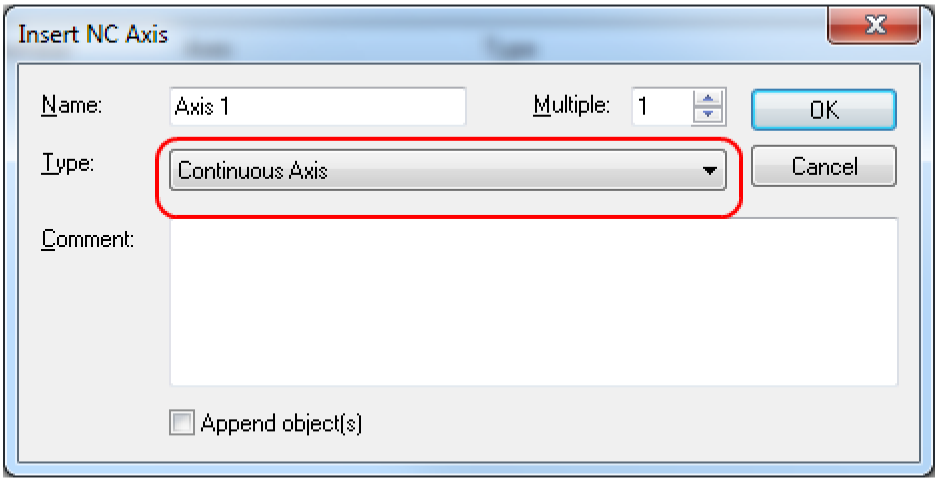


Fig. 166: *Selecting and confirming the axis type*

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

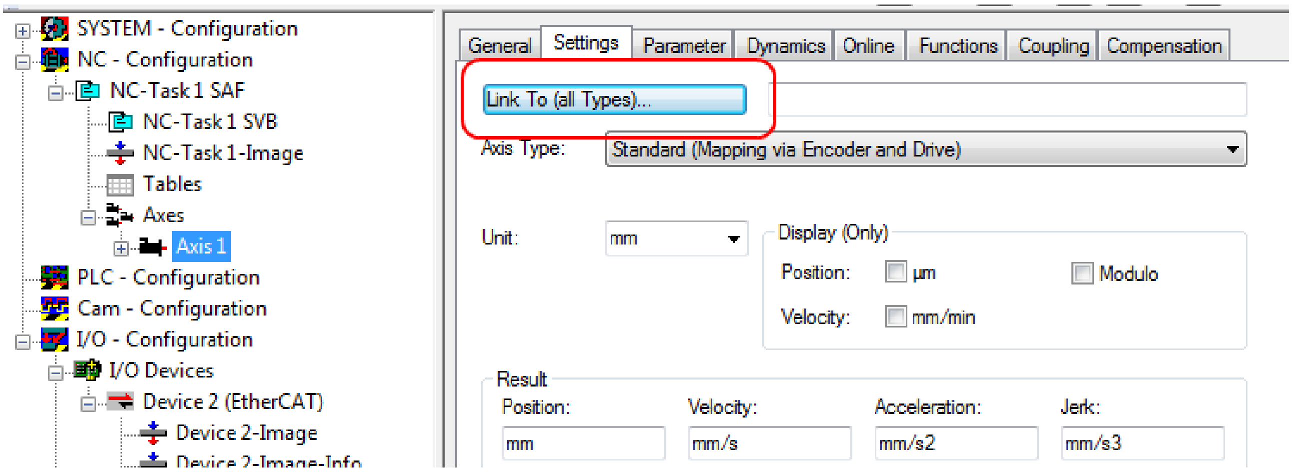


Fig. 167: *Linking the axis with the terminal*

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

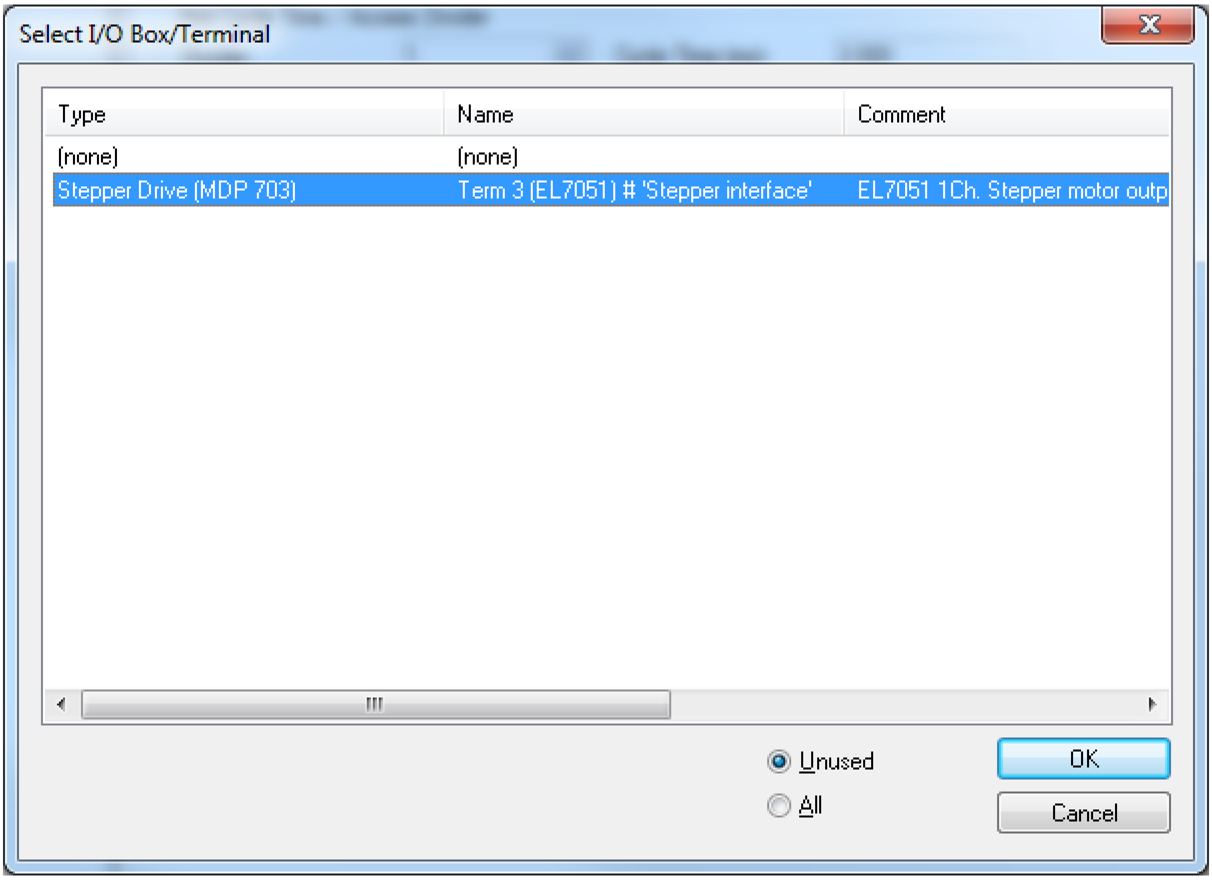


Fig. 168: *Selecting the right terminal*

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

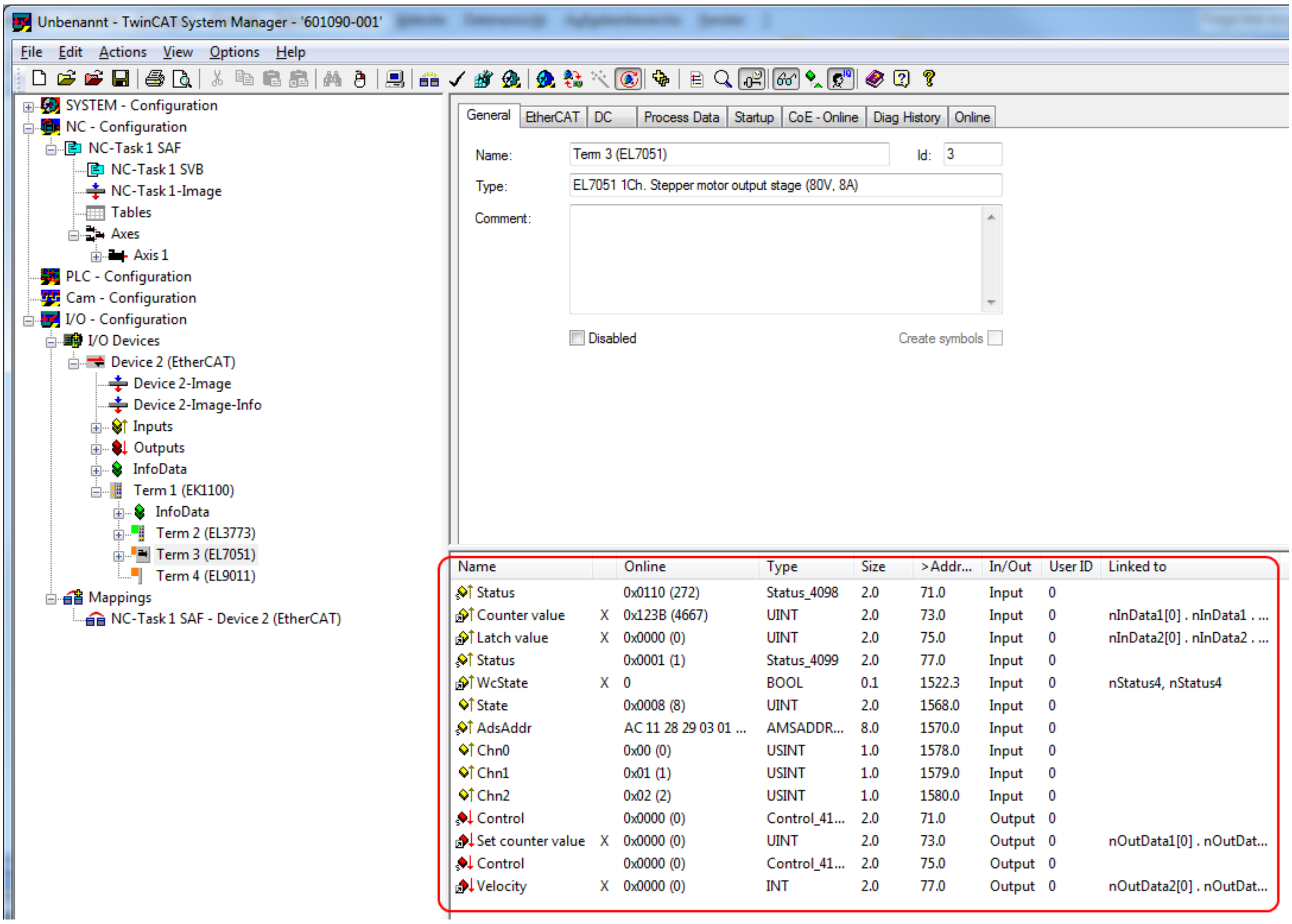


Fig. 169: *Automatic linking of all main variables*

- Several parameters have to be set before the motor can be started up. The values can be found in sections "CoE settings [▶ 144]" and "NC settings".
  Please set these parameters before continuing with the motor commissioning procedure.

5.4.3 Configuring the main parameters - Settings in the CoE register

The specified data apply to an AS 1050-0120 stepper motor and are intended as an example. For other motors the values may vary, depending on the application.

Adaptation of current and voltage

NOTE

The motor may overheat!

In order to prevent overheating of the connected motor it is important to adapt the current and voltage output from the stepper interface to the motor.

To this end set the index 8010:01 [▶ 214] "Maximum current" and 8010:03 [▶ 214] "Nominal voltage" in the CoE register to suitable values (see Fig. "*Adaptation of current and voltage*").

Reduced current can be set in index 8010:02 [▶ 214]. This reduces the coil current when at a standstill (and therefore the power dissipation). Please note that the torque is also reduced.

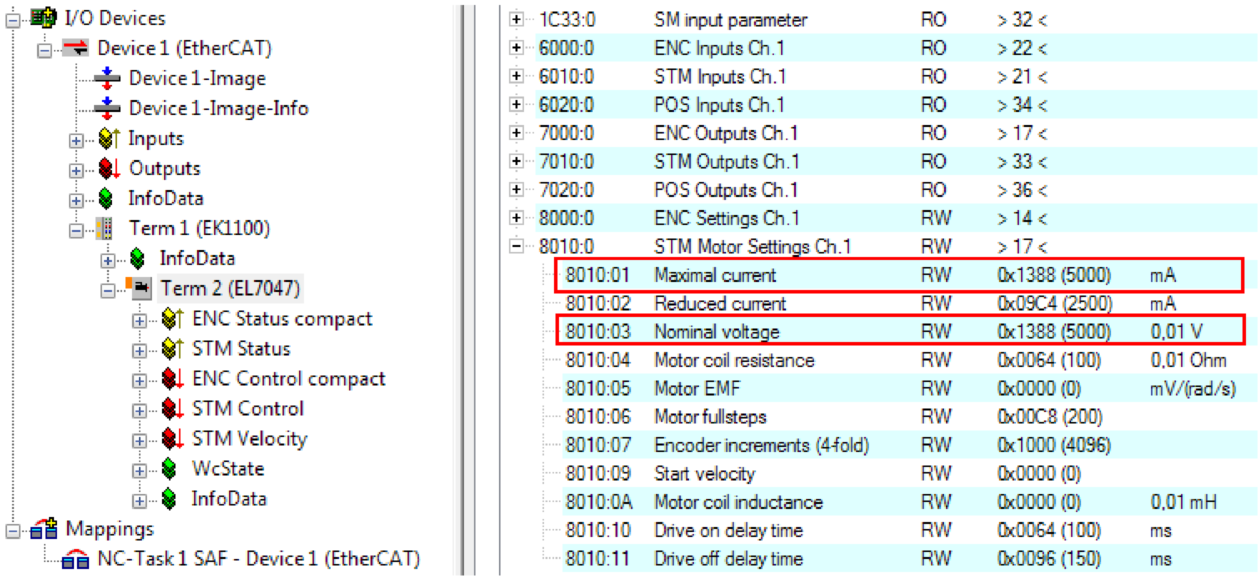


Fig. 170: *Adaptation of current and voltage*

Base frequency selection

Microstepping is set to 1/64 and cannot be changed. However, the base frequency can be changed (default: 2000). To this end select the terminal and select the *CoE Online* tab. Change the base frequency by double-clicking on the index 8012:05 [▸ 215] "Speed range" (Fig. "*Setting the base frequency*").

> **Adjusting the reference velocity**
>
> The base frequency is directly linked to the reference speed of the TwinCAT NC, so that the reference speed [▸ 139] always has to be adapted when the base frequency is changed.

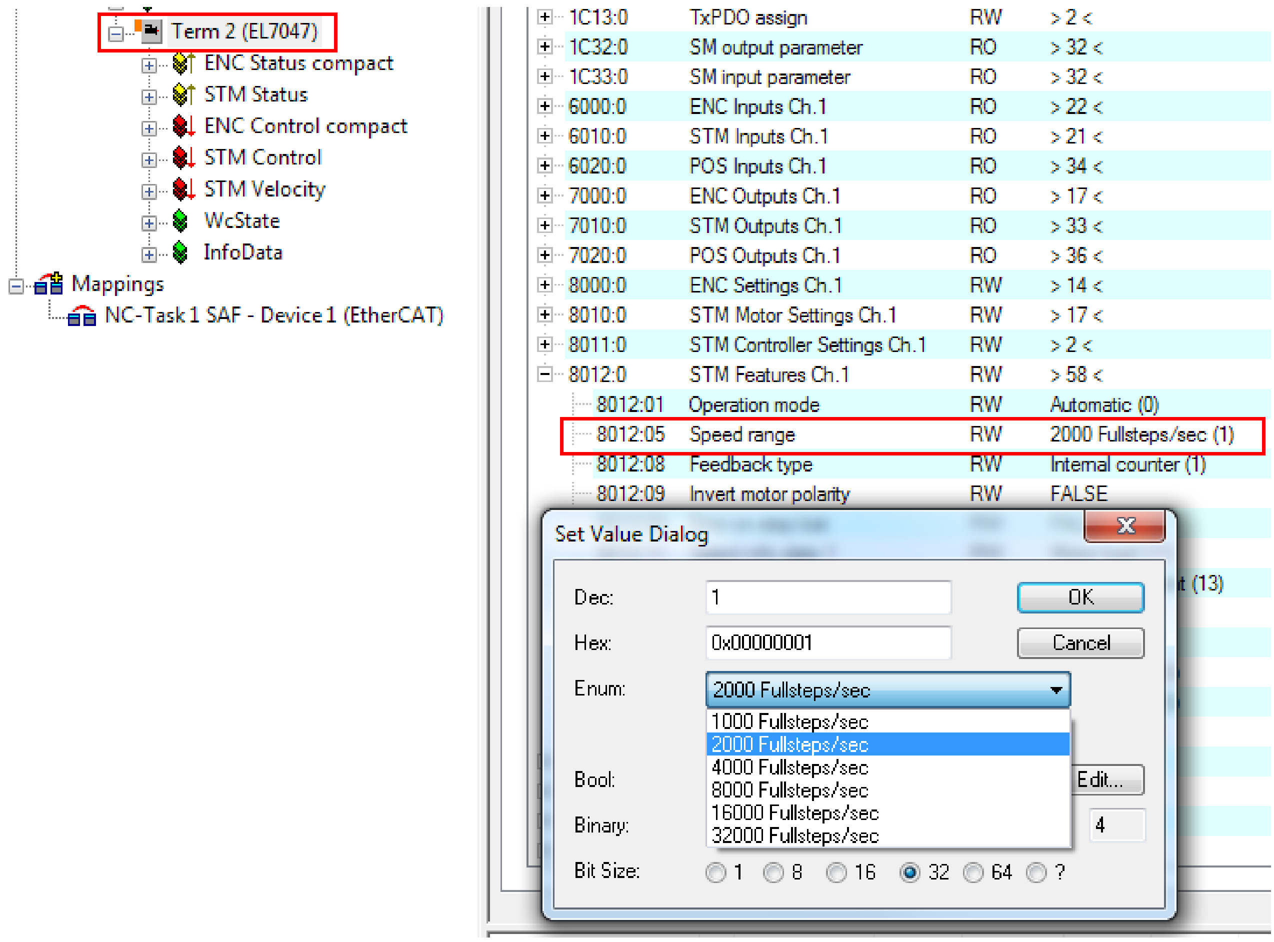


Fig. 171: *Setting the base frequency*

Selecting the feedback system (only for the module with encoder connections)

Two feedback system options are available for selection:

- Encoder: Use external encoder for position feedback
- Internal Counter **(default)**: Use internal counter for position feedback

CoE "Feedback type"

By default, the stepper module is set to internal counter. If an external encoder is used, the setting must be changed by double-clicking on the index 8012:08 [▶ 215] "Feedback type" in the Enum menu (Fig. "Selecting the feedback system").

Adaptation of the scaling factor

The feedback system is directly related to the scaling factor [▶ 139] of the TwinCAT NC, so that the scaling factor always has to be adjusted when the feedback system is changed.

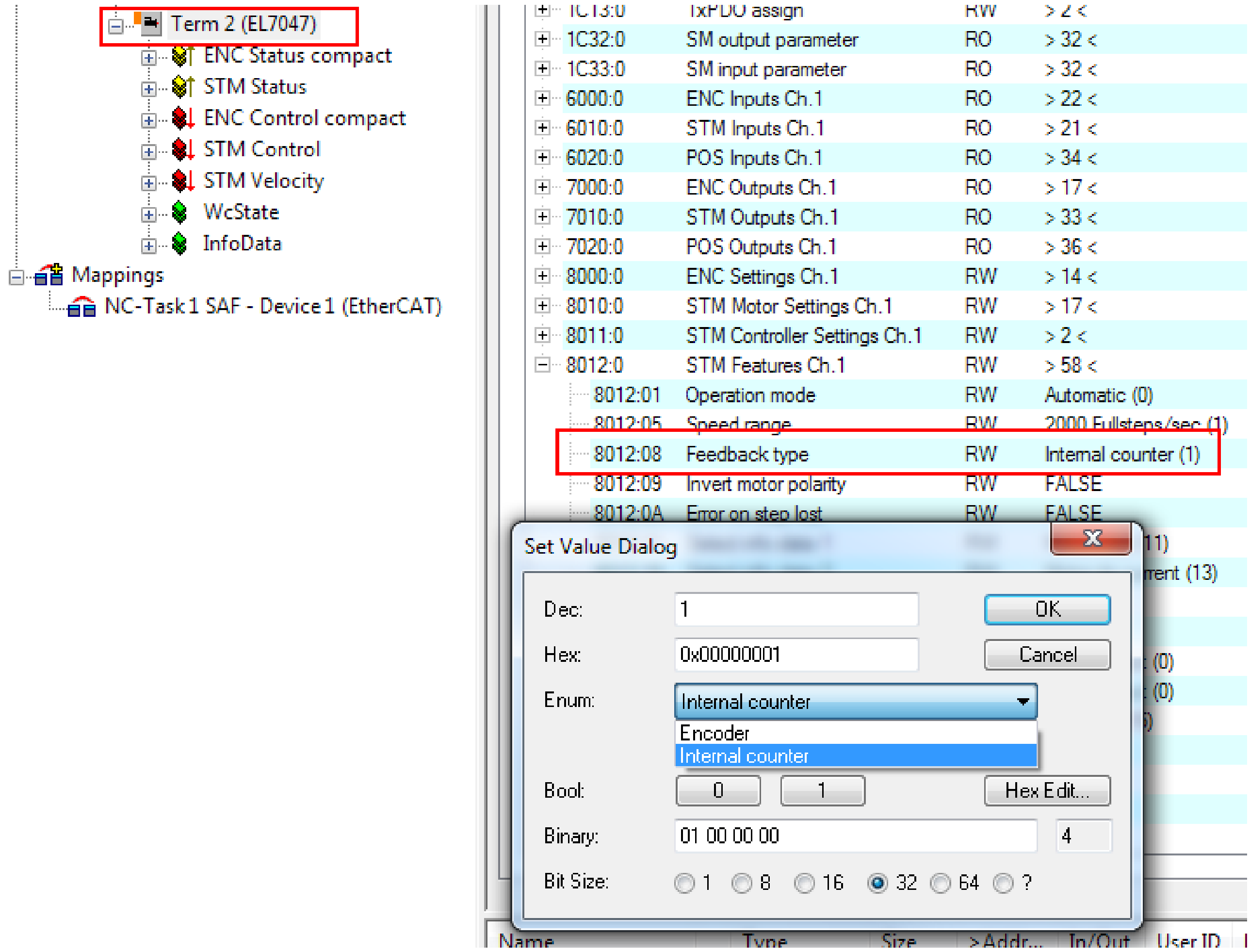


Fig. 172: *Selecting the feedback system*

5.4.4 Configuring the main parameter - Selecting the reference velocity

The specified data apply to an AS 1050-0120 stepper motor and are intended as an example. For other motors the values may vary, depending on the application.

The maximum velocity can be calculated from the base frequency and the motor frequency.

v_{max}= *base frequency / motor frequency = (2000 full steps / s) / (200 full steps / rev) = 10 revolutions / s*

The reference velocity can be calculated by multiplying the maximum velocity with the distance per revolution.

v_{ref} = *10 revolutions / s x 360° = 3600 °/ s*

Adjusting the reference velocity

The base frequency is directly linked to the reference speed of the TwinCAT NC, so that the reference speed always has to be adapted when the base frequency is changed.

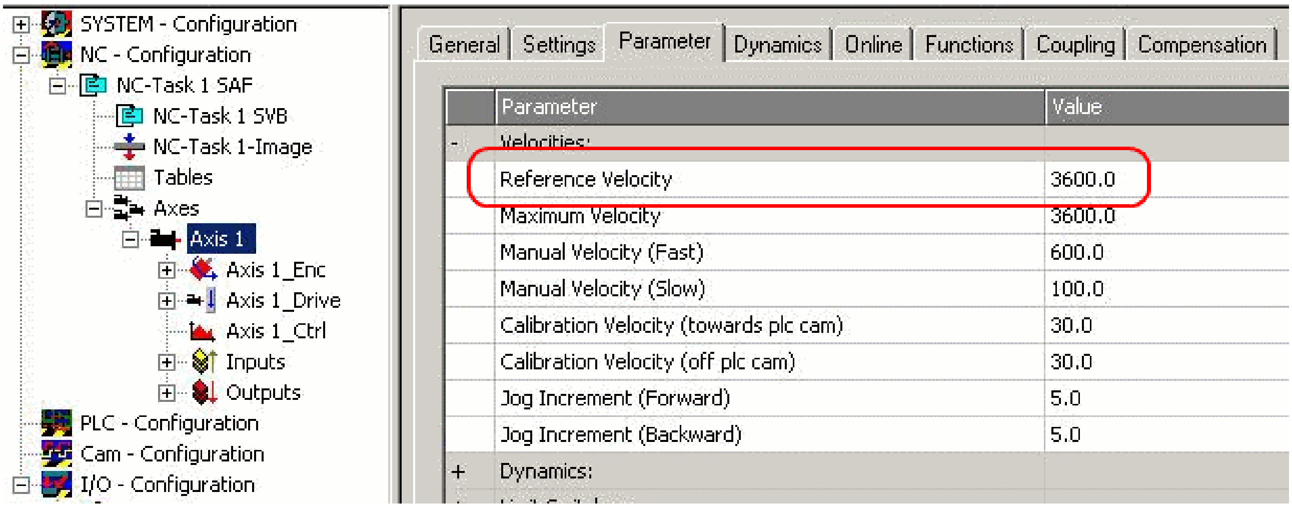


Fig. 173: Ex70x1_KONFIG_ref_velo

Reference velocity parameter

Dead time compensation

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

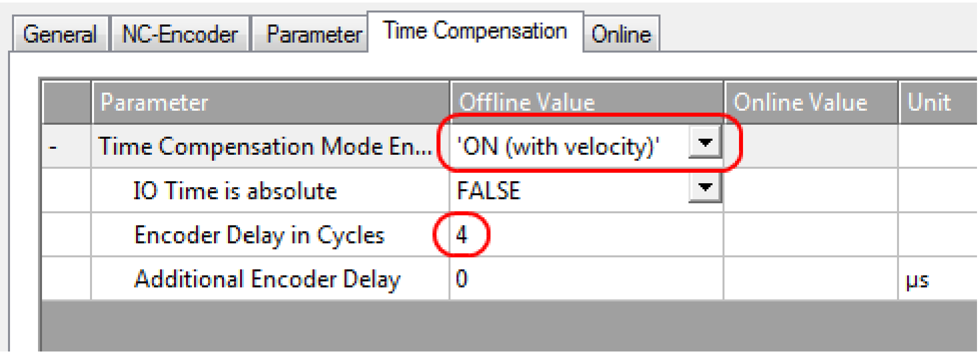


Fig. 174: Dead time compensation parameter

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.

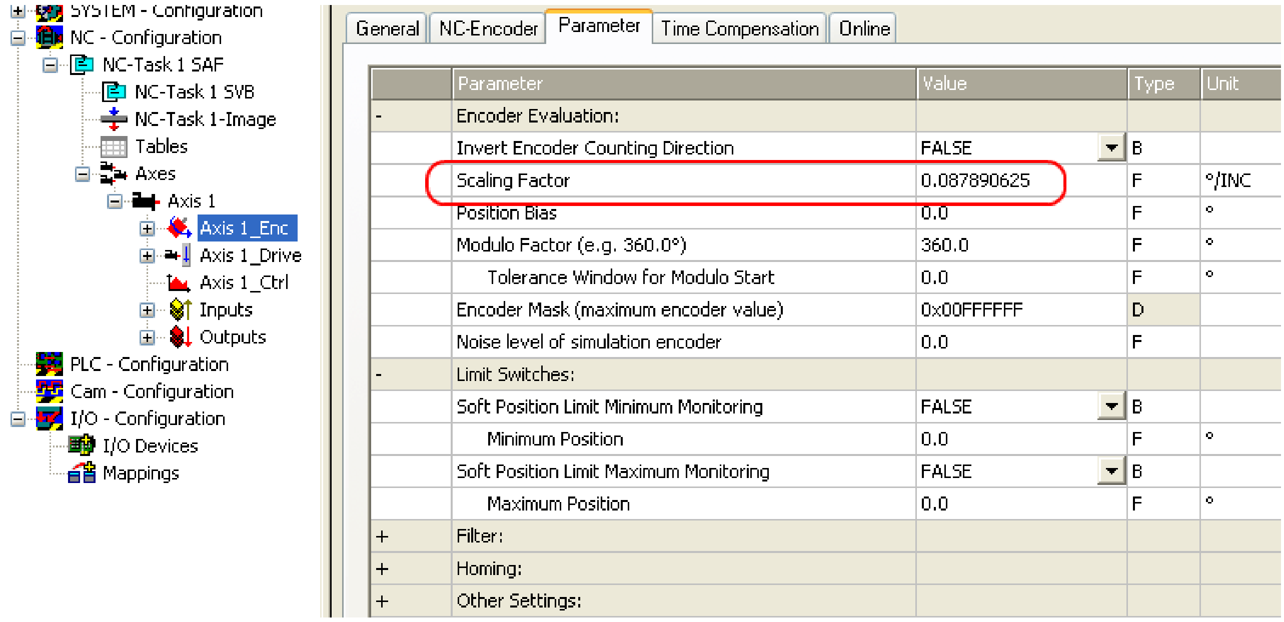


Fig. 175: *Setting the Scaling Factor*

Adaptation of the scaling factor

The feedback system is directly related to the scaling factor of the TwinCAT NC, so that the scaling factor always has to be adjusted when the feedback system [▶ 144] is changed.

Calculation of the scaling factor

with encoder, 4-fold evaluation:

SF = distance per revolution / (increments x 4) = 360° / (1024 x 4) = 0.087890625 ° / INC

without encoder:

SF = distance per revolution / (full steps x microsteps) = 360° / (200 x 64) = 0.028125 ° / INC

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.

NOTE

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.

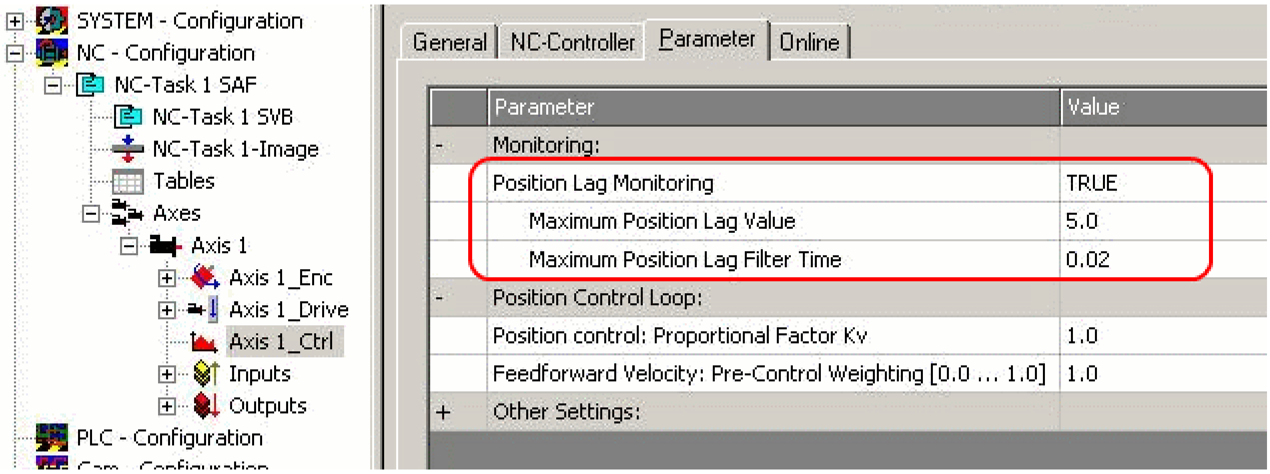


Fig. 176: *Position lag monitoring parameters*

K_v factors

In the NC two proportional factors K_v can be set under "Axis 1_Ctrl " in tab "Parameter". First select the position controller *Type* with two P constants (with K_a) under the “NC Controller” tab. The two P constants are for the *Standstill* range and for the *Moving* range (see Fig. *"Setting the proportional factor K_v"*). The factors can be used to set the start-up torque and the braking torque to a different value than the drive torque. The threshold value can be set directly below (Position control: Velocity threshold V dyn) between 0.0 (0%) and 1.0 (100%). Fig. *"Velocity ramp with K factor limit values"* shows speed ramp with thresholds of 30%. The K_v factor for Standstill (t_1 and t_3) can be different than the Kv factor for Moving (t_2). In this case the same factor was used, since for stepper motors this function is less crucial than for DC motors.

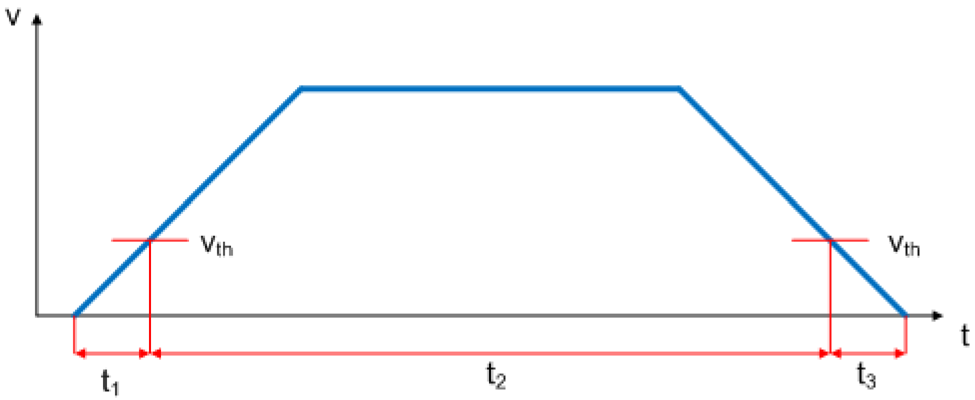


Fig. 177: *Speed ramp with K factor limit values*

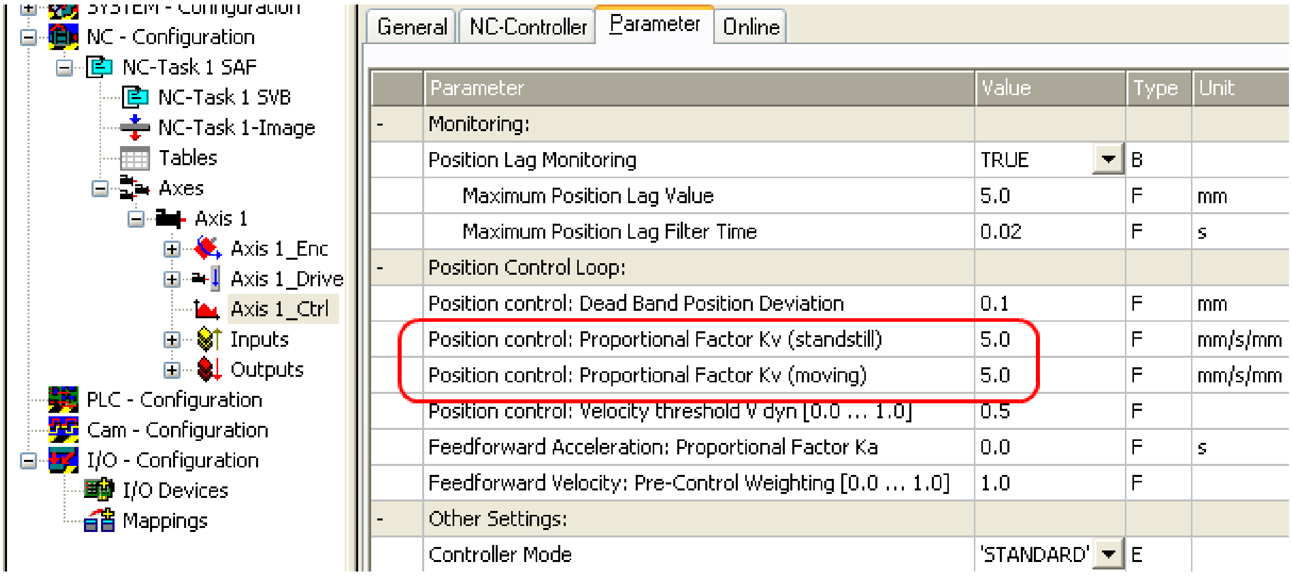


Fig. 178: *Setting the proportional factor* K_v

Dead band for position errors

Microstepping can be used to target 200 * 64 = 12800 positions. Since the encoder can only scan 1024 * 4 = 4096 positions, positions between two encoder scan points may not be picked up correctly, in which case the terminal will control around this position The dead band for position errors is a tolerance range within which the position is regarded as reached (Fig. "*Dead band for position errors*").

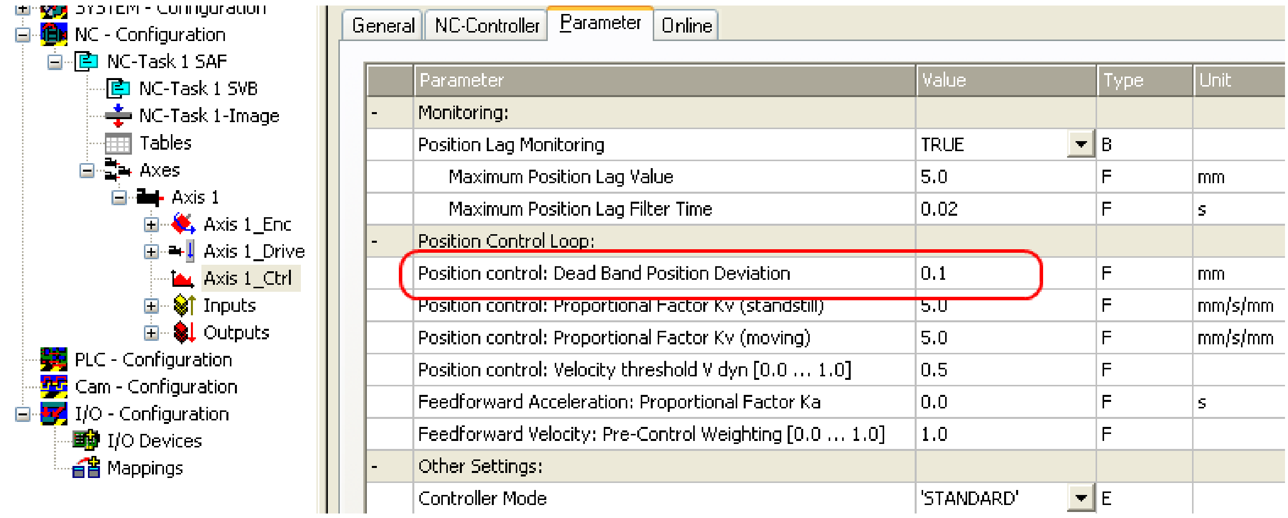


Fig. 179: *Dead band for position errors*

Setting the acceleration time

In order to pass through any resonances that may occur as quickly as possible, the ramps for the acceleration time and the deceleration time should be as steep as possible.

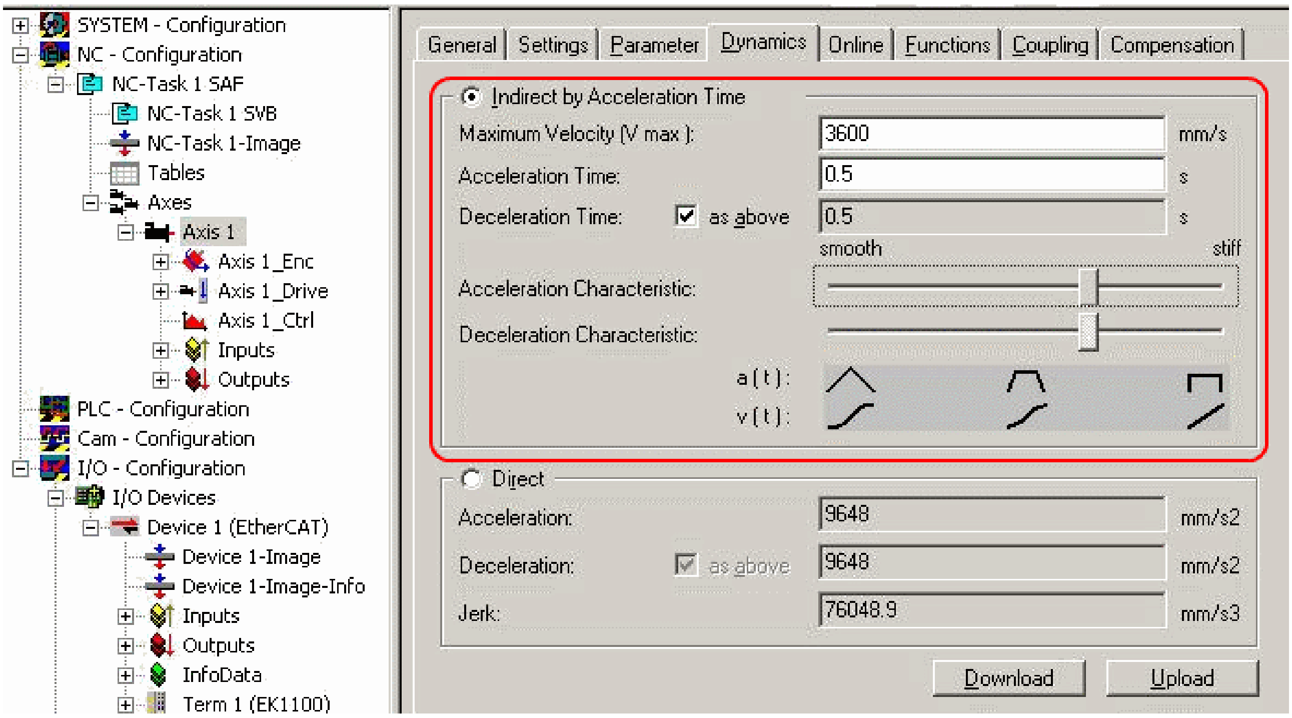


Fig. 180: *Setting the acceleration time*

NOTE

ATTENTION: Use a buffer capacitor terminal (EL9570) for short deceleration ramps.

Very short deceleration ramps may lead to temporarily increased feedback. In this case the terminal would report an error. To prevent this, one should connect a buffer capacitor terminal (EL9570) with a suitable ballast resistance (e.g. 10 Ohm) in parallel with the power supply (50 V) of the motor in order to absorb energy being fed back.

5.4.5 Application example

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 on the Beckhoff Website and installing it according to the installation instructions.

Motor control with visualization

Sample program:

https://infosys.beckhoff.com/content/1033/el70x7/Resources/zip/1308655627.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.

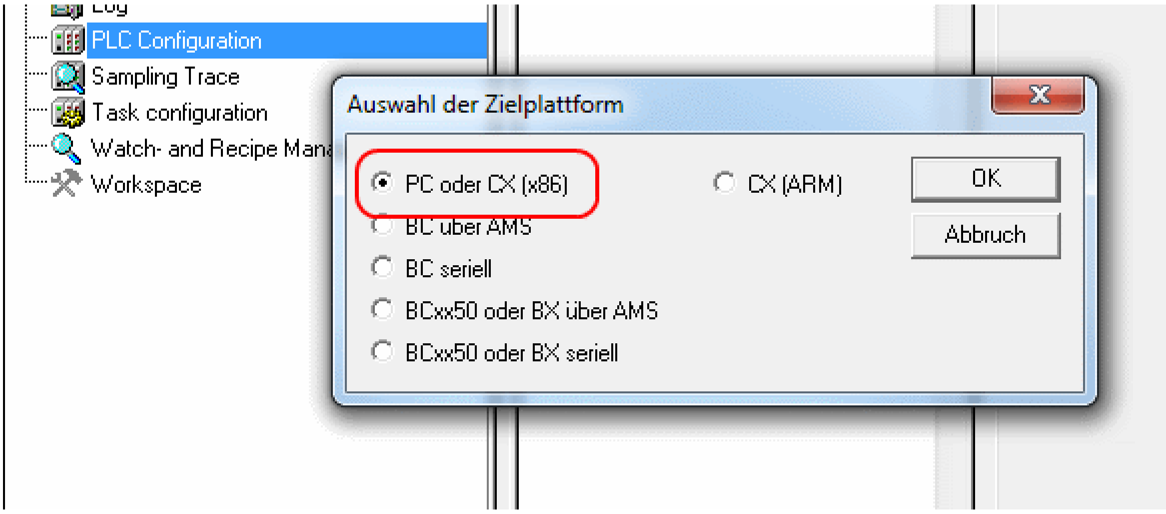


Fig. 181: *Selection of the target platform*

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.

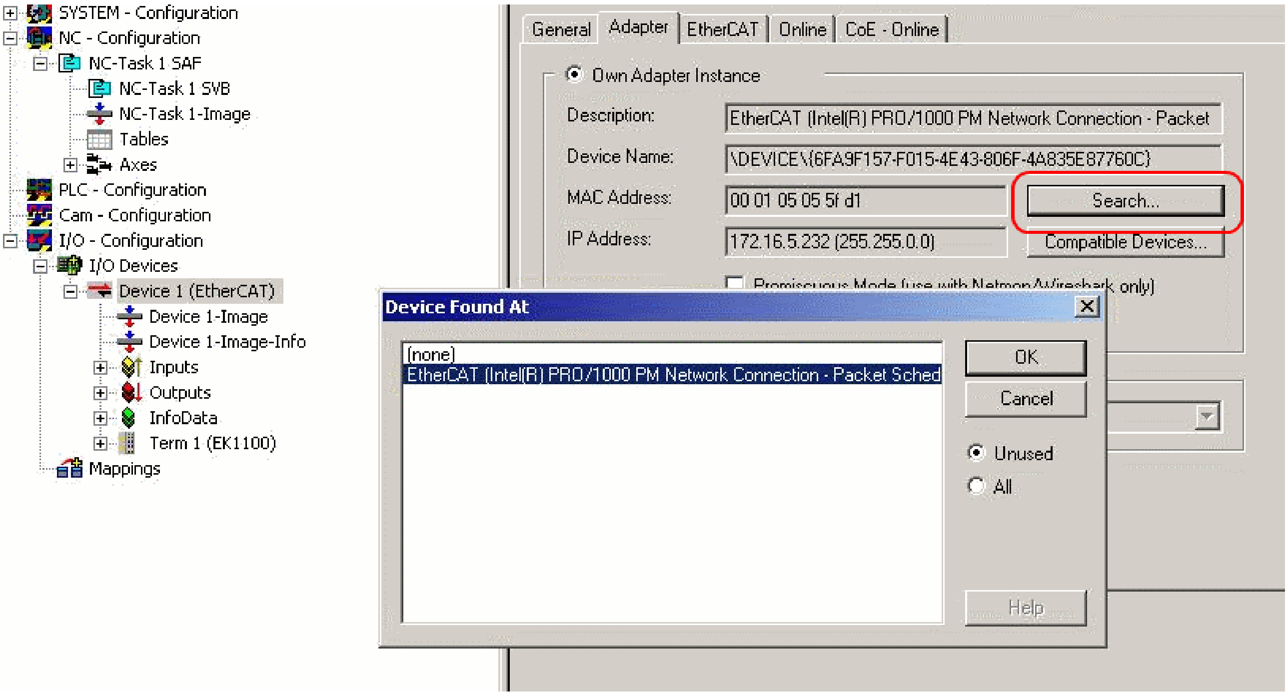


Fig. 182: *Selecting the MAC address*

- 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.

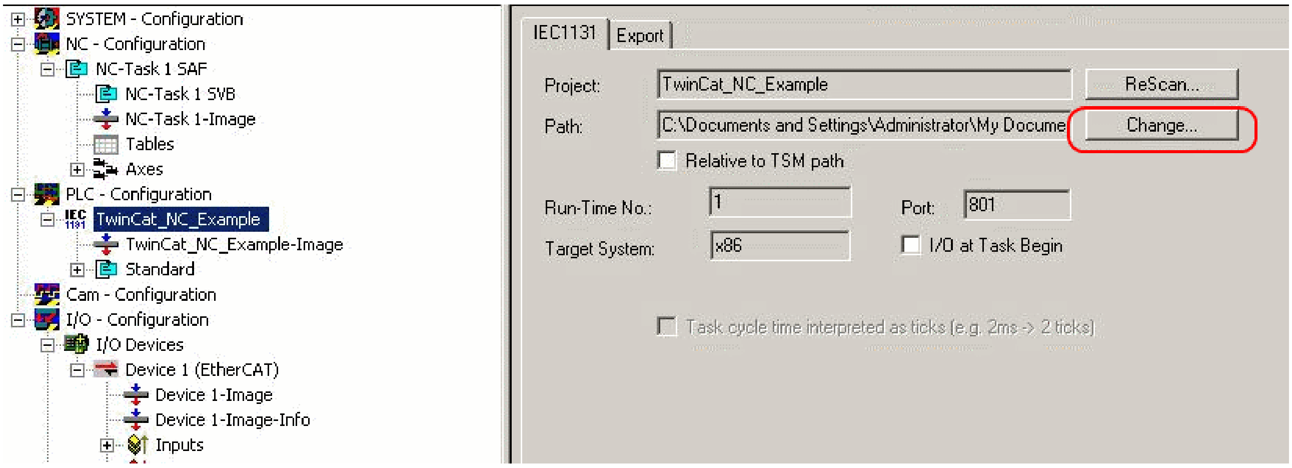


Fig. 183: *Changing the PLC path*

- 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 [▶ 139]“.

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

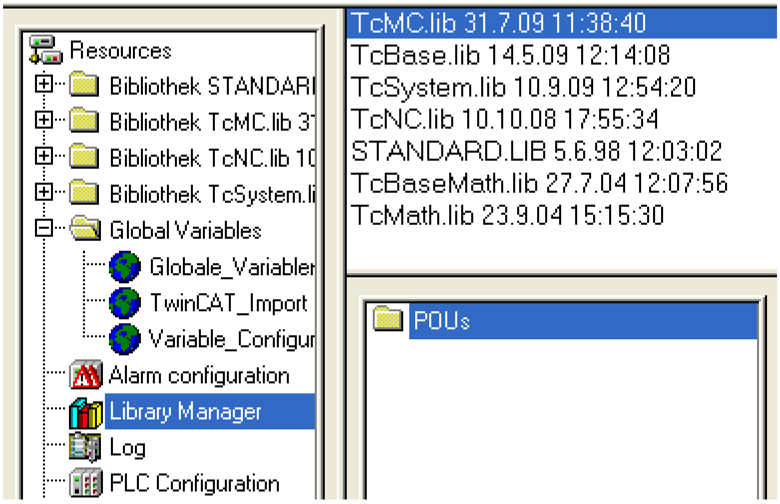


Fig. 184: *Required libraries*

Subsequently certain global variables are declared (see Fig.5). The data types PLCTONC_AXLESTRUCT and NCTOPLC_AXLESTRUCT deal with the communication between the PLC and the NC.

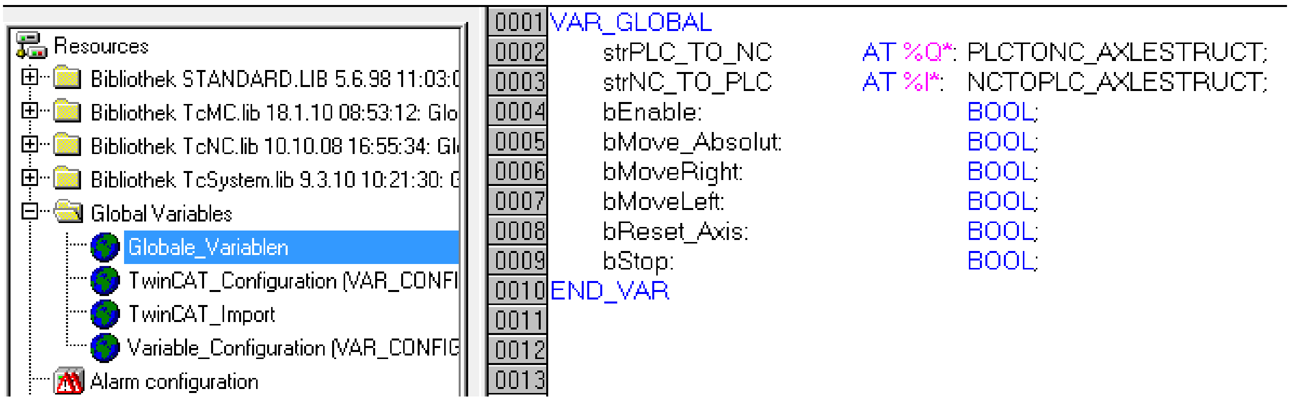


Fig. 185: *Global variables*

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.

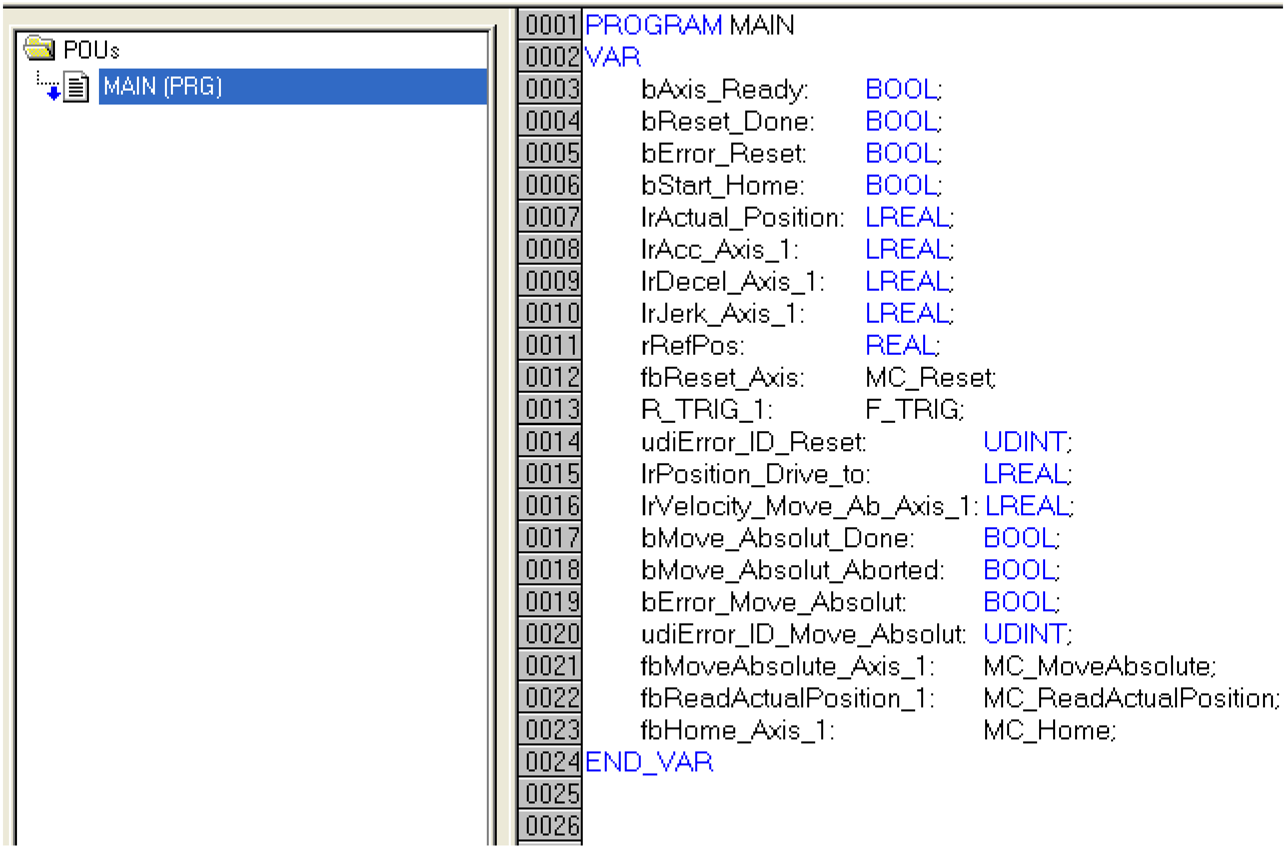


Fig. 186: *Local variables*

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

Bausteine
MAIN (PRG)

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

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

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

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

IF fbMoveAbsolute_Axis_1.Done THEN
    bMove_Absolut := FALSE;
END_IF

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

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

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

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

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

Fig. 187: *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.

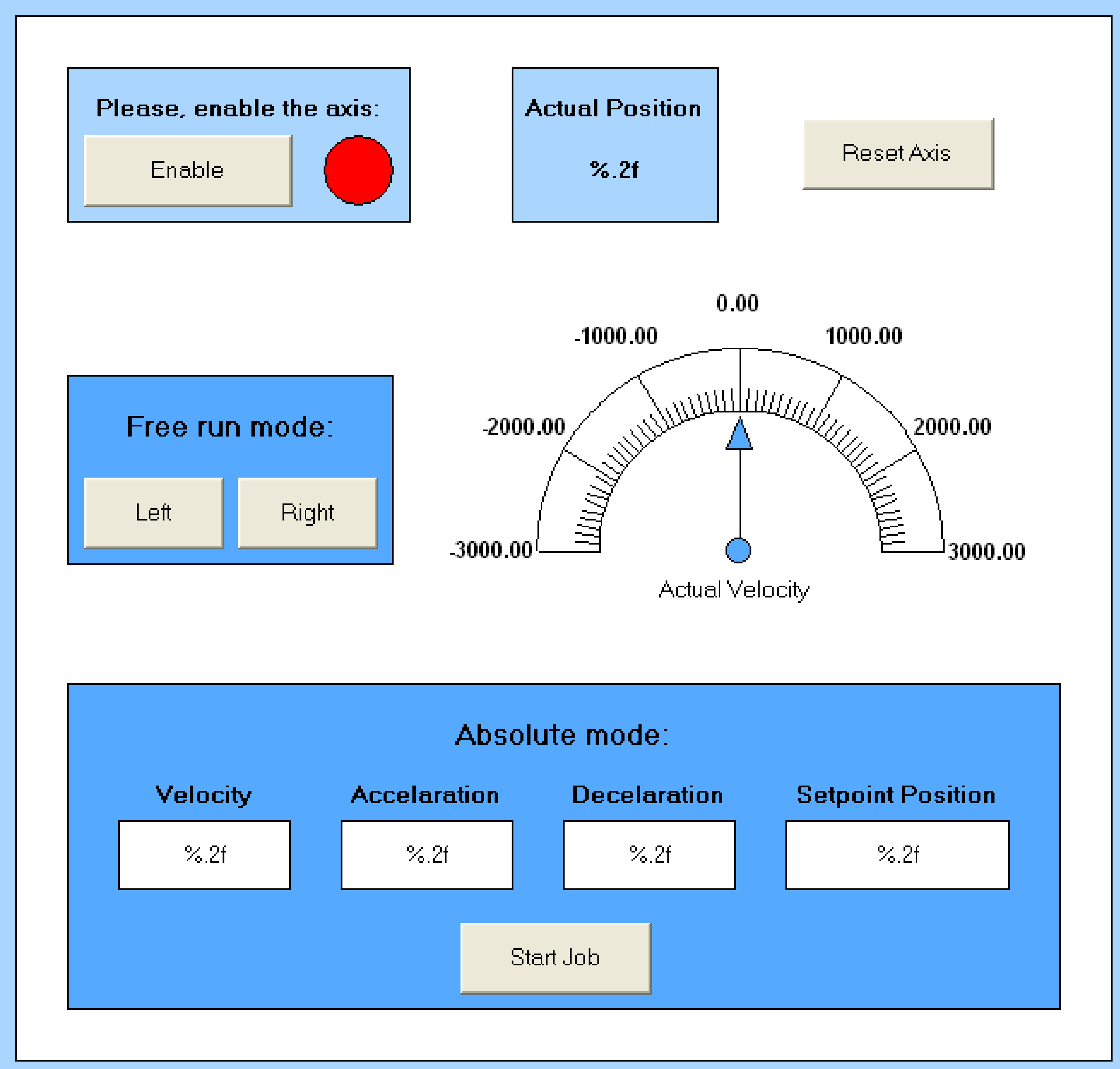


Fig. 188: *Visualization*

Information on function blocks and data types

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

5.5 Operating modes

5.5.1 Overview

The modes *Velocity direct*, *Position controller*, *Ext. Velocity mode*, *Ext. Position mode* and *Velocity sensorless* are supported. The operating mode is set in the CoE list in index 8012:01 [▸ 215] (Operating Mode). In the respective process data the user can additionally select the respective Predefined PDO Assignment [▸ 138]. All required variables are then in the process data.
The Predefined PDO Assignments Positioning interface and the compact Positioning interface can be used to realise an additional path control based on the positioning controller.

Automatic

Automatic mode is the default setting for the EL70x7. This operating mode is selected, the EL70x7 recognizes the set predefined PDO assignment and automatically selects between *Velocity direct* and *Position controller* so that the interplay between predefined PDO assignment and the matching mode is automatically guaranteed. If the user switches, for example, from Predefined PDO Assignment *Velocity control* auf *Position control*, the EL70x7 recognizes this and automatically switches from operating mode *Velocity direct* to *Position controller*.
The extended modes are not implemented in *Automatic* mode.
If the extended modes are not required, is the recommended to use *Automatic* mode.

Velocity direct

In *Velocity direct* mode, the EL70x7 operates in the cyclic velocity interface. A defined velocity can be set via the *STM Velocity* variable.

Position controller

In *Position controller* mode, the EL70x7 operates in the cyclical position interface. A defined position can be set via the *STM Position* variable.

Extended Velocity mode

In the *Extended Velocity* mode, the EL70x7 operates in the cyclic velocity interface with a field-oriented control. A defined velocity can be set via the *STM Velocity* variable.

Extended Position mode

In *Extended Position controller* mode, the EL70x7 operates in the cyclic velocity interface with a field-oriented control. A defined position can be set via the *STM Position* variable.

Velocity sensorless

In *Velocity sensorless* mode, the EL70x7 operates in the cyclic velocity interface. In this mode, above a minimum speed the motor current without encoder is controlled load-dependent. A defined velocity can be set via the STM Velocity variable.

Positioning interface

The position control loop is usually closed with the aid of TwinCAT NC. The *Positioning interface* can be used to transfer travel commands via the PLC directly to the terminal. The position control loop is closed by the terminal. This can be advantageous in simple, price-sensitive applications, since no TwinCAT NC licence is required. Only a very short TC cycle time is required, so that the controller load is reduced. However, the accuracy and the possibility of synchronization to other drive terminals and modules in the system is severely restricted.

Notes regarding the individual operating modes

The following matrix shows an overview of the limitations of individual operating modes.
It shows whether the operating mode supports third-party motors or only Beckhoff motors and whether or not an encoder is required. It also shows which operating mode performs a commutation determination operation after the axis is enabled.
The shaft moves minimally in both directions. This must be taken into account in the application.

	Automatic	Velocity direct	Position controller	Extended Velocity mode	Extended Position mode	Velocity sensorless
Beckhoff Motor (AS10xx)	x	x	x	x	x	x
Third-party motor	x	x	x	-	-	-
With encoder	x	x	x	x	x	-
Without encoder	x	x	x	-	-	x
Commutation determination required	-	-	-	x	x	-

Overview of the limitations of individual operating modes

Advantages of the individual operating modes

The following matrix shows the advantages of the individual operating modes.
With *Velocity sensorless* the velocity controller cannot be set "too hard". This has a slight effect on travel dynamics. The modes *Velocity direct* and *Position controller* offer very good travel dynamics for a stepper motor. However, significantly better travel dynamics, approaching that of a servomotor, can be achieved with the *Extended modes,* thanks to the field-oriented control.

	Automatic	Velocity direct	Position controller	Extended Velocity mode	Extended Position mode	Velocity sensorless
Control dynamics	+	+	+	++	++	**o**
Step loss recognition	x	x	x	Step losses are avoided	Step losses are avoided	**-**
Load angle recognition	x	x	x	always 90°	always 90°	-
Positioning interface	depending on mode selection	-	x	-	x	-
Load-dependent current	-	-	-	x	x	yes, if velo > $velo_{min}$
Energy efficiency	o	o	o	++	++	o, +

Overview of the advantages of individual operating modes

Required parameter settings for the individual operating modes

The following matrix provides an overview of the parameters required for the individual operating modes.
Motor XML files are provided online for all supported Beckhoff motors. The corresponding file can be inserted in the startup list. This file presets the parameters in an optimum manner. A little fine tuning may be beneficial, depending on the application.

	Velocity direct	Position contr.	Extended velocity mode	Extended position mode	Velocity sensor-less	Load angle recognition	Step loss recognition	
							with encoder	without encoder
Index 8010:03 Nominal voltage	X	x	x	x	x	x	x	x
Index 8010:04 Motor coil resistance		x		x				
Index 8010:05 Motor EMF					x	x		x
Index 8010:0A Motor coil inductance					x	x		x
Index 8011:01 Kp factor (curr.)	x	x	x	x	x	x	x	x
Index 8011:02 KI factor (curr.)	x	x	x	x	x	x	x	x
Index 8014:01 Feed forward (pos.)		x		x				
Index 8014:02 Kp factor (pos.)		x		x				
Index 8014:03 Kp factor (velo.)			x	x	x			
Index 8014:04 Tn (velo.)			x	x	x			

Overview of parameter settings for individual operating modes

5.5.2 Velocity direct

In Velocity direct mode, the EL70x7 operates in the cyclic velocity interface. A defined velocity can be set via the *STM Velocity* variable.

Prerequisites

- This mode can be used with a connected encoder or with the internal counter (without encoder).

- The process data can be transferred with TwinCAT NC or directly from the PLC.

Step by Step

- Add the terminal to the configuration as described in the section TwinCAT configuration settings – manual [▶ 103] or – Online scan [▶ 108].
- Link the terminal with the NC as described in section Integration into the NC configuration [▶ 139] (if TwinCAT NC is used).
- Configure the EL70x7
    - automatically - import the XML motor file into the startup directory as described in section Settings in the CoE - automatic [▶ 140].
    - manually - configure the parameters as described in section Settings in the CoE - manual [▶ 140].
- Set the operating mode in the CoE directory to Velocity direct [▶ 215], Fig. *"Velocity direct mode".*

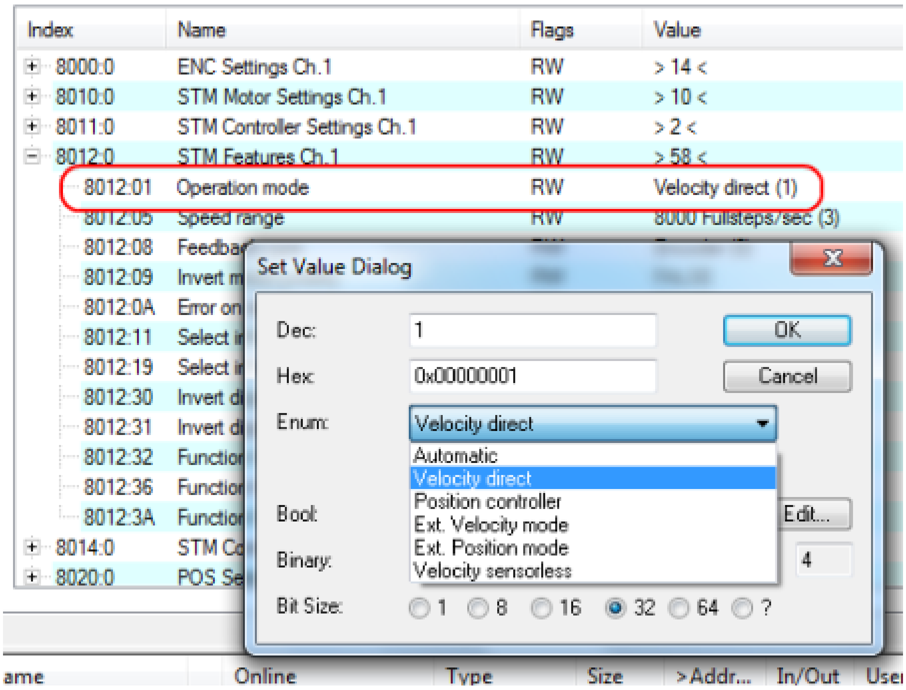


Fig. 189: *Velocity direct mode*

- Under Predefined PDO Assignments [▶ 138] select Velocity control, Velocity control compact or Velocity control compact with info data, Fig. *"Predefined PDO Assignment: Velocity control compact".*

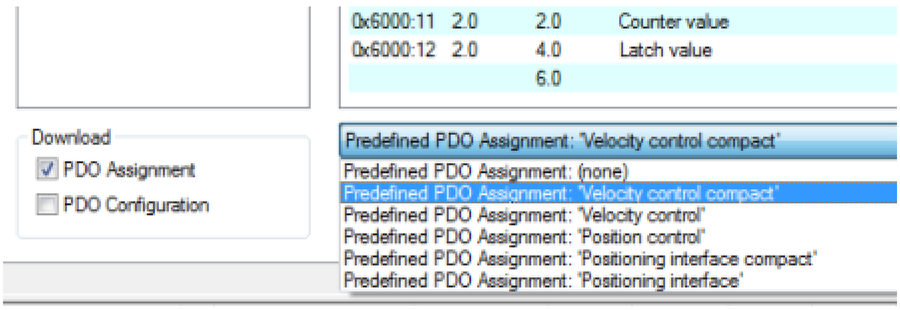


Fig. 190: *Predefined PDO Assignment: Velocity control compact*

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. Here you have two options.

◦ 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.
Tick all options and set override to 100% (see Fig. *"Enabling the axis in the NC"*). The axis is then ready.

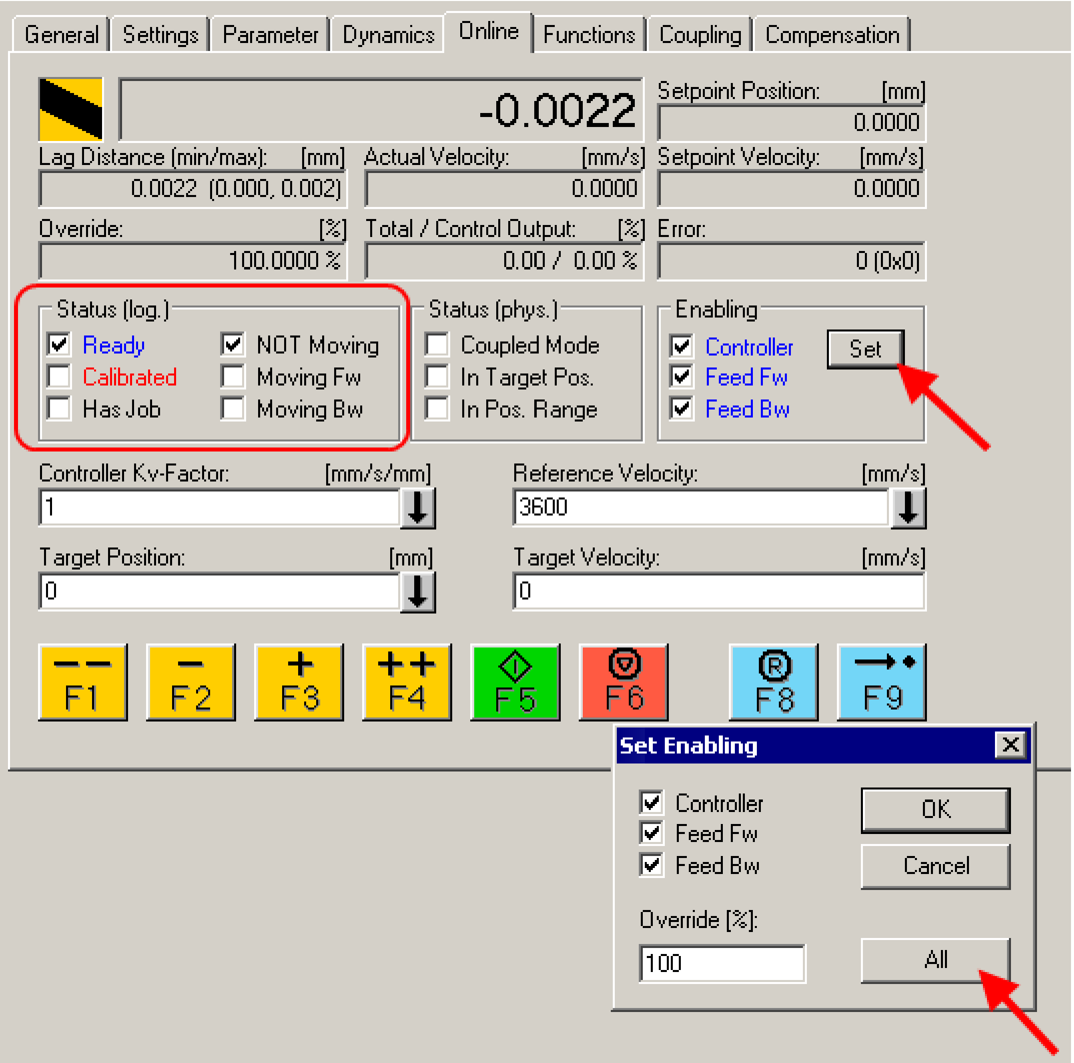


Fig. 191: *Enabling the axis in the NC*

- If you don't use the TwinCAT NC.
In this case you must run through the State Machine manually. Set the variable 0 x 7010:01 [▸ 220] Enable to 1 (true), Fig. *"Enabling the axis manually"*.

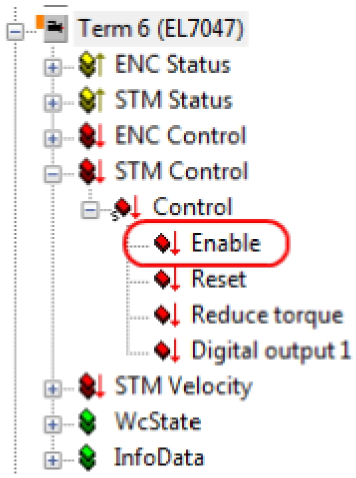


Fig. 192: *Enabling the axis manually*

- A defined velocity can be entered via the cyclic variable STM velocity (Fig. *"Entering the velocity"*). The speed is specified in % of the speed range (index 8012:05 [▸ 215]). The value + 32767 corresponds to 100%, the value -32767 corresponds to -100%.

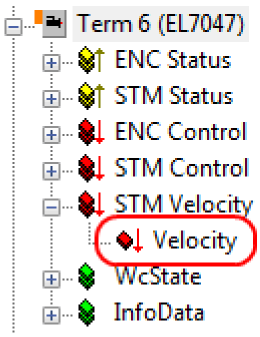


Fig. 193: *Entering the velocity*

5.5.3 Position controller

In *Position controller* mode, the EL70x7 operates in the cyclical position interface. A defined position can be set via the *STM Position* variable.

Notes

- This mode can be used with a connected encoder or with the internal counter (without encoder).
- The process data can be transferred with TwinCAT NC or directly from the PLC (Positioning interface).
- Third-party motors are supported

Step by Step

- Add the terminal to the configuration as described in the section TwinCAT configuration settings – manual [▸ 103] or – Online scan [▸ 108].
- Link the terminal with the NC as described in section Integration into the NC configuration [▸ 139] (if TwinCAT NC is used).
- Configure the EL70x7
  - automatically - import the XML motor file into the startup directory as described in section Settings in the CoE - automatic [▸ 140].
  - manually - configure the parameters as described in section Settings in the CoE - manual [▸ 140].
- Set the operating mode in the CoE directory to Position controller [▸ 215], Fig. *"Position controller mode"*.

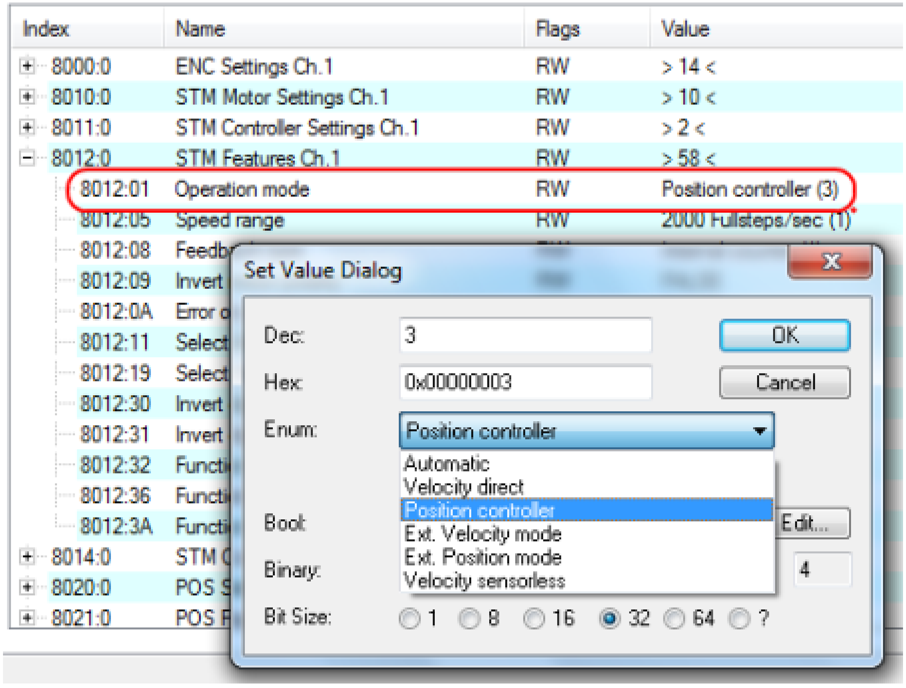


Fig. 194: *Position controller mode*

- Under Predefined PDO Assignments [▶ 138] select Position control, Positioning interface compact, Positioning interface or Positioning interface with info data , Fig. *"Predefined PDO Assignment: Position control"*.

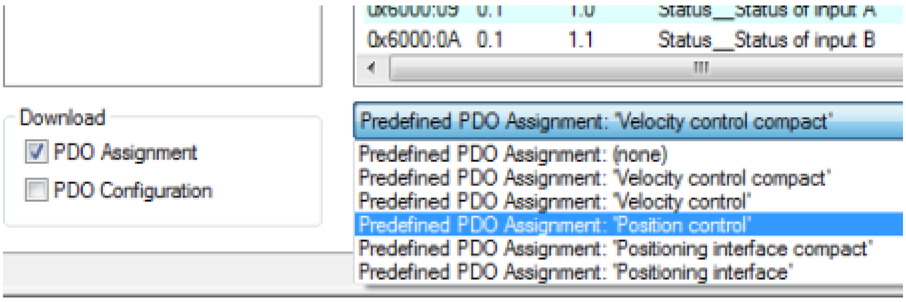


Fig. 195: *Predefined PDO Assignment: Position control*

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. Here you have two options.
  - 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.
    Tick all options and set override to 100% (see Fig. *"Enabling the axis in the NC"*). The axis is then ready.

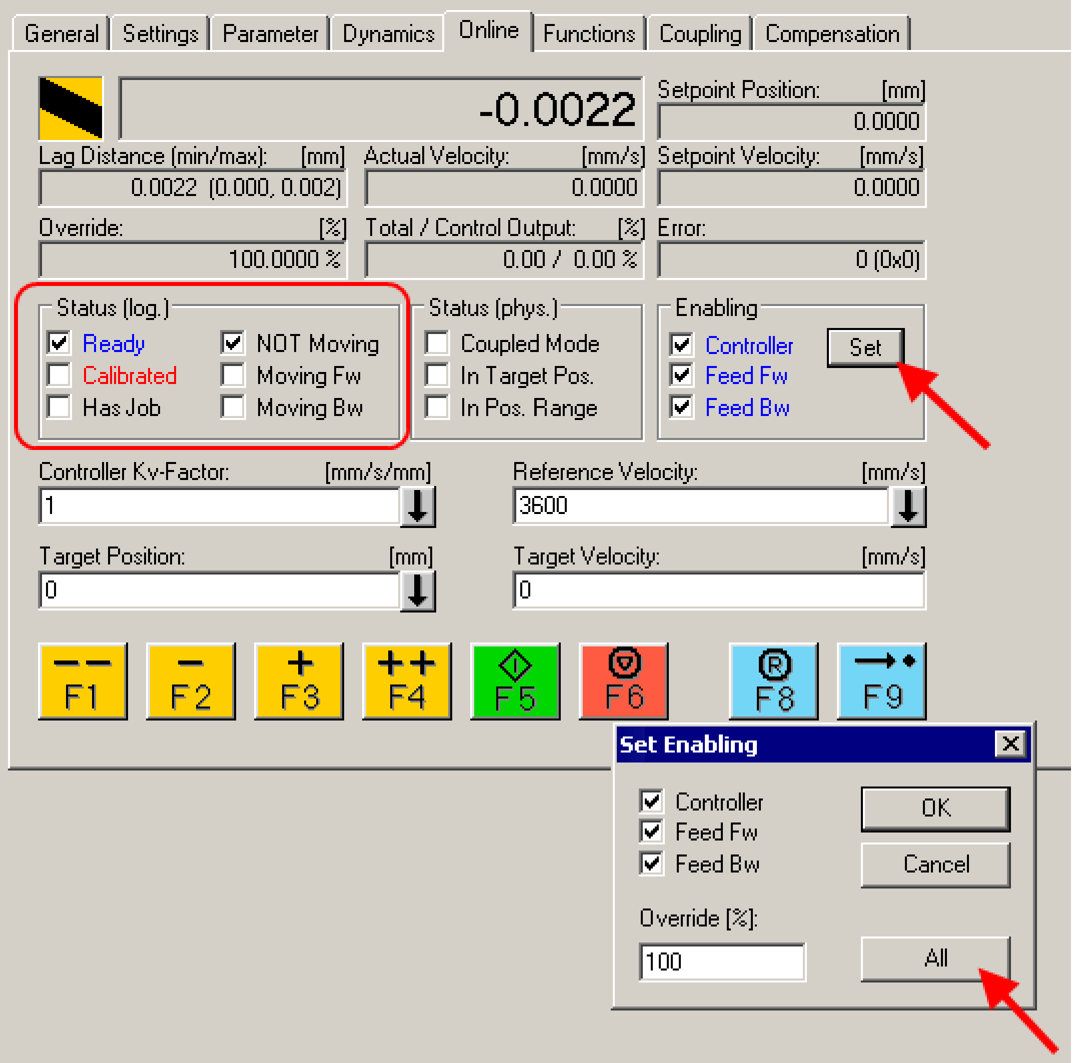


Fig. 196: *Enabling the axis in the NC*

- If you don’t use the TwinCAT NC.
  In this case you must run through the State Machine manually. Set the variable 0 x 7010:01 [▸ 220] Enable to 1 (true), Fig. *"Enabling the axis manually"*.

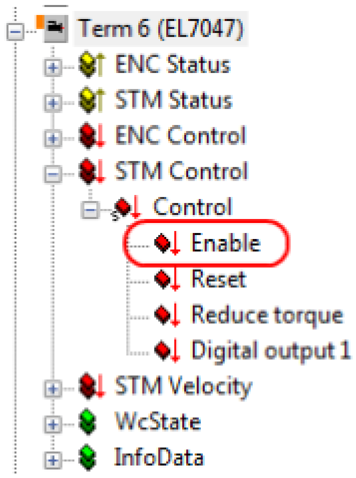


Fig. 197: *Enabling the axis manually*

- A defined position can be entered via the cyclic variable *STM Position* (Fig. *"Entering the position"*).
  The position is specified in increments and depends on the selected feedback (index 8012:0A [▸ 215]).
  For an AS10xx motor with internal counter, 12,800 (64-fold microstepping * 200 full steps of the

AS10xx motor) corresponds to one full turn. With an external encoder the value depends on the encoder. For an AS10xx motor encoder with 1024 INC/revolution, 4,096 (1024 INC/rev * 4-fold evaluation) corresponds to one full turn.

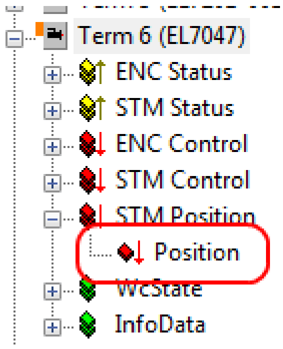


Fig. 198: *Entering the position*

5.5.4 Extended Velocity mode

In *Extended Velocity* mode the EL70x7 operates in the cyclic velocity interface with field-oriented control. A defined velocity can be set via the *STM Velocity* variable.

Notes

- This operating mode can be only be used when an encoder with sufficiently high resolution (min. 4000 [INC/360°]) is connected.
- Only stepper motors from Beckhoff Automation GmbH (AS10xx) are supported.
- TwinCAT NC is required.
- When this mode is enabled commutation determination is required, since the shaft requires a degree of clearance. To this end the shaft moves a few degrees right and left.

Step by Step

- Add the terminal to the configuration as described in the section TwinCAT configuration settings – manual [▶ 103] or – Online scan [▶ 108].
- Link the terminal with the NC as described in section Integration into the NC configuration [▶ 139] (if TwinCAT NC is used).
- Configure the EL70x7
    - automatically - import the XML motor file into the startup directory as described in section Settings in the CoE - automatic [▶ 140].
    - manually - configure the parameters as described in section Settings in the CoE - manual [▶ 140].
- Set the operating mode in the CoE directory to Extended Velocity mode [▶ 215], Fig. *"Extended Velocity mode".*

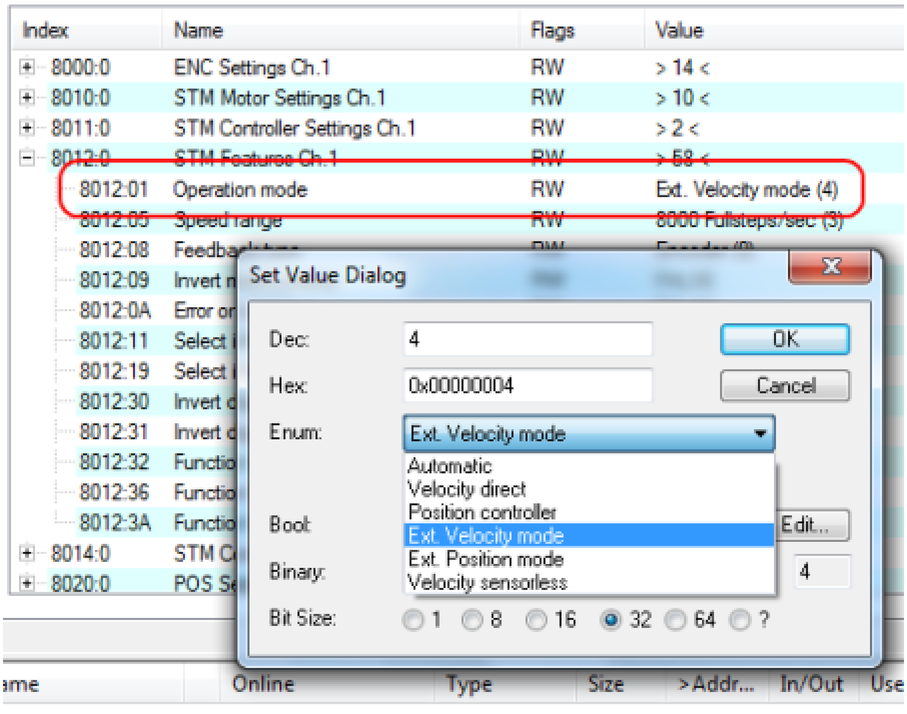


Fig. 199: *Extended Velocity mode*

- Under Predefined PDO Assignments [▶ 138] select Velocity control or Velocity control compact or Velocity control with info data, Fig. *"Predefined PDO Assignment: Velocity control compact"*.

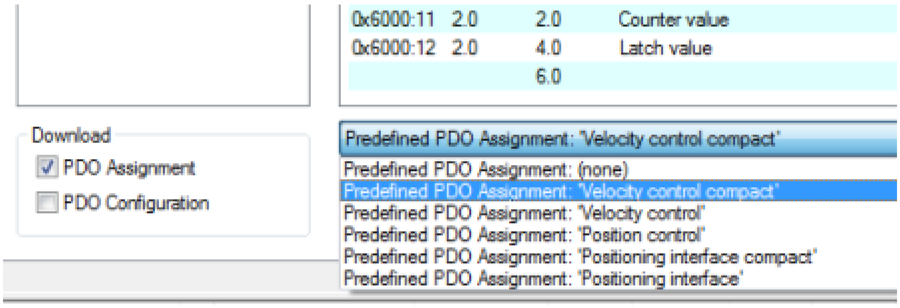


Fig. 200: *Predefined PDO Assignment: Velocity control compact*

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. Here you have two options.
    - 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.
      Tick all options and set override to 100% (see Fig. *"Enabling the axis in the NC"*). The axis is then ready.

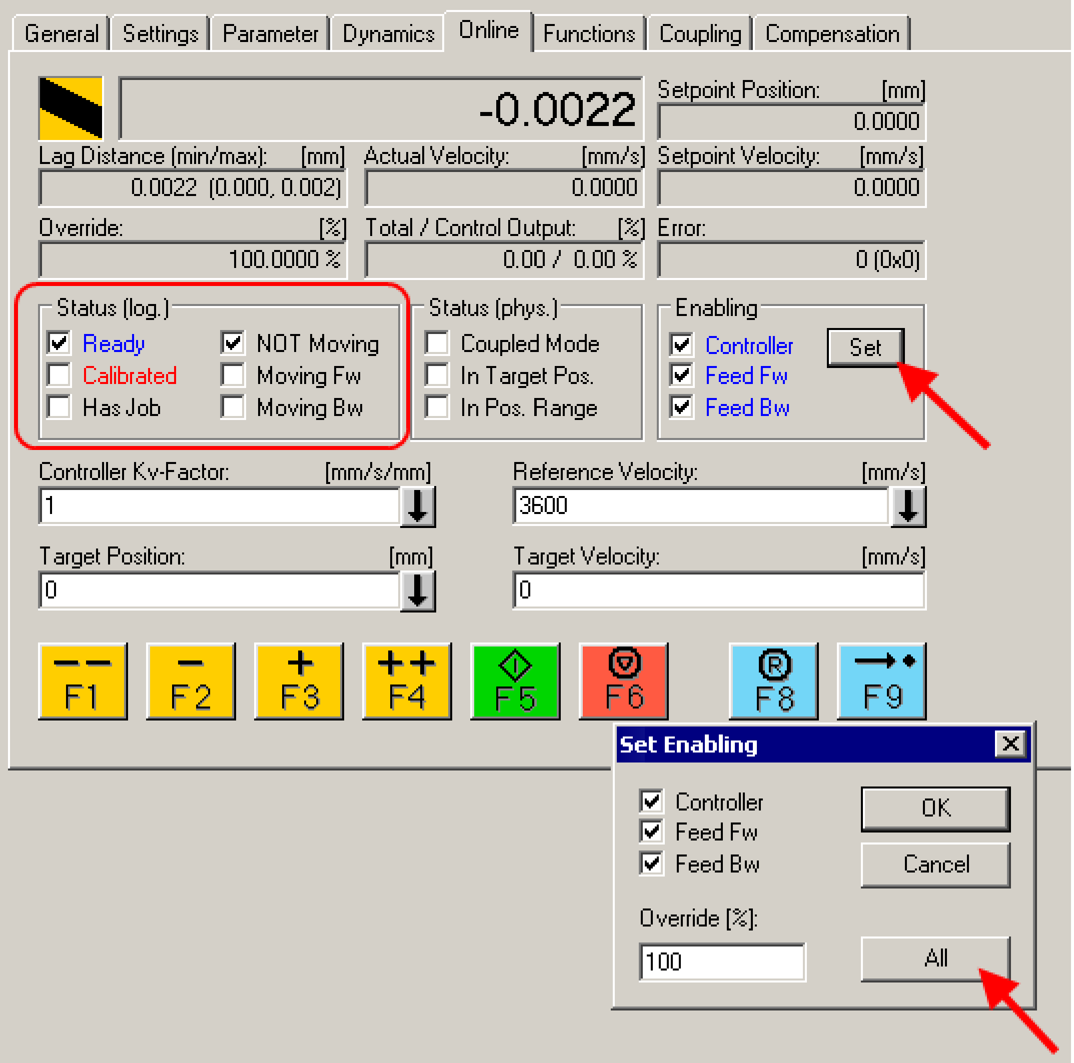


Fig. 201: *Enabling the axis in the NC*

- If you don't use the TwinCAT NC.
  In this case you must run through the State Machine manually. Set the variable 0 x 7010:01 [▸ 220] Enable to 1 (true), Fig. *"Enabling the axis manually"*.

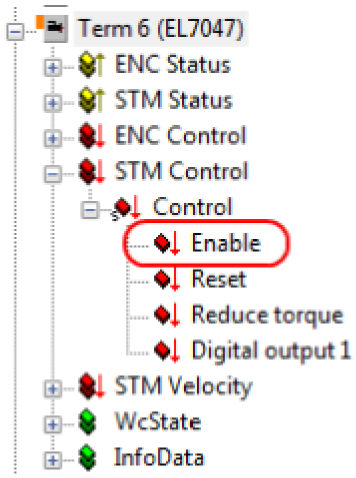


Fig. 202: *Enabling the axis manually*

- A defined velocity can be entered via the cyclic variable STM velocity (Fig. *"Entering the velocity"*). The speed is specified in % of the speed range (index 8012:05 [▸ 215]). The value + 32767 corresponds to 100%, the value -32767 corresponds to -100%.

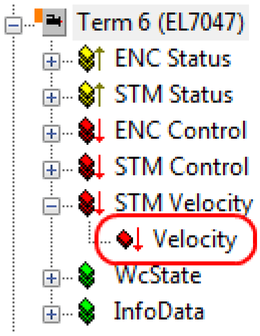


Fig. 203: *Entering the velocity*

5.5.5 Extended Position mode

In *Extended Position* mode the EL70x7 operates in the cyclic position interface with field-oriented control. A defined position can be set via the *STM Position* variable.

Notes

- This operating mode can be only be used when an encoder with sufficiently high resolution (min. 4000 [INC/360°]) is connected.
- Only stepper motors from Beckhoff (AS10xx) are supported.
- TwinCAT NC is not required.
- When this mode is enabled commutation determination is required, since the shaft requires a degree of clearance. To this end the shaft moves a few degrees right and left.

Step by Step

- Add the terminal to the configuration as described in the section TwinCAT configuration settings – manual [▶ 103] or – Online scan [▶ 108].
- Link the terminal with the NC as described in section Integration into the NC configuration [▶ 139] (if TwinCAT NC is used).
- Configure the EL70x7
  - automatically - import the XML motor file into the startup directory as described in section Settings in the CoE - automatic [▶ 140].
  - manually - configure the parameters as described in section Settings in the CoE - manual [▶ 140].
- Set the operating mode in the CoE directory to Ext. Position mode [▶ 215], Fig. *"Ext. Position mode".*

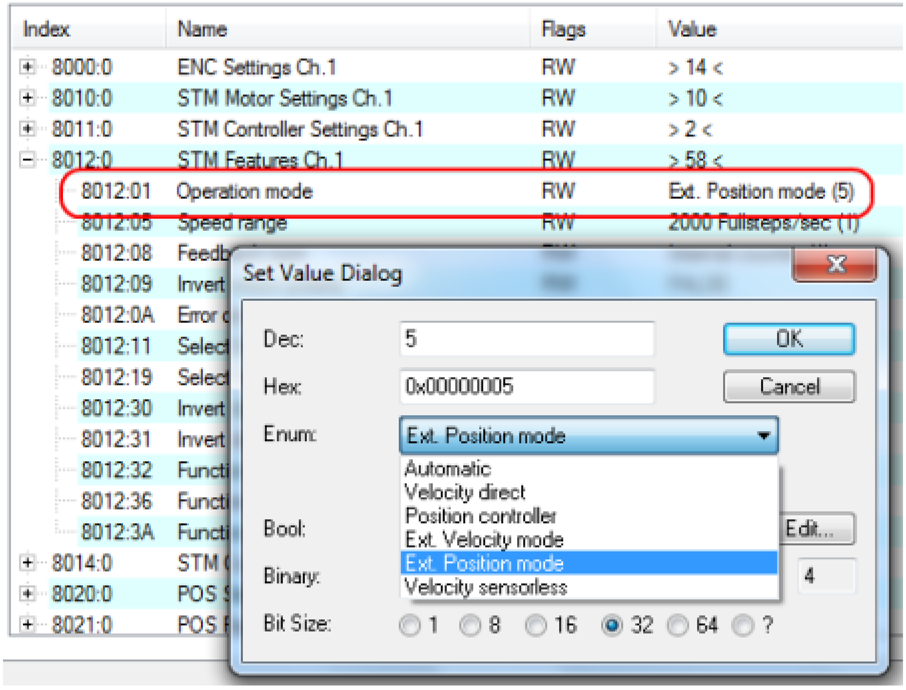


Fig. 204: *Ext. Position mode*

- Under Predefined PDO Assignments [▶ 138] select Position control, Positioning interface compact, Positioning interface or Positioning interface with info data , Fig. *"Predefined PDO Assignment: Position control"*.

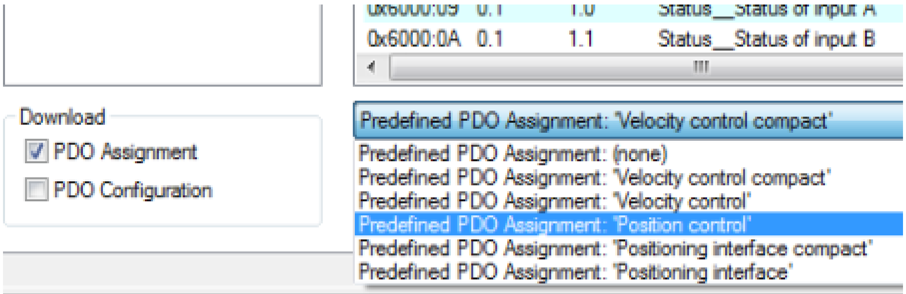


Fig. 205: *Predefined PDO Assignment: Position control*

- Activate the configuration (Ctrl+Shift+F4).
- Run through the State Machine of the terminal. Here you have two options.
  - 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.
    Tick all options and set override to 100% (see Fig. *"Enabling the axis in the NC"*). The axis is then ready.

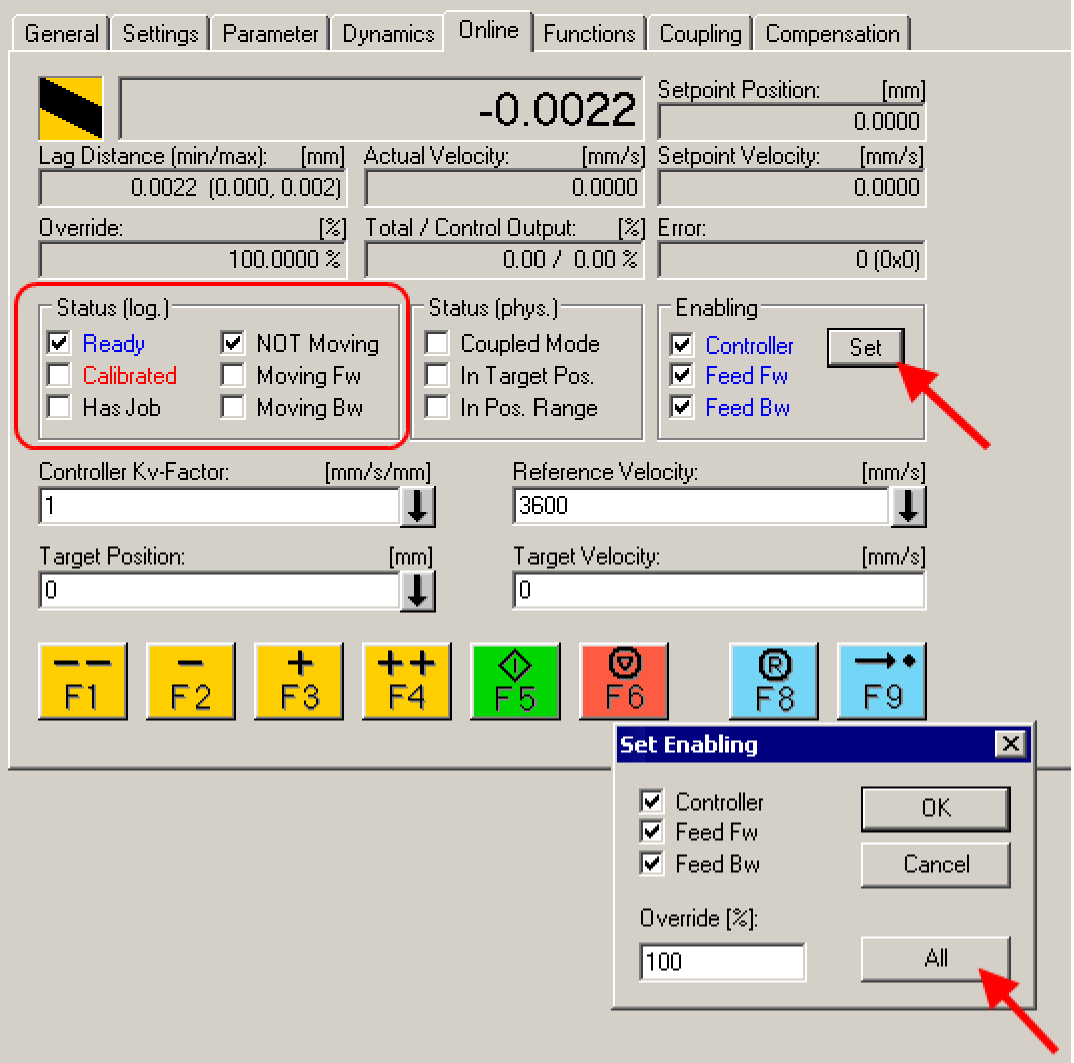


Fig. 206: *Enabling the axis in the NC*

- If you don't use the TwinCAT NC.
  In this case you must run through the State Machine manually. Set the variable 0 x 7010:01 [▶ 220] Enable to 1 (true), Fig. *"Enabling the axis manually"*.

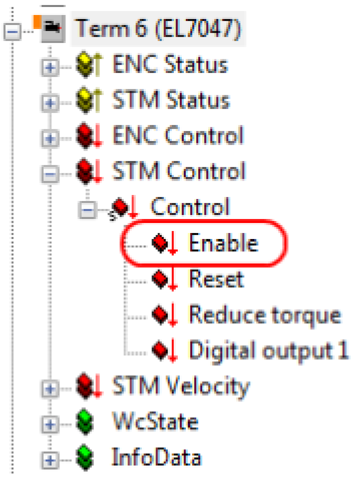


Fig. 207: *Enabling the axis manually*

- A defined position can be entered via the cyclic variable *STM Position* (Fig. *"Entering the position"*). This mode is only supported with AS10xx motors with corresponding 1024 INC/rev encoders. The position is specified in increments. 4096 (1024 INC/rev * 4-fold evaluation) corresponds to one full turn.

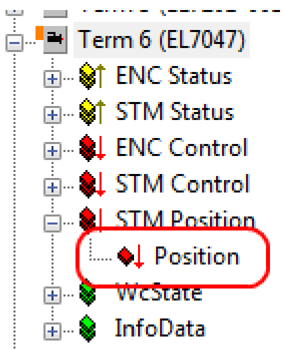


Fig. 208: *Entering the position*

5.5.6 Basic principles: "Positioning interface"

The "*Positioning interface"* offers the user a possibility to implement travel commands directly on the terminal.

Predefined PDO Assignment

The "Predefined PDO Assignment [▸ 138]" enables a simplified selection of the process data. Select the function “*Positioning interface*” or “*Positioning interface compact*” in the lower part of the Process data tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.

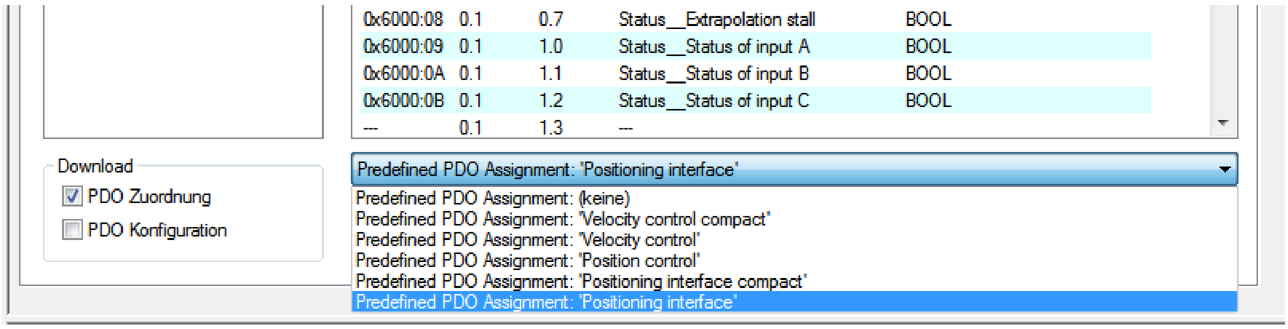


Fig. 209: *Predefined PDO Assignment*

Parameter set

Two objects are at the user’s disposal in the CoE for the configuration – the “POS Settings” (Index 8020 [▸ 217]) and the “POS Features” (Index 8021 [▸ 218]).

Index	Name	Flags	Wert
8020:0	POS Settings Ch.1	RW	> 15 <
8020:01	Velocity min.	RW	100
8020:02	Velocity max.	RW	10000
8020:03	Acceleration pos.	RW	0x03E8 (1000)
8020:04	Acceleration neg.	RW	0x03E8 (1000)
8020:05	Deceleration pos.	RW	0x03E8 (1000)
8020:06	Deceleration neg.	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0064 (100)
8020:08	Calibration position	RW	0x00000000 (0)
8020:09	Calibration velocity (towards plc cam)	RW	200
8020:0A	Calibration Velocity (off plc cam)	RW	50
8020:0B	Target window	RW	0x0014 (20)
8020:0C	In-Target timeout	RW	0x03E8 (1000)
8020:0D	Dead time compensation	RW	50
8020:0E	Modulo factor	RW	0x00000000 (0)
8020:0F	Modulo tolerance window	RW	0x00000000 (0)
8021:0	POS Features Ch.1	RW	> 20 <
8021:01	Start type	RW	Absolute (1)
8021:11	Time information	RW	Elapsed time (0)
8021:13	Invert calibration cam search direction	RW	TRUE
8021:14	Invert sync impulse search direction	RW	FALSE

Fig. 210: *Settings objects in the CoE*

POS Settings

Velocity min.:

For reasons of performance when ramping down to the target position, the terminal needs a safety margin of 0.5%. That means that, depending on the maximum velocity reached and the configured deceleration, the time is calculated at which the deceleration ramp begins. In order to always reach the destination reliably, 0.5% is subtracted from the position determined. If the deceleration ramp has ended and the destination has not yet been reached, the terminal drives at the velocity "*Velocity min.*" to the destination. It must be configured in such a way that the motor is able to stop abruptly and without a step loss at this velocity.

Velocity max.:

The maximum velocity with which the motor drives during a travel command.

"Speed range" (index 8012:05 [▶ 215]) [applies to EL70x7]

Velocity min./max. are standardised to the configured "Speed range" (Index 8012:05). This means that for a "Speed range" of 4000 full steps/second, for example, for a speed output of 100% (i.e. 4000 full steps/second) 10,000 should be entered under "Velocity max.", and 5,000 for 50% (i.e. 2000 full steps/second).

Acceleration pos.:

Acceleration time in the positive direction of rotation.

The 5 parameters for acceleration also refer to the set "*Speed range*" and are given in ms. With a setting of 1000, the terminal accelerates the motor from 0 to 100% in 1000 ms. At a speed of 50% the acceleration time is linearly reduced to half accordingly.

Acceleration neg.:

Acceleration time in the negative direction of rotation.

Deceleration pos.:

Deceleration time in the positive direction of rotation.

Deceleration neg.:

Deceleration time in the negative direction of rotation.

Emergency deceleration:

Emergency deceleration time (both directions of rotation). If “*Emergency stop*” is set in the appropriate PDO, the motor is stopped within this time.

Calibration position:

The current counter value is loaded with this value after calibration.

Calibration velocity (towards plc cam):

Velocity with which the motor travels towards the cam during calibration.

Calibration velocity (off plc cam):

Velocity with which the motor travels away from the cam during calibration.

Target window:

Target window of the travel distance control. “*In-Target*” is set if the motor comes to a stop within this target window.

In-Target timeout:

“*In-Target*” is not set if the motor is not within the target window after the expiry of the travel distance control after this set time. This condition can be recognised only by checking the falling edge of “*Busy*”.

Dead time compensation:

Compensation of the internal propagation delays. This parameter does not have to be changed with standard applications.

Modulo factor:

The “*Modulo factor*” is referred to for the calculation of the target position and the direction of rotation in the modulo operating modes. It refers to the controlled system.

Modulo tolerance window:

Tolerance window for the determination of the start condition of the modulo operating modes.

POS Features

Start type:

The “*Start type*” specifies the type of calculation used to determine the target position (see below [▶ 177]).

Time information:

The meaning of the “*Actual drive time*” displayed is configured by this parameter. At present this value cannot be changed, since there are no further selection options. The elapsed time of the travel command is displayed.

Invert calibration cam search direction:

In relation to a positive direction of rotation, the direction of the search for the calibration cam is configured here (travel towards the cam).

Invert sync impulse search direction:

In relation to a positive direction of rotation, the direction of the search is configured here in accordance with the HW sync pulse (travel away from the cam).

Information and diagnostic data

Information and diagnostic data

Via the information and diagnostic data, the user can obtain a more exact statement about which error occurred during a travel command.

Index	Name	Flags	Wert
9020:0	POS Info data Ch.1	RO	> 3 <
9020:01	Status word	RO	0x0000 (0)
9020:03	State (drive controller)	RO	Idle (1)
A010:0	STM Diag data Ch.1	RO	> 17 <
A020:0	POS Diag data Ch.1	RO	> 3 <
A020:01	Command rejected	RO	FALSE
A020:02	Command aborted	RO	FALSE
A020:03	Target overrun	RO	FALSE

Fig. 211: *Diagnostic objects in the CoE*

POS Info data

Status word:

The “*Status word*” reflects the status bits used in *Index A020* in a data word, in order to be able to process them more simply in the PLC. The positions of the bits correspond to the number of the subindex-1.

Bit 0: Command rejected
Bit 1: Command aborded
Bit 2: Target overrun

State (drive controller):

The current status of the internal state machine is displayed here (see below [▶ 177]).

POS Diag data:

Command rejected:

A dynamic change of the target position is not accepted each time by the terminal, since this is then not possible. The new command is rejected in this case and indicated by the setting of this bit.

These 3 diagnostic bits are transmitted synchronously to the controller by setting “*Warning*” in the PDO.

Command aborted:

If the current travel command is prematurely aborted due to an internal error or by an “Emergency stop”.

Target overrun:

In the case of a dynamic change of the target position, the change may take place at a relatively late point in time. The consequence of this may be that a change in the direction of rotation is necessary and that the new target position may be overrun. "*Target overrun*" is set if this occurs.

States of the internal state machine

States of the internal state machine

The state (drive controller) (Index 9020:03 [▸ 223]) provides information about the current state of the internal state machine. For diagnostic purposes this can be read out by the PLC for the propagation delay. The internal cycle works constantly with 250 µs. A connected PLC cycle is very probably slower (e.g. 1 ms). For this reason it may be the case that some states are not visible at all in the PLC, since these will sometimes run through only one internal cycle.

Name	ID	Description
INIT	0x0000	Initialisation/preparation for the next travel command
IDLE	0x0001	Wait for the next travel command
START	0x0010	The new command is evaluated and the corresponding calculations are performed
ACCEL	0x0011	Acceleration phase
CONST	0x0012	Constant phase
DECEL	0x0013	Deceleration phase
EMCY	0x0020	An "*Emergency stop*" has been triggered
STOP	0x0021	The motor has stopped
CALI_START	0x0100	Start of a calibration command
CALI_GO_CAM	0x0110	The motor is being driven towards the cam
CALI_ON_CAM	0x0111	The cam has been reached
CALI_GO_SYNC	0x0120	The motor is being driven in the direction of the HW sync pulse
CALI_LEAVE_CAM	0x0121	The motor is being driven away from the cam
CALI_STOP	0x0130	End of the calibration phase
CALIBRATED	0x0140	The motor is calibrated
NOT_CALIBRATED	0x0141	The motor is not calibrated
PRE_TARGET	0x1000	The set position has been reached; the position controller "pulls" the motor further into the target; "*In-Target timeout*" is started here
TARGET	0x1001	The motor has reached the target window within the timeout
TARGET_RESTART	0x1002	A dynamic change of the target position is processed here
END	0x2000	End of the positioning phase
WARNING	0x4000	A warning state occurred during the travel command; this is processed here
ERROR	0x8000	An error state occurred during the travel command; this is processed here
UNDEFINED	0xFFFF	Undefined state (can occur, for example, if the driver stage has no control voltage)

States of the internal state machine

Standard sequence of a travel command

Standard sequence of a travel command

The "normally" sequence of a travel command is shown in the following flow diagram.
Coarse distinction is made between these four stages:

StartUp:

Test the system and the ready status of the motor.

Start positioning:

Write all variables and calculate the desired target position with the appropriate “*Start type*”. Subsequently, start the travel command.

Evaluate status:

Monitor the terminal state and, if necessary, dynamically change the target position.

Error handling:

In case of error, procure the necessary information from the CoE and evaluate it.

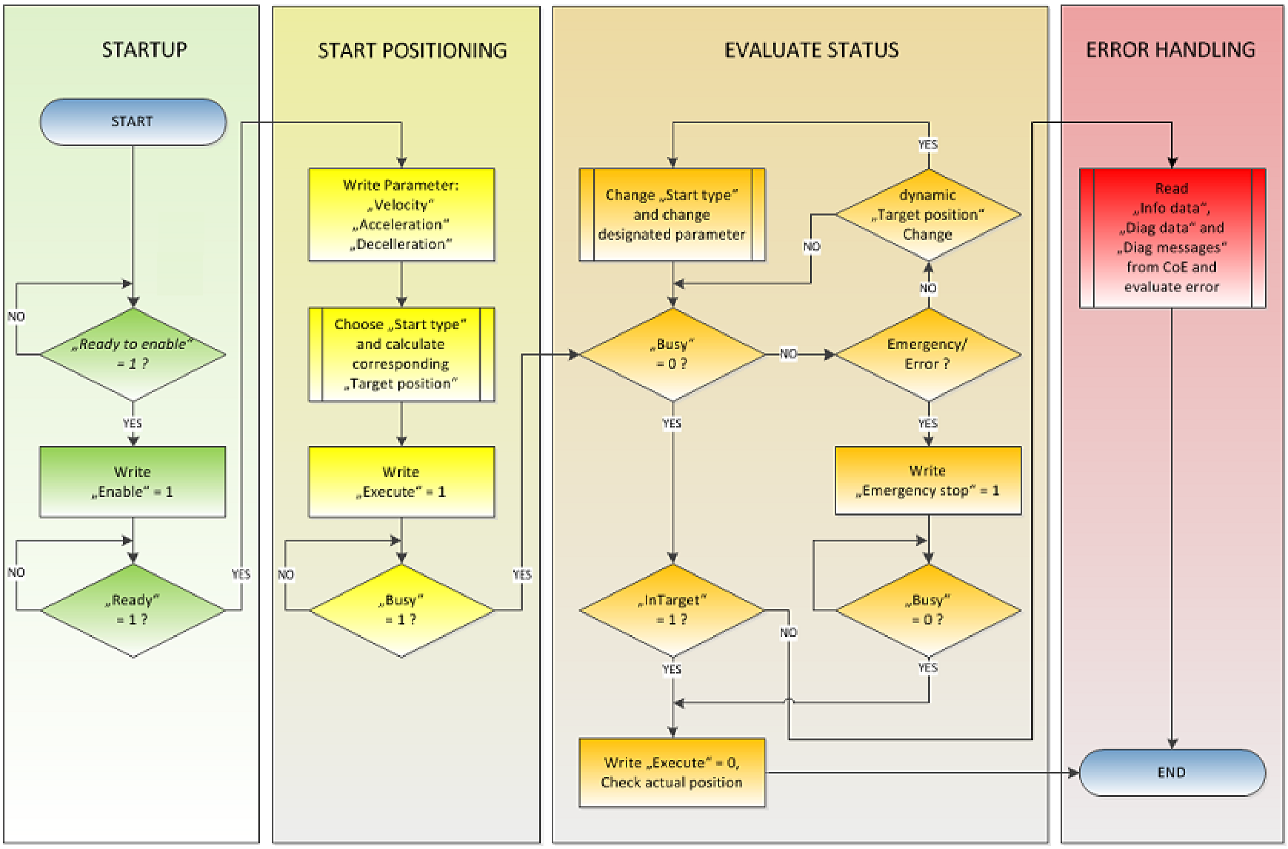


Fig. 212: *Flow diagram for a travel command*

Start types

The “*Positioning interface*” offers different types of positioning. The following table contains all commands supported; these are divided into 4 groups.

Name	Com-mand	Group	Description
ABSOLUTE	0x0001	Standard [▶ 178]	Absolute positioning to a specified target position
RELATIVE	0x0002		Relative positioning to a calculated target position; a specified position difference is added to the current position
ENDLESS_PLUS	0x0003		Endless travel in the positive direction of rotation (direct specification of a speed)
ENDLESS_MINUS	0x0004		Endless travel in the negative direction of rotation (direct specification of a speed)
ADDITIVE	0x0006		Additive positioning to a calculated target position; a specified position difference is added to the last target position
ABSOLUTE_CHANGE	0x1001	Standard Ext. [▶ 180]	Dynamic change of the target position during a travel command to a new absolute position
RELATIVE_CHANGE	0x1002		Dynamic change of the target position during a travel command to a new relative position (the current changing position value is used here also)
ADDITIVE_CHANGE	0x1006		Dynamic change of the target position during a travel command to a new additive position (the last target position is used here)
MODULO_SHORT	0x0105	Modulo [▶ 181]	Modulo positioning along the shortest path to the modulo position (positive or negative), calculated by the "*Modulo factor*" (Index 8020:0E [▶ 217])
MODULO_SHORT_EXT	0x0115		Modulo positioning along the shortest path to the modulo position; the "*Modulo tolerance window*" (Index 8020:0F [▶ 217]) is ignored
MODULO_PLUS	0x0205		Modulo positioning in the positive direction of rotation to the calculated modulo position
MODULO_PLUS_EXT	0x0215		Modulo positioning in the positive direction of rotation to the calculated modulo position; the "*Modulo tolerance window*" is ignored
MODULO_MINUS	0x0305		Modulo positioning in the negative direction of rotation to the calculated modulo position
MODULO_MINUS_EXT	0x0315		Modulo positioning in the negative direction of rotation to the calculated modulo position; the "*Modulo tolerance window*" is ignored
MODULO_CURRENT	0x0405		Modulo positioning in the last direction of rotation to the calculated modulo position
MODULO_CURRENT_EXT	0x0415		Modulo positioning in the last direction of rotation to the calculated modulo position; the "*Modulo tolerance window*" is ignored
CALI_PLC_CAM	0x6000	Calibration [▶ 180]	Start a calibration with cam (digital inputs)
CALI_HW_SYNC	0x6100		start a calibration with cam and HW sync pulse (C-track)
SET_CALIBRATION	0x6E00		Manually set the terminal to "*Calibrated*"
SET_CALIBRATION_AUTO	0x6E01		Automatically set the terminal to "*Calibrated*" on the first rising edge on "*Enable*"
CLEAR_CALIBRATION	0x6F00		Manually delete the calibration

Supported "Start types" of the "Positioning interface"

ABSOLUTE:

The absolute positioning represents the simplest positioning case. A position B is specified and travelled to from the start point A.

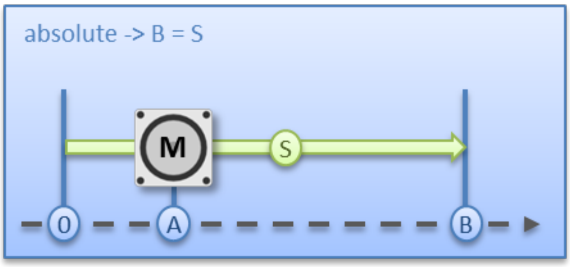


Fig. 213: *Absolute positioning*

RELATIVE:

In relative positioning, the user specifies a position delta S, which is added to the current position A, producing the target position B.

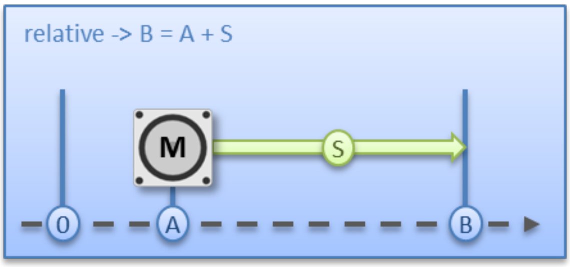


Fig. 214: *Relative positioning*

ENDLESS_PLUS / ENDLESS_MINUS:

The two start types "*ENDLESS_PLUS*" and "*ENDLESS_MINUS*" offer the possibility in the "*Positioning interface*" to specify a direct motor velocity in order to travel endlessly in the positive or negative direction with the specified accelerations.

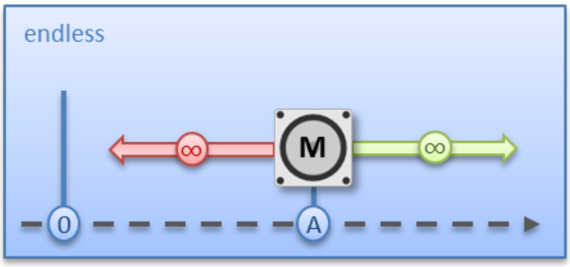


Fig. 215: *Endless travel*

ADDITIVE:

For additive positioning, the position delta S specified by the user is added to the target position E used for the last travel command in order to calculate the target position B.

This kind of positioning resembles the relative positioning, but there is a difference. If the last travel command was completed successfully, the new target position is the same. If there was an error, however, be it that the motor entered a stall state or an "*Emergency stop*" was triggered, the current position is arbitrary and not foreseeable. The user now has the advantage that he can use the last target position for the calculation of the following target position.

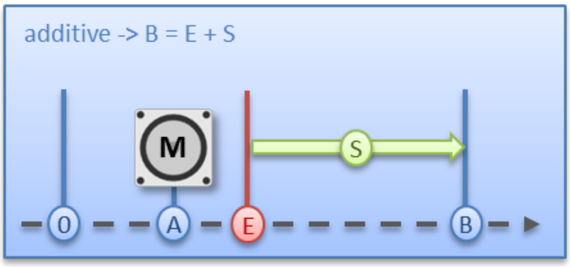


Fig. 216: *Additive positioning*

ABSOLUTE_CHANGE / RELATIVE_CHANGE / ADDITIVE_CHANGE:

These three kinds of positioning are completely identical to those described above. The important difference thereby is that the user uses these commands during an active travel command in order to dynamically specify a new target position.

The same rules and conditions apply as to the “normal” start types. “*ABSOLUTE_CHANGE*” and “*ADDITIVE_CHANGE*” are unique in the calculation of the target position i.e. in absolute positioning an absolute position is specified and in additive positioning a position delta is added to the momentarily active target position.

NOTE

Caution when using the “RELATIVE_CHANGE” positioning

The change by means of "RELATIVE_CHANGE" must be used with caution, since the current position of the motor is also used here as the start position. Due to propagation delays in the system, the position indicated in the PDO never corresponds to the actual position of the motor! Therefore a difference to the desired target position always results in the calculation of the transferred position delta.

Time of the change of the target position

A change of the target position cannot take place at an arbitrary point in time. If the calculation of the output parameters shows that the new target position cannot be readily reached, the command is rejected by the terminal and the “Command rejected [▶ 175]” bit is set. This is the case, for example, at standstill (since the terminal expects a standard positioning here) and in the acceleration phase (since at this point the braking time cannot be calculated yet).

CALI_PLC_CAM / CALI_HW_SYNC / SET_CALIBRATION / SET_CALIBRATION_AUTO / CLEAR_CALIBRATION:

The simplest calibration case is calibration by cam only (connected to one digital input).

Here, the motor travels in the 1st step with velocity 1 (Index 0x8020:09 [▶ 217]) in direction 1 (Index 0x8021:13 [▶ 218]) towards the cam. Subsequently, in the 2nd step, it travels with velocity 2 (Index 0x8020:0A [▶ 217]) in direction 2 (Index 8021:14 [▶ 218]) away from the cam. After the "*In-Target timeout"* (Index 8020:0C [▶ 217]) has elapsed, the calibration position (Index 0x8020:08 [▶ 217]) is taken on by the terminal as the current position.

NOTE

Observe the switching hysteresis of the cam switch

With this simple calibration it must be noted that the position detection of the cam is only exact to a certain degree. The digital inputs are not interrupt-controlled and are “only” polled. The internal propagation delays may therefore result in a system-related position difference.

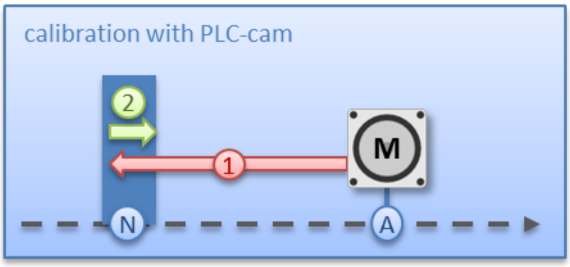


Fig. 217: *Calibration with cam*

For a more precise calibration, an HW sync pulse (C-track) is used in addition to the cam. This calibration proceeds in exactly the same way as described above, up to the point at which the motor travels away from the cam. The travel is not stopped immediately; instead, the sync pulse is awaited. Subsequently, the "*In-Target timeout*" runs down again and the calibration position is taken on by the terminal as the current position.

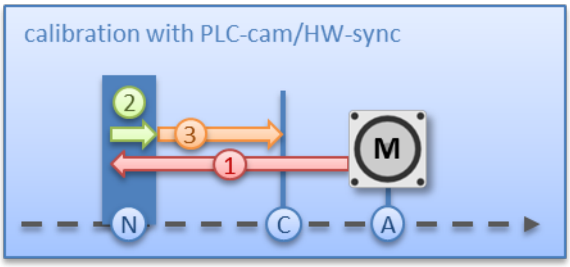


Fig. 218: *Calibration with cam and C-track*

If calibration by hardware is not possible due to the circumstances of the application, the user can also set the "*Calibrated*" bit manually or automatically. The manual setting or deletion takes place with the commands "*SET_CALIBRATION*" and "*CLEAR_CALIBRATION*".

It is simpler, however, if the standard start types (Index 0x8021:01 [▸ 218]) are set to "*SET_CALIBRATION_AUTO*". The "*Calibrated*" bit will now be set automatically by the first rising edge on "*Enable*". The command is conceived only for this purpose; therefore, it does not make sense to use it via the synchronous data exchange.

MODULO:

The modulo position of the axis is a piece of additional information about the absolute axis position. Modulo positioning represents the required target position in a different way. Contrary to the standard types of positioning, the modulo positioning has several pitfalls, since the desired target position can be interpreted differently.

The modulo positioning refers in principle to the "*Modulo factor*" (Index 0x8020:0E [▸ 217]), which can be set in the CoE. In the following examples, a rotary axis with a "*Modulo factor*" equivalent to 360 degrees is assumed.

The "*Modulo tolerance window*" (Index 0x8020:0F [▸ 217]) defines a position window around the current modulo target position of the axis. The window width is twice the specified value (set position ± tolerance value). A detailed description of the tolerance window is provided below.

The positioning of an axis is always referenced to its current actual position. The actual position of an axis is normally the target position of the last travel command. Under certain circumstances (incorrect positioning due to the axis stalling, or a very coarse resolution of the connected encoder), however, a position not expected by the user may arise. If this possibility is not considered, subsequent positioning may lead to unexpected behaviour.

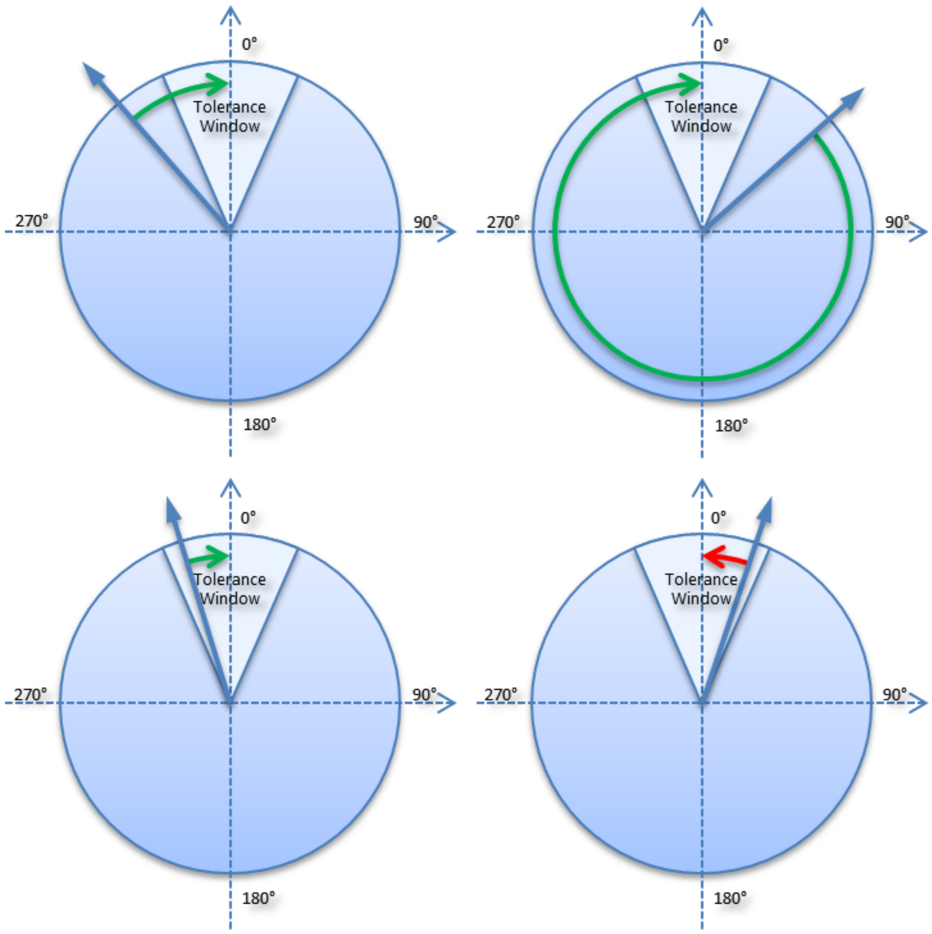


Fig. 219: *Effect of the modulo tolerance window - modulo target position 0° in positive direction*

Example:

An axis is positioned to 0°, with the result that subsequently the actual position of the axis is exactly 0°. A further modulo travel command to 360° in *positive direction* results in a full turn, with the subsequent modulo position of the axis of once again being exactly 0°. If the axis comes to a stop somewhat in front of or behind the target position for mechanical reasons, the next travel command does not behave as one would expect. If the actual position lies slightly below 0° (see fig. 9, below left), a new travel command to 0° in the *positive direction* leads only to a minimal movement. The deviation that arose beforehand is compensated and the position is subsequently exactly 0° once more. If the position lies slightly above 0°, however, the same travel command leads to a full revolution in order to reach the exact position of 0° again. This problem occurs if complete turns by 360° or multiples of 360° were initiated. For positioning to an angle that is significantly different from the current modulo position, the travel command is unambiguous.

In order to solve the problem, a “*Modulo tolerance window*” (Index 0x8020:0F [▶ 217]) can be parameterized. This ensures that small deviations from the position that are within the window do not lead to different axis behavior. If, for example, a window of 1° is parameterized, in the case described above the axis will behave identically, as long the actual position is between 359° and 1°. If the position exceeds 0° by less than 1°, the axis is re-positioned in *positive direction* at a modulo start. In both cases, a target position of 0° therefore leads to minimum movement to exactly 0°. A target position of 360° leads to a full turn in both cases.

For values that are within the window range, the modulo tolerance window can therefore lead to movements against the specified direction. For small windows this is usually not a problem, because system deviations between set and actual position are compensated in both directions. This means that the tolerance window may also be used for axes that may only be moved in one direction due to their construction.

Modulo positioning by less than one turn

Modulo positioning from a starting position to a non-identical target position is unambiguous and requires no special consideration. A modulo target position in the range [0 ≤; position < 360] reaches the required target in less than one whole turn. No motion occurs if target position and starting position are identical. Target positions of more than 360 ° lead to one or more full turns before the axis travels to the required target position.

For a movement from 270° to 0°, a modulo target position of 0° (not 360°) should therefore be specified, because 360° is outside the basic range and would lead to an additional turn.

The modulo positioning distinguishes between three direction specifications: *positive direction, negative direction* and *along the shortest path* (*MODULO_PLUS, MODULO_MINUS, MODULO_SHORT*). For positioning along the shortest path, target positions of more than 360° are not sensible, because the movement towards the target is always direct. In contrast to positive or negative direction, it is therefore not possible to carry out several turns before the axis moves to the target.

NOTE

Only basic periods of less than 360° are permitted

For modulo positioning with start type "MODULO_SHORT", only modulo target positions within the basic period (e.g. less than 360°) are permitted, otherwise an error is returned.

Positioning without the modulo tolerance window

The Modulo tolerance window” (Index 0x8020:0F [▶ 217]) is always taken into account in the “normal” types of modulo positioning. However, this is less desirable in some situations. In order to eliminate this "disadvantage", the comparable start types "MODULO_SHORT_EXT", "MODULO_PLUS_EXT", "MODULO_MINUS_EXT" and "MODULO_CURRENT_EXT" can be used, which ignore the modulo tolerance window.

The following table shows some positioning examples:

Modulo start type	Absolute start position	Modulo target position	Relative travel path	Absolute end position	Modulo end position
MODULO_PLUS	90°	0°	270°	360°	0°
MODULO_PLUS	90°	360°	630°	720°	0°
MODULO_PLUS	90°	720°	990°	1080°	0°
MODULO_MINUS	90°	0°	-90°	0°	0°
MODULO_MINUS	90°	360°	-450°	-360°	0°
MODULO_MINUS	90°	720°	-810°	-720°	0°
MODULO_SHORT	90°	0°	-90°	0°	0°

Examples of modulo positioning with less than one revolution

Modulo positioning with full turns

In principle, modulo positioning by one or full turns are no different than positioning to an angle that differs from the starting position. No motion occurs if target position and starting position are identical. For a full turn, 360° has to be added to the starting position. The behaviour described in the example [▶ 182] shows that special attention must be paid to positionings with whole revolutions. The following table shows positioning examples for a starting position of approximately 90°. The modulo tolerance window is set to 1° here. Special cases for which the starting position is outside this window are identified.

Modulo start type	Absolute start position	Modulo target position	Relative travel path	Absolute end position	Modulo end position	Note
MODULO_PLUS	90.00°	90.00°	0.00°	90.00°	90.00°	
MODULO_PLUS	90.90°	90.00°	-0.90°	90.00°	90.00°	
MODULO_PLUS	91.10°	90.00°	358.90°	450.00°	90.00°	outside TF
MODULO_PLUS	89.10°	90.00°	0.90°	90.00°	90.00°	
MODULO_PLUS	88.90°	90.00°	1.10°	90.00°	90.00°	outside TF
MODULO_PLUS	90.00°	450.00	360.00°	450.00°	90.00°	
MODULO_PLUS	90.90°	450.00°	359.10°	450.00°	90.00°	
MODULO_PLUS	91.10°	450.00°	718.90°	810.00°	90.00°	outside TF
MODULO_PLUS	89.10°	450.00°	360.90°	450.00°	90.00°	
MODULO_PLUS	88.90°	450.00°	361.10°	450.00°	90.00°	outside TF
MODULO_PLUS	90.00°	810.00	720.00°	810.00°	90.00°	
MODULO_PLUS	90.90°	810.00	719.10°	810.00°	90.00°	
MODULO_PLUS	91.10°	810.00	1078.90°	1170.00°	90.00°	outside TF
MODULO_PLUS	89.10°	810.00	720.90°	810.00°	90.00°	
MODULO_PLUS	88.90°	810.00	721.10°	810.00°	90.00°	outside TF
MODULO_MINUS	90.00°	90.00°	0.00°	90.00°	90.00°	
MODULO_MINUS	90.90°	90.00°	-0.90°	90.00°	90.00°	
MODULO_MINUS	91.10°	90.00°	-1.10°	90.00°	90.00°	outside TF
MODULO_MINUS	89.10°	90.00°	0.90°	90.00°	90.00°	
MODULO_MINUS	88.90°	90.00°	-358.90°	-270.00°	90.00°	outside TF
MODULO_MINUS	90.00°	450.00°	-360.00°	-270.00°	90.00°	
MODULO_MINUS	90.90°	450.00°	-360.90°	-270.00°	90.00°	
MODULO_MINUS	91.10°	450.00°	-361.10°	-270.00°	90.00°	outside TF
MODULO_MINUS	89.10°	450.00°	-359.10°	-270.00°	90.00°	
MODULO_MINUS	88.90°	450.00°	-718.90°	-630.00°	90.00°	outside TF
MODULO_MINUS	90.00°	810.00°	-720.00°	-630.00°	90.00°	
MODULO_MINUS	90.90°	810.00°	-720.90°	-630.00°	90.00°	
MODULO_MINUS	91.10°	810.00°	-721.10°	-630.00°	90.00°	outside TF
MODULO_MINUS	89.10°	810.00°	-719.10°	-630.00°	90.00°	
MODULO_MINUS	88.90°	810.00°	-1078.90°	-990.00°	90.00°	outside TF

Examples of modulo positioning with whole revolutions

Examples of two travel commands with a dynamic change of the target position

Without overrun of the target position

Time	POS Outputs	POS Inputs	Description
t1:	Execute = 1 Target position = 200000 Velocity = 2000 Start type = 0x0001 Acceleration = 1000 Deceleration = 1000	Busy = 1 Accelerate = 1	- Specification of the first parameter - Start of the acceleration phase
t2:		Accelerate = 0	- End of the acceleration phase
t3:	Target position = 100000 Velocity = 1500 Start type = 0x1001 Acceleration = 2000 Deceleration = 2000		- Change of the parameters - Activation by new start types
t4:		Decelerate = 1	- Start of the deceleration phase
t5:	Execute = 0	Busy = 0 In-Target = 1 Decelerate = 0	- End of the deceleration phase - Motor is at the new target position
t6 - t9:			- Absolute travel back to the start position 0

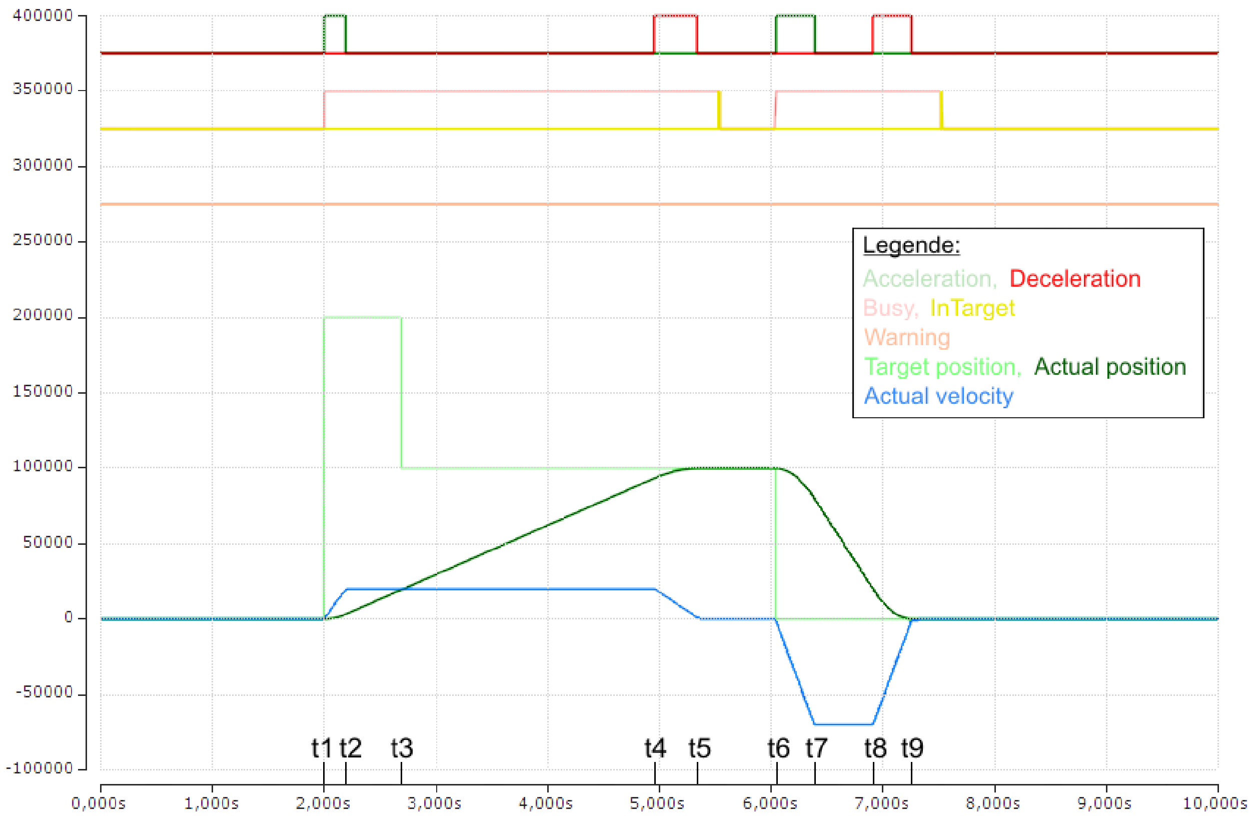


Fig. 220: *Scope recording of a travel command with a dynamic change of the target position, without overrunning the target position*
(The axis scaling refers only to the positions, not to the speed or the status bits)

With overrun of the target position

Time	POS Outputs	POS Inputs	Description
t1:	Execute = 1 Target position = 200000 Velocity = 5000 Start type = 0x0001 Acceleration = 3000 Deceleration = 5000	Busy = 1 Accelerate = 1	- Specification of the 1st parameter - Start of the 1st acceleration phase
t2:		Accelerate = 0	- End of the 1st acceleration phase
t3:	Target position = 100000 Velocity = 1500 Start type = 0x1001 Acceleration = 1000 Deceleration = 2000	Warning = 1 Decelerate = 1	- Change of the parameters - Activation by new start types - Warning of overrunning the target position - Start of the 1st deceleration phase
t4:		Accelerate = 1 Decelerate = 0	- End of the 1st deceleration phase - Start of the 2nd acceleration phase in the opposite direction
t5:		Accelerate = 0 Decelerate = 1	- End of the 2nd acceleration phase - Start of the 2nd deceleration phase
t6:	Execute = 0	Busy = 0 In-Target = 1 Decelerate = 0	- End of the 2nd deceleration phase - Motor is at the new target position
t7 - t10:			- Absolute travel back to the start position 0

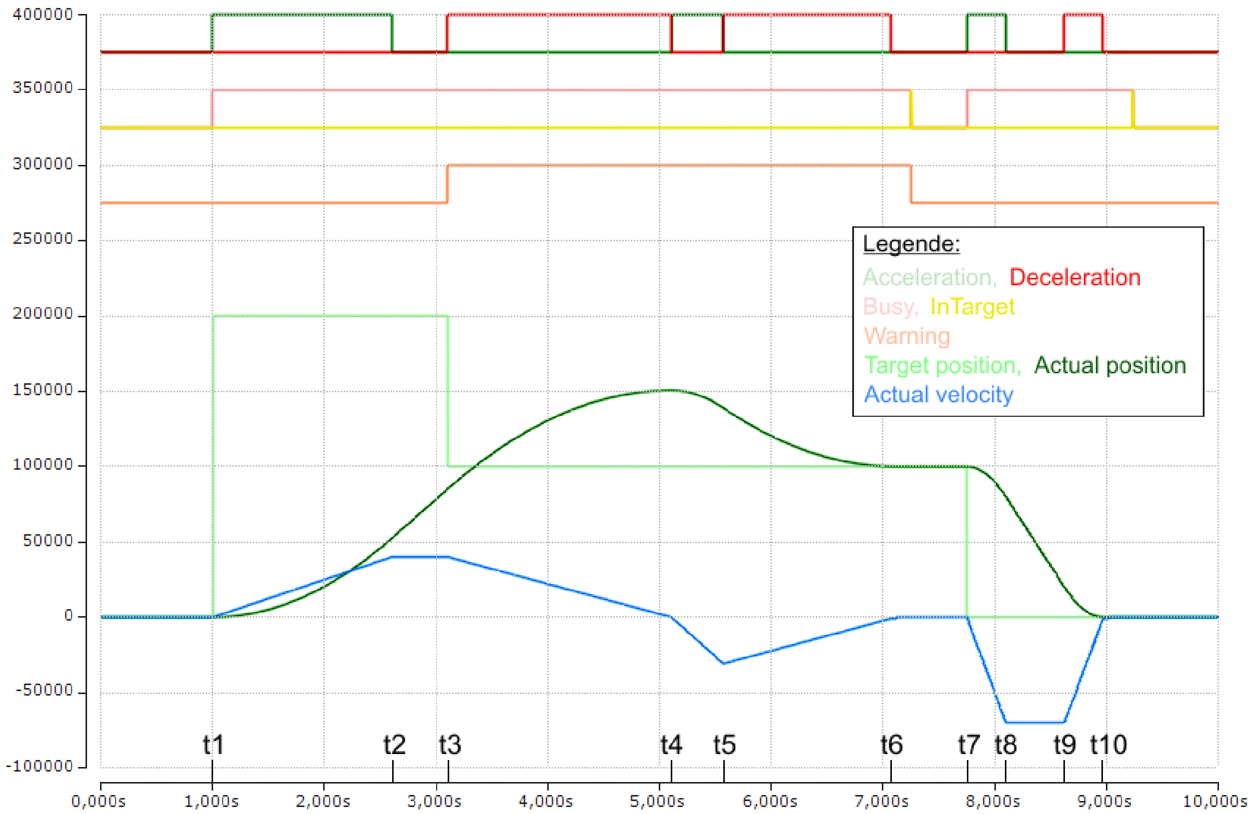


Fig. 221: *Scope recording of a travel command with a dynamic change of the target position, with overrunning of the final target position*
(The axis scaling refers only to the positions, not to the speed or the status bits)

6 Configuration by means of the TwinCAT System Manager

6.1 EL7037

6.1.1 Object description and parameterization - Profile-specific objects

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information [▸ 34] 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

NOTE

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.

Introduction

The CoE overview contains objects for different intended applications:

Object overview

Restore object

Index 1011 Restore default parameters

Index	Name	Meaning	Data type	Flags	Default value
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})

Configuration data

Index 8000 ENC Settings Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8000:0	ENC Settings Ch.1	Maximum subindex	UINT8	RO	0x0E (14_{dec})
8000:08	Disable filter	Deactivates the input filters.	BOOLEAN	RW	0x00 (0_{dec})
8000:0A	Enable micro increments	The lower 8 bits of the counter value are extrapolated.	BOOLEAN	RW	0x00 (0_{dec})
8000:0E	Reversion of rotation	Activates reversion of rotation of the encoder.	BOOLEAN	RW	0x00 (0_{dec})

Index 8010 STM Motor Settings Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8010:0	STM Motor Settings Ch.1		UINT8	RO	0x11 (17_{dec})
8010:01	Maximal current	Maximum permanent motor coil current **Unit**: 1 mA	UINT16	RW	0x1388 (5000_{dec})
8010:02	Reduced current	Reduced coil current **Unit**: 1 mA	UINT16	RW	0x09C4 (2500_{dec})
8010:03	Nominal voltage	Nominal voltage (supply voltage) of the motor **Unit**: 10 mV	UINT16	RW	0x1388 (5000_{dec})
8010:04	Motor coil resistance	Internal resistance of the motor **Unit**: 10 mOhm	UINT16	RW	0x0064 (100_{dec})
8010:05	Motor EMF	Countervoltage of the motor **Unit**: 1 mV / (rad/s)	UINT16	RW	0x0000 (0_{dec})
8010:06	Motor fullsteps	Number of full motor steps	UINT16	RW	0x00C8 (200_{dec})
8010:07	Encoder increments (4-fold)	Number of encoder increments per revolution with quadruple evaluation	UINT16	RW	0x1000 (4096_{dec})
8010:09	Start velocity	Minimum starting velocity of the motor **Unit**: 10000 corresponds to 100% [▶ 161]	UINT16	RW	0x0000 (0_{dec})
8010:0A	Motor coil inductance	Inductance of the motor **Unit**: 0.01 mH	UINT16	RW	0x0000 (0_{dec})
8010:10	Drive on delay time	Delay between activation of driver stage and „ready = 1“	UINT16	RW	0x0064 (100dez)
8010:11	Drive off delay time	Delay between deactivation of driver stage and „ready = 0“	UINT16	RW	0x0096 (150dez)

Index 8011 STM Controller Settings Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8011:0	STM Controller Settings Ch.1	Maximum subindex	UINT8	RO	0x02 (2_{dec})
8011:01	Kp factor (curr.)	Kp control factor of the current controller	UINT16	RW	0x0096 (150_{dec})
8011:02	Ki factor (curr.)	Ki control factor of the current controller	UINT16	RW	0x000A (10_{dec})

Index 8012 STM Features Ch.1 (part 1)

Index	Name	Meaning	Data type	Flags	Default value
8012:0	STM Features Ch.1	Maximum subindex	UINT8	RO	0x3A (58_{dec})
8012:01	Operation mode	permitted values:	BIT4	RW	0x00 (0_{dec})
		0: Automatic			
		1: Velocity direct			
		3: Position controller			
		4: Ext. Velocity mode			
		5: Ext. Position mode			
		6: Velocity sensorless			
8012:05	Speed range	permitted values:	BIT3	RW	0x01 (1_{dec})
		0: 1000 Fullsteps/sec			
		1: 2000 Fullsteps/sec			
		2: 4000 Fullsteps/sec			
		3: 8000 Fullsteps/sec			
		4: 16000 Fullsteps/sec			
		5: 32000 Fullsteps/sec			
8012:08	Feedback type	permitted values:	BIT1	RW	0x01 (1_{dec})
		0: Encoder			
		1: Internal counter			
8012:09	Invert motor polarity	Invert the direction of rotation of the motor	BOOLEAN	RW	0x00 (0_{dec})
8012:0A	Error on step lost	Error on loss of step	BOOLEAN	RW	0x00 (0_{dec})
8012:11	Select info data 1	permitted values:	UINT8	RW	0x0B (11_{dec})
		0: Status word			
		7: Motor velocity			
		11: Motor load			
		13: Motor dc current			
		101: Internal temperature			
		103: Control voltage			
		104: Motor supply voltage			
		150: Drive - Status word			
		151: Drive – State			
		152: Drive - Position lag (low word)			
		153: Drive - Position lag (high word)			

Index 8012 STM Features Ch.1 (part 2)

Index	Name	Meaning	Data type	Flags	Default value
8012:19	Select info data 2	permitted values: 0: Status word 7: Motor velocity 11: Motor load 13: Motor dc current 101: Internal temperature 103: Control voltage 104: Motor supply voltage 150: Drive - Status word 151: Drive - State 152: Drive - Position lag (low word) 153: Drive - Position lag (high word)	UINT8	RW	0x0D (13_{dec})
8012:30	Invert digital input 1	Invert digital input	BOOLEAN	RW	0x00 (0_{dec})
8012:31	Invert digital input 2	Invert digital input	BOOLEAN	RW	0x00 (0_{dec})
8012:32	Function for input 1	permitted values: 0: Normal input 1: Hardware enable 2: PLC cam	BIT4	RW	0x00 (0_{dec})
8012:36	Function for input 2	permitted values: 0: Normal input 1: Hardware enable 2: PLC cam	BIT4	RW	0x00 (0_{dec})
8012:3A	Function for output 1	permitted values: 0: Normal output 1: Break (linked with driver enable) 15: Disabled	BIT4	RW	0x0F (15_{dec})

Index 8014 STM Controller Settings 3 Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8014:0	STM Controller Settings 3 Ch.1	Maximum subindex	UINT8	RO	0x09 (9_{dec})
8014:01	Feed forward (pos.)	Pilot control of the position controller	UINT32	RW	0x000186A0 (100000_{dec})
8014:02	Kp factor (pos.)	Kp control factor of the position controller	UINT16	RW	0x01F4 (500_{dec})
8014:03	Kp factor (velo.)	Kp control factor of the velocity controller **Unit**: 0.1 mA / (rad/s)	UINT32	RW	0x00000032 (50_{dec})
8014:04	Tn (velo.)	Time constant Tn of the velocity controller **Unit**: 0.01 ms	UINT16	RW	0xC350 (50000_{dec})
8014:05	Sensorless param 1	First parameter (sensorless control)	UINT16	RW	0x0000 (0_{dec})
8014:06	Sensorless param 2	Second parameter (sensorless control)	UINT16	RW	0x0000 (0_{dec})
8014:07	Cross over velocity 1	First velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0_{dec})
8014:08	Cross over velocity 2	Second velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0_{dec})
8014:09	Cross over velocity 3	Third velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0_{dec})

Index 8020 POS Settings Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8020:0	POS Settings Ch.1	Maximum subindex	UINT8	RO	0x10 (16_{dec})
8020:01	Velocity min.	Minimum set velocity (range: 0-10000)	INT16	RW	0x0064 (100_{dec})
8020:02	Velocity max.	Maximum set velocity (range: 0-10000)	INT16	RW	0x2710 (10000_{dec})
8020:03	Acceleration pos.	Acceleration in positive direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:04	Acceleration neg.	Acceleration in negative direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:05	Deceleration pos.	Deceleration in positive direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:06	Deceleration neg.	Deceleration in negative direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:07	Emergency deceleration	Emergency deceleration (both directions of rotation) **Unit**: 1 ms	UINT16	RW	0x0064 (100_{dec})
8020:08	Calibration position	Calibration position	UINT32	RW	0x00000000 (0_{dec})
8020:09	Calibration velocity (towards plc cam)	Calibration velocity towards the cam (range: 0-10000)	INT16	RW	0x0064 (100_{dec})
8020:0A	Calibration Velocity (off plc cam)	Calibration velocity away from the cam (range: 0-10000)	INT16	RW	0x000A (10_{dec})
8020:0B	Target window	Target window	UINT16	RW	0x000A (10_{dec})
8020:0C	In-Target timeout	Target position timeout **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:0D	Dead time compensation	Dead time compensation **Unit**: 1 µs	INT16	RW	0x0032 (50_{dec})
8020:0E	Modulo factor	Modulo factor/position	UINT32	RW	0x00000000 (0_{dec})
8020:0F	Modulo tolerance window	Tolerance window for modulo positioning	UINT32	RW	0x00000000 (0_{dec})
8020:10	Position lag max.	Maximum allowable step error	UINT16	RW	0x0000 (0_{dec})

Index 8021 POS Features Ch.1

Index	Name	Meaning	Data type	Flags	Default value
8021:0	POS Features Ch.1	Maximum subindex	UINT8	RO	0x16 (22_{dec})
8021:01	Start type	permitted values:	UINT16	RW	0x0001 (1_{dec})
		0: Idle			
		1: Absolute			
		2: Relative			
		3: Endless plus			
		4: Endless minus			
		6: Additive			
		24832: Calibration (Hardware sync)			
		24576: Calibration (Plc cam)			
		28416: Calibration (Clear manual)			
		28160: Calibration (Set manual)			
		28161: Calibration (Set manual auto)			
		1029: Modulo current			
		773: Modulo minus			
		517: Modulo plus			
		261: Modulo short			
8021:11	Time information	permitted values:	BIT2	RW	0x00 (0_{dec})
		0: Elapsed time current drive time since start of the travel command			
8021:13	Invert calibration cam search direction	Inversion of the direction of rotation towards the cam	BOOLEAN	RW	0x01 (1_{dec})
8021:14	Invert sync impulse search direction	Inversion of the direction of rotation away from the cam	BOOLEAN	RW	0x00 (0_{dec})
8021:15	Emergency stop on position lag error	Triggers an emergency stop if the maximum following error is exceeded	BOOLEAN	RW	0x00 (0_{dec})
8021:16	Enhanced diag history	Provides detailed messages about the status of the positioning interface in the diag history	BOOLEAN	RW	0x00 (0_{dec})

Command object

Index FB00 STM Command

Index	Name	Meaning	Data type	Flags	Default value
FB00:0	STM Command	Maximum subindex	UINT8	RO	0x03 (3_{dec})
FB00:01	Request	Requesting a command 0x8000: Software reset	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Status of the command	UINT8	RO	0x00 (0_{dec})
		0: No error, without return value			
		1: No error, with return value			
		2: With error, without return value			
		3: With error, with return value			
		... reserved			
		255: Command execution active			
FB00:03	Response	Return value of the executed command	OCTET-STRING[4]	RO	{0}

Input data

Index 6000 ENC Inputs Ch.1

Index	Name	Meaning	Data type	Flags	Default value
6000:0	ENC Inputs Ch.1	Maximum subindex	UINT8	RO	0x16 (22_{dec})
6000:01	Latch C valid	The counter value was latched with the C track.	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Latch extern valid	The counter value was stored via the external latch.	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Set counter done	The counter was set.	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Counter underflow	Counter underflow.	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Counter overflow	Counter overflow.	BOOLEAN	RO	0x00 (0_{dec})
6000:08	Extrapolation stall	The extrapolated part of the counter is invalid	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Status of input A	Status of the A-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0A	Status of input B	Status of the B-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0B	Status of input C	Status of the C-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0D	Status of extern latch	Status of the ext. latch input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	0x00 (0_{dec})
6000:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6000:11	Counter value	The counter value.	UINT32	RO	0x00000000 (0_{dec})
6000:12	Latch value	The latch value.	UINT32	RO	0x00000000 (0_{dec})
6000:16	Timestamp	Time stamp of the last counter change.	UINT32	RO	0x00000000 (0_{dec})

Index 6010 STM Inputs Ch.1

Index	Name	Meaning	Data type	Flags	Default value
6010:0	STM Inputs Ch.1	Maximum subindex	UINT8	RO	0x15 (21_{dec})
6010:01	Ready to enable	Driver stage is ready for enabling	BOOLEAN	RO	0x00 (0_{dec})
6010:02	Ready	Driver stage is ready for operation	BOOLEAN	RO	0x00 (0_{dec})
6010:03	Warning	a warning has occurred (see index A010 [▸ 198]	BOOLEAN	RO	0x00 (0_{dec})
6010:04	Error	an error has occurred (see index A010 [▸ 198]))	BOOLEAN	RO	0x00 (0_{dec})
6010:05	Moving positive	Driver stage is activated in positive direction	BOOLEAN	RO	0x00 (0_{dec})
6010:06	Moving negative	Driver stage is activated in negative direction	BOOLEAN	RO	0x00 (0_{dec})
6010:07	Torque reduced	Reduced torque is active	BOOLEAN	RO	0x00 (0_{dec})
6010:08	Motor stall	A loss of step has occurred	BOOLEAN	RO	0x00 (0_{dec})
6010:0C	Digital input 1	Digital input 1	BOOLEAN	RO	0x00 (0_{dec})
6010:0D	Digital input 2	Digital input 2	BOOLEAN	RO	0x00 (0_{dec})
6010:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	0x00 (0_{dec})
6010:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6010:11	Info data 1	Synchronous information (selection via subindex 8012:11 [▸ 189])	UINT16	RO	0x0000 (0_{dec})
6010:12	Info data 2	Synchronous information (selection via subindex 8012:19 [▸ 189])	UINT16	RO	0x0000 (0_{dec})
6010:13	Motor load	Current motor load **Unit:** 0.01°	INT16	RO	0x0000 (0_{dec})
6010:14	Internal position	Internal microstep position	UINT32	RO	0x00000000 (0_{dec})
6010:15	External position	Encoder position	UINT32	RO	0x00000000 (0_{dec})

Index 6020 POS Inputs Ch.1

Index	Name	Meaning	Data type	Flags	Default value
6020:0	POS Inputs Ch.1	Maximum subindex	UINT8	RO	0x22 (34_{dec})
6020:01	Busy	A current travel command is active	BOOLEAN	RO	0x00 (0_{dec})
6020:02	In-Target	Motor has arrived at target	BOOLEAN	RO	0x00 (0_{dec})
6020:03	Warning	A warning has occurred	BOOLEAN	RO	0x00 (0_{dec})
6020:04	Error	an error has occurred	BOOLEAN	RO	0x00 (0_{dec})
6020:05	Calibrated	Motor is calibrated	BOOLEAN	RO	0x00 (0_{dec})
6020:06	Accelerate	Motor is in the acceleration phase	BOOLEAN	RO	0x00 (0_{dec})
6020:07	Decelerate	Motor is in the deceleration phase	BOOLEAN	RO	0x00 (0_{dec})
6020:11	Actual position	Current target position of the travel command generator	UINT32	RO	0x00000000 (0_{dec})
6020:21	Actual velocity	Current set velocity of the travel command generator	INT16	RO	0x0000 (0_{dec})
6020:22	Actual drive time	Travel command time information (see subindex 8021:11 [▸ 192])	UINT32	RO	0x00000000 (0_{dec})
6020:23	Actual position lag	Lag of position	UINT32	RO	0x00000000 (0_{dec})

Output data

Index 7000 ENC Outputs (compact) Ch.1

Index	Name	Meaning	Data type	Flags	Default value
7000:0	ENC Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:01	Enable latch C	Activate latching via the C-track.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Enable latch extern on positive edge	Activate external latch with positive edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Set counter	Set the counter value.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Enable latch extern on negative edge	Activate external latch with negative edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:11	Set counter value	This is the counter value to be set via "Set counter".	UINT16	RO	0x0000 (0_{dec})

Index 7000 ENC Outputs Ch.1

Index	Name	Meaning	Data type	Flags	Default value
7000:0	ENC Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:01	Enable latch C	Activate latching via the C-track.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Enable latch extern on positive edge	Activate external latch with positive edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Set counter	Set the counter value.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Enable latch extern on negative edge	Activate external latch with negative edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:11	Set counter value	This is the counter value to be set via "Set counter".	UINT32	RO	0x00000000 (0_{dec})

Index 7010 STM Outputs Ch.1

Index	Name	Meaning	Data type	Flags	Default value
7010:0	STM Outputs Ch.1	Maximum subindex	UINT8	RO	0x21 (33_{dec})
7010:01	Enable	activates the output stage	BOOLEAN	RO	0x00 (0_{dec})
7010:02	Reset	all errors that may have occurred are reset by setting this bit (rising edge)	BOOLEAN	RO	0x00 (0_{dec})
7010:03	Reduce torque	Reduced torque (coil current) is active (see subindex 8010:02 [▸ 188])	BOOLEAN	RO	0x00 (0_{dec})
7010:0C	Digital output 1	Digital output 1	BOOLEAN	RO	0x00 (0_{dec})
7010:11	Position	Set position specification **Unit:** Increments [▸ 163]	UINT32	RO	0x00000000 (0_{dec})
7010:21	Velocity	Set velocity specification **Unit:** +/- 32767 corresponds to +/- 100% [▸ 161]	INT16	RO	0x0000 (0_{dec})

Index 7020 POS Outputs Ch.1 (part 1)

Index	Name	Meaning	Data type	Flags	Default value
7020:0	POS Outputs Ch.1	Maximum subindex	UINT8	RO	0x24 (33_{dec})
7020:01	Execute	Start travel command (rising edge), or prematurely abort travel command (falling edge)	BOOLEAN	RO	0x00 (0_{dec})
7020:02	Emergency Stop	Prematurely abort travel command with an emergency ramp (rising edge)	BOOLEAN	RO	0x00 (0_{dec})
7020:11	Target position	Specification of the target position	UINT32	RO	0x00000000 (0_{dec})
7020:21	Velocity	Specification of the maximum set velocity	INT16	RO	0x0000 (0_{dec})

Index 7020 POS Outputs Ch.1 (part 2)

Index	Name	Meaning	Data type	Flags	Default value
7020:22	Start type				
	0x0000 Idle: No travel command is being executed		UINT16	RO	0x0000 (0$_{dec}$)
	0x0001 Absolute: Absolute target position		UINT16	RO	0x0000 (0$_{dec}$)
	0x1001 Absolute (Change): Change during an active travel command		UINT16	RO	0x0000 (0$_{dec}$)
	0x0002 Relative: Target position relative to the current position		UINT16	RO	0x0000 (0$_{dec}$)
	0x1002 Relative (Change): Change during an active travel command		UINT16	RO	0x0000 (0$_{dec}$)
	0x0003 Endless plus: Endless driving in positive direction of rotation		UINT16	RO	0x0000 (0$_{dec}$)
	0x0004 Endless minus: Endless driving in negative direction of rotation		UINT16	RO	0x0000 (0$_{dec}$)
	0x0105 Modulo short: Shortest distance to the next modulo position		UINT16	RO	0x0000 (0$_{dec}$)
	0x0115 Modulo short extended: Shortest distance to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0$_{dec}$)
	0x0205 Modulo plus: Drive in positive direction of rotation to the next modulo position		UINT16	RO	0x0000 (0$_{dec}$)
	0x0215 Modulo plus extended: Drive in positive direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0$_{dec}$)
	0x0305 Modulo minus: Drive in negative direction of rotation to the next modulo position		UINT16	RO	0x0000 (0$_{dec}$)
	0x0315 Modulo minus extended: Drive in negative direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0$_{dec}$)
	0x0405 Modulo current: Drive in the last implemented direction of rotation to the next modulo position		UINT16	RO	0x0000 (0$_{dec}$)
	0x0415 Modulo current extended: Drive in the last implemented direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0$_{dec}$)
	0x0006 Additive: New target position relative/additive to the last target position		UINT16	RO	0x0000 (0$_{dec}$)
	0x1006 Additive (Change): Change during an active travel command		UINT16	RO	0x0000 (0$_{dec}$)
	0x6000 Calibration, PLC cam: Calibration with cam		UINT16	RO	0x0000 (0$_{dec}$)
	0x6100 Calibration, HW sync: Calibration with cam and C-track		UINT16	RO	0x0000 (0$_{dec}$)
	0x6E00 Calibration, set manual: Set calibration manually		UINT16	RO	0x0000 (0$_{dec}$)
	0x6E01 Calibration, set manual auto: Set automatic calibration, for "Enable = 1"		UINT16	RO	0x0000 (0$_{dec}$)
	0x6F00 Calibration, clear manual: Clear calibration manually		UINT16	RO	0x0000 (0$_{dec}$)
7020:23	Acceleration	Acceleration specification	UINT16	RO	0x0000 (0$_{dec}$)
7020:24	Deceleration	Deceleration specification	UINT16	RO	0x0000 (0$_{dec}$)

Index 7021 POS Outputs 2 Ch.1 (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7020:0	POS Outputs Ch.1	Maximum subindex	UINT8	RO	0x24 (36$_{dec}$)
7020:03	Enable auto start	Enable auto start	BOOLEAN	RO	0x00 (0$_{dec}$)
7020:11	Target position	Specification of the target position	UINT32	RO	0x00000000 (0$_{dec}$)
7020:21	Velocity	Specification of the maximum set velocity	INT16	RO	0x0000 (0$_{dec}$)

Index 7021 POS Outputs 2 Ch.1 (part 2)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7021:22	Start type				
	0x0000 Idle: No travel command is being executed		UINT16	RO	0x0000 (0_{dec})
	0x0001 Absolute: Absolute target position		UINT16	RO	0x0000 (0_{dec})
	0x1001 Absolute (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0002 Relative: Target position relative to the current position		UINT16	RO	0x0000 (0_{dec})
	0x1002 Relative (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0003 Endless plus: Endless driving in positive direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0004 Endless minus: Endless driving in negative direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0105 Modulo short: Shortest distance to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0115 Modulo short extended: Shortest distance to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0205 Modulo plus: Drive in positive direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0215 Modulo plus extended: Drive in positive direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0305 Modulo minus: Drive in negative direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0315 Modulo minus extended: Drive in negative direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0405 Modulo current: Drive in the last implemented direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0415 Modulo current extended: Drive in the last implemented direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0006 Additive: New target position relative/additive to the last target position		UINT16	RO	0x0000 (0_{dec})
	0x1006 Additive (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x6000 Calibration, PLC cam: Calibration with cam		UINT16	RO	0x0000 (0_{dec})
	0x6100 Calibration, HW sync: Calibration with cam and C-track		UINT16	RO	0x0000 (0_{dec})
	0x6E00 Calibration, set manual: Set calibration manually		UINT16	RO	0x0000 (0_{dec})
	0x6E01 Calibration, set manual auto: Set automatic calibration, for "Enable = 1"		UINT16	RO	0x0000 (0_{dec})
	0x6F00 Calibration, clear manual: Clear calibration manually		UINT16	RO	0x0000 (0_{dec})
7021:23	Acceleration	Acceleration specification	UINT16	RO	0x0000 (0_{dec})
7021:24	Deceleration	Deceleration specification	UINT16	RO	0x0000 (0_{dec})

Information / diagnostic data (channel specific)

Index 9010 STM Info data Ch.1

Index	Name	Meaning	Data type	Flags	Default value
9010:0	STM Info data Ch.1	Maximum subindex	UINT8	RO	0x13 (19_{dec})
9010:01	Status word	Status word (see index A010 [▶ 198])	UINT16	RO	0x0000 (0_{dec})
9010:08	Motor velocity	Current motor velocity	INT16	RO	0x0000 (0_{dec})
9010:09	Internal position	Internal position (micro increments)	UINT32	RO	0x00000000 (0_{dec})
9010:0B	Motor load	Current motor load **Unit**: 0.01°	INT16	RO	0x0000 (0_{dec})
9010:0D	Motor dc current	Current motor current (DC vector) **Unit**: 1 mA	INT16	RO	0x0000 (0_{dec})
9010:0E	Tn (curr.)	Internally calculated time constant of the current controller **Unit**: 0.01 ms	UINT16	RO	0x0000 (0_{dec})
9010:13	External position	External position (connected encoder)	UINT32	RO	0x00000000 (0_{dec})

Index 9020 POS Info data Ch.1

Index	Name	Meaning	Data type	Flags	Default value
9020:0	POS Info data Ch.1	Maximum subindex	UINT8	RO	0x04 (4_{dec})
9020:01	Status word	Status word	UINT16	RO	0x0000 (0_{dec})
9020:03	State (drive controller)	permitted values:	UINT16	RO	0x0000 (0_{dec})
		0: Init			
		1: Idle			
		272: Go cam			
		273: On cam			
		16: Start			
		17: Acceleration			
		18: Constant			
		19: Deceleration			
		288: Go sync impulse			
		289: Leave cam			
		4096: Pre target			
		4097: In target			
		32: Emergency Stop			
		33: Normal stop			
		304: Calibration stop			
		8192: Drive end			
		8193: Wait for init			
		320: Is calibrated			
		321: Not calibrated			
		16384: Drive warning			
		32768: Error			
		65535: Undefined			
		256: Calibration start			
9020:04	Actual position lag	Current step error	INT32	RO	0x00000000 (0_{dec})

Index A010 STM Diag data Ch.1

Index	Name	Meaning	Data type	Flags	Default value
A010:0	STM Diag data Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
A010:01	Saturated	Driver stage operates with maximum duty cycle	BOOLEAN	RO	0x00 (0_{dec})
A010:02	Over temperature	Internal terminal temperature is greater than 80 °C	BOOLEAN	RO	0x00 (0_{dec})
A010:03	Torque overload	Duty cycle output at 100%	BOOLEAN	RO	0x00 (0_{dec})
A010:04	Under voltage	Supply voltage less than 7 V	BOOLEAN	RO	0x00 (0_{dec})
A010:05	Over voltage	Supply voltage 10 % higher than the nominal voltage (see 8010:03 [▶ 188])	BOOLEAN	RO	0x00 (0_{dec})
A010:06	Short circuit	Short circuit of motor coil	BOOLEAN	RO	0x00 (0_{dec})
A010:08	No control power	No power supply to driver stage	BOOLEAN	RO	0x00 (0_{dec})
A010:09	Misc error	• Initialization failed or • Internal terminal temperature is higher than 100 °C (see F80F:05 [▶ 199])	BOOLEAN	RO	0x00 (0_{dec})
A010:0A	Configuration	CoE change has not yet been adopted into the current configuration	BOOLEAN	RO	0x00 (0_{dec})
A010:0B	Motor stall	A loss of step has occurred	BOOLEAN	RO	0x00 (0_{dec})
A010:11	Actual operation mode	permitted values:	BIT4	RO	0x00 (0_{dec})
		0: Automatic			
		1: Velocity direct			
		2: Velocity controller			
		3: Position controller			
		4: Ext. Velocity mode			
		5: Ext. Position mode			
		6: Velocity sensorless			

Index A020 POS Diag data Ch.1

Index	Name	Meaning	Data type	Flags	Default value
A020:0	POS Diag data Ch.1	Maximum subindex	UINT8	RO	0x06 (6_{dec})
A020:01	Command rejected	Travel command was rejected	BOOLEAN	RO	0x00 (0_{dec})
A020:02	Command aborted	Travel command was aborted	BOOLEAN	RO	0x00 (0_{dec})
A020:03	Target overrun	Target position was overrun in the opposite direction	BOOLEAN	RO	0x00 (0_{dec})
A020:04	Target timeout	The target window was not reached within the in-target timeout	BOOLEAN	RO	0x00 (0_{dec})
A020:05	Position lag	The maximum following error was exceeded	BOOLEAN	RO	0x00 (0_{dec})
A020:06	Emergency Stop	An emergency stop was triggered (automatic or manual)	BOOLEAN	RO	0x00 (0_{dec})

Manufacturer configuration data (device-specific)

Index F80F STM Vendor data

Index	Name	Meaning	Data type	Flags	Default value
F80F:0	STM Vendor data	Maximum subindex	UINT8	RO	0x05 (5_{dec})
F80F:04	Warning temperature	Temperature warning threshold **Unit**: 1 °C	INT8	RW	0x50 (80_{dec})
F80F:05	Switch off temperature	Switch-off temperature **Unit**: 1 °C	INT8	RW	0x64 (100_{dec})

Information / diagnostic data (device-specific)

Index F900 STM Info data

Index	Name	Meaning	Data type	Flags	Default value
F900:0	STM Info data	Maximum subindex	UINT8	RO	0x06 (6_{dec})
F900:01	Software version (driver)	Software version of the output driver	STRING	RO	
F900:02	Internal temperature	Internal terminal temperature **Unit**: 1 °C	INT8	RO	0x00 (0_{dec})
F900:04	Control voltage	Control voltage **Unit**: 1 mV, 10 mV with field-oriented control	UINT16	RO	0x0000 (0_{dec})
F900:05	Motor supply voltage	Motor supply voltage **Unit**: 1 mV, 10 mV with field-oriented control	UINT16	RO	0x0000 (0_{dec})
F900:06	Cycle time	Current EtherCAT cycle time **Unit**: 1 µs	UINT16	RO	0x0000 (0_{dec})

Index F010 Module list

Index	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x03 (3_{dec})
F010:01	SubIndex 001	Encoder profile number	UINT32	RW	0x000001FF (511_{dec})
F010:02	SubIndex 002	Stepper motor profile number	UINT32	RW	0x000002BF (703_{dec})
F010:03	SubIndex 003	Positioning interface profile number	UINT32	RW	0x000002C0 (704_{dec})

Index F081 Download revision

Index	Name	Meaning	Data type	Flags	Default value
F081:0	Download revision	Maximum subindex	UINT8	RO	0x01 (1_{dec})
F081:01	Revision number	Revision number	UINT32	RW	0x00000000 (0_{dec})

Index FB40 Memory interface

Index	Name	Meaning	Data type	Flags	Default value
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}

6.1.2 Object description and parameterization - standard objects

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.

Standard objects (0x1000-0x1FFF)

Index 1000 Device type

Index	Name	Meaning	Data type	Flags	Default value
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	Name	Meaning	Data type	Flags	Default value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL7037

Index 1009 Hardware version

Index	Name	Meaning	Data type	Flags	Default value
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	

Index 100A Software version

Index	Name	Meaning	Data type	Flags	Default value
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

Index 1018 Identity

Index	Name	Meaning	Data type	Flags	Default value
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	0x1B873052 (461844562_{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	Name	Meaning	Data type	Flags	Default value
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 10F3 Diagnosis History

Index	Name	Meaning	Data type	Flags	Default value
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	Name	Meaning	Data type	Flags	Default value
10F8:0	Actual Time Stamp	Timestamp	UINT64	RO	

Index 1400 ENC RxPDO-Par Control compact

Index	Name	Meaning	Data type	Flags	Default value
1400:0	ENC RxPDO-Par Control compact	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[6]	RO	01 16 00 00 00 00

Index 1401 ENC RxPDO-Par Control

Index	Name	Meaning	Data type	Flags	Default value
1401:0	ENC RxPDO-Par Control	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[6]	RO	00 16 00 00 00 00

Index 1403 STM RxPDO-Par Position

Index	Name	Meaning	Data type	Flags	Default value
1403:0	STM RxPDO-Par Position	PDO Parameter RxPDO 4	UINT8	RO	0x06 (6_{dec})
1403:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 4	OCTET-STRING[6]	RO	04 16 05 16 06 16

Index 1404 STM RxPDO-Par Velocity

Index	Name	Meaning	Data type	Flags	Default value
1404:0	STM RxPDO-Par Velocity	PDO Parameter RxPDO 5	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5	OCTET-STRING[6]	RO	03 16 05 16 06 16

Index 1405 POS RxPDO-Par Control compact

Index	Name	Meaning	Data type	Flags	Default value
1405:0	POS RxPDO-Par Control compact	PDO Parameter RxPDO 6	UINT8	RO	0x06 (6_{dec})
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6	OCTET-STRING[6]	RO	03 16 04 16 06 16

Index 1406 POS RxPDO-Par Control

Index	Name	Meaning	Data type	Flags	Default value
1406:0	POS RxPDO-Par Control	PDO Parameter RxPDO 7	UINT8	RO	0x06 (6_{dec})
1406:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 7	OCTET-STRING[6]	RO	03 16 04 16 05 16

Index 1407 POS RxPDO-Par Control 2

Index	Name	Meaning	Data type	Flags	Default
1407:0	POS RxPDO-Par Control 2	PDO Parameter RxPDO 8	UINT8	RO	0x06 (6_{dec})
1407:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 8	OCTET-STRING[6]	RO	03 16 04 16 05 16

Index 1600 ENC RxPDO-Map Control compact

Index	Name	Meaning	Data type	Flags	Default value
1600:0	ENC RxPDO-Map Control compact	PDO Mapping RxPDO 1	UINT8	RO	0x06 (6_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (1 bits align)	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set counter))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (13 bits align)	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x0000:00, 12
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 16

Index 1601 ENC RxPDO-Map Control

Index	Name	Meaning	Data type	Flags	Default value
1601:0	ENC RxPDO-Map Control	PDO Mapping RxPDO 2	UINT8	RO	0x06 (6_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (1 bits align)	UINT32	RO	0x7000:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set counter))	UINT32	RO	0x7000:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (13 bits align)	UINT32	RO	0x7000:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x0000:00, 12
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 32

Index 1602 STM RxPDO-Map Control

Index	Name	Meaning	Data type	Flags	Default value
1602:0	STM RxPDO-Map Control	PDO Mapping RxPDO 3	UINT8	RO	0x06 (6_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x01 (Enable))	UINT32	RO	0x7010:01, 1
1602:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x02 (Reset))	UINT32	RO	0x7010:02, 1
1602:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x03 (Reduce torque))	UINT32	RO	0x7010:03, 1
1602:04	SubIndex 004	4. PDO Mapping entry (13 bits align)	UINT32	RO	0x0000:00, 8
1602:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x0C (Digital output 1))	UINT32	RO	0x7010:0C, 1
1602:06	SubIndex 006	6. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4

Index 1603 STM RxPDO-Map Position

Index	Name	Meaning	Data type	Flags	Default value
1603:0	STM RxPDO-Map Position	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x11 (Position))	UINT32	RO	0x7010:11, 32

Index 1604 STM RxPDO-Map Velocity

Index	Name	Meaning	Data type	Flags	Default value
1604:0	STM RxPDO-Map Velocity	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7010:21, 16

Index 1605 POS RxPDO-Map Control compact

Index	Name	Meaning	Data type	Flags	Default value
1605:0	POS RxPDO-Map Control compact	PDO Mapping RxPDO 6	UINT8	RO	0x04 (4_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x01 (Execute))	UINT32	RO	0x7020:01, 1
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x02 (Emergency stop))	UINT32	RO	0x7020:02, 1
1605:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7020:11, 32

Index 1606 POS RxPDO-Map Control

Index	Name	Meaning	Data type	Flags	Default value
1606:0	POS RxPDO-Map Control	PDO Mapping RxPDO 7	UINT8	RO	0x08 (8_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x01 (Execute))	UINT32	RO	0x7020:01, 1
1606:02	SubIndex 002	2. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x02 (Emergency stop))	UINT32	RO	0x7020:02, 1
1606:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1606:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7020:11, 32
1606:05	SubIndex 005	5. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7020:21, 16
1606:06	SubIndex 006	6. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x22 (Start type))	UINT32	RO	0x7020:22, 16
1606:07	SubIndex 007	7. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x23 (Acceleration))	UINT32	RO	0x7020:23, 16
1606:08	SubIndex 008	8. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x24 (Deceleration))	UINT32	RO	0x7020:24, 16

Index 1607 POS RxPDO-Map Control 2

Index	Name	Meaning	Data type	Flags	Default
1606:0	POS RxPDO-Map Control	PDO Mapping RxPDO 7	UINT8	RO	0x08 (8_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00,2
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7021 (POS Outputs 2 Ch.1), entry 0x03 (Enable auto start))	UINT32	RO	0x7021:03, 1
1607:03	SubIndex 003	3. PDO Mapping entry (13 bits align)	UINT32	RO	0x0000:00, 13
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7021:11, 32
1607:05	SubIndex 005	5. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7021:21, 16
1607:06	SubIndex 006	6. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x22 (Start type))	UINT32	RO	0x7021:22, 16
1607:07	SubIndex 007	7. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x23 (Acceleration))	UINT32	RO	0x7021:23, 16
1607:08	SubIndex 008	8. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x24 (Deceleration))	UINT32	RO	0x7021:24, 16

Index 1800 ENC TxPDO-Par Status compact

Index	Name	Meaning	Data type	Flags	Default value
1800:0	ENC TxPDO-Par Status compact	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[2]	RO	01 1A

Index 1801 ENC TxPDO-Par Status

Index	Name	Meaning	Data type	Flags	Default value
1801:0	ENC TxPDO-Par Status	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2	OCTET-STRING[2]	RO	00 1A

Index 1806 POS TxPDO-Par Status compact

Index	Name	Meaning	Data type	Flags	Default value
1806:0	POS TxPDO-Par Status compact	PDO parameter TxPDO 7	UINT8	RO	0x06 (6_{dec})
1806:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 7	OCTET-STRING[2]	RO	07 1A

Index 1807 POS TxPDO-Par Status

Index	Name	Meaning	Data type	Flags	Default value
1807:0	POS TxPDO-Par Status	PDO parameter TxPDO 8	UINT8	RO	0x06 (6_{dec})
1807:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 8	OCTET-STRING[2]	RO	06 1A

Index 1A00 ENC TxPDO-Map Status compact

Index	Name	Meaning	Data type	Flags	Default value
1A00:0	ENC TxPDO-Map Status compact	PDO Mapping TxPDO 1	UINT8	RO	0x11 (17_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6000:0D, 1
1A00:0D	SubIndex 013	13. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x0000:00, 1
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:10, 1
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:11, 16
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 16

Index 1A01 ENC TxPDO-Map Status

Index	Name	Meaning	Data type	Flags	Default value
1A01:0	ENC TxPDO-Map Status	PDO Mapping TxPDO 2	UINT8	RO	0x11 (17_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A01:02	SubIndex 002	2. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:02, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A01:0B	SubIndex 011	11. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 1
1A01:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6000:0D, 1
1A01:0D	SubIndex 013	13. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A01:0E	SubIndex 014	14. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x0000:00, 1
1A01:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:10, 1
1A01:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:11, 32
1A01:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 32

Index 1A02 ENC TxPDO-Map Timest. compact

Index	Name	Meaning	Data type	Flags	Default value
1A02:0	ENC TxPDO-Map Timest. compact	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 32

Index 1A03 STM TxPDO-Map Status

Index	Name	Meaning	Data type	Flags	Default value
1A03:0	STM TxPDO-Map Status	PDO Mapping TxPDO 4	UINT8	RO	0x0E (14_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x01 (Ready to enable))	UINT32	RO	0x6010:01, 1
1A03:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x02 (Ready))	UINT32	RO	0x6010:02, 1
1A03:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6010:03, 1
1A03:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6010:04, 1
1A03:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x05 (Moving positive))	UINT32	RO	0x6010:05, 1
1A03:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x06 (Moving negative))	UINT32	RO	0x6010:06, 1
1A03:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x07 (Torque reduced))	UINT32	RO	0x6010:07, 1
1A03:08	SubIndex 008	8. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x08 (Motor stall))	UINT32	RO	0x6010:08, 1
1A03:09	SubIndex 009	9. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 3
1A03:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6010:0C, 1
1A03:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x6010:0D, 1
1A03:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6010:0E, 1
1A03:0D	SubIndex 013	13. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A03:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6010:10, 1

Index 1A04 STM TxPDO-Map Synchron info data

Index	Name	Meaning	Data type	Flags	Default value
1A04:0	STM TxPDO-Map Synchron info data	PDO Mapping TxPDO 5	UINT8	RO	0x02 (2_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x11 (Info data 1))	UINT32	RO	0x6010:11, 16
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x12 (Info data 2))	UINT32	RO	0x6010:12, 16

Index 1A05 STM TxPDO-Map Motor load

Index	Name	Meaning	Data type	Flags	Default value
1A05:0	STM TxPDO-Map Motor load	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x13 (Motor load))	UINT32	RO	0x6010:13, 16

Index 1A06 POS TxPDO-Map Status compact

Index	Name	Meaning	Data type	Flags	Default value
1A06:0	POS TxPDO-Map Status compact	PDO Mapping TxPDO 7	UINT8	RO	0x08 (8_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x01 (Busy))	UINT32	RO	0x6020:01, 1
1A06:02	SubIndex 002	2. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x02 (In-Target))	UINT32	RO	0x6020:02, 1
1A06:03	SubIndex 003	3. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6020:03, 1
1A06:04	SubIndex 004	4. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6020:04, 1
1A06:05	SubIndex 005	5. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x05 (Calibrated))	UINT32	RO	0x6020:05, 1
1A06:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x06 (Accelerate))	UINT32	RO	0x6020:06, 1
1A06:07	SubIndex 007	7. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x07 (Decelerate))	UINT32	RO	0x6020:07, 1
1A06:08	SubIndex 008	8. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9

Index 1A07 POS TxPDO-Map Status

Index	Name	Meaning	Data type	Flags	Default value
1A07:0	POS TxPDO-Map Status	PDO Mapping TxPDO 8	UINT8	RO	0x0B (11_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x01 (Busy))	UINT32	RO	0x6020:01, 1
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x02 (In-Target))	UINT32	RO	0x6020:02, 1
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6020:03, 1
1A07:04	SubIndex 004	4. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6020:04, 1
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x05 (Calibrated))	UINT32	RO	0x6020:05, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x06 (Accelerate))	UINT32	RO	0x6020:06, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x07 (Decelerate))	UINT32	RO	0x6020:07, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A07:09	SubIndex 009	9. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x11 (Actual position))	UINT32	RO	0x6020:11, 32
1A07:0A	SubIndex 010	10. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x21 (Actual velocity))	UINT32	RO	0x6020:21, 16
1A07:0B	SubIndex 011	11. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x22 (Actual drive time))	UINT32	RO	0x6020:22, 32

Index 1A08 STM TxPDO-Map Internal position

Index	Name	Meaning	Data type	Flags	Default
1A08:0	STM TxPDO-Map Internal position	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x14 (Internal position))	UINT32	RO	0x6010:14, 32

Index 1A09 STM TxPDO-Map External position

Index	Name	Meaning	Data type	Flags	Default
1A09:0	STM TxPDO-Map External position	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x15 (External position))	UINT32	RO	0x6010:15, 32

Index 1A0A POS TxPDO-Map Actual position lag

Index	Name	Meaning	Data type	Flags	Default
1A0A:0	POS TxPDO-Map Actual position lag	PDO Mapping TxPDO 11	UINT8	RO	0x01 (1_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x23 (Actual position lag))	UINT32	RO	0x6020:23, 32

Index 1C00 Sync manager type

Index	Name	Meaning	Data type	Flags	Default value
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	Name	Meaning	Data type	Flags	Default value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x03 (3_{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	0x1602 (5634_{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})

Index 1C13 TxPDO assign

Index	Name	Meaning	Data type	Flags	Default value
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	0x1A03 (6659_{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})
1C32:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C32:07	Subindex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C32:08	Subindex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter (part 1)

Index	Name	Meaning	Data type	Flags	Default value
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 2 event • 2: DC-Mode - Synchronous with SYNC0 Event • 3: DC-Mode - Synchronous with SYNC1 event	UINT16	RW	0x0001 (1_{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	0x000F4240 (1000000_{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 1C32:08 [▶ 210])	UINT16	RO	0x0C07 (3079_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{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	Min. time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})

Index 1C32 SM output parameter (part2)

Index	Name	Meaning	Data type	Flags	Default value
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03, 1C32:05, 1C32:06, 1C32:07, 1C32:09, 1C33:03, 1C33:06, and 1C33:09 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	Max. 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})
1C32:14	Frame repeat time		UINT32	RW	0x00000000 (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 (part 1)

Index	Name	Meaning	Data type	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 1C32:02 [▸ 210]	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous with SM 3 event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08 [▸ 210] or 1C33:08 [▸ 211])	UINT16	RO	0x0C07 (3079_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▸ 210]	UINT32	RO	0x0003D090 (250000_{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	Min. time between SYNC1 event and output of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})

Index 1C33 SM input parameter (part 2)

Index	Name	Meaning	Data type	Flags	Default value
1C33:08	Command	as 1C32:08 [▸ 210]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Max. time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 1C32:11 [▸ 210]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 1C32:12 [▸ 210]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 1C32:13 [▸ 210]	UINT16	RO	0x0000 (0_{dec})
1C33:14	Frame repeat time	as 1C32:14 [▸ 210]	UINT32	RW	0x00000000 (0_{dec})
1C33:20	Sync error	as 1C32:32 [▸ 210]	BOOLEAN	RO	0x00 (0_{dec})

Index F000 Modular device profile

Index	Name	Meaning	Data type	Flags	Default value
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	0x0003 (3_{dec})

Index F008 Code word

Index	Name	Meaning	Data type	Flags	Default value
F008:0	Code word	see note! [▸ 35]	UINT32	RW	0x00000000 (0_{dec})

6.2 EL7047

6.2.1 Object description and parameterization - Profile-specific objects

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

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

NOTE

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.

Introduction

The CoE overview contains objects for different intended applications:

Object overview
• Restore object [▶ 213]
• Configuration data [▶ 214]
• Command object [▶ 218]
• Input data [▶ 219]
• Output data [▶ 220]
• Information / diagnostic data (channel specific) [▶ 222]
• Manufacturer configuration data (device-specific) [▶ 224]
• Information / diagnostic data (device-specific) [▶ 224]
• Standard objects [▶ 225]

Restore object

Index 1011 Restore default parameters

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

Configuration data

Index 8000 ENC Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8000:0	ENC Settings Ch.1	Maximum subindex	UINT8	RO	0x0E (14_{dec})
8000:08	Disable filter	Deactivates the input filters.	BOOLEAN	RW	0x00 (0_{dec})
8000:0A	Enable micro increments	The lower 8 bits of the counter value are extrapolated.	BOOLEAN	RW	0x00 (0_{dec})
8000:0E	Reversion of rotation	Activates reversion of rotation of the encoder.	BOOLEAN	RW	0x00 (0_{dec})

Index 8010 STM Motor Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8010:0	STM Motor Settings Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
8010:01	Maximal current	Maximum permanent motor coil current **Unit**: 1 mA	UINT16	RW	0x1388 (5000_{dec})
8010:02	Reduced current	Reduced coil current **Unit**: 1 mA	UINT16	RW	0x09C4 (2500_{dec})
8010:03	Nominal voltage	Nominal voltage (supply voltage) of the motor **Unit**: 10 mV	UINT16	RW	0x1388 (5000_{dec})
8010:04	Motor coil resistance	Internal resistance of the motor **Unit**: 10 mOhm	UINT16	RW	0x0064 (100_{dec})
8010:05	Motor EMF	Countervoltage of the motor **Unit**: 1 mV / (rad/s)	UINT16	RW	0x0000 (0_{dec})
8010:06	Motor fullsteps	Number of full motor steps	UINT16	RW	0x00C8 (200_{dec})
8010:07	Encoder increments (4-fold)	Number of encoder increments per revolution with quadruple evaluation	UINT16	RW	0x1000 (4096_{dec})
8010:09	Start velocity	Minimum starting velocity of the motor **Unit**: 10000 corresponds to 100% [▶ 161]	UINT16	RW	0x0000 (0_{dec})
8010:0A	Motor coil inductance	Inductance of the motor **Unit**: 0.01 mH	UINT16	RW	0x0000 (0_{dec})
8010:10	Drive on delay time	Delay between activation of driver stage and „ready = 1“	UINT16	RW	0x0064 (100dez)
8010:11	Drive off delay time	Delay between deactivation of driver stage and „ready = 0“	UINT16	RW	0x0096 (150dez)

Index 8011 STM Controller Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8011:0	STM Controller Settings Ch.1	Maximum subindex	UINT8	RO	0x02 (2_{dec})
8011:01	Kp factor (curr.)	Kp control factor of the current controller	UINT16	RW	0x0096 (150_{dec})
8011:02	Ki factor (curr.)	Ki control factor of the current controller	UINT16	RW	0x000A (10_{dec})

Index 8012 STM Features Ch.1 (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
8012:0	STM Features Ch.1	Maximum subindex	UINT8	RO	0x3A (58_{dec})
8012:01	Operation mode	permitted values:	BIT4	RW	0x00 (0_{dec})
		0: Automatic			
		1: Velocity direct			
		3: Position controller			
		4: Ext. Velocity mode			
		5: Ext. Position mode			
		6: Velocity sensorless			
8012:05	Speed range	permitted values:	BIT3	RW	0x01 (1_{dec})
		0: 1000 Fullsteps/sec			
		1: 2000 Fullsteps/sec			
		2: 4000 Fullsteps/sec			
		3: 8000 Fullsteps/sec			
		4: 16000 Fullsteps/sec			
		5: 32000 Fullsteps/sec			
8012:08	Feedback type	permitted values:	BIT1	RW	0x01 (1_{dec})
		0: Encoder			
		1: Internal counter			
8012:09	Invert motor polarity	Invert the direction of rotation of the motor	BOOLEAN	RW	0x00 (0_{dec})
8012:0A	Error on step lost	Error on loss of step	BOOLEAN	RW	0x00 (0_{dec})
8012:11	Select info data 1	permitted values:	UINT8	RW	0x0B (11_{dec})
		0: Status word			
		7: Motor velocity			
		11: Motor load			
		13: Motor dc current			
		101: Internal temperature			
		103: Control voltage			
		104: Motor supply voltage			
		150: Drive - Status word			
		151: Drive – State			
		152: Drive - Position lag (low word)			
		153: Drive - Position lag (high word)			

Index 8012 STM Features Ch.1 (part 2)

Index (hex)	Name	Meaning	Data type	Flags	Default value
8012:19	Select info data 2	permitted values:	UINT8	RW	0x0D (13$_{dec}$)
		0: Status word			
		7: Motor velocity			
		11: Motor load			
		13: Motor dc current			
		101: Internal temperature			
		103: Control voltage			
		104: Motor supply voltage			
		150: Drive - Status word			
		151: Drive - State			
		152: Drive - Position lag (low word)			
		153: Drive - Position lag (high word)			
8012:30	Invert digital input 1	Invert digital input	BOOLEAN	RW	0x00 (0$_{dec}$)
8012:31	Invert digital input 2	Invert digital input	BOOLEAN	RW	0x00 (0$_{dec}$)
8012:32	Function for input 1	permitted values:	BIT4	RW	0x00 (0$_{dec}$)
		0: Normal input			
		1: Hardware enable			
		2: PLC cam			
8012:36	Function for input 2	permitted values:	BIT4	RW	0x00 (0$_{dec}$)
		0: Normal input			
		1: Hardware enable			
		2: PLC cam			
8012:3A	Function for output 1	permitted values:	BIT4	RW	0x0F (15$_{dec}$)
		0: Normal output			
		1: Break (linked with driver enable)			
		15: Disabled			

Index 8014 STM Controller Settings 3 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8014:0	STM Controller Settings 3 Ch.1	Maximum subindex	UINT8	RO	0x09 (9$_{dec}$)
8014:01	Feed forward (pos.)	Pilot control of the position controller	UINT32	RW	0x000186A0 (100000$_{dec}$)
8014:02	Kp factor (pos.)	Kp control factor of the position controller	UINT16	RW	0x01F4 (500$_{dec}$)
8014:03	Kp factor (velo.)	Kp control factor of the velocity controller **Unit**: 0.1 mA / (rad/s)	UINT32	RW	0x00000032 (50$_{dec}$)
8014:04	Tn (velo.)	Time constant Tn of the velocity controller **Unit**: 0.01 ms	UINT16	RW	0xC350 (50000$_{dec}$)
8014:05	Sensorless param 1	First parameter (sensorless control)	UINT16	RW	0x0000 (0$_{dec}$)
8014:06	Sensorless param 2	Second parameter (sensorless control)	UINT16	RW	0x0000 (0$_{dec}$)
8014:07	Cross over velocity 1	First velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0$_{dec}$)
8014:08	Cross over velocity 2	Second velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0$_{dec}$)
8014:09	Cross over velocity 3	Third velocity transition (sensorless control) **Unit**: 0.1 rad/s	UINT16	RW	0x0000 (0$_{dec}$)

Index 8020 POS Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8020:0	POS Settings Ch.1	Maximum subindex	UINT8	RO	0x10 (16_{dec})
8020:01	Velocity min.	Minimum set velocity (range: 0-10000)	INT16	RW	0x0064 (100_{dec})
8020:02	Velocity max.	Maximum set velocity (range: 0-10000)	INT16	RW	0x2710 (10000_{dec})
8020:03	Acceleration pos.	Acceleration in positive direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:04	Acceleration neg.	Acceleration in negative direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:05	Deceleration pos.	Deceleration in positive direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:06	Deceleration neg.	Deceleration in negative direction of rotation **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:07	Emergency deceleration	Emergency deceleration (both directions of rotation) **Unit**: 1 ms	UINT16	RW	0x0064 (100_{dec})
8020:08	Calibration position	Calibration position	UINT32	RW	0x00000000 (0_{dec})
8020:09	Calibration velocity (towards plc cam)	Calibration velocity towards the cam (range: 0-10000)	INT16	RW	0x0064 (100_{dec})
8020:0A	Calibration Velocity (off plc cam)	Calibration velocity away from the cam (range: 0-10000)	INT16	RW	0x000A (10_{dec})
8020:0B	Target window	Target window	UINT16	RW	0x000A (10_{dec})
8020:0C	In-Target timeout	Target position timeout **Unit**: 1 ms	UINT16	RW	0x03E8 (1000_{dec})
8020:0D	Dead time compensation	Dead time compensation **Unit**: 1 µs	INT16	RW	0x0032 (50_{dec})
8020:0E	Modulo factor	Modulo factor/position	UINT32	RW	0x00000000 (0_{dec})
8020:0F	Modulo tolerance window	Tolerance window for modulo positioning	UINT32	RW	0x00000000 (0_{dec})
8020:10	Position lag max.	Maximum allowable step error	UINT16	RW	0x0000 (0_{dec})

Index 8021 POS Features Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
8021:0	POS Features Ch.1	Maximum subindex	UINT8	RO	0x16 (22$_{dec}$)
8021:01	Start type	permitted values:	UINT16	RW	0x0001 (1$_{dec}$)
		0: Idle			
		1: Absolute			
		2: Relative			
		3: Endless plus			
		4: Endless minus			
		6: Additive			
		24832: Calibration (Hardware sync)			
		24576: Calibration (Plc cam)			
		28416: Calibration (Clear manual)			
		28160: Calibration (Set manual)			
		28161: Calibration (Set manual auto)			
		1029: Modulo current			
		773: Modulo minus			
		517: Modulo plus			
		261: Modulo short			
8021:11	Time information	permitted values:	BIT2	RW	0x00 (0$_{dec}$)
		0: Elapsed time current drive time since start of the travel command			
8021:13	Invert calibration cam search direction	Inversion of the direction of rotation towards the cam	BOOLEAN	RW	0x01 (1$_{dec}$)
8021:14	Invert sync impulse search direction	Inversion of the direction of rotation away from the cam	BOOLEAN	RW	0x00 (0$_{dec}$)
8021:15	Emergency stop on position lag error	Triggers an emergency stop if the maximum following error is exceeded.	BOOLEAN	RW	0x00 (0$_{dec}$)
8021:16	Enhanced diag history	Provides detailed messages about the status of the positioning interface in the diag history.	BOOLEAN	RW	0x00 (0$_{dec}$)

Command object

Index FB00 STM Command

Index (hex)	Name	Meaning	Data type	Flags	Default value
FB00:0	STM Command	Maximum subindex	UINT8	RO	0x03 (3$_{dec}$)
FB00:01	Request	Requesting a command 0x8000: Software reset	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Status of the command	UINT8	RO	0x00 (0$_{dec}$)
		0: No error, without return value			
		1: No error, with return value			
		2: With error, without return value			
		3: With error, with return value			
		... reserved			
		255: Command execution active			
FB00:03	Response	Return value of the executed command	OCTET-STRING[4]	RO	{0}

Input data

Index 6000 ENC Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
6000:0	ENC Inputs Ch.1	Maximum subindex	UINT8	RO	0x16 (22_{dec})
6000:01	Latch C valid	The counter value was latched with the C track.	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Latch extern valid	The counter value was stored via the external latch.	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Set counter done	The counter was set.	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Counter underflow	Counter underflow.	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Counter overflow	Counter overflow.	BOOLEAN	RO	0x00 (0_{dec})
6000:08	Extrapolation stall	The extrapolated part of the counter is invalid.	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Status of input A	Status of the A-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0A	Status of input B	Status of the B-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0B	Status of input C	Status of the C-input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0D	Status of extern latch	Status of the ext. latch input.	BOOLEAN	RO	0x00 (0_{dec})
6000:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	0x00 (0_{dec})
6000:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6000:11	Counter value	The counter value.	UINT32	RO	0x00000000 (0_{dec})
6000:12	Latch value	The latch value.	UINT32	RO	0x00000000 (0_{dec})
6000:16	Timestamp	Time stamp of the last counter change.	UINT32	RO	0x00000000 (0_{dec})

Index 6010 STM Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
6010:0	STM Inputs Ch.1	Maximum subindex	UINT8	RO	0x15 (21_{dec})
6010:01	Ready to enable	Driver stage is ready for enabling	BOOLEAN	RO	0x00 (0_{dec})
6010:02	Ready	Driver stage is ready for operation	BOOLEAN	RO	0x00 (0_{dec})
6010:03	Warning	A warning has occurred (see index 0xA010 [▶ 223]).	BOOLEAN	RO	0x00 (0_{dec})
6010:04	Error	An error has occurred (see index 0xA010 [▶ 223]).	BOOLEAN	RO	0x00 (0_{dec})
6010:05	Moving positive	Driver stage is activated in positive direction.	BOOLEAN	RO	0x00 (0_{dec})
6010:06	Moving negative	Driver stage is activated in negative direction.	BOOLEAN	RO	0x00 (0_{dec})
6010:07	Torque reduced	Reduced torque is active.	BOOLEAN	RO	0x00 (0_{dec})
6010:08	Motor stall	A loss of step has occurred.	BOOLEAN	RO	0x00 (0_{dec})
6010:0C	Digital input 1	Digital input 1	BOOLEAN	RO	0x00 (0_{dec})
6010:0D	Digital input 2	Digital input 2	BOOLEAN	RO	0x00 (0_{dec})
6010:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	0x00 (0_{dec})
6010:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6010:11	Info data 1	Synchronous information (selection via subindex 0x8012:11 [▶ 215])	UINT16	RO	0x0000 (0_{dec})
6010:12	Info data 2	Synchronous information (selection via subindex 0x8012:19 [▶ 215])	UINT16	RO	0x0000 (0_{dec})
6010:13	Motor load	Current motor load **Unit:** 0.01°	INT16	RO	0x0000 (0_{dec})
6010:14	Internal position	Internal microstep position	UINT32	RO	0x00000000 (0_{dec})
6010:15	External position	Encoder position	UINT32	RO	0x00000000 (0_{dec})

Index 6020 POS Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
6020:0	POS Inputs Ch.1	Maximum subindex	UINT8	RO	0x16 (22_{dec})
6020:01	Busy	A current travel command is active.	BOOLEAN	RO	0x00 (0_{dec})
6020:02	In-Target	Motor has arrived at target.	BOOLEAN	RO	0x00 (0_{dec})
6020:03	Warning	A warning has occurred.	BOOLEAN	RO	0x00 (0_{dec})
6020:04	Error	An error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
6020:05	Calibrated	The Motor is calibrated.	BOOLEAN	RO	0x00 (0_{dec})
6020:06	Accelerate	The Motor is in the acceleration phase.	BOOLEAN	RO	0x00 (0_{dec})
6020:07	Decelerate	The Motor is in the deceleration phase.	BOOLEAN	RO	0x00 (0_{dec})
6020:11	Actual position	Current target position of the travel command generator	UINT32	RO	0x00000000 (0_{dec})
6020:21	Actual velocity	Current set velocity of the travel command generator	INT16	RO	0x0000 (0_{dec})
6020:22	Actual drive time	Travel command time information (see subindex 0x8021:11 [▸ 218])	UINT32	RO	0x00000000 (0_{dec})
6020:23	Actual position lag	Lag of position	UINT32	RO	0x00000000 (0_{dec})

Output data

Index 7000 ENC Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
7000:0	ENC Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:01	Enable latch C	Activate latching via the C-track.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Enable latch extern on positive edge	Activate external latch with positive edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:03	Set counter	Set the counter value.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Enable latch extern on negative edge	Activate external latch with negative edge.	BOOLEAN	RO	0x00 (0_{dec})
7000:11	Set counter value	This is the counter value to be set via "Set counter".	UINT32	RO	0x00000000 (0_{dec})

Index 7010 STM Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
7010:0	STM Outputs Ch.1	Maximum subindex	UINT8	RO	0x21 (33_{dec})
7010:01	Enable	Activates the output stage	BOOLEAN	RO	0x00 (0_{dec})
7010:02	Reset	All errors that may have occurred are reset by setting this bit (rising edge).	BOOLEAN	RO	0x00 (0_{dec})
7010:03	Reduce torque	Reduced torque (coil current) is active (see subindex 0x8010:02 [▸ 214]).	BOOLEAN	RO	0x00 (0_{dec})
7010:0C	Digital output 1	Digital output 1	BOOLEAN	RO	0x00 (0_{dec})
7010:11	Position	Set position specification **Unit:** Increments [▸ 163]	UINT32	RO	0x00000000 (0_{dec})
7010:21	Velocity	Set velocity specification **Unit:** +/- 32767 corresponds to +/- 100% [▸ 161]	INT16	RO	0x0000 (0_{dec})

Index 7020 POS Outputs Ch.1 (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7020:0	POS Outputs Ch.1	Maximum subindex	UINT8	RO	0x24 (36_{dec})
7020:01	Execute	Start travel command (rising edge), or prematurely abort travel command (falling edge)	BOOLEAN	RO	0x00 (0_{dec})
7020:02	Emergency Stop	Prematurely abort travel command with an emergency ramp (rising edge)	BOOLEAN	RO	0x00 (0_{dec})
7020:11	Target position	Specification of the target position	UINT32	RO	0x00000000 (0_{dec})
7020:21	Velocity	Specification of the maximum set velocity	INT16	RO	0x0000 (0_{dec})

Index 7020 POS Outputs Ch.1 (part 2)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7020:22	Start type				
	0x0000 Idle: No travel command is being executed		UINT16	RO	0x0000 (0_{dec})
	0x0001 Absolute: Absolute target position		UINT16	RO	0x0000 (0_{dec})
	0x1001 Absolute (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0002 Relative: Target position relative to the current position		UINT16	RO	0x0000 (0_{dec})
	0x1002 Relative (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0003 Endless plus: Endless driving in positive direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0004 Endless minus: Endless driving in negative direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0105 Modulo short: Shortest distance to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0115 Modulo short extended: Shortest distance to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0205 Modulo plus: Drive in positive direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0215 Modulo plus extended: Drive in positive direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0305 Modulo minus: Drive in negative direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0315 Modulo minus extended: Drive in negative direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0405 Modulo current: Drive in the last implemented direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0415 Modulo current extended: Drive in the last implemented direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0006 Additive: New target position relative/additive to the last target position		UINT16	RO	0x0000 (0_{dec})
	0x1006 Additive (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x6000 Calibration, PLC cam: Calibration with cam		UINT16	RO	0x0000 (0_{dec})
	0x6100 Calibration, HW sync: Calibration with cam and C-track		UINT16	RO	0x0000 (0_{dec})
	0x6E00 Calibration, set manual: Set calibration manually		UINT16	RO	0x0000 (0_{dec})
	0x6E01 Calibration, set manual auto: Set automatic calibration, for "Enable = 1"		UINT16	RO	0x0000 (0_{dec})
	0x6F00 Calibration, clear manual: Clear calibration manually		UINT16	RO	0x0000 (0_{dec})
7020:23	Acceleration	Acceleration specification	UINT16	RO	0x0000 (0_{dec})
7020:24	Deceleration	Deceleration specification	UINT16	RO	0x0000 (0_{dec})

Index 7021 POS Outputs 2 Ch.1 (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7020:0	POS Outputs Ch.1	Maximum subindex	UINT8	RO	0x24 (36_{dec})
7020:03	Enable auto start	Enable auto start	BOOLEAN	RO	0x00 (0_{dec})
7020:11	Target position	Specification of the target position	UINT32	RO	0x00000000 (0_{dec})
7020:21	Velocity	Specification of the maximum set velocity	INT16	RO	0x0000 (0_{dec})

Index 7021 POS Outputs 2 Ch.1 (part 2)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7021:22	Start type				
	0x0000 Idle: No travel command is being executed		UINT16	RO	0x0000 (0_{dec})
	0x0001 Absolute: Absolute target position		UINT16	RO	0x0000 (0_{dec})
	0x1001 Absolute (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0002 Relative: Target position relative to the current position		UINT16	RO	0x0000 (0_{dec})
	0x1002 Relative (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x0003 Endless plus: Endless driving in positive direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0004 Endless minus: Endless driving in negative direction of rotation		UINT16	RO	0x0000 (0_{dec})
	0x0105 Modulo short: Shortest distance to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0115 Modulo short extended: Shortest distance to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0205 Modulo plus: Drive in positive direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0215 Modulo plus extended: Drive in positive direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0305 Modulo minus: Drive in negative direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0315 Modulo minus extended: Drive in negative direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0405 Modulo current: Drive in the last implemented direction of rotation to the next modulo position		UINT16	RO	0x0000 (0_{dec})
	0x0415 Modulo current extended: Drive in the last implemented direction of rotation to the next modulo position (without modulo window)		UINT16	RO	0x0000 (0_{dec})
	0x0006 Additive: New target position relative/additive to the last target position		UINT16	RO	0x0000 (0_{dec})
	0x1006 Additive (Change): Change during an active travel command		UINT16	RO	0x0000 (0_{dec})
	0x6000 Calibration, PLC cam: Calibration with cam		UINT16	RO	0x0000 (0_{dec})
	0x6100 Calibration, HW sync: Calibration with cam and C-track		UINT16	RO	0x0000 (0_{dec})
	0x6E00 Calibration, set manual: Set calibration manually		UINT16	RO	0x0000 (0_{dec})
	0x6E01 Calibration, set manual auto: Set automatic calibration, for "Enable = 1"		UINT16	RO	0x0000 (0_{dec})
	0x6F00 Calibration, clear manual: Clear calibration manually		UINT16	RO	0x0000 (0_{dec})
7021:23	Acceleration	Acceleration specification	UINT16	RO	0x0000 (0_{dec})
7021:24	Deceleration	Deceleration specification	UINT16	RO	0x0000 (0_{dec})

Information / diagnostic data (channel specific)

Index 9010 STM Info data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
9010:0	STM Info data Ch.1	Maximum subindex	UINT8	RO	0x13 (19_{dec})
9010:01	Status word	Status word (see index 0xA010 [▸ 223])	UINT16	RO	0x0000 (0_{dec})
9010:08	Motor velocity	Current motor velocity	INT16	RO	0x0000 (0_{dec})
9010:09	Internal position	Internal position (micro increments)	UINT32	RO	0x00000000 (0_{dec})
9010:0B	Motor load	Current motor load **Unit**: 0.01°	INT16	RO	0x0000 (0_{dec})
9010:0D	Motor dc current	Current motor current (DC vector) **Unit**: 1 mA	INT16	RO	0x0000 (0_{dec})
9010:0E	Tn (curr.)	Internally calculated time constant of the current controller **Unit**: 0.01 ms	UINT16	RO	0x0000 (0_{dec})
9010:13	External position	External position (connected encoder)	UINT32	RO	0x00000000 (0_{dec})

Index 9020 POS Info data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
9020:0	POS Info data Ch.1	Maximum subindex	UINT8	RO	0x04 (4_{dec})
9020:01	Status word	Status word	UINT16	RO	0x0000 (0_{dec})
9020:03	State (drive controller)	permitted values:	UINT16	RO	0x0000 (0_{dec})
		0: Init			
		1: Idle			
		272: Go cam			
		273: On cam			
		16: Start			
		17: Acceleration			
		18: Constant			
		19: Deceleration			
		288: Go sync impulse			
		289: Leave cam			
		4096: Pre target			
		4097: In target			
		32: Emergency Stop			
		33: Normal stop			
		304: Calibration stop			
		8192: Drive end			
		8193: Wait for init			
		320: Is calibrated			
		321: Not calibrated			
		16384: Drive warning			
		32768: Error			
		65535: Undefined			
		256: Calibration start			
9020:04	Actual position lag	Current step error	INT32	RO	0x00000000 (0_{dec})

Index A010 STM Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
A010:0	STM Diag data Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
A010:01	Saturated	Driver stage operates with maximum duty cycle.	BOOLEAN	RO	0x00 (0_{dec})
A010:02	Over temperature	Internal terminal temperature is greater than 80 °C.	BOOLEAN	RO	0x00 (0_{dec})
A010:03	Torque overload	Duty cycle output at 100%	BOOLEAN	RO	0x00 (0_{dec})
A010:04	Under voltage	Supply voltage less than 7 V	BOOLEAN	RO	0x00 (0_{dec})
A010:05	Over voltage	Supply voltage 10 % higher than the nominal voltage (see 0x8010:03)	BOOLEAN	RO	0x00 (0_{dec})
A010:06	Short circuit	Short circuit of motor coil	BOOLEAN	RO	0x00 (0_{dec})
A010:08	No control power	No power supply to driver stage	BOOLEAN	RO	0x00 (0_{dec})
A010:09	Misc error	• Initialization failed or • Internal terminal temperature is higher than 100 °C (see 0xF80F:05).	BOOLEAN	RO	0x00 (0_{dec})
A010:0A	Configuration	CoE change has not yet been adopted into the current configuration.	BOOLEAN	RO	0x00 (0_{dec})
A010:0B	Motor stall	A loss of step has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A010:11	Actual operation mode	permitted values:	BIT4	RO	0x00 (0_{dec})
		0: Automatic			
		1: Velocity direct			
		2: Velocity controller			
		3: Position controller			
		4: Ext. Velocity mode			
		5: Ext. Position mode			
		6: Velocity sensorless			

Index A020 POS Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default value
A020:0	POS Diag data Ch.1	Maximum subindex	UINT8	RO	0x06 (6_{dec})
A020:01	Command rejected	Travel command was rejected.	BOOLEAN	RO	0x00 (0_{dec})
A020:02	Command aborted	Travel command was aborted.	BOOLEAN	RO	0x00 (0_{dec})
A020:03	Target overrun	Target position was overrun in the opposite direction.	BOOLEAN	RO	0x00 (0_{dec})
A020:04	Target timeout	The target window was not reached within the in-target timeout.	BOOLEAN	RO	0x00 (0_{dec})
A020:05	Position lag	The maximum following error was exceeded.	BOOLEAN	RO	0x00 (0_{dec})
A020:06	Emergency Stop	An emergency stop was triggered (automatic or manual).	BOOLEAN	RO	0x00 (0_{dec})

Manufacturer configuration data (device-specific)

Index F80F STM Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default value
F80F:0	STM Vendor data	Maximum subindex	UINT8	RO	0x05 (5_{dec})
F80F:04	Warning temperature	Temperature warning threshold **Unit**: 1 °C	INT8	RW	0x50 (80_{dec})
F80F:05	Switch off temperature	Switch-off temperature **Unit**: 1 °C	INT8	RW	0x64 (100_{dec})

Information / diagnostic data (device-specific)

Index F900 STM Info data

Index (hex)	Name	Meaning	Data type	Flags	Default value
F900:0	STM Info data	Maximum subindex	UINT8	RO	0x06 (6_{dec})
F900:01	Software version (driver)	Software version of the output driver	STRING	RO	
F900:02	Internal temperature	Internal terminal temperature **Unit**: 1 °C	INT8	RO	0x00 (0_{dec})
F900:04	Control voltage	Control voltage **Unit**: 1 mV, 10 mV with field-oriented control	UINT16	RO	0x0000 (0_{dec})
F900:05	Motor supply voltage	Motor supply voltage **Unit**: 1 mV, 10 mV with field-oriented control	UINT16	RO	0x0000 (0_{dec})
F900:06	Cycle time	Current EtherCAT cycle time **Unit**: 1 µs	UINT16	RO	0x0000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x03 (3_{dec})
F010:01	SubIndex 001	Encoder profile number	UINT32	RW	0x000001FF (511_{dec})
F010:02	SubIndex 002	Stepper motor profile number	UINT32	RW	0x000002BF (703_{dec})
F010:03	SubIndex 003	Positioning interface profile number	UINT32	RW	0x000002C0 (704_{dec})

Index F081 Download revision

Index (hex)	Name	Meaning	Data type	Flags	Default value
F081:0	Download revision	Maximum subindex	UINT8	RO	0x01 (1_{dec})
F081:01	Revision number	Revision number	UINT32	RW	0x00000000 (0_{dec})

Index FB40 Memory interface

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

6.2.2 Object description and parameterization - standard objects

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.

Standard objects (0x1000-0x1FFF)

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default value
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 value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL7047

Index 1009 Hardware version

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

Index 100ASoftware version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x1B873052 (461844562_{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 value
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 10F3 Diagnosis History

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

Index 1400 ENC RxPDO-Par Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1400:0	ENC RxPDO-Par Control compact	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1.	OCTET-STRING[6]	RO	01 16 00 00 00 00

Index 1401 ENC RxPDO-Par Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1401:0	ENC RxPDO-Par Control	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2.	OCTET-STRING[6]	RO	00 16 00 00 00 00

Index 1403 STM RxPDO-Par Position

Index (hex)	Name	Meaning	Data type	Flags	Default value
1403:0	STM RxPDO-Par Position	PDO Parameter RxPDO 4	UINT8	RO	0x06 (6_{dec})
1403:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 4.	OCTET-STRING[6]	RO	04 16 05 16 06 16

Index 1404 STM RxPDO-Par Velocity

Index (hex)	Name	Meaning	Data type	Flags	Default value
1404:0	STM RxPDO-Par Velocity	PDO Parameter RxPDO 5	UINT8	RO	0x06 (6_{dec})
1404:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 5.	OCTET-STRING[6]	RO	03 16 05 16 06 16

Index 1405 POS RxPDO-Par Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1405:0	POS RxPDO-Par Control compact	PDO Parameter RxPDO 6	UINT8	RO	0x06 (6_{dec})
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6.	OCTET-STRING[6]	RO	03 16 04 16 06 16

Index 1406 POS RxPDO-Par Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1406:0	POS RxPDO-Par Control	PDO Parameter RxPDO 7	UINT8	RO	0x06 (6_{dec})
1406:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 7.	OCTET-STRING[6]	RO	03 16 04 16 05 16

Index 1407 POS RxPDO-Par Control 2

Index	Name	Meaning	Data type	Flags	Default
1407:0	POS RxPDO-Par Control 2	PDO Parameter RxPDO 8	UINT8	RO	0x06 (6_{dec})
1407:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 8	OCTET-STRING[6]	RO	03 16 04 16 05 16

Index 1600 ENC RxPDO-Map Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1600:0	ENC RxPDO-Map Control compact	PDO Mapping RxPDO 1	UINT8	RO	0x06 (6_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set counter))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 16

Index 1601 ENC RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1601:0	ENC RxPDO-Map Control	PDO Mapping RxPDO 2	UINT8	RO	0x06 (6_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x01 (Enable latch C))	UINT32	RO	0x7000:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x03 (Set counter))	UINT32	RO	0x7000:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7000:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7000:11, 32

Index 1602 STM RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1602:0	STM RxPDO-Map Control	PDO Mapping RxPDO 3	UINT8	RO	0x06 (6_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x01 (Enable))	UINT32	RO	0x7010:01, 1
1602:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x02 (Reset))	UINT32	RO	0x7010:02, 1
1602:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x03 (Reduce torque))	UINT32	RO	0x7010:03, 1
1602:04	SubIndex 004	4. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1602:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x0C (Digital output 1))	UINT32	RO	0x7010:0C, 1
1602:06	SubIndex 006	6. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4

Index 1603 STM RxPDO-Map Position

Index (hex)	Name	Meaning	Data type	Flags	Default value
1603:0	STM RxPDO-Map Position	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x11 (Position))	UINT32	RO	0x7010:11, 32

Index 1604 STM RxPDO-Map Velocity

Index (hex)	Name	Meaning	Data type	Flags	Default value
1604:0	STM RxPDO-Map Velocity	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (STM Outputs Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7010:21, 16

Index 1605 POS RxPDO-Map Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1605:0	POS RxPDO-Map Control compact	PDO Mapping RxPDO 6	UINT8	RO	0x04 (4_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x01 (Execute))	UINT32	RO	0x7020:01, 1
1605:02	SubIndex 002	2. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x02 (Emergency stop))	UINT32	RO	0x7020:02, 1
1605:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1605:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7020:11, 32

Index 1606 POS RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1606:0	POS RxPDO-Map Control	PDO Mapping RxPDO 7	UINT8	RO	0x08 (8_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x01 (Execute))	UINT32	RO	0x7020:01, 1
1606:02	SubIndex 002	2. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x02 (Emergency stop))	UINT32	RO	0x7020:02, 1
1606:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1606:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7020:11, 32
1606:05	SubIndex 005	5. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7020:21, 16
1606:06	SubIndex 006	6. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x22 (Start type))	UINT32	RO	0x7020:22, 16
1606:07	SubIndex 007	7. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x23 (Acceleration))	UINT32	RO	0x7020:23, 16
1606:08	SubIndex 008	8. PDO Mapping entry (object 0x7020 (POS Outputs Ch.1), entry 0x24 (Deceleration))	UINT32	RO	0x7020:24, 16

Index 1607 POS RxPDO-Map Control 2

Index	Name	Meaning	Data type	Flags	Default
1606:0	POS RxPDO-Map Control	PDO Mapping RxPDO 7	UINT8	RO	0x08 (8_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00,2
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7021 (POS Outputs 2 Ch.1), entry 0x03 (Enable auto start))	UINT32	RO	0x7021:03, 1
1607:03	SubIndex 003	3. PDO Mapping entry (13 bits align)	UINT32	RO	0x0000:00, 13
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x11 (Target position))	UINT32	RO	0x7021:11, 32
1607:05	SubIndex 005	5. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x21 (Velocity))	UINT32	RO	0x7021:21, 16
1607:06	SubIndex 006	6. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x22 (Start type))	UINT32	RO	0x7021:22, 16
1607:07	SubIndex 007	7. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x23 (Acceleration))	UINT32	RO	0x7021:23, 16
1607:08	SubIndex 008	8. PDO Mapping entry (object 0x7020 (POS Outputs 2 Ch.1), entry 0x24 (Deceleration))	UINT32	RO	0x7021:24, 16

Index 1800 ENC TxPDO-Par Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1800:0	ENC TxPDO-Par Status compact	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1.	OCTET-STRING[2]	RO	01 1A

Index 1801 ENC TxPDO-Par Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1801:0	ENC TxPDO-Par Status	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2.	OCTET-STRING[2]	RO	00 1A

Index 1806 POS TxPDO-Par Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1806:0	POS TxPDO-Par Status compact	PDO parameter TxPDO 7	UINT8	RO	0x06 (6_{dec})
1806:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 7.	OCTET-STRING[2]	RO	07 1A

Index 1807 POS TxPDO-Par Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1807:0	POS TxPDO-Par Status	PDO parameter TxPDO 8	UINT8	RO	0x06 (6_{dec})
1807:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 8.	OCTET-STRING[2]	RO	06 1A

Index 1A00 ENC TxPDO-Map Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A00:0	ENC TxPDO-Map Status compact	PDO Mapping TxPDO 1	UINT8	RO	0x11 (17_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0D (Sync error))	UINT32	RO	0x6000:0D, 1
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0E (Status of extern latch))	UINT32	RO	0x6000:0E, 1
1A00:0E	SubIndex 014	14. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6000:10, 1
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 16
1A00:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 16

Index 1A01 ENC TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A01:0	ENC TxPDO-Map Status	PDO Mapping TxPDO 2	UINT8	RO	0x11 (17_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x01 (Latch C valid))	UINT32	RO	0x6000:01, 1
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:02, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6000:03, 1
1A01:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x04 (Counter underflow))	UINT32	RO	0x6000:04, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x05 (Counter overflow))	UINT32	RO	0x6000:05, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A01:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0B (Status of input C))	UINT32	RO	0x6000:0B, 1
1A01:0B	SubIndex 011	11. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 1
1A01:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0D (Status of extern latch))	UINT32	RO	0x6000:0D, 1
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6000:0E, 1
1A01:0E	SubIndex 014	14. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A01:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6000:10, 1
1A01:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6000:11, 32
1A01:11	SubIndex 017	17. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6000:12, 32

Index 1A02 ENC TxPDO-Map Timest. compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A02:0	ENC TxPDO-Map Timest. compact	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 32

Index 1A03 STM TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A03:0	STM TxPDO-Map Status	PDO Mapping TxPDO 4	UINT8	RO	0x0E (14_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x01 (Ready to enable))	UINT32	RO	0x6010:01, 1
1A03:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x02 (Ready))	UINT32	RO	0x6010:02, 1
1A03:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6010:03, 1
1A03:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6010:04, 1
1A03:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x05 (Moving positive))	UINT32	RO	0x6010:05, 1
1A03:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x06 (Moving negative))	UINT32	RO	0x6010:06, 1
1A03:07	SubIndex 007	7. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x07 (Torque reduced))	UINT32	RO	0x6010:07, 1
1A03:08	SubIndex 008	8. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x08 (Motor stall))	UINT32	RO	0x6010:08, 1
1A03:09	SubIndex 009	9. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A03:0A	SubIndex 010	10. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6010:0C, 1
1A03:0B	SubIndex 011	11. PDO Mapping entry (1 bits align)	UINT32	RO	0x6010:0D, 1
1A03:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6010:0E, 1
1A03:0D	SubIndex 013	13. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A03:0E	SubIndex 014	14. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6010:10, 1

Index 1A04 STM TxPDO-Map Synchron info data

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A04:0	STM TxPDO-Map Synchron info data	PDO Mapping TxPDO 5	UINT8	RO	0x02 (2_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x11 (Info data 1))	UINT32	RO	0x6010:11, 16
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x12 (Info data 2))	UINT32	RO	0x6010:12, 16

Index 1A05 STM TxPDO-Map Motor load

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A05:0	STM TxPDO-Map Motor load	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x13 (Motor load))	UINT32	RO	0x6010:13, 16

Index 1A06 POS TxPDO-Map Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A06:0	POS TxPDO-Map Status compact	PDO Mapping TxPDO 7	UINT8	RO	0x08 (8_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x01 (Busy))	UINT32	RO	0x6020:01, 1
1A06:02	SubIndex 002	2. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x02 (In-Target))	UINT32	RO	0x6020:02, 1
1A06:03	SubIndex 003	3. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6020:03, 1
1A06:04	SubIndex 004	4. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6020:04, 1
1A06:05	SubIndex 005	5. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x05 (Calibrated))	UINT32	RO	0x6020:05, 1
1A06:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x06 (Accelerate))	UINT32	RO	0x6020:06, 1
1A06:07	SubIndex 007	7. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x07 (Decelerate))	UINT32	RO	0x6020:07, 1
1A06:08	SubIndex 008	8. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9

Index 1A07 POS TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A07:0	POS TxPDO-Map Status	PDO Mapping TxPDO 8	UINT8	RO	0x0B (11_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x01 (Busy))	UINT32	RO	0x6020:01, 1
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x02 (In-Target))	UINT32	RO	0x6020:02, 1
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x03 (Warning))	UINT32	RO	0x6020:03, 1
1A07:04	SubIndex 004	4. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x04 (Error))	UINT32	RO	0x6020:04, 1
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x05 (Calibrated))	UINT32	RO	0x6020:05, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x06 (Accelerate))	UINT32	RO	0x6020:06, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x07 (Decelerate))	UINT32	RO	0x6020:07, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A07:09	SubIndex 009	9. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x11 (Actual position))	UINT32	RO	0x6020:11, 32
1A07:0A	SubIndex 010	10. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x21 (Actual velocity))	UINT32	RO	0x6020:21, 16
1A07:0B	SubIndex 011	11. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x22 (Actual drive time))	UINT32	RO	0x6020:22, 32

Index 1A08 STM TxPDO-Map Internal position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	STM TxPDO-Map Internal position	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x14 (Internal position))	UINT32	RO	0x6010:14, 32

Index 1A09 STM TxPDO-Map External position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	STM TxPDO-Map External position	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (STM Inputs Ch.1), entry 0x15 (External position))	UINT32	RO	0x6010:15, 32

Index 1A0A POS TxPDO-Map Actual position lag

Index	Name	Meaning	Data type	Flags	Default
1A0A:0	POS TxPDO-Map Actual position lag	PDO Mapping TxPDO 11	UINT8	RO	0x01 (1_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (POS Inputs Ch.1), entry 0x23 (Actual position lag))	UINT32	RO	0x6020:23, 32

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default value
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 value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x03 (3_{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	0x1602 (5634_{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1604 (5636_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x1A03 (6659_{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})
1C32:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 2 event • 2: DC-Mode - Synchronous with SYNC0 Event • 3: DC-Mode - Synchronous with SYNC1 event	UINT16	RW	0x0001 (1_{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	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x0C07 (3079_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{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	Min. time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})

Index 1C32 SM output parameter (part 2)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C32:03, 1C32:05, 1C32:06, 1C32:07, 1C32:09, 1C33:03, 1C33:06, and 1C33:09 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})
1C32:14	Frame repeat time		UINT32	RW	0x00000000 (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 (part 1)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 0x1C32:02 [▸ 235]	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • 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 [▸ 235] or 0x1C33:08 [▸ 236])	UINT16	RO	0x0C07 (3079_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▸ 235]	UINT32	RO	0x0003D090 (250000_{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	Min. time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})

Index 1C33 SM input parameter (part 2)

Index	Name	Meaning	Data type	Flags	Default value
1C33:08	Command	as 0x1C32:08 [▸ 235]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Max. time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 0x1C32:11 [▸ 235]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▸ 235]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▸ 235]	UINT16	RO	0x0000 (0_{dec})
1C33:14	Frame repeat time	as 1C32:14 [▸ 210]	UINT32	RW	0x00000000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▸ 235]	BOOLEAN	RO	0x00 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x0003 (3_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default value
F008:0	Code word	see note! [▸ 35]	UINT32	RW	0x00000000 (0_{dec})

Also see about this

Object description and parameterization - standard objects [▸ 235]

7 Error correction

7.1 Diagnostics – basic principles of diag messages

DiagMessages designates a system for the transmission of messages from the EtherCAT Slave to the EtherCAT Master/TwinCAT. The messages are stored by the device in its own CoE under 0x10F3 and can be read by the application or the System Manager. An error message referenced via a code is output for each event stored in the device (warning, error, status change).

Definition

The *DiagMessages* system is defined in the ETG (EtherCAT Technology Group) in the guideline ETG.1020, chapter 13 “Diagnosis handling”. It is used so that pre-defined or flexible diagnostic messages can be conveyed from the EtherCAT Slave to the Master. In accordance with the ETG, the process can therefore be implemented supplier-independently. Support is optional. The firmware can store up to 250 DiagMessages in its own CoE.

Each DiagMessage consists of

- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

The DiagMessages are explained in text form in the ESI/XML file belonging to the EtherCAT device: on the basis of the Text ID contained in the DiagMessage, the corresponding plain text message can be found in the languages contained in the ESI/XML. In the case of Beckhoff products these are usually German and English.

Via the entry *NewMessagesAvailable* the user receives information that new messages are available.

DiagMessages can be confirmed in the device: the last/latest unconfirmed message can be confirmed by the user.

In the CoE both the control entries and the history itself can be found in the CoE object 0x10F3:

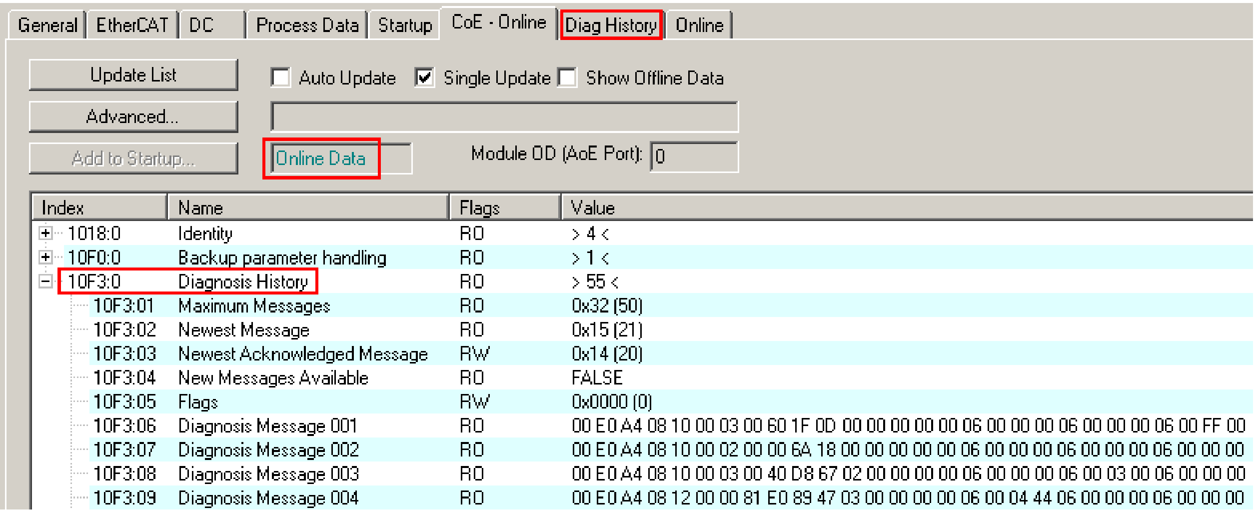


Fig. 222: DiagMessages in the CoE

The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

The DiagMessages system is to be used above all during the commissioning of the plant. The diagnostic values e.g. in the StatusWord of the device (if available) are helpful for online diagnosis during the subsequent continuous operation.

TwinCAT System Manager implementation

From TwinCAT 2.11 DiagMessages, if available, are displayed in the device's own interface. Operation (collection, confirmation) also takes place via this interface.

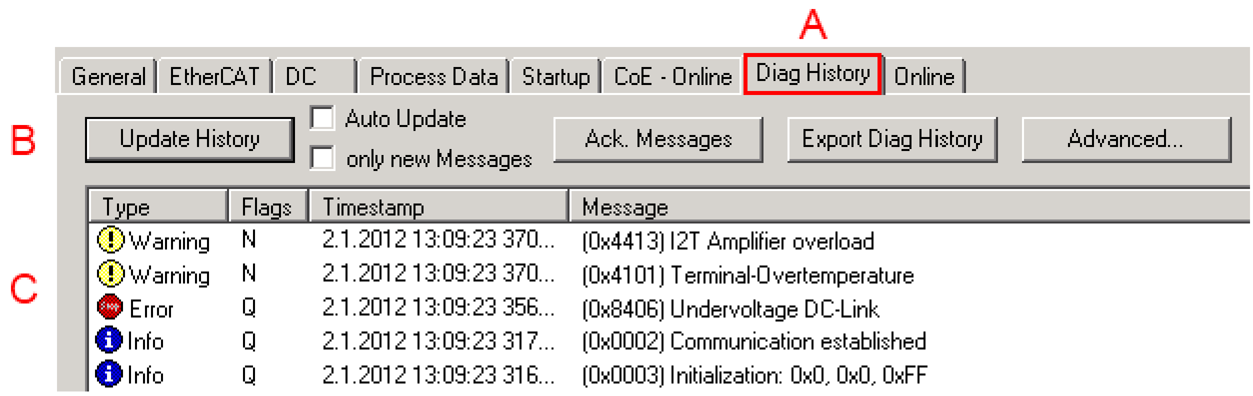


Fig. 223: Implementation of the DiagMessage system in the TwinCAT System Manager

The operating buttons (B) and the history read out (C) can be seen on the Diag History tab (A). The components of the message:

- Info/Warning/Error
- Acknowledge flag (N = unconfirmed, Q = confirmed)
- Time stamp
- Text ID
- Plain text message according to ESI/XML data

The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

Since TwinCAT 3.1 build 4022 DiagMessages send by the terminal are shown by the TwinCAT ADS Logger. Given that DiagMessages are represented IO- comprehensive at one place, commissioning will be simplified. In addition, the logger output could be stored into a data file – hence DiagMessages are available long-term for analysis.

DiagMessages are actually only available locally in CoE 0x10F3 in the terminal and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT Terminals are set by default to report the presence of a DiagMessage as emergency via EtherCAT; the event logger can then retrieve the DiagMessage. The function is activated in the terminal via 0x10F3:05, so such terminals have the following entry in the StartUp list by default:

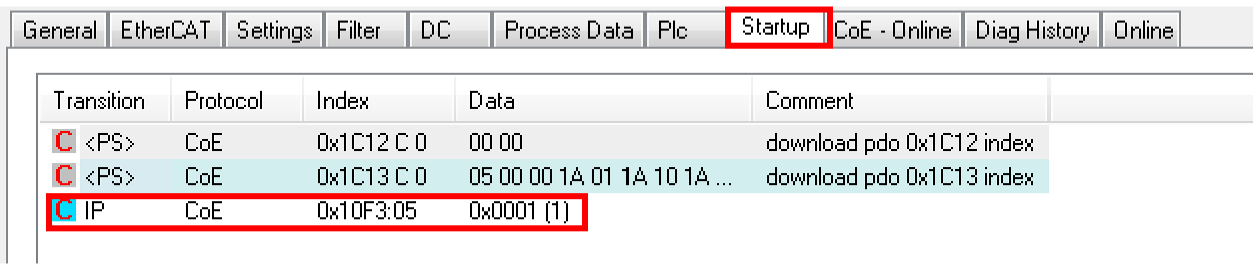


Fig. 224: Startup List

If the function is to be deactivated because, for example, many messages come in or the EventLogger is not used, the StartUp entry can be deleted or set to 0.

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

The time stamp is obtained from the local clock of the terminal at the time of the event. The time is usually the distributed clock time (DC) from register x910.

Please note: When EtherCAT is started, the DC time in the reference clock is set to the same time as the local IPC/TwinCAT time. From this moment the DC time may differ from the IPC time, since the IPC time is not adjusted. Significant time differences may develop after several weeks of operation without a EtherCAT restart. As a remedy, external synchronization of the DC time can be used, or a manual correction calculation can be applied, as required: The current DC time can be determined via the EtherCAT master or from register x901 of the DC slave.

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 2: reserved 1: Info 4: Warning 8: Error	0: System 1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: I/O general 7: reserved	Error number

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

Overview of text IDs

Specific text IDs are listed in the device documentation.

Text ID	Type	Place	Text Message	Additional comment
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Connection established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1000	Information	System	Information: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1012	Information	System	EtherCAT state change Init - PreOp	
0x1021	Information	System	EtherCAT state change PreOp - Init	
0x1024	Information	System	EtherCAT state change PreOp - Safe-Op	
0x1042	Information	System	EtherCAT state change SafeOp - PreOp	
0x1048	Information	System	EtherCAT state change SafeOp - Op	
0x1084	Information	System	EtherCAT state change Op - SafeOp	
0x1100	Information	General	Detection of operation mode completed: 0x%X, %d	Detection of the mode of operation ended
0x1135	Information	General	Cycle time o.k.: %d	Cycle time OK
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved manually
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved automatically
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted
0x117F	Information	General	Information: 0x%X, 0x%X, 0x%X	Information
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation.
0x1300	Information	Encoder	Position set: %d, %d	Position set - StartInputhandler
0x1303	Information	Encoder	Encoder Supply ok	Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully completed
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command
0x1400	Information	Drive	Drive is calibrated: %d, %d	Drive is calibrated
0x1401	Information	Drive	Actual drive state: 0x%X, %d	Current drive status
0x1705	Information		CPU usage returns in normal range (< 85%%)	Processor load is back in the normal range
0x1706	Information		Channel is not in saturation anymore	Channel is no longer in saturation
0x1707	Information		Channel is not in overload anymore	Channel is no longer overloaded
0x170A	Information		No channel range error anymore	A measuring range error is no longer active
0x170C	Information		Calibration data saved	Calibration data were saved
0x170D	Information		Calibration data will be applied and saved after sending the command “0x5AFE”	Calibration data are not applied and saved until the command "0x5AFE" is sent.

Text ID	Type	Place	Text Message	Additional comment
0x2000	Information	System	%s: %s	
0x2001	Information	System	%s: Network link lost	Network connection lost
0x2002	Information	System	%s: Network link detected	Network connection found
0x2003	Information	System	%s: no valid IP Configuration - Dhcp client started	Invalid IP configuration
0x2004	Information	System	%s: valid IP Configuration (IP: %d.%d.%d.%d) assigned by Dhcp server %d.%d.%d.%d	Valid IP configuration, assigned by the DHCP server
0x2005	Information	System	%s: Dhcp client timed out	DHCP client timeout
0x2006	Information	System	%s: Duplicate IP Address detected (%d.%d.%d.%d)	Duplicate IP address found
0x2007	Information	System	%s: UDP handler initialized	UDP handler initialized
0x2008	Information	System	%s: TCP handler initialized	TCP handler initialized
0x2009	Information	System	%s: No more free TCP sockets available	No free TCP sockets available.

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d.%d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.

Text ID	Type	Place	Text Message	Additional comment
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x %X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. • The EtherCAT connection was interrupted during operation. • The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	• There is no voltage applied to the power contacts. • A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel. %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel. %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x %X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing

Text ID	Type	Place	Text Message	Additional comment
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

7.2 Notes on Diag Messages associated with Motor Terminals

„Ack. Message“ Button

The ‚Ack. Message’ button has no effect on the Drive State Machine of the Motor terminals, pressing the button does not make an axis reset.
The Drive State Machine has no influence on the error list, an axis reset also does not remove any entries from the error list, however, this can be done by pressing the ‚Ack. Message’ button.

8 Appendix

8.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

8.2.1 Device description ESI file/XML

NOTE

Attention regarding update of the ESI description/EEPROM

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

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

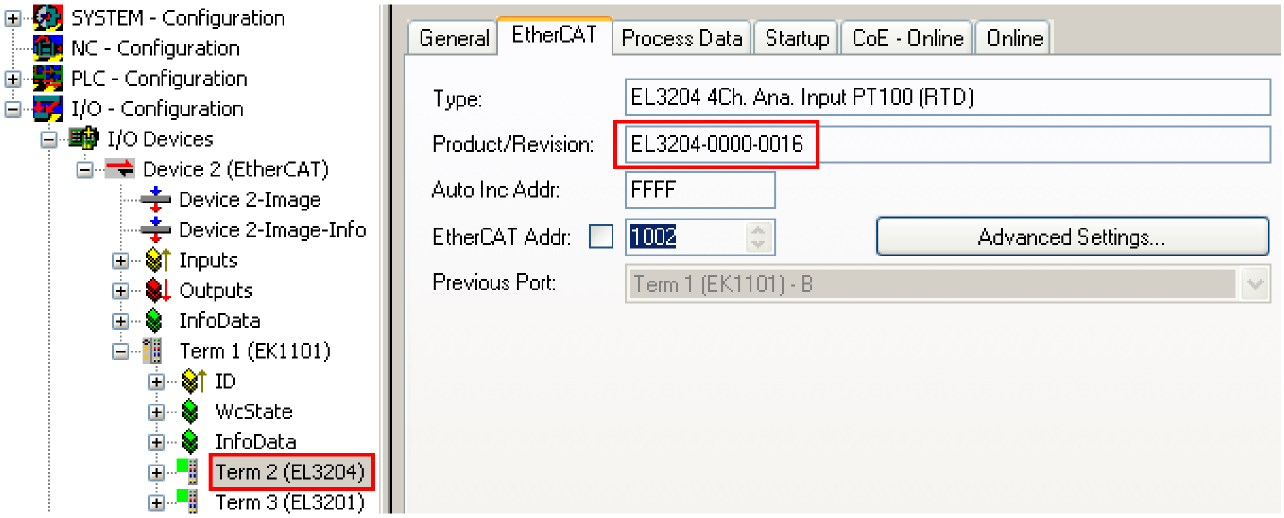


Fig. 225: Device identifier consisting of name EL3204-0000 and revision -0016

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

For further information on this, please refer to the EtherCAT system documentation.

Update of XML/ESI description

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

Display of ESI slave identifier

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

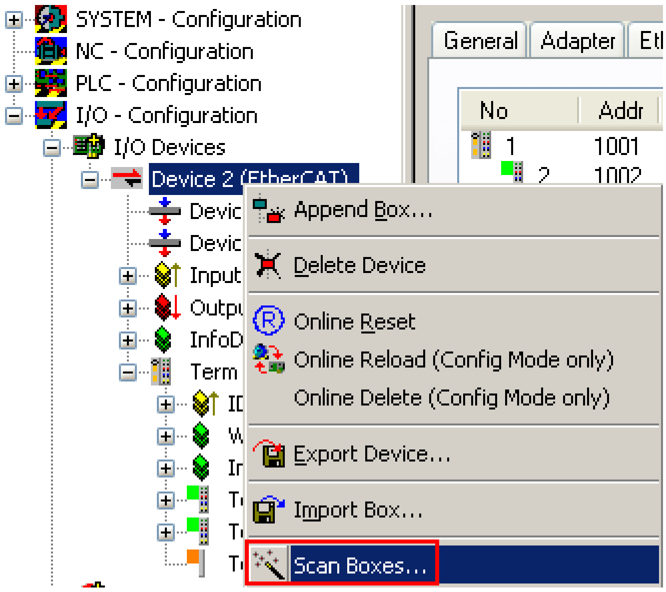


Fig. 226: Scan the subordinate field by right-clicking on the EtherCAT device

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

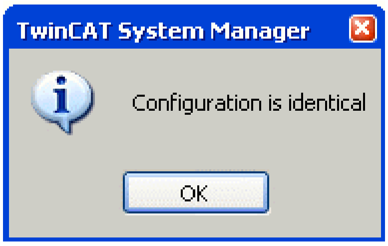


Fig. 227: Configuration is identical

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

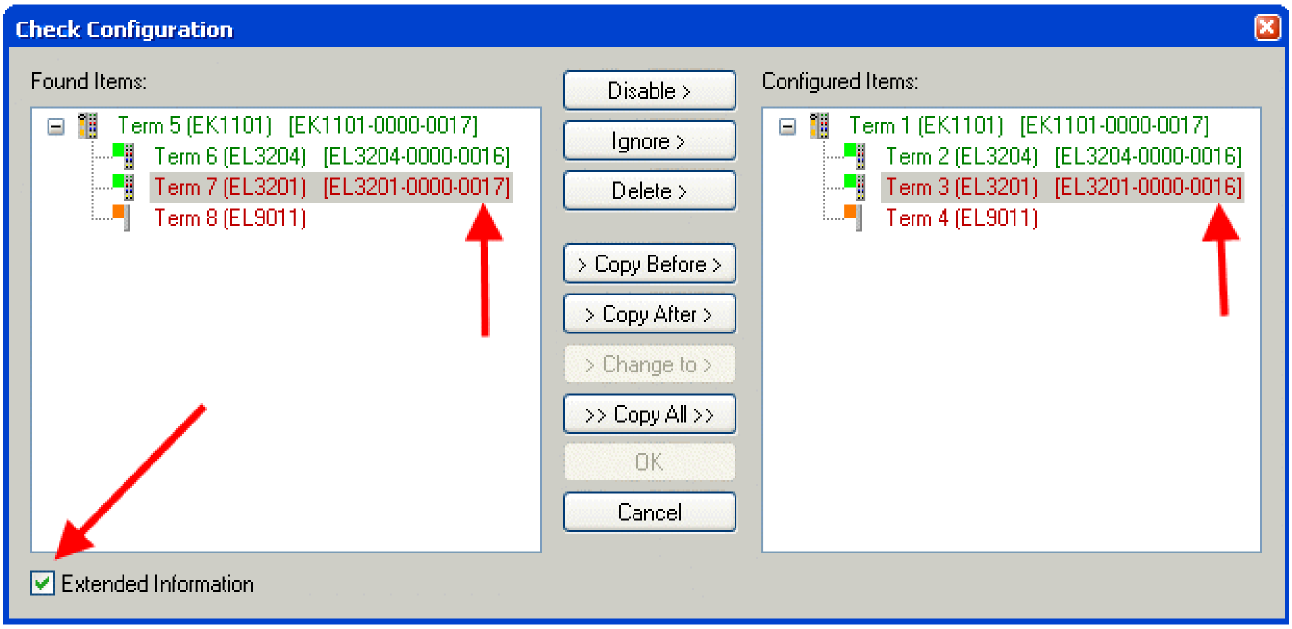


Fig. 228: Change dialog

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

Changing the ESI slave identifier

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

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

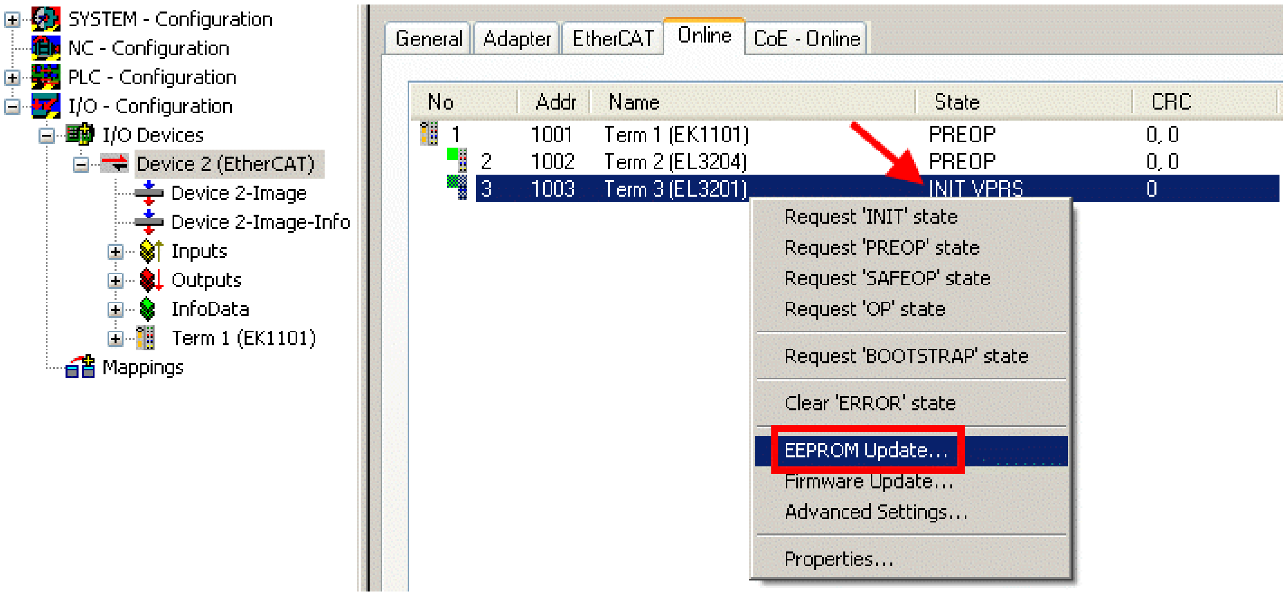


Fig. 229: EEPROM Update

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

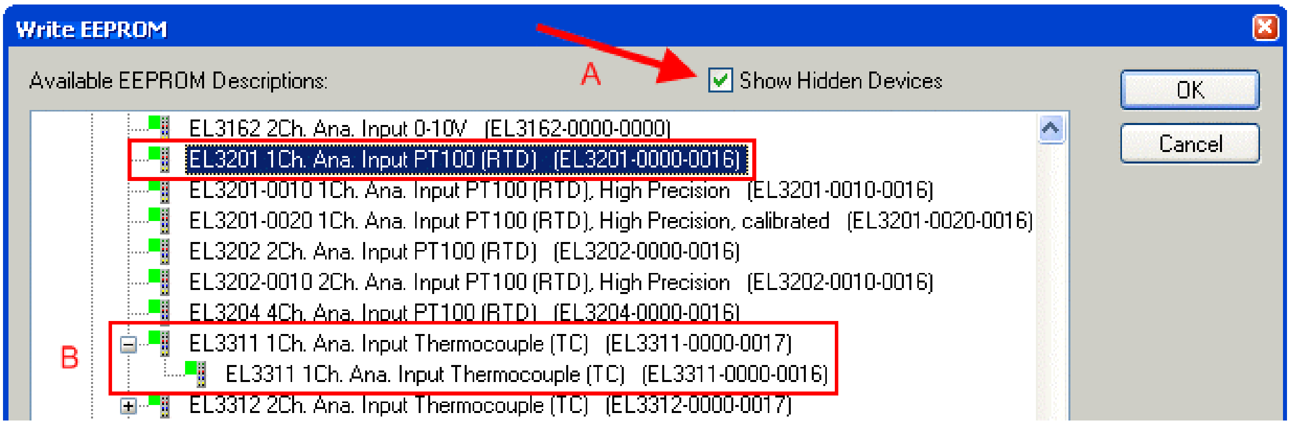


Fig. 230: Selecting the new ESI

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

The change only takes effect after a restart.

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

8.2.2 Firmware explanation

Determining the firmware version

Determining the version on laser inscription

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

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

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

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

Determining the version via the System Manager

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

CoE Online and Offline CoE

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

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

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

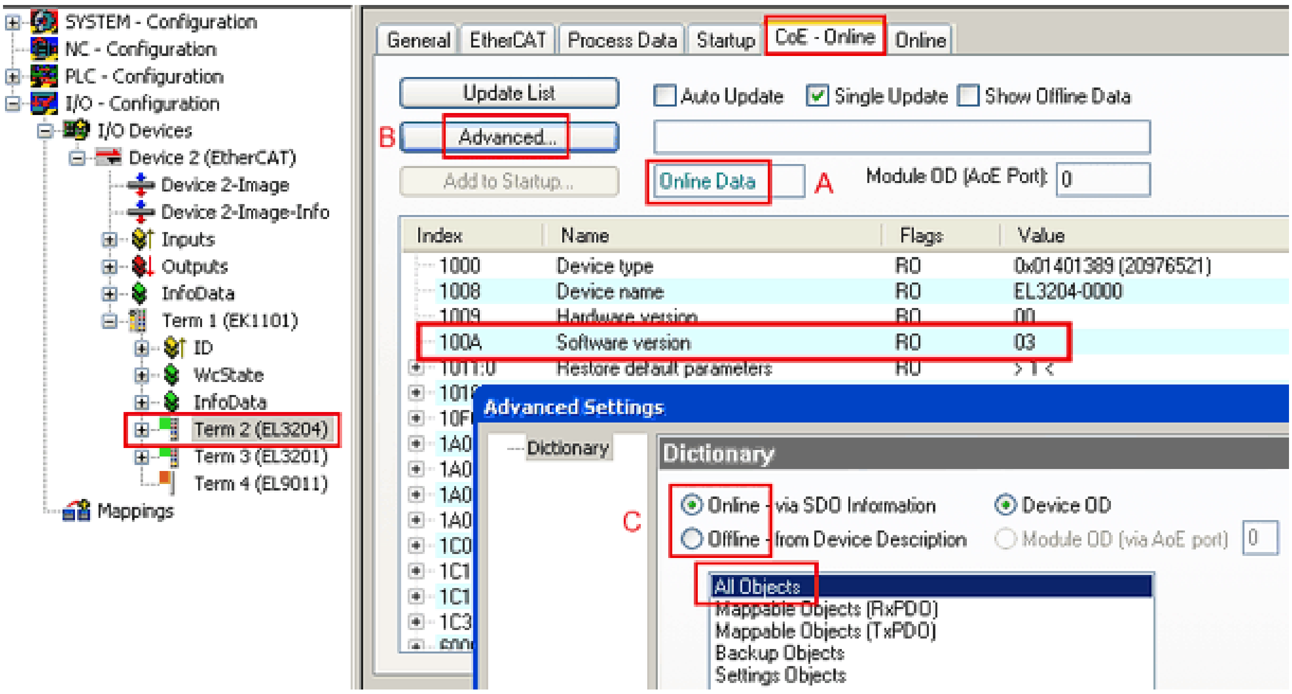


Fig. 231: Display of EL3204 firmware version

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*.

8.2.3 Updating controller firmware *.efw

CoE directory

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

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

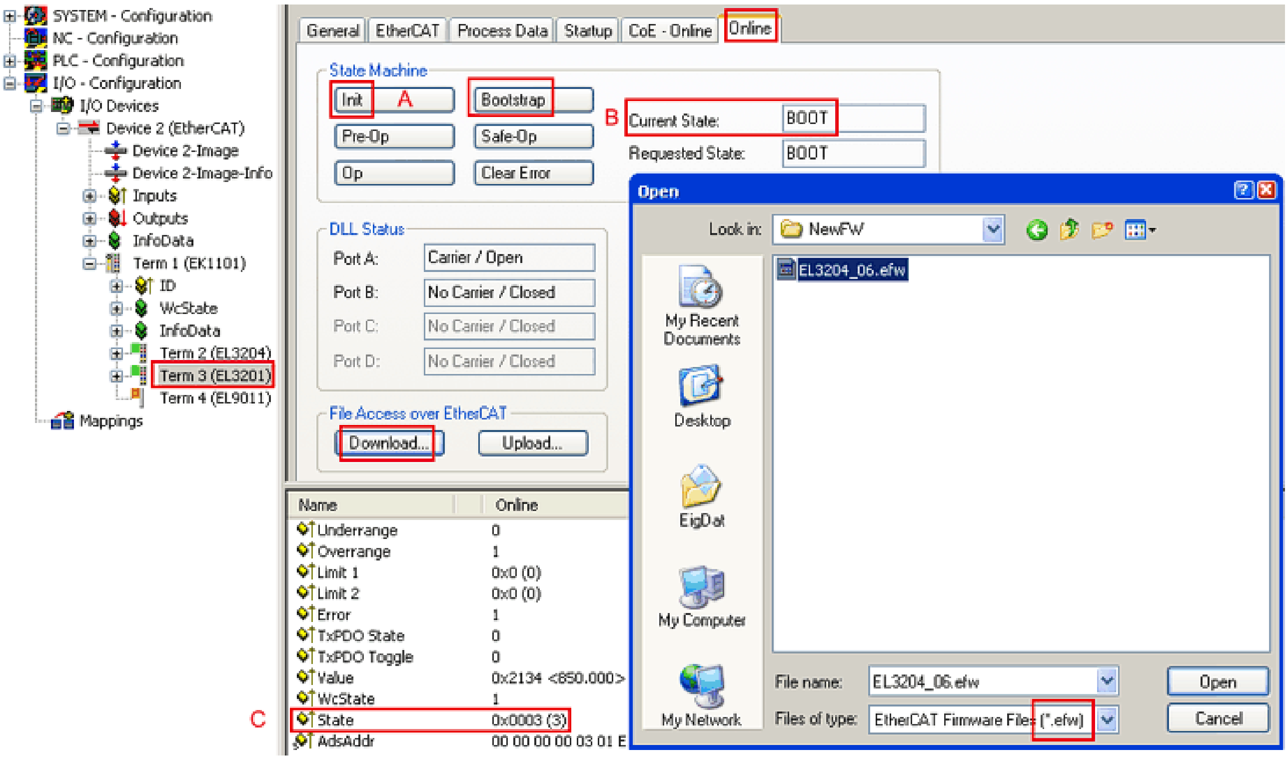
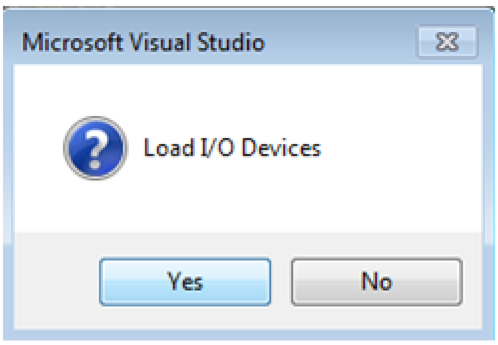
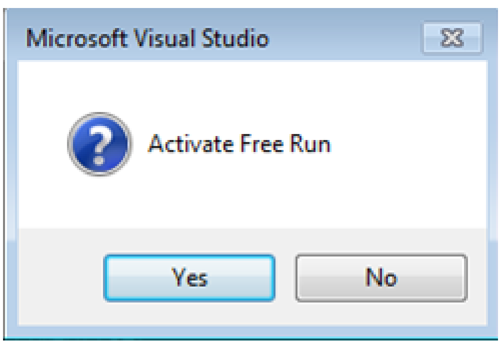


Fig. 232: Firmware Update

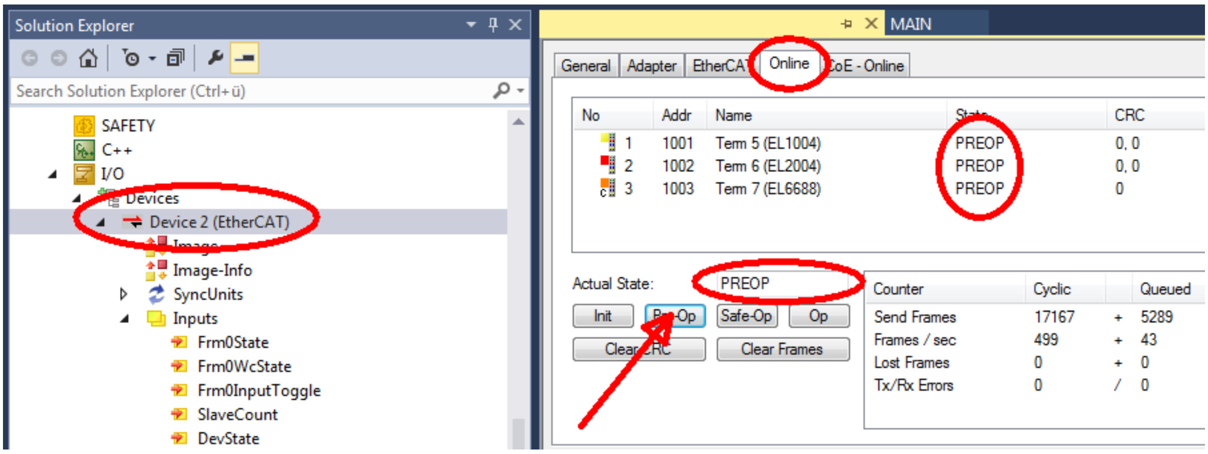
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

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

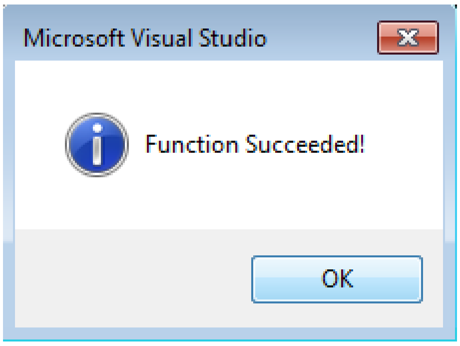




- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A pass word will not be neccessary usually.



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

8.2.4 FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

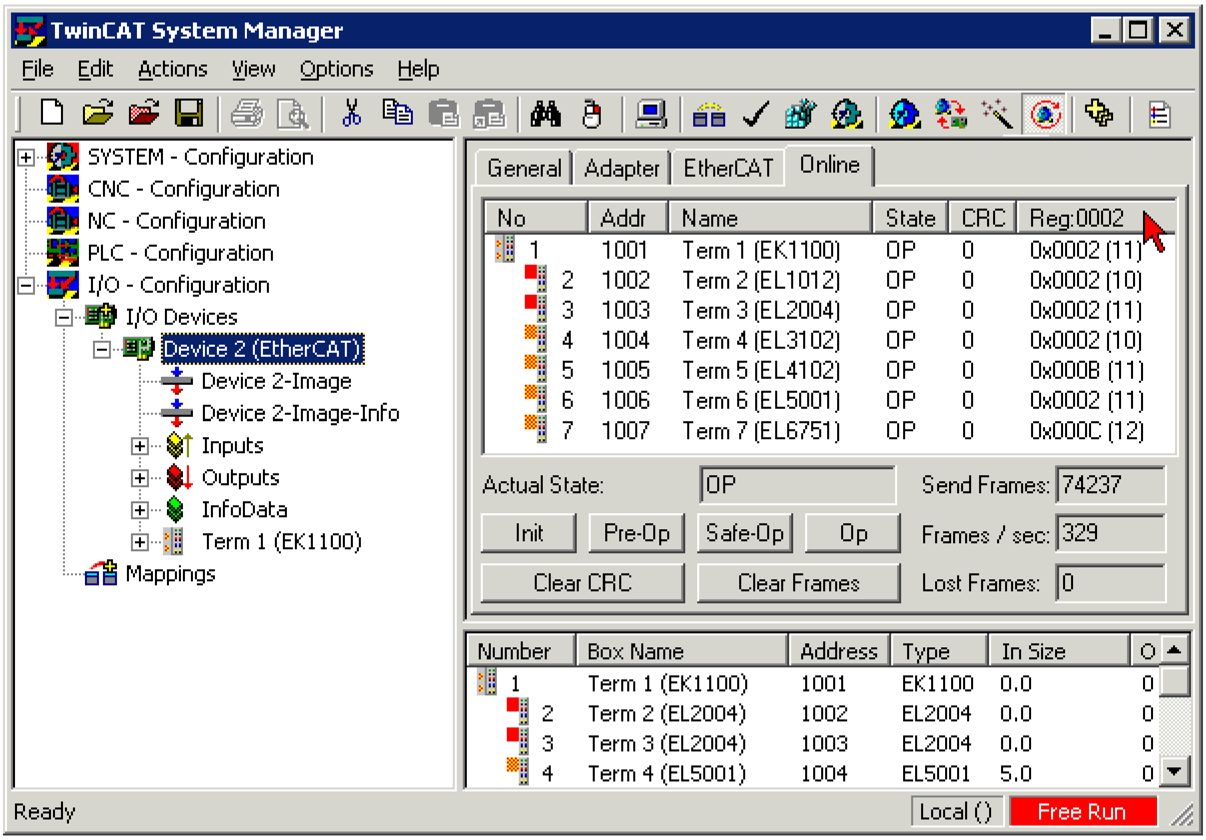


Fig. 233: FPGA firmware version definition

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

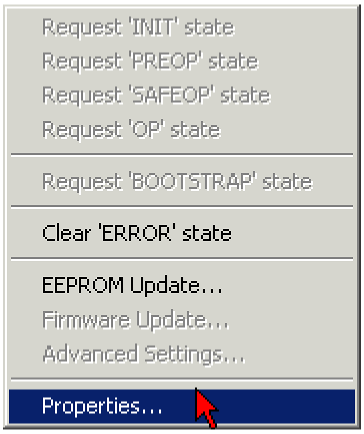


Fig. 234: Context menu *Properties*

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.

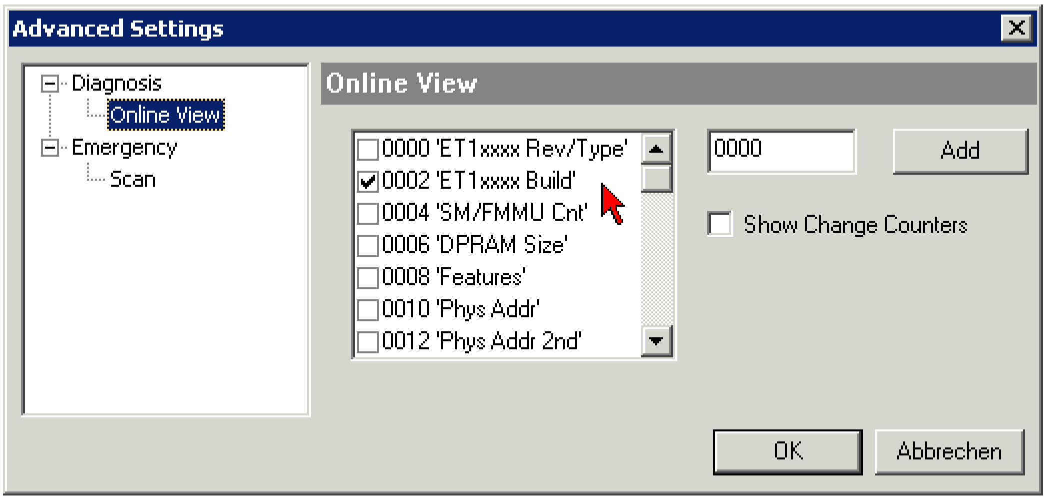


Fig. 235: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

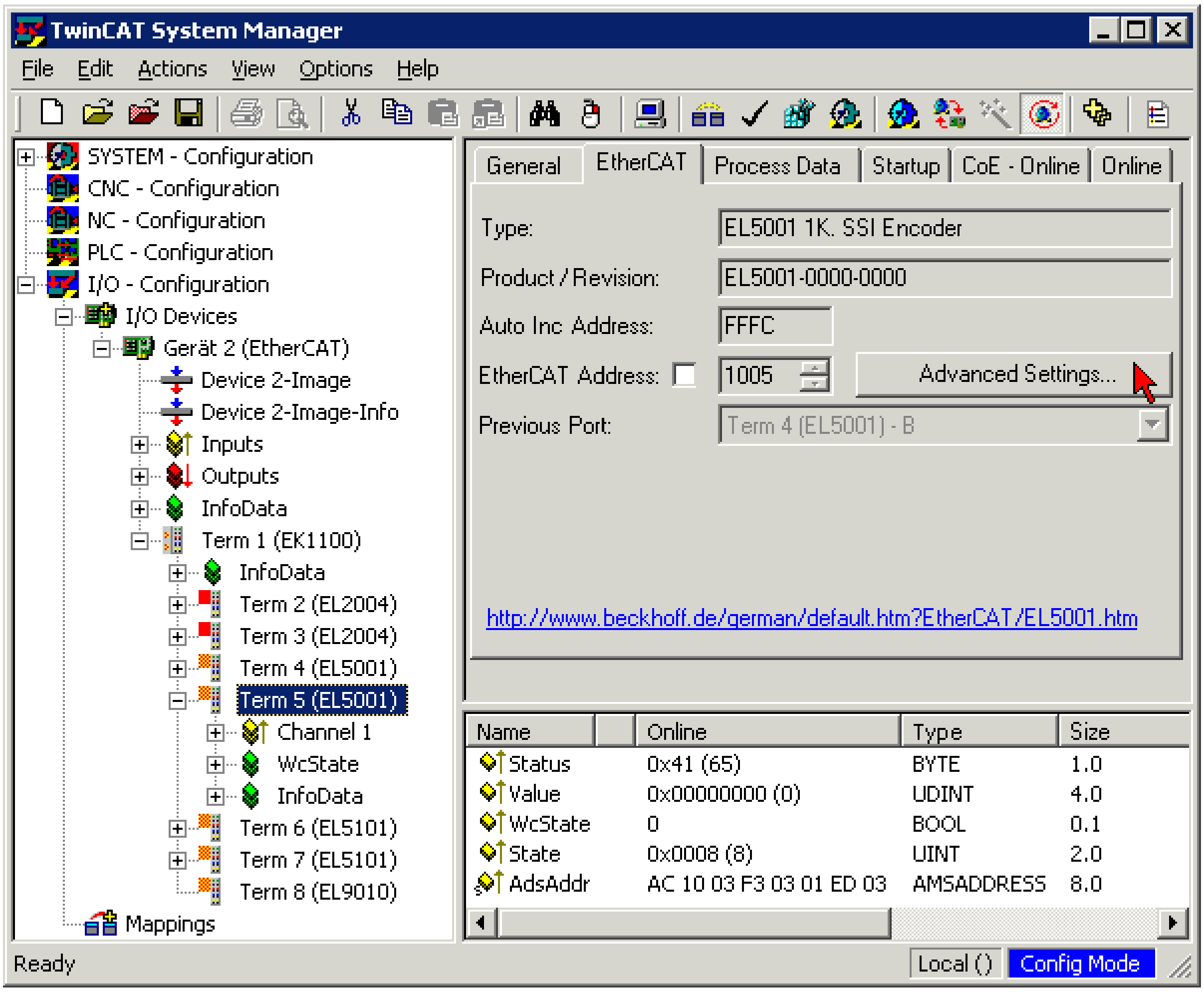
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

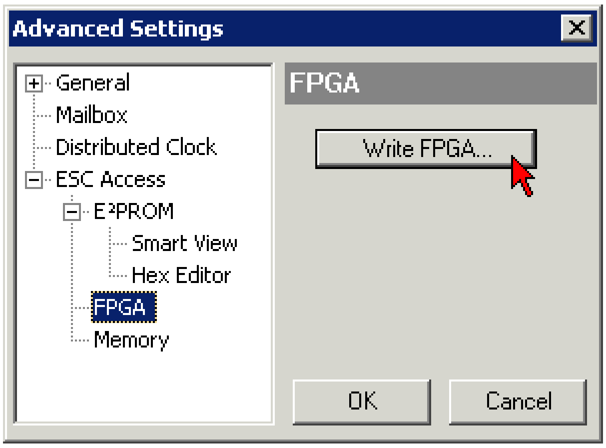
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

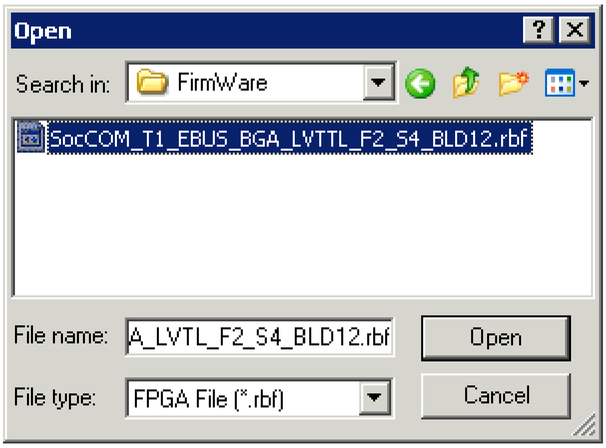
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab:



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



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



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

NOTE

Risk of damage to the device!

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

8.2.5 Simultaneous updating of several EtherCAT devices

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

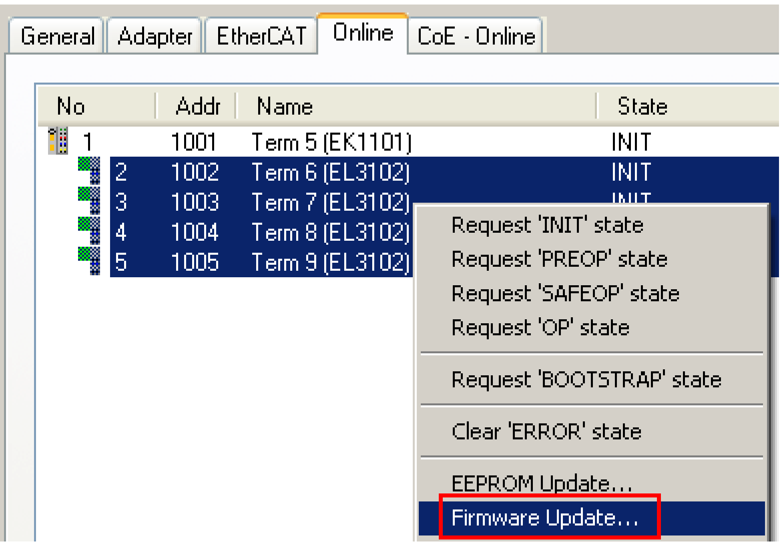


Fig. 236: Multiple selection and firmware update

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

8.3 Firmware compatibility

Beckhoff EtherCAT devices will be equipped with the last available firmware on delivery. There are certain dependencies between firmware and hardware, not all combinations are compatible.

The overview below shows which firmware is compatible to a certain hardware status.

Notice

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

NOTE

Damage of devices possible!

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

EL7037			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00 - 05*	01	EL7037-0000-0016	2015/02
	02	EL7037-0000-0017	2015/07
	03	EL7037-0000-0018	2016/06
	04*	EL7037-0000-0019	2017/03

EL7047			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01 – 09*	01	EL7047-0000-0016	2014/07
	02	EL7047-0000-0017	2015/01
	03	EL7047-0000-0018	2015/08
	04	EL7047-0000-0019	2016/06
	05	EL7047-0000-0020	2017/03
07 – 11*	06*	EL7047-0000-0021	2019/04

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

8.4 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*)

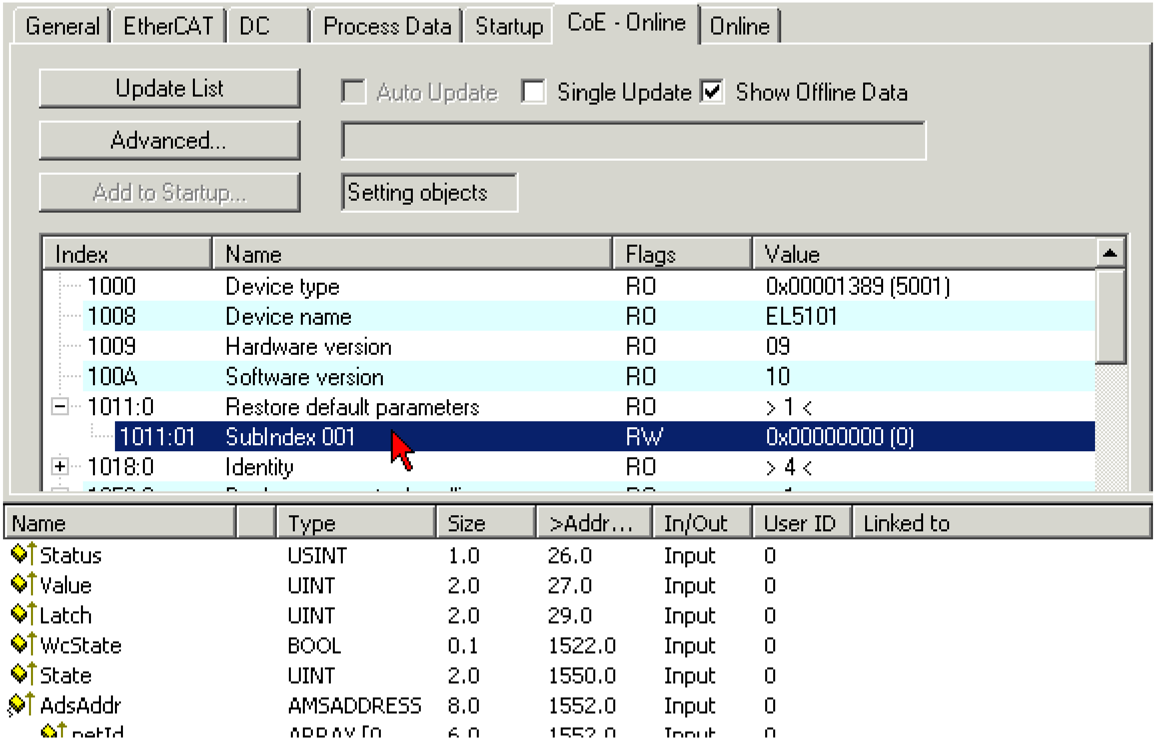


Fig. 237: Selecting the *Restore default parameters* PDO

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.

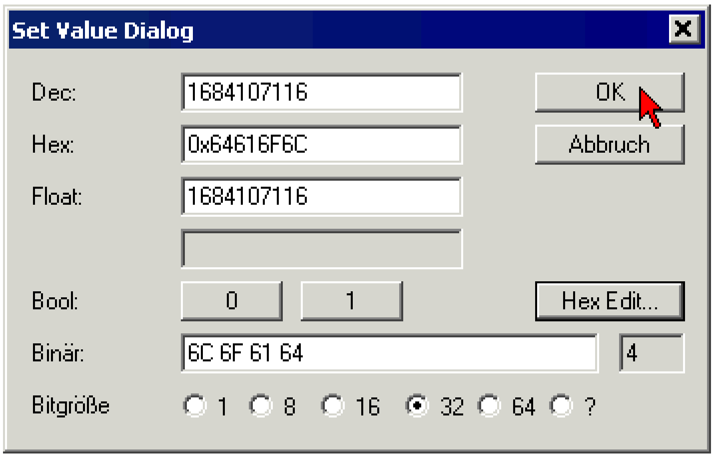


Fig. 238: Entering a restore value in the Set Value dialog

Alternative restore value

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

8.5 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 5246 963 0
Fax:	+49 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 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

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

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

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

Table of figures

More Information:
beckhoff.de/EL7037/

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