BECKHOFF New Automation Technology

Documentation | EN

EPP7041-x002

Stepper motor module 48 V DC with incremental encoder

2023-02-01 | Version: 1.1

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 Safety instructions 6
1.3 Documentation Issue Status 7
2 Product group: EtherCAT P Box modules 8
3 Product overview 9
3.1 Module overview 9
3.2 Introduction 10
3.3 Technical data 11
3.4 Scope of supply 13
3.5 Process image 14
3.5.1 "Predefined PDO Assignments" 14
3.5.2 Process data objects 16
3.6 Technology 21
3.6.1 Stepper motor parameters 21
3.6.2 Selecting a stepper motor 23
4 Mounting and connections 24
4.1 Mounting 24
4.1.1 Dimensions 24
4.1.2 Fixing 25
4.2 Functional earth (FE) 25
4.3 Connections 26
4.3.1 Overview 26
4.3.2 EtherCAT P 27
4.3.3 Stepper motor connection: X01 31
4.3.4 DC link voltage input: X02 33
4.3.5 Digital inputs and outputs: X03 34
4.3.6 Incremental encoders: X04 35
4.3.7 Status LEDs 36
4.4 Disposal 37
5 Commissioning and configuration 38
5.1 Integrating EPP7041 into a TwinCAT project 38
5.2 Parameterizing EPP7041 39
5.2.1 Open the parameter directory (CoE) 39
5.2.2 Setting important motor parameters 40
5.2.3 Setting other important parameters 42
5.3 Setting the operating mode 43
5.3.1 Operating modes 44
5.4 Parameterizing the NC axis 45
5.4.1 Parameterizing the encoder 47
5.4.2 Parameterizing the controller 49
5.5 Performing a test run 51
5.5.1 Test run with TwinCAT NC 51

5.5.2 Test run without the TwinCAT NC ... 52
5.6 Further applications ... 53
5.6.1 Using the "Positioning Interface" ... 53
5.6.2 Linking an NC axis with EPP7041-x002 ... 68
5.6.3 Restoring the delivery state ... 69
5.7 Decommissioning ... 70
6 Diagnosis ... 71
6.1 Diagnostics – basic principles of diag messages ... 71
6.2 Diag Messages of EtherCAT devices for drive technology ... 80
7 CoE parameters ... 81
7.1 Object directory ... 81
7.2 Data format of CoE parameters ... 83
7.3 Object description ... 84
7.3.1 Objects for parameterization ... 84
7.3.2 Standard objects ... 88
8 Appendix ... 90
8.1 General operating conditions ... 90
8.2 Accessories ... 91
8.3 Version identification of EtherCAT devices ... 92
8.3.1 General notes on marking ... 92
8.3.2 Version identification of EP/EPI/EPP/ER/ERI boxes ... 93
8.3.3 Beckhoff Identification Code (BIC) ... 94
8.3.4 Electronic access to the BIC (eBIC) ... 96
8.4 Support and Service ... 98

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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
1.1	• EtherCAT P status LEDs updated
1.0	• First release

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH	Example with D no. 29 10 02 01:
WW - week of production (calendar week)	29 - week of production 29
YY - year of production	10 - year of production 2010
FF - firmware version	02 - firmware version 02
HH - hardware version	01 - hardware version 01

Further information on this topic: Version identification of EtherCAT devices [▶ 92].

2 Product group: EtherCAT P Box modules

EtherCAT P

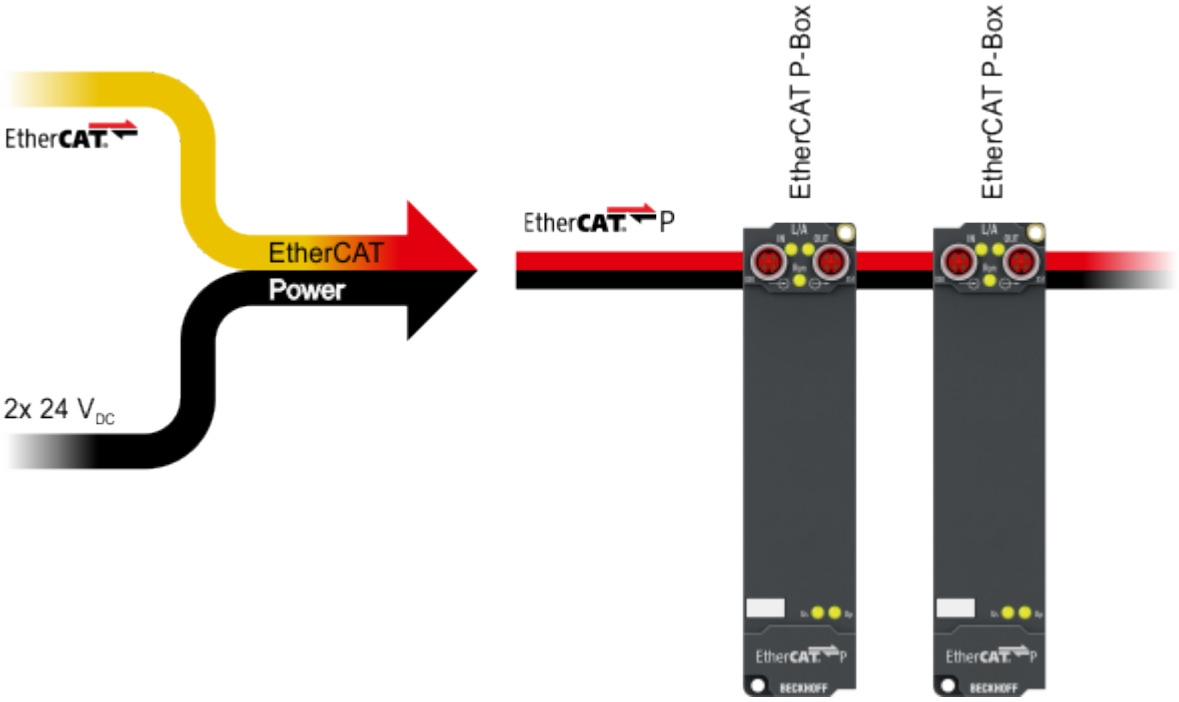
EtherCAT P supplements the EtherCAT technology with a process in which communication and supply voltages are transmitted on a common line. All EtherCAT properties are retained with this process.

Two supply voltages are transmitted per EtherCAT P line. The supply voltages are electrically isolated from each other and can therefore be switched individually. The nominal supply voltage for both is 24 V_{DC}.

EtherCAT P uses the same cable structure as EtherCAT: a 4-core Ethernet cable with M8 connectors. The connectors are mechanically coded so that EtherCAT connectors and EtherCAT P connectors cannot be interchanged.

EtherCAT P Box modules

EtherCAT P Box modules are EtherCAT P slaves with degree of protection IP67. They are designed for operation in wet, dirty or dusty industrial environments.



EtherCAT basics

A detailed description of the EtherCAT system can be found in the EtherCAT system documentation.

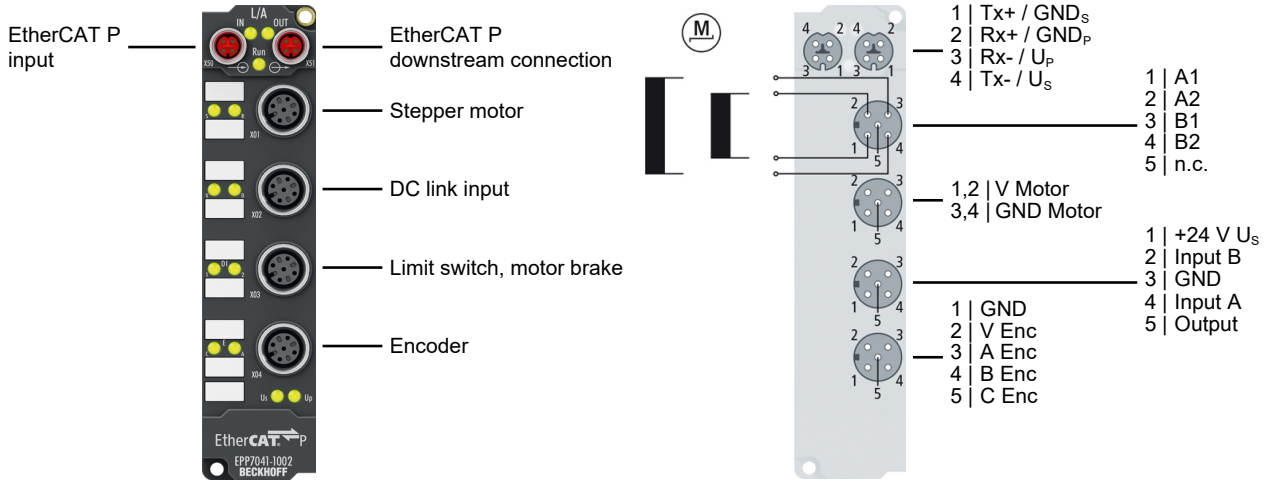
3 Product overview

3.1 Module overview

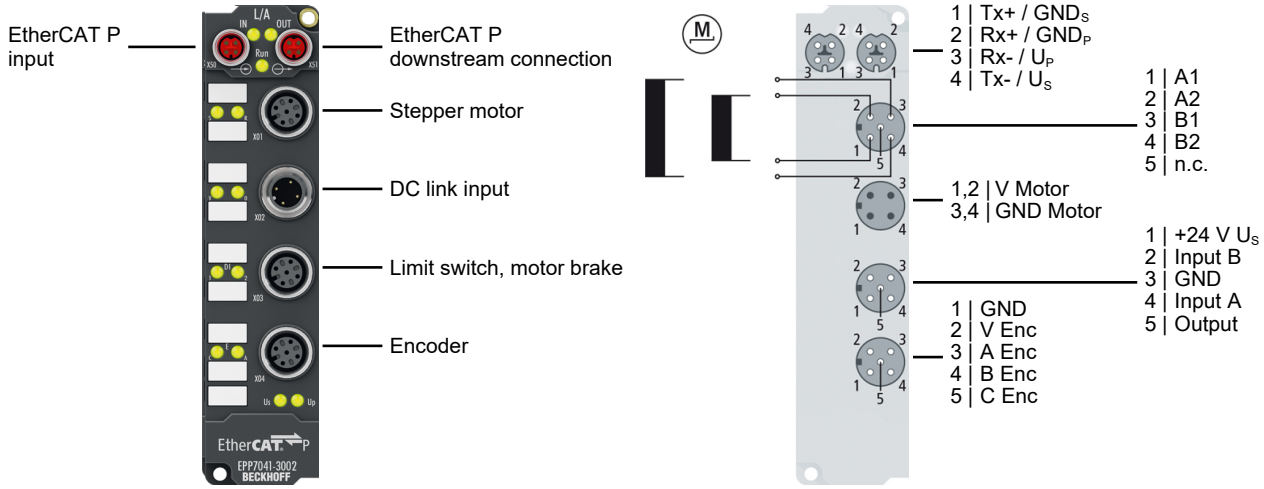
Module	Rated current per phase	Peak current per phase	Microstepping
EPP7041-1002	1.0 A	1.5 A	up to 64-fold
EPP7041-3002	3.5 A	5.0 A	up to 256-fold

3.2 Introduction

EPP7041-1002



EPP7041-3002



Schrittmotormodul 48 V_{DC} mit Inkremental-Encoder

The EPP7041-1002 EtherCAT P Box is intended for the direct connection of different Stepper Motors. The PWM output stages for two motor coils with compact design are located in the module together with two inputs for limit switches and cover a wide voltage and current range. The EPP7041-1002 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. Connection of an incremental encoder enables a simple servo axis to be realised. Two digital inputs and a digital 0.5 A output enable connection of end switches and a motor brake.

The -3002 variant is particularly suitable for applications that are subject to unsteady motor operation due to natural resonance of the motor and the moved mass.

Quick links

Technical data [▶ 11]
Connections [▶ 26]
Commissioning [▶ 38]

3.3 Technical data

All values are typical values over the entire temperature range, unless stated otherwise.

EtherCAT P	
Connection	2 x M8 socket, 4-pin, P-coded, red
Distributed Clocks	yes

Supply voltages	
Connection	See EtherCAT P connection
U_S nominal voltage	24 V_{DC} (-15 % / +20 %)
U_S sum current: $I_{S,sum}$	max. 3 A
Current consumption from U_S	100 mA + current consumption of the encoder + current consumption of the motor brake + auxiliary voltage for the limit switches
U_P nominal voltage	24 V_{DC} (-15 % / +20 %)
U_P sum current: $I_{P,sum}$	max. 3 A
Current consumption from U_P	None. U_P is only forwarded.

Stepper motor	EPP7041-1002	EPP7041-3002
Motor type	2-phase stepper motor, unipolar or bipolar	
Connection	1 x M12 socket	
DC link voltage	8…48 V_{DC}	
Connection for the DC link voltage	M12 socket	M12 connector
Nominal current per phase	1.0 A	3.5 A
Peak current per phase	1.5 A	5.0 A at 50 °C
Step frequency	max. 32000 full steps per second	
Microstepping	up to 64-fold	up to 256-fold
Current controller frequency	approx. 30 kHz	
Resolution	approx. 5000 positions per revolution in typical applications	
Protective functions	Overload protection, short-circuit protection	

Encoder input	
Number	1
encoder type	Incremental encoder with single-ended output drivers
Connection	1x M12 socket, 5-pin
Encoder supply	24 V_{DC} from the control voltage U_S max. 0.5 A, not short-circuit proof
signals	A, B, C; single-ended (C = reference pulse / zero pulse)
Signal voltage "0"	-3…2 V
Signal voltage "1"	3.5…28 V
Pulse frequency	max. 400,000 increments per second (quadruple evaluation)

Digital inputs for limit switches	
Number	2
Connection	M12 socket
Nominal voltage high level	24 V_{DC}
Signal voltage "0"	-3…2 V
Signal voltage "1"	3.5…28 V
Input current	5 mA

Digital output for the motor brake	
Connection	M12 socket
Output voltage high level	24 V_{DC} from the control voltage U_S
Output current	max. 0.5 A

Housing data	
Dimensions W x H x D	30 mm x 126 mm x 26.5 mm (without connectors)
Weight	approx. 165 g
Installation position	variable
Material	PA6 (polyamide)

Environmental conditions	
Ambient temperature during operation	-25 … +60 °C
Ambient temperature during storage	-40 … +85 °C
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 conforms to EN 60529

Approvals	
Approvals	CE, UL in preparation

Additional tests

The devices have undergone the following additional tests:

Test	Explanation
Vibration	10 frequency sweeps in 3 axes
	5 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
	35 g, 11 ms

Also see about this

Technical data [▶ 12]

3.4 Scope of supply

Make sure that the following components are included in the scope of delivery:

- 1x EPP7041-x002
- 2x protective cap for EtherCAT P socket, M8, red (pre-assembled)
- 10x labels, blank (1 strip of 10)

Pre-assembled protective caps do not ensure IP67 protection

Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

Ensure that the protective caps are correctly seated to ensure IP67 protection.

3.5 Process image

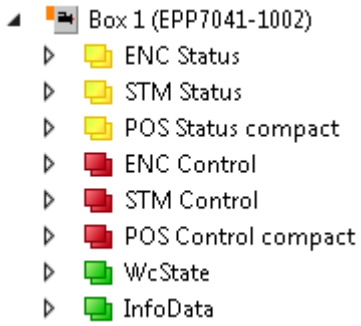
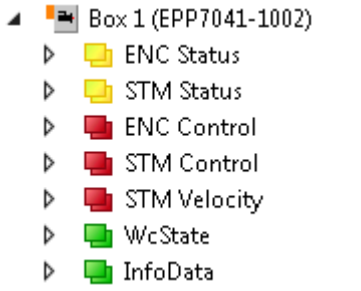
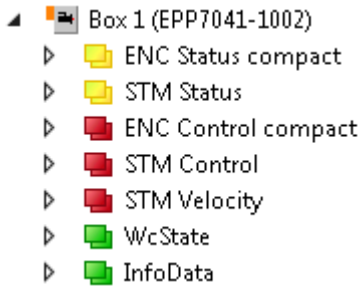
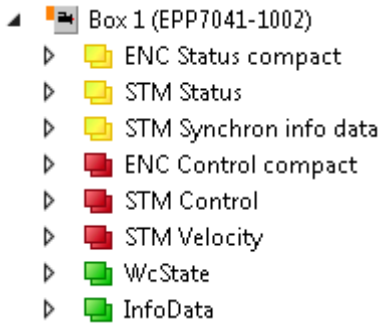
The scope of the process image is adjustable.

EP7047-1032 has several predefined variants of the process image: "Predefined PDO Assignments". Select the "Predefined PDO Assignment" according to the operating mode.

The factory default setting is "Velocity control compact" [▶ 15].

3.5.1 "Predefined PDO Assignments"

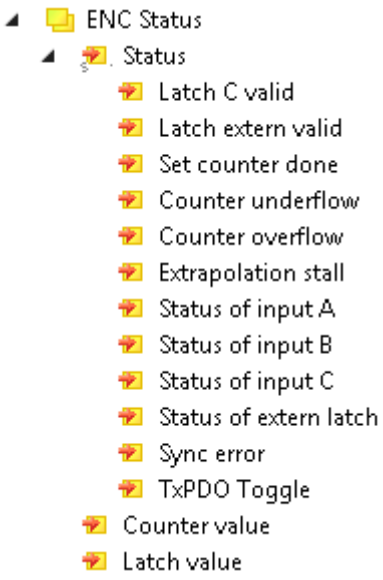
Name	Process image	Process data objects
Position control	Box 1 (EPP7041-1002) ENC Status STM Status ENC Control STM Control STM Position WcState InfoData	ENC Status [▶ 16] STM Status [▶ 18] ENC Control [▶ 19] STM Control [▶ 20] STM Position [▶ 20]
Positioning interface	Box 1 (EPP7041-1002) ENC Status STM Status POS Status ENC Control STM Control POS Control WcState InfoData	ENC Status [▶ 16] STM Status [▶ 18] POS Status [▶ 17] ENC Control [▶ 19] STM Control [▶ 20] POS Control [▶ 19]
Positioning interface (Auto start)	Box 1 (EPP7041-1002) ENC Status STM Status POS Status ENC Control STM Control POS Control POS Control 2 WcState InfoData	ENC Status [▶ 16] STM Status [▶ 18] POS Status [▶ 17] ENC Control [▶ 19] STM Control [▶ 20] POS Control [▶ 19] POS Control 2 [▶ 19]
Positioning interface (Auto start) with info data	Box 1 (EPP7041-1002) ENC Status STM Status STM Synchron info data POS Status ENC Control STM Control POS Control POS Control 2 WcState InfoData	ENC Status [▶ 16] STM Status [▶ 18] STM Synchron info data [▶ 18] POS Status [▶ 17] ENC Control [▶ 19] STM Control [▶ 20] POS Control [▶ 19] POS Control 2 [▶ 19]

Name	Process image	Process data objects
Positioning interface compact		ENC Status [▶ 16] STM Status [▶ 18] POS Status compact [▶ 17] ENC Control [▶ 19] STM Control [▶ 20] POS Control compact [▶ 19]
Positioning interface with info data		ENC Status [▶ 16] STM Status [▶ 18] STM Synchron info data [▶ 18] POS Status [▶ 17] ENC Control [▶ 19] STM Control [▶ 20] POS Control [▶ 19]
Velocity control		ENC Status [▶ 16] STM Status [▶ 18] ENC Control [▶ 19] STM Control [▶ 20] STM Velocity [▶ 20]
Velocity control compact (*Factory setting*)		ENC Status compact [▶ 16] STM Status [▶ 18] ENC Control compact [▶ 19] STM Control [▶ 20] STM Velocity [▶ 20]
Velocity control compact with info data		ENC Status compact [▶ 16] STM Status [▶ 18] STM Synchron info data [▶ 18] ENC Control compact [▶ 19] STM Control [▶ 20] STM Velocity [▶ 20]

3.5.2 Process data objects

3.5.2.1 "ENC status"

"ENC Status" contains the status variables of the encoder input. "ENC" is the abbreviation for "Encoder".



State

- **Latch C valid**: A signal edge has been detected at encoder signal "C". As a result, the "Counter value" was written to the variable "Latch value" at the time of the signal edge.[1)]
- **Latch extern valid**: A signal edge was detected at the latch input. As a result, the counter value was written to the variable "Latch value" at the time of the signal edge.[1)]
- **Set counter done**: The value from "Set counter value" was written to the variable "Counter value" after setting of "Set counter" (ENC Control).
- **Counter underflow**: The counter value "Counter value" has fallen below the value 0.
- **Counter overflow**: The counter value "Counter value" has exceeded the maximum value.
- **Extrapolation stall**: The extrapolated part of the counter is invalid ("Micro increments").
- **Status of input A**: current signal level of encoder signal "A"
- **Status of input B**: current signal level of encoder signal "B"
- **Status of input C**: current signal level of encoder signal "C"
- **Status of extern latch**: current signal level at the latch input
- **Sync error**: Distributed Clocks synchronization error in the previous cycle.
- **TxPDO Toggle**: This bit is inverted each time an input data update occurs.

Counter value: The current counter value.

Latch value: Counter value stored at the time of the last signal edge at latch input or encoder signal "C".[1)]

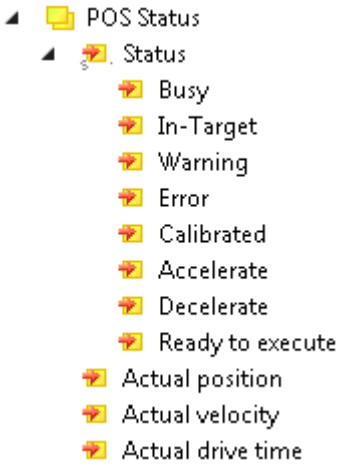
[1)] The latch function is deactivated in the factory setting. You may activate and configure the latch function in process data object "ENC Control" [▶ 19] or "ENC Control compact" [▶ 19].

3.5.2.2 "ENC Status compact"

This process data object is identical with "ENC status" [▶ 16], see there.

3.5.2.3 "POS Status"

"POS Status" contains the status variables of the Positioning Interface [▶ 53].



Status

- **Busy**: A motion command is active.
- **In-Target**: The target position of the motion command has been reached.
- **Warning**: Warning message.
- **Error**: Error message.
- **Calibrated**: The motor is calibrated.
- **Accelerate**: The motor accelerates.
- **Decelerate**: The motor brakes.
- **Ready to execute**: Ready for a motion command.

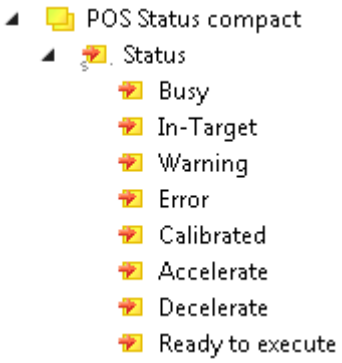
Actual position: current set position

Actual velocity: current set velocity

Actual drive time: the elapsed time of the motion command.

3.5.2.4 "POS Status compact"

"POS Status compact" contains the status variables of the Positioning Interface [▶ 53].

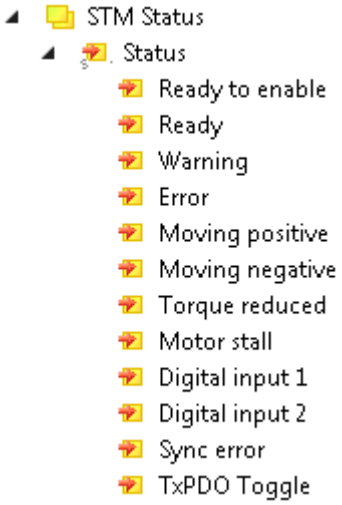


Status

This variable is identical to the "Status" variable in the process data object "POS Status [▶ 17]". See there.

3.5.2.5 "STM Status"

„STM Status" contains the status bits of the stepper motor output stage. "STM" is the abbreviation for "Stepper Motor".



Ready to enable: The output stage can be enabled. See output variable "Enable" in the process data object STM Control [▶ 20].

Ready: The output stage is enabled.

Warning: Warning message.

Error: Error message. The output stage was switched off due to an error. You can acknowledge the error message with the output variable "Reset" in the process data object STM Control [▶ 20]

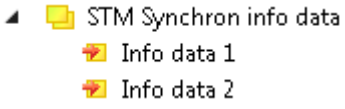
Moving positive: The speed is greater than 0.

Moving negative: The speed is less than 0.

Motor stall: A loss of step has occurred.

3.5.2.6 "STM Synchronous info data"

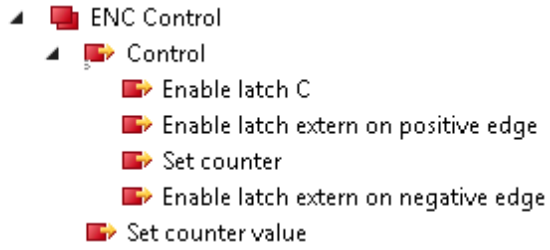
"STM" is the abbreviation for "Stepper Motor".



Info data *n*: Additional information from the box. You can select what information these variables should contain:

- Parameter 8012:11$_{hex}$ Select info data 1 [▶ 86]
- Parameter 8012:19$_{hex}$ Select info data 2 [▶ 86]

3.5.2.7 "ENC Control"



Enable latch C: Activate edge trigger for encoder input "C".

Enable latch extern on positive edge: Activate edge trigger for positive signal edges at latch input.

Set counter: Accept the value of the variable "Set counter value" as the current counter value.

Enable latch extern on negative edge: Activate edge trigger for negative signal edges at latch input.

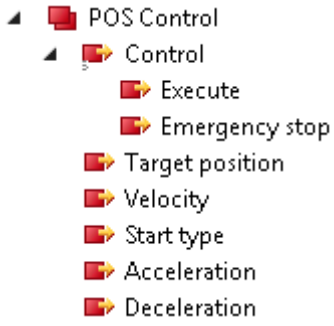
Set counter value: Default value for "Set counter".

3.5.2.8 "ENC Control compact"

This process data object is identical with "ENC Control" [▶ 19].

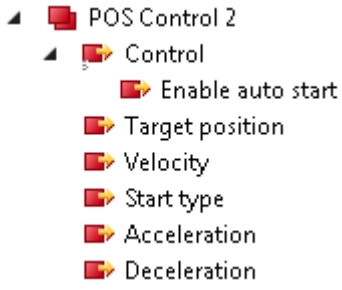
3.5.2.9 "POS Control"

This process data object contains variables for controlling the Positioning Interface [▶ 53].



3.5.2.10 "POS Control 2"

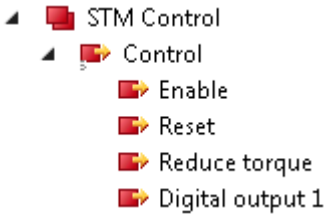
This process data object contains variables for controlling the Positioning Interface [▶ 53].



3.5.2.11 "POS Control compact"

This process data object contains variables for controlling the Positioning Interface [▶ 53].

3.5.2.12 "STM Control"



Enable: Enable output stage.

Reset: Acknowledge error message, reset error status. See input variable "Error" in the process data object STM Status [▶ 18]

3.5.2.13 "STM Position"

Position: Position setpoint.
Specify the position setpoint in increments.

Conversion from degrees (°) to increments: See below.

Conversion of position setpoints

The formula for converting a position setpoint from degrees (°) to increments depends on whether you are using an encoder.

- If you are not using an encoder (feedback type [▶ 86] = "Internal counter"), use this formula:

$$Position = \frac{\theta_{set} \times 64}{\varphi}$$

Position: Setpoint [increments]

Θ_{set}: Setpoint [°]

φ: Step angle of the motor [°]
(for AS10xx stepper motors: φ = 1.8 °)

- If you are using an encoder (feedback type [▶ 86] = "Encoder"), use this formula:

$$Position = \frac{\theta_{set} \times PPR}{90}$$

Position: Setpoint [increments]

Θ_{set}: Setpoint [°]

PPR: Resolution of the encoder [increments/revolution]
(for AS10xx stepper motors: inc = 1024)

3.5.2.14 "STM Velocity"

Velocity: Speed setpoint in % of the parameter "Speed range" [▶ 42].
32767_{dec} corresponds to 100%, -32767_{dec} corresponds to -100%.

Conversion of speed setpoints

$$Velocity = 196602 \times \frac{n_{set}}{\varphi \times f_{max}}$$

Velocity: Setpoint [increments/s]

n_{set}: Setpoint [rpm]

φ: Step angle of the motor [°]
(for AS10xx stepper motors: φ = 1.8 °)

f_{max}: "Speed range" [▶ 42] [full steps/s]

The speed setpoint can be positive or negative, depending on the desired direction of rotation of the motor.

3.6 Technology

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. 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.

3.6.1 Stepper motor parameters

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.

Speed

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

Nominal voltage, supply voltage and winding resistance

In steady state, the rated current flows at rated voltage, depending on the winding resistance. This voltage should not be confused with the supply voltage of the power output stage.
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.

Number of phases

Motors with 2 to 5 phases are common. EPP7041-x002 supports 2-phase motors. 4-phase motors are basically 2-phase motors with separately fed out winding ends. They can be connected directly to EPP7041-x002.

Resonance

At certain speeds, stepper motors run less smoothly. This phenomenon is particularly pronounced if the motor runs without load. Under certain circumstances, it may even stop. This is caused by resonance. A distinction can roughly be made between

- resonances in the lower frequency range up to approx. 250 Hz 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.

EPP7041-x002 prevents this effect by means of a SinCos-shaped current profile for almost all standard motors. The rotor is not switched from step to step, i.e. it no longer jumps to the next position, but instead passes through intermediate steps (microsteps), i.e. the rotor is gently guided from one step to the next. The usual loss of torque at certain speeds is avoided, and operation can be optimized for the particular application. This means that the lower speed range, where particularly high torque is available, can be fully utilized.

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 improving positioning accuracy is the microstepping function. 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.

3.6.2 Selecting a stepper motor

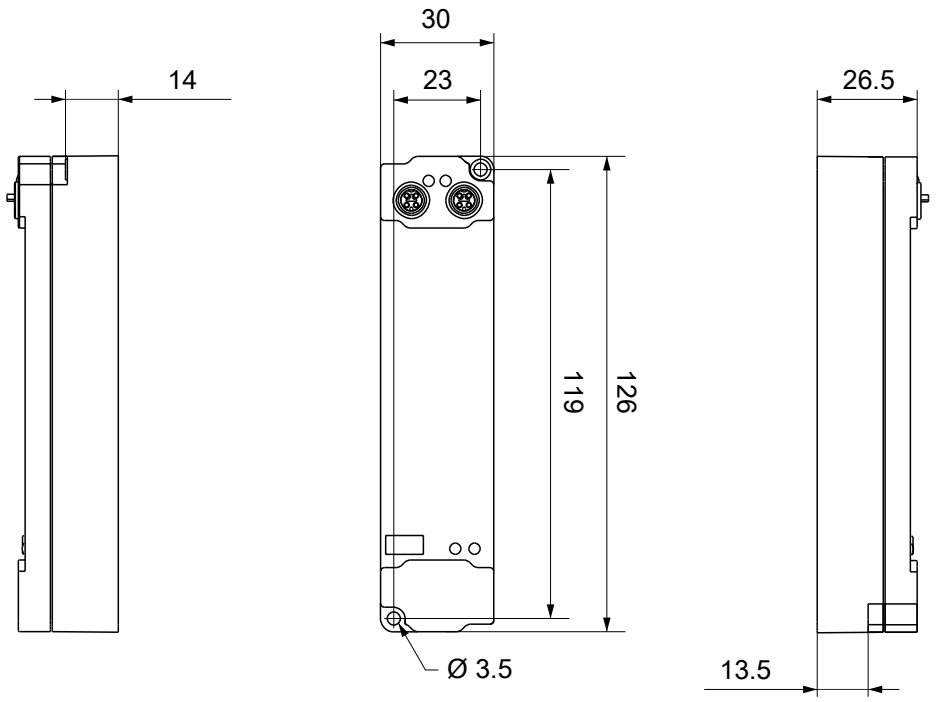
Specifying the stepper motor

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. Microstepping 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. EPP7041-x002 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.

4 Mounting and connections

4.1 Mounting

4.1.1 Dimensions



All dimensions are given in millimeters.
The drawing is not true to scale.

Housing features

Housing material	PA6 (polyamide)
Sealing compound	polyurethane
Mounting	two mounting holes Ø 3.5 mm for M3
Metal parts	brass, nickel-plated
Contacts	CuZn, gold-plated
Installation position	variable
Protection class	IP65, IP66, IP67 (conforms to EN 60529) when screwed together
Dimensions (H x W x D)	approx. 126 x 30 x 26.5 mm (without connectors)

4.1.2 Fixing

NOTE

Dirt during assembly

Dirty connectors can lead to malfunctions. Protection class IP67 can only be guaranteed if all cables and connectors are connected.

- Protect the plug connectors against dirt during the assembly.

Mount the module with two M3 screws on the mounting holes in the corners of the module. The mounting holes have no thread.

4.2 Functional earth (FE)

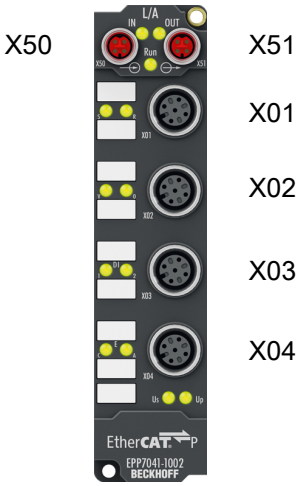
The upper mounting holes also serves as a connection for functional earth (FE).

Make sure that the box is grounded to low impedance via the functional earth (FE) connection. You can achieve this, for example, by mounting the box on a grounded machine bed.

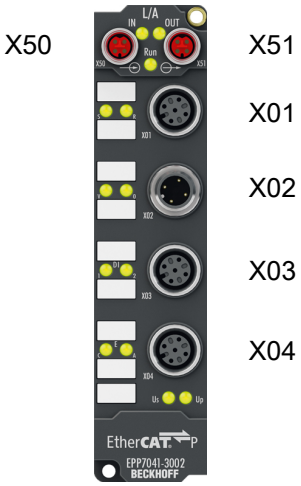
Fig. 1: Connection for functional earth (FE)

4.3 Connections

4.3.1 Overview



EPP7041-1002



EPP7041-3002

Name	Function	Connector type	Tightening torque
X01	Stepper motor connection [▶ 31]	M12 socket	0.6 Nm [1]
X02	DC link voltage input [▶ 33]	M12 socket (EPP7041-1002) M12 plug (EPP7041-3002)	0.6 Nm [1]
X03	Digital inputs for limit switches [▶ 34] Digital output for the motor brake [▶ 34]	M12 socket	0.6 Nm [1]
X04	Encoder connection [▶ 35]	M12 socket	0.6 Nm [1]
X50	EtherCAT P input [▶ 27]	M8 socket, p-coded	0.4 Nm [1]
X51	EtherCAT P downstream connection [▶ 27]	M8 socket, p-coded	0.4 Nm [1]

[1] Mount plugs on these connectors using a torque wrench, e.g. ZB8801 from Beckhoff.

Protective caps

- Seal unused connectors with protective caps.
- Ensure the correct seating of pre-assembled protective caps.
  Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

4.3.2 EtherCAT P

⚠ WARNING

Power supply from SELV/PELV power supply unit!

SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply the EtherCAT P Power Sourcing Device (PSD).

Notes:

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

EtherCAT P transmits two supply voltages:

- **Control voltage U_S**
  The following sub-functions are supplied from the control voltage U_S:
  - the fieldbus
  - the processor logic
  - typically the inputs and the sensors if the EtherCAT P Box has inputs.
- **Peripheral voltage U_P**
  The digital outputs are typically supplied from the peripheral voltage U_P for EtherCAT P Box modules with digital outputs. U_P can be supplied separately. If U_P is switched off, the fieldbus function, the function of the inputs and the supply of the sensors are maintained.

The exact assignment of U_S and U_P can be found in the pin assignment of the I/O connections.

Redirection of the supply voltages

The supply voltages are passed on internally from the "IN" connection to the "OUT" connection. Hence, the supply voltages U_S and U_P can be passed from one EtherCAT P Box to the next EtherCAT P Box in a simple manner.

NOTE

Note the maximum current.

Ensure that the maximum permitted current of 3 A for the M8 connectors is not exceeded when redirecting EtherCAT P.

4.3.2.1 Connectors

NOTE

Risk of damage to the device!

Bring the EtherCAT/EtherCAT P system into a safe, powered down state before starting installation, disassembly or wiring of the modules!

Two M8 sockets at the upper end of the modules are provided for supply and downstream connection of EtherCAT P:

- IN: left M8 socket for EtherCAT P supply
- OUT: right M8 socket for downstream connection of EtherCAT P

The metal threads of the M8 EtherCAT P sockets are internally linked to the FE connection via high impedance RC combination. See chapter Functional earth (FE) [▶ 25].

Fig. 2: Connectors for EtherCAT P

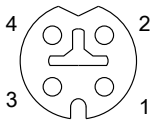


Fig. 3: M8 socket, p-coded

Contact	Signal	Voltage	Core color [1)]
1	Tx +	GND_S	yellow
2	Rx +	GND_P	white
3	Rx -	U_P: peripheral voltage, +24 V_{DC}	blue
4	Tx -	U_S: control voltage, +24 V_{DC}	orange
Housing	Shield	Shield	Shield

[1)] The core colors apply to EtherCAT P cables and ECP cables from Beckhoff.

4.3.2.2 Status LEDs

4.3.2.2.1 Supply voltages

EtherCAT P Box modules indicate the status of the supply voltages via two status LEDs. The status LEDs are labeled with the designations of the supply voltages: Us and Up.

LED	Display	Meaning
U_S (control voltage)	off	The supply voltage U_S is not available.
	green illuminated	The supply voltage U_S is available.
U_P (peripheral voltage)	off	The supply voltage U_P is not available.
	green illuminated	The supply voltage U_P is available.

4.3.2.2.2 EtherCAT

L/A (Link/Act)

A green LED labeled "L/A" or "Link/Act" is located next to each EtherCAT/EtherCAT P socket. The LED indicates the communication state of the respective socket:

LED	Meaning
off	no connection to the connected EtherCAT device
lit	LINK: connection to the connected EtherCAT device
flashes	ACT: communication with the connected EtherCAT device

Run

Each EtherCAT slave has a green LED labelled "Run". The LED signals the status of the slave in the EtherCAT network:

LED	Meaning
off	Slave is in "Init" state
flashes uniformly	Slave is in "Pre-Operational“ state
flashes sporadically	Slave is in "Safe-Operational" state
lit	Slave is in "Operational" state

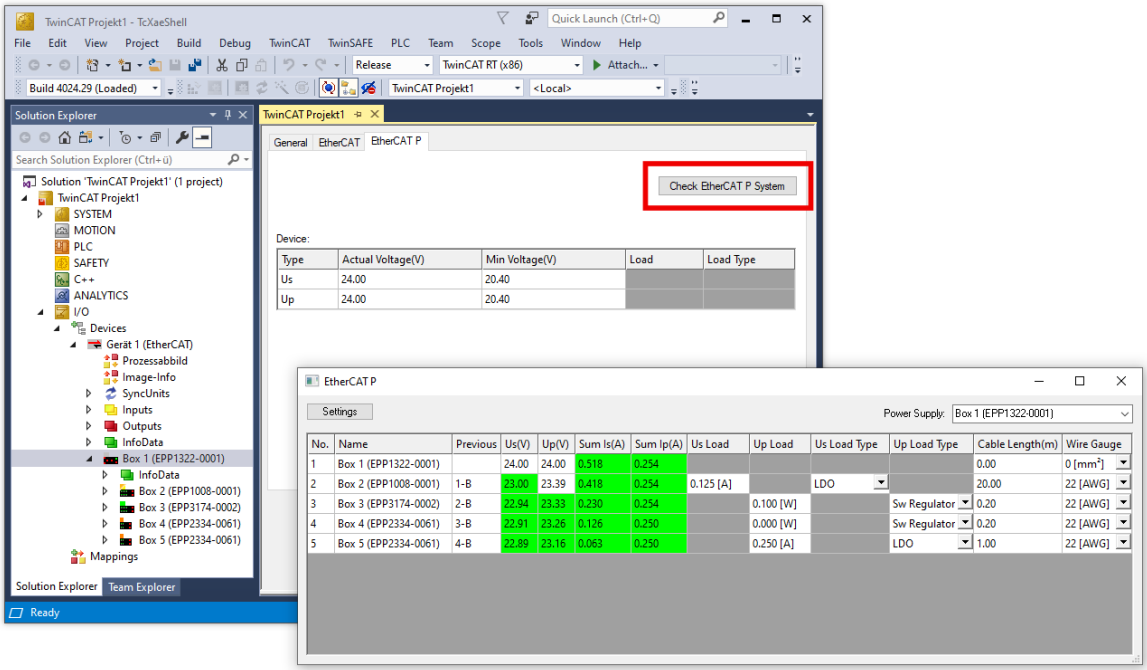
Description of the EtherCAT slave states

4.3.2.3 Conductor losses

Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage.
Variations in the voltage of the power supply unit must also be taken into account.

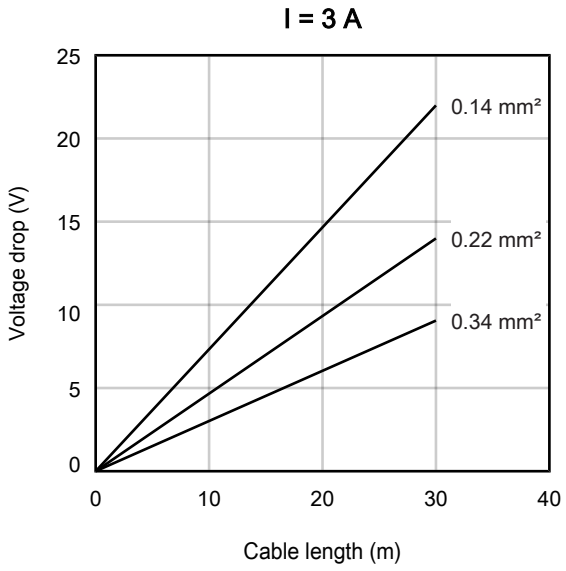
Planning tool for EtherCAT P

You can plan cable lengths, voltages and currents of your EtherCAT P system using TwinCAT 3. The requirement for this is TwinCAT 3 Build 4020 or higher.



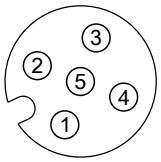
Further information can be found in the quick start guide IO configuration in TwinCAT in chapter "Configuration of EtherCAT P via TwinCAT".

Voltage drop on the supply line



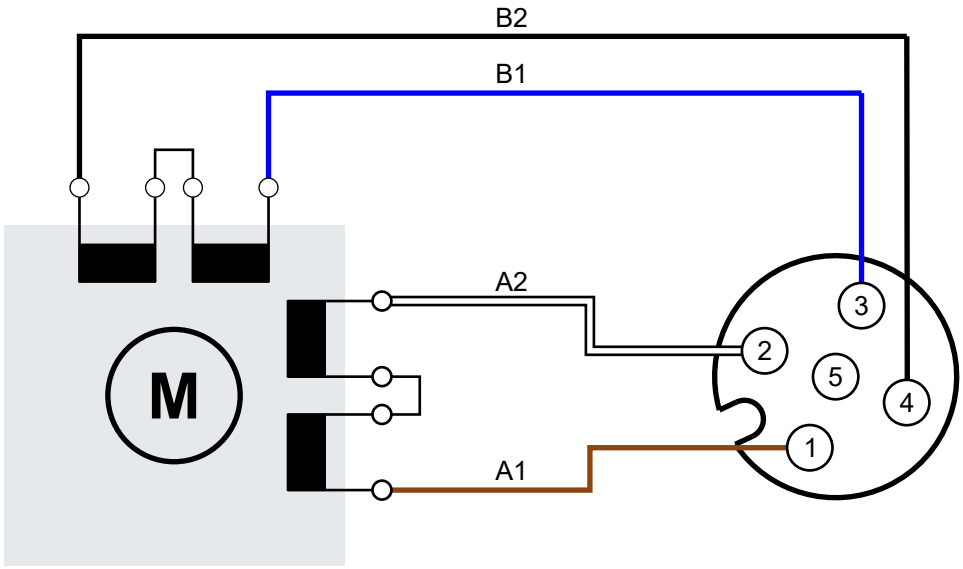
4.3.3 Stepper motor connection: X01

Pin assignment

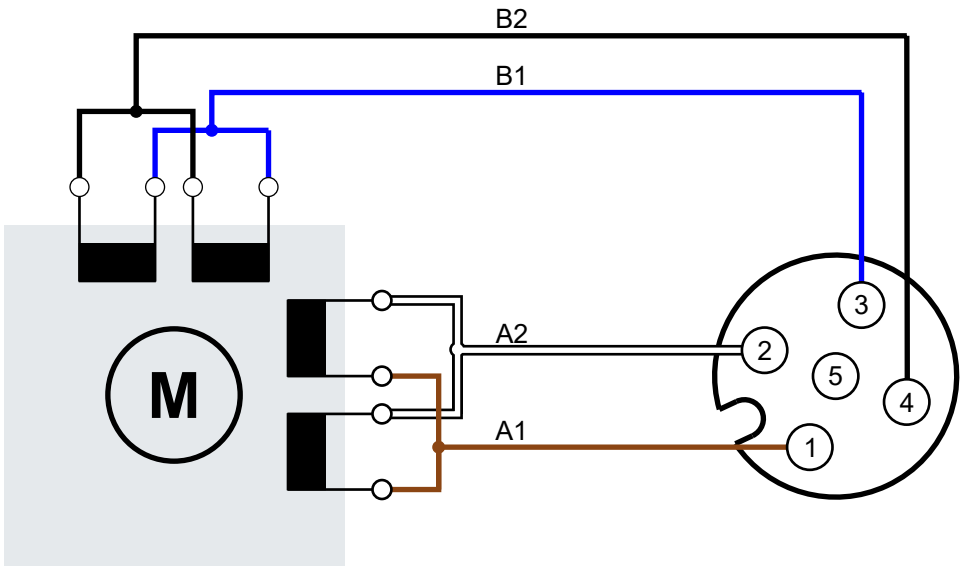
M12 socket	Pin	Function	Symbol	Core color [1]
	1	Motor winding A	A1	brown
	2		A2	white
	3	Motor winding B	B1	blue
	4		B2	black
	5	n.c.	n.c.	gray

[1] The core colors apply to M12 cables from Beckhoff: ZK2000-5xxx, ZK2000-6xxx, ZK2000-7xxx

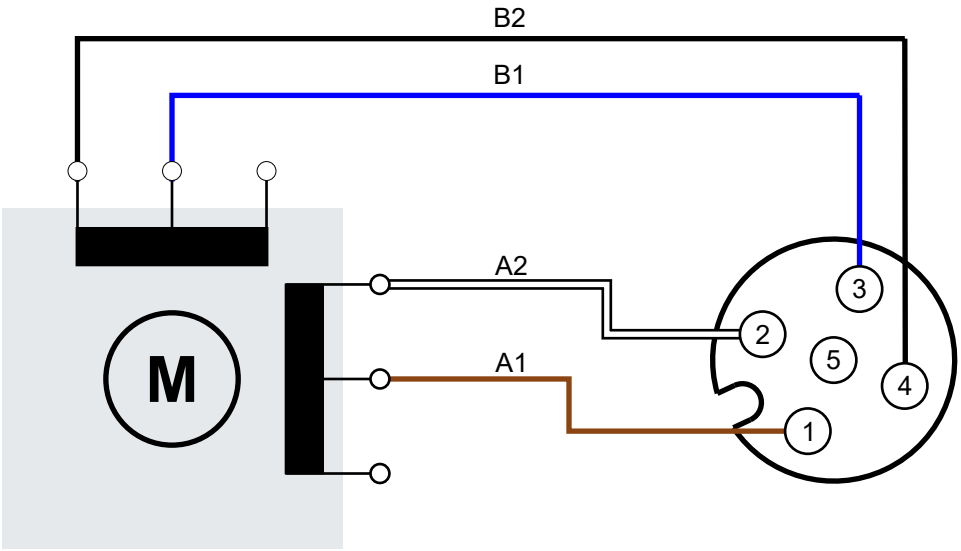
Connection example: Bipolar stepper motor, serial connection



Connection example: Bipolar stepper motor, parallel connection



Connection example: Unipolar stepper motor



In unipolar stepper motors only half of each winding is energized.

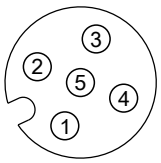
4.3.4 DC link voltage input: X02

NOTE

The DC link voltage input is not protected against polarity reversal.

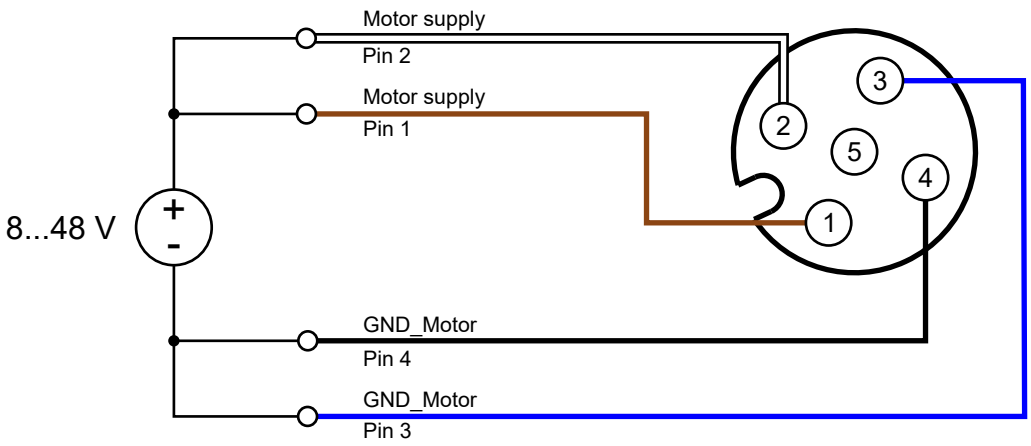
Defect possible through polarity reversal.

Pin assignment

M12 socket	Pin	Function	Symbol	Core color [1]
	1	DC link voltage	Motor supply	brown
	2	DC link voltage	Motor supply	white
	3	Ground	GND_Motor	blue
	4	Ground	GND_Motor	black
	5	n.c.	n.c.	gray

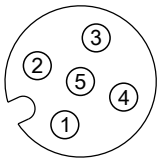
[1] The core colors apply to M12 cables from Beckhoff: ZK2000-5xxx, ZK2000-6xxx, ZK2000-7xxx

Connection example



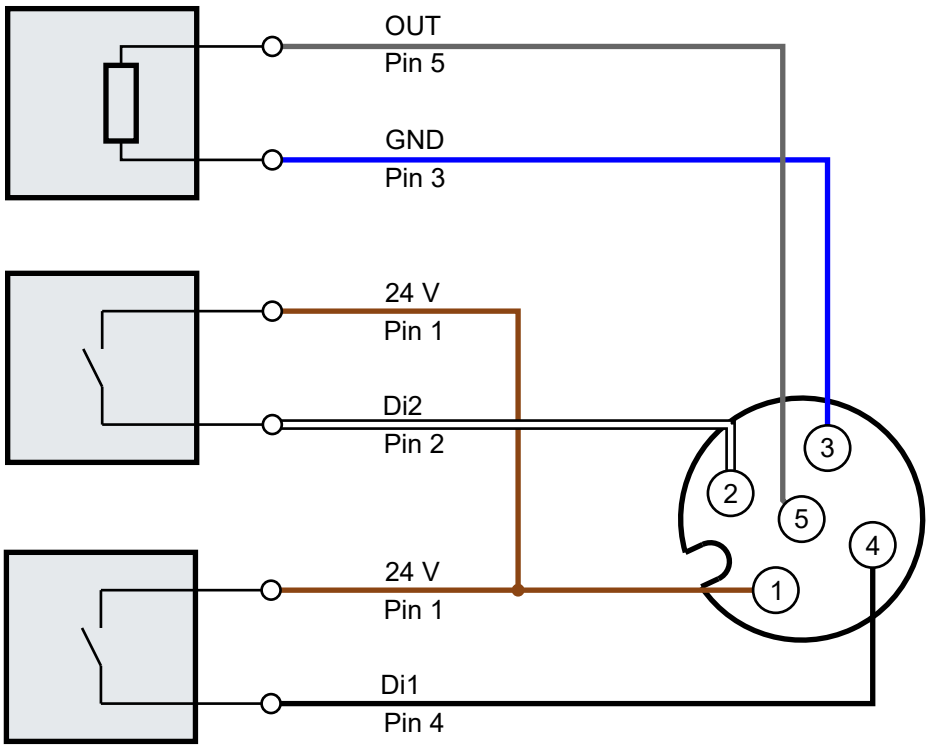
4.3.5 Digital inputs and outputs: X03

Pin assignment

M12 socket	Pin	Function	Symbol	Core color [1]
	1	Auxiliary voltage 24 V_{DC}	+24 V Us	brown
	2	Digital input B	Input B	white
	3	Ground	GND	blue
	4	Digital input A	Input A	black
	5	Digital output	OUT	gray

[1] The core colors apply to M12 cables from Beckhoff: ZK2000-5xxx, ZK2000-6xxx, ZK2000-7xxx

Connection example



4.3.6 Incremental encoders: X04

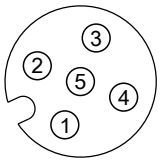
NOTE

The encoder supply voltage is not short-circuit proof

Risk of damage due to short circuit.

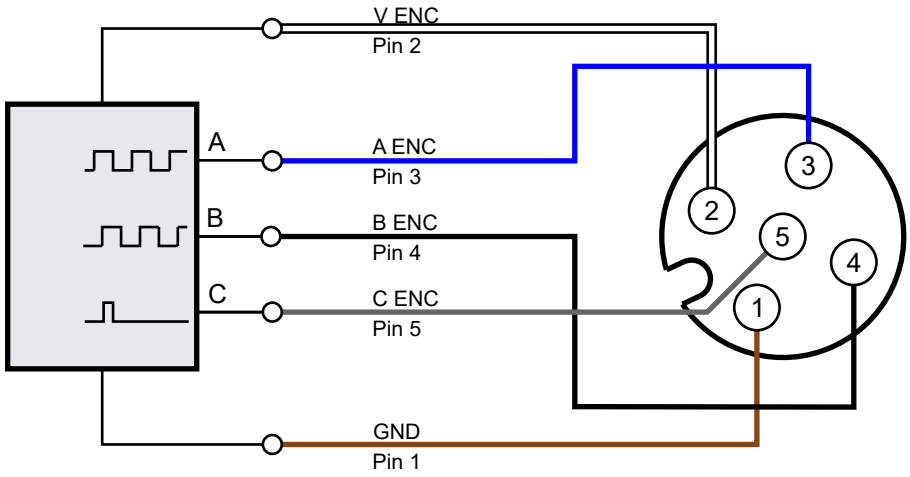
- Avoid short-circuiting the encoder supply voltage.

Pin assignment

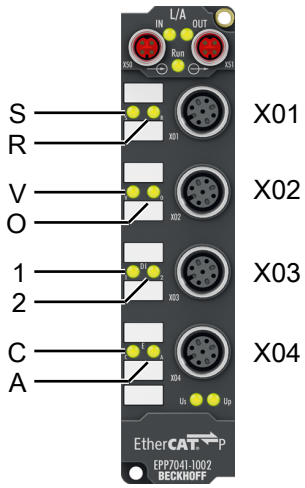
M12 socket	Pin	Function	Symbol	Core color [1]
	1	GND	GND	brown
	2	Encoder supply 24 V_{DC}	V ENC	white
	3	Signal input A	A ENC	blue
	4	Signal input B	B ENC	black
	5	Reference pulse / zero pulse	C ENC	gray

[1] The wire colors apply to M12 encoder cables from Beckhoff: ZK4000-5100-2xxx, ZK4000-5151-0xxx.

Connection example



4.3.7 Status LEDs



Connection	LED	Display	Meaning
X01: Stepper motor	S	green	The output stage is enabled.
		yellow	If motor is disabled: Motor control in standby
			If motor is enabled: warning, configuration error. Check the motor status.
		red	Error. Check the Diag Messages [▶ 71]
	R	green	Motor is turning
		red	internal error
X02: DC link input	V	off	The DC link voltage is not present.
		green	The DC link voltage is present.
	O	off	The digital output gives a low level.
		green	The digital output gives a high level.
X03: digital inputs digital output	1	off	Low level at digital input 1.
		green	High level at digital input 1.
	2	off	Low level at digital input 2.
		green	High level at digital input 2.
X04: Encoder	C	flashes	Encoder track C
	A	flashes	Encoder track A

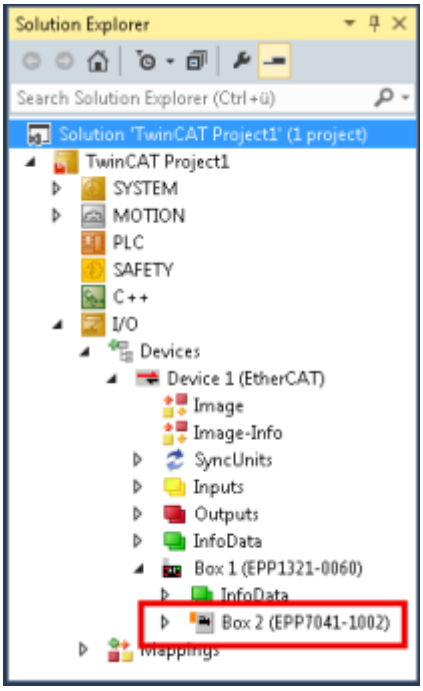
4.4 Disposal

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

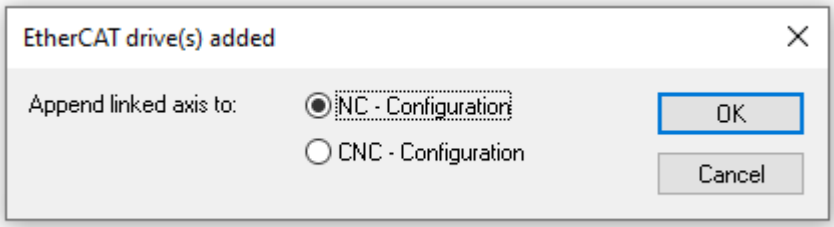
5 Commissioning and configuration

5.1 Integrating EPP7041 into a TwinCAT project

1. Integrate EPP7041 as an IO module in TwinCAT (Quick Start Guide).



⇨ A dialog box appears:



You now have two options:

- Click "OK" (recommended) …
  - … if you want to use the TwinCAT NC functions and you have not yet created the axis to be controlled in the current TwinCAT project.
- Click "Cancel" …
  - … if you do not want to use the TwinCAT NC functions.
  - … if you have already created the axis to be controlled in TwinCAT.

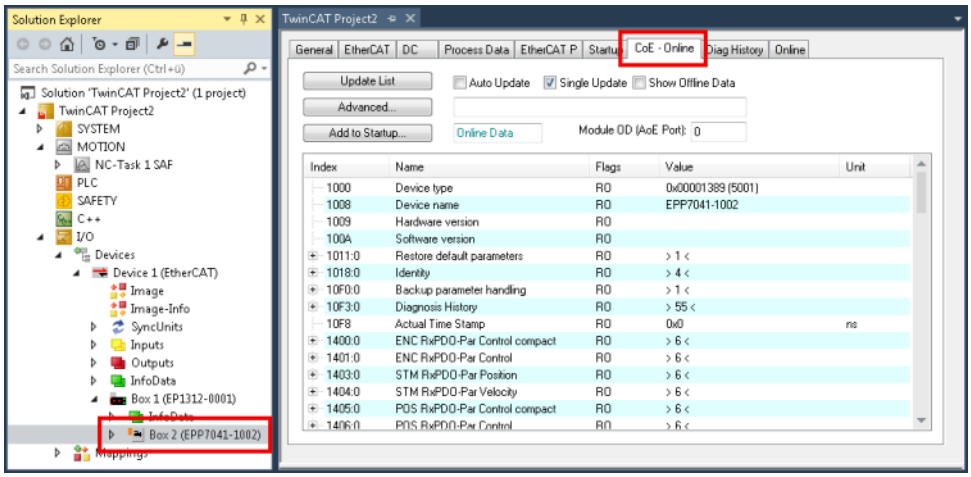
Note: This information is not binding. In other words, you can link an NC axis with EPP7041 at a later stage or disconnect the link.

When you click "OK":

- In the Solution Explorer under the entry "MOTION", a new NC task "NC-Task 1 SAF" is created if no NC task is available there yet.
- A new NC axis is created in the NC task under "Axes": "Axis*n*".
- The newly created NC axis is automatically linked to EPP7041.

5.2 Parameterizing EPP7041

5.2.1 Open the parameter directory (CoE)



1. In the Solution Explorer: Double-click EPP7041.
2. Click on the "CoE - Online" tab.

⇨ You now see the CoE directory of EPP7041. Here you can check and change parameter values.

Resetting parameters to factory settings

If you do not know whether parameters have already been changed by the present EPP7041, you can reset all parameters to the factory settings [▶ 69].

5.2.2 Setting important motor parameters

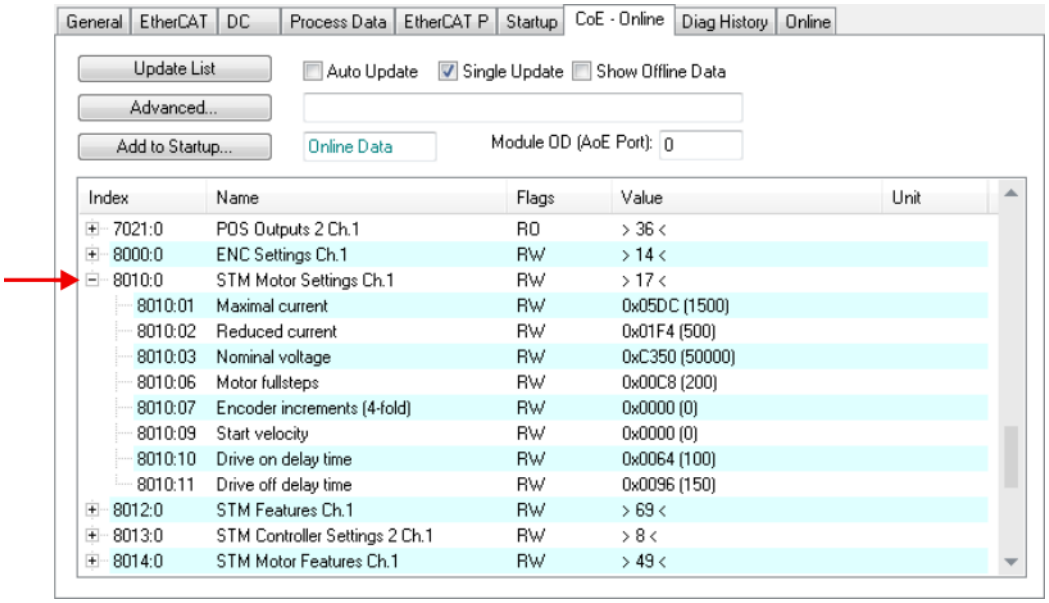
NOTE

Some motor parameters are not fault-tolerant

Incorrect motor parameters can result in damage.

- Take care when setting the motor parameters.

The motor parameters are stored in CoE object 8010_{hex}.



To ensure safe commissioning, it is sufficient to set the following parameters correctly. Further motor parameters are described under CoE object 8010_{hex}: STM Motor Settings Ch.1 [▶ 84].

8010:01 "Maximal current"

The maximum current that the current controller outputs per motor winding.

Unit: mA
Factory setting: 1500_{dec} (EPP7041-1002) / 5000_{dec} (EPP7041-3002)

The maximum value that should be entered here is the nominal motor current. The nominal current can usually be found in the data sheet of the motor.

8010:02 "Reduced current"

Winding current at motor standstill.

Unit: mA
Factory setting: 500_{dec} (EPP7041-1002) / 2500_{dec} (EPP7041-3002)

Criteria for setting this parameter:

- A lower value results in a lower power loss when the motor is at standstill.
- A higher value leads to a higher breakdown torque when the motor is at standstill.

8010:03 "Nominal voltage"

The DC link voltage you connect to X02 [▶ 33].

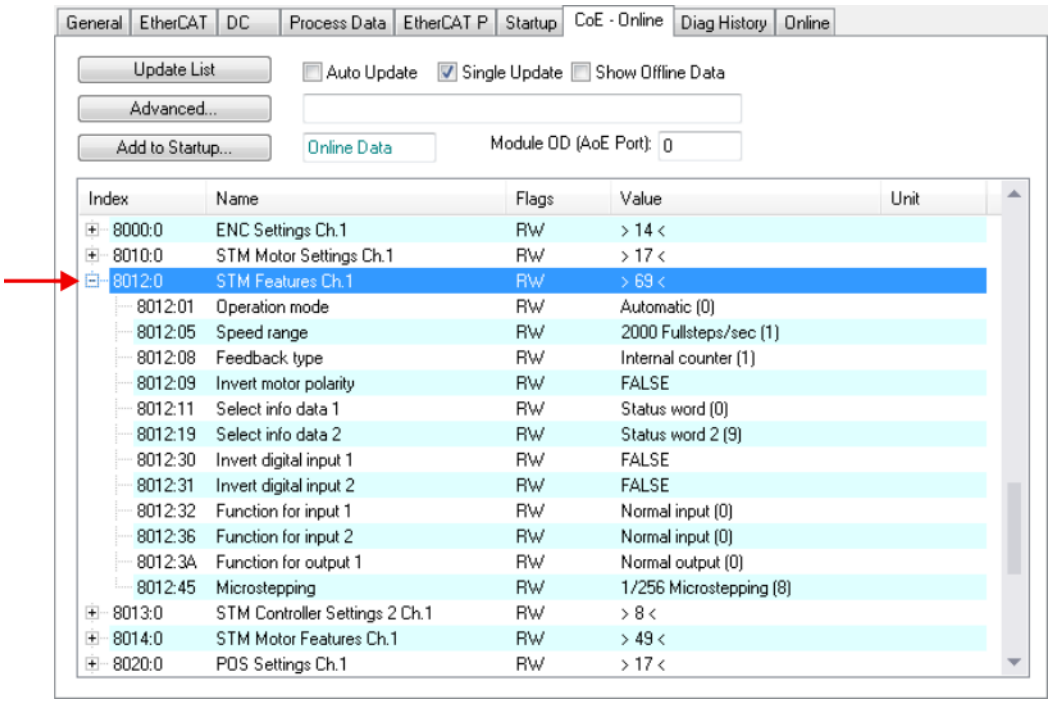
Risk of confusion: DC link voltage and nominal motor voltage

- Do *not* enter the nominal motor voltage here.

Unit: 1 mV
Factory setting: 50000_{dec}

5.2.3 Setting other important parameters

Other important parameters are stored in CoE object 8012_{hex}.



8012:05 "Speed range"

When changing "Speed range": adjust "Reference velocity"

Recalculate the parameter "Reference velocity" [▶ 45] if you have changed the parameter "Speed range".

The "Speed range" parameter has several functions:

- Upper limit of the output step frequency.
- Reference value for speed setpoints:
  Speed setpoints are given in % of the "Speed range".

Unit: Full steps per second
Factory setting: "2000 Fullsteps/sec"

The following formula can be used to determine the maximum achievable speed for a "Speed range":

$$n_{max} = f_{max} \times \frac{\varphi}{6}$$

n_{max}: Maximum achievable speed [rpm]
f_{max}: "Speed range" [full steps/s]
φ: Step angle of the motor [°]

8012:08 "Feedback type"

When changing the "Feedback type": adjust the "Scaling factor"

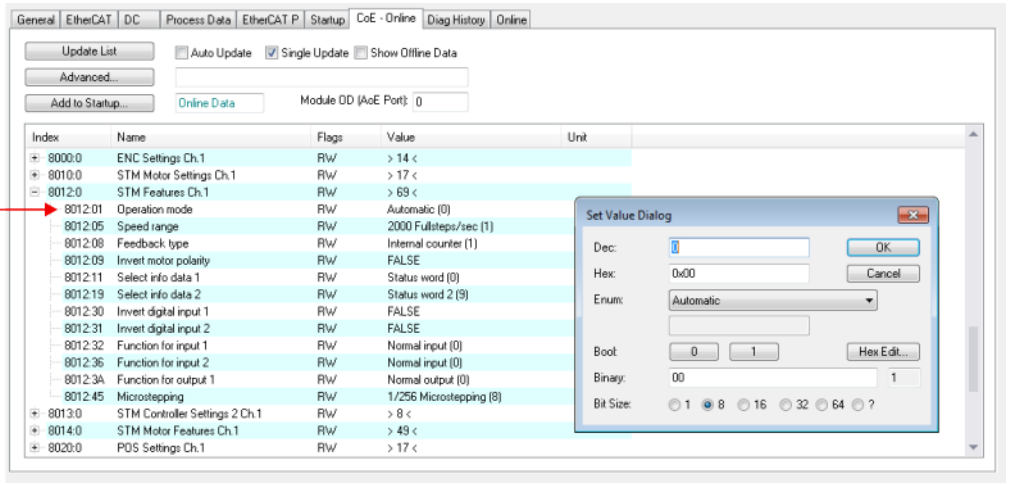
Recalculate the parameter Scaling factor [▶ 47] if you have changed the "Feedback type" parameter.

Factory setting: "Internal counter"

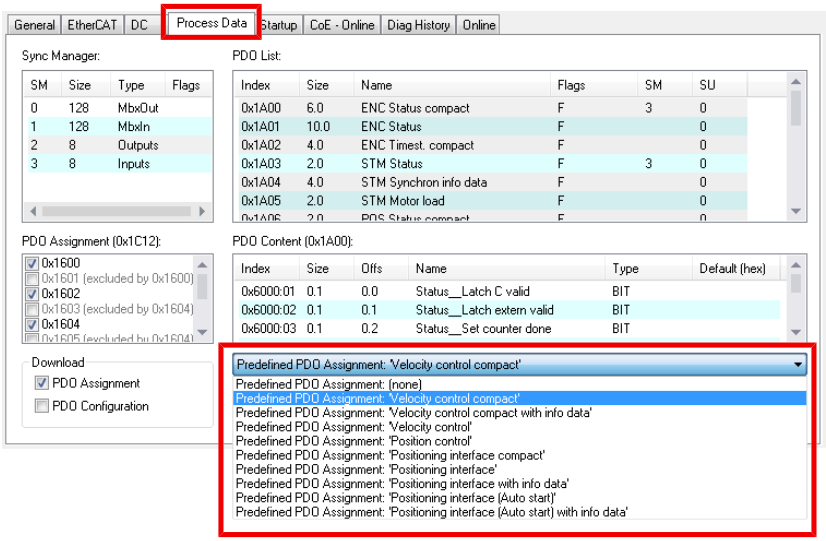
- If you are using an encoder, set this parameter to "Encoder". Parameterize the encoder [▶ 47].
- Otherwise set this parameter to "Internal counter".

5.3 Setting the operating mode

1. Decide which operation mode [▶ 44] is required for your application.
2. Set the operation mode via CoE parameter 8012:01$_{hex}$.



3. Click on the "Process data" tab.



4. Select a suitable "Predefined PDO Assignment" for the selected operating mode.
   Suitable "Predefined PDO Assignments" for the individual operating modes can be found in chapter Operating modes [▶ 44].
   Note:
   - if you have set the "Automatic" operating mode, the selection of the "Predefined PDO Assignment" determines the actual operating mode.
   - if you select "Positioning Interface [...]", the link to an NC axis is broken.

5.3.1 Operating modes

The "Automatic" operation mode is selected in the factory setting.

	Operation mode		
	Velocity direct	**Velocity controller**	**Position controller**
Setpoint type	Speed	Speed	Position
Possible "Predefined PDO Assignments"	• Velocity control [▶ 15] • Velocity control compact [▶ 15] • Velocity control compact with info data [▶ 15]		Position control [▶ 14]

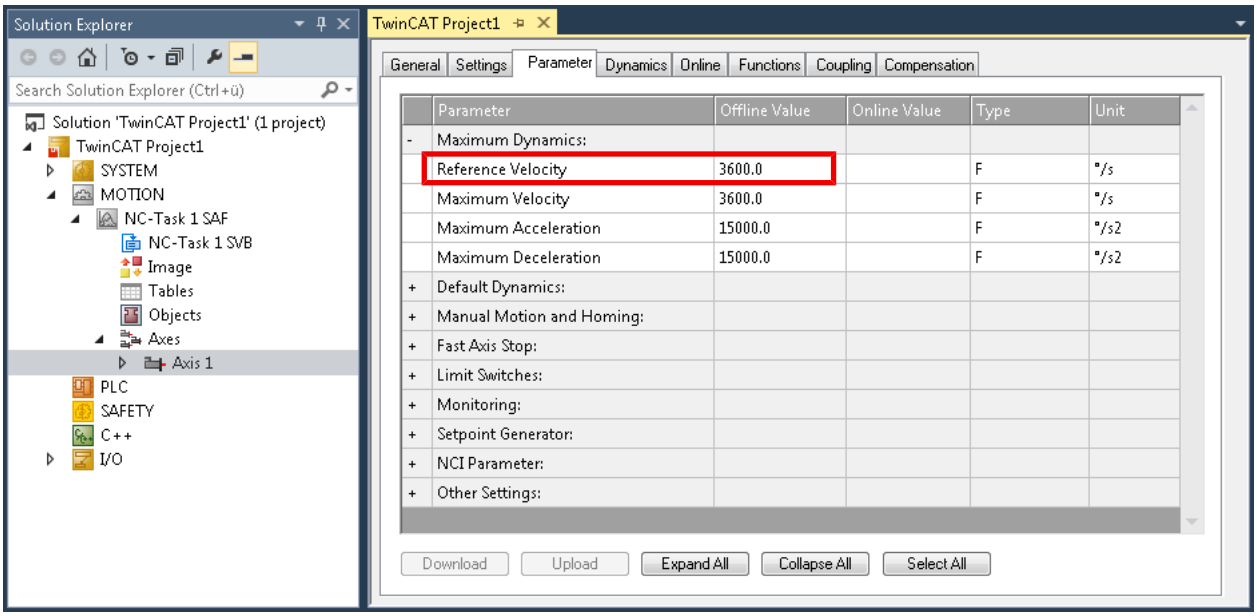
5.3.1.1 "Automatic" operating mode

If the "Automatic" operation mode is set, EPP7041-x002 selects the actual operation mode according to the set "Predefined PDO assignment":

The operation mode that EPP7041-x002 has automatically selected is set in CoE parameter A010:11.

5.4 Parameterizing the NC axis

Parameter "Reference Velocity"



Unit: °/s
Factory setting: 2200_{dec}

Calculate the "Reference Velocity" using this formula:

$$n_{ref} = f_{max} \times \varphi$$

n_{ref}: "Reference Velocity" [°/s]

f_{max}: "Speed range" [▶ 42] [full steps/s]

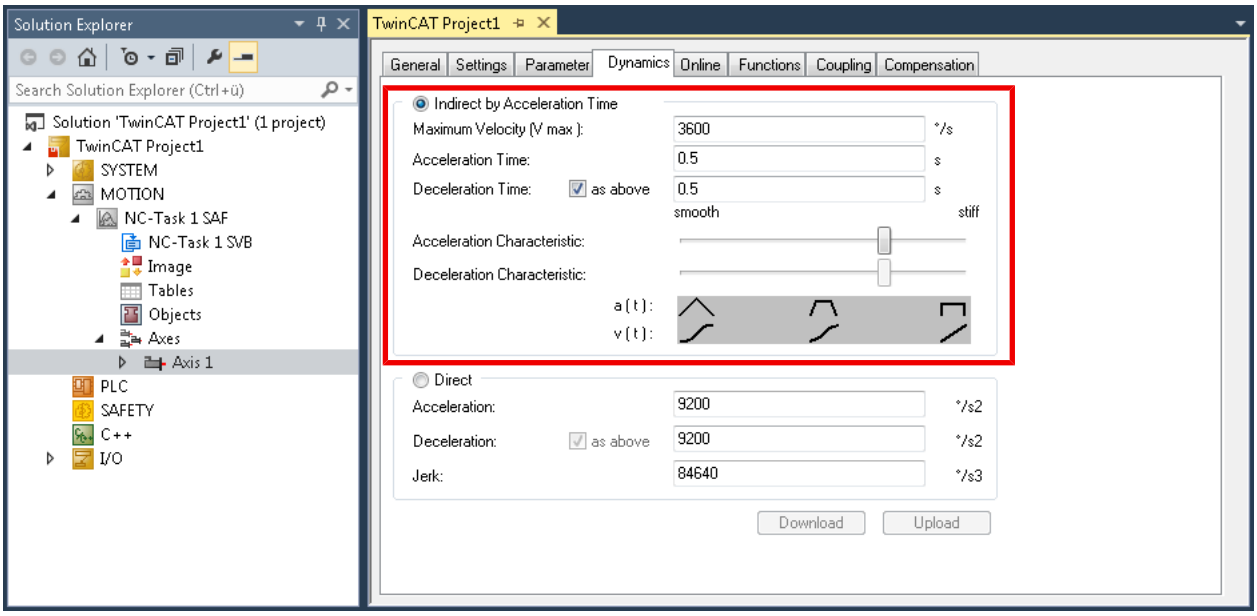
φ: Step angle of the motor [°]

Example for an AS1050-0120 motor:

$$n_{ref} = 2000 \frac{1}{s} \times 1{,}8\,° = 3600 \frac{°}{s}$$

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.



NOTE

Short braking times can lead to overvoltages in the DC link.

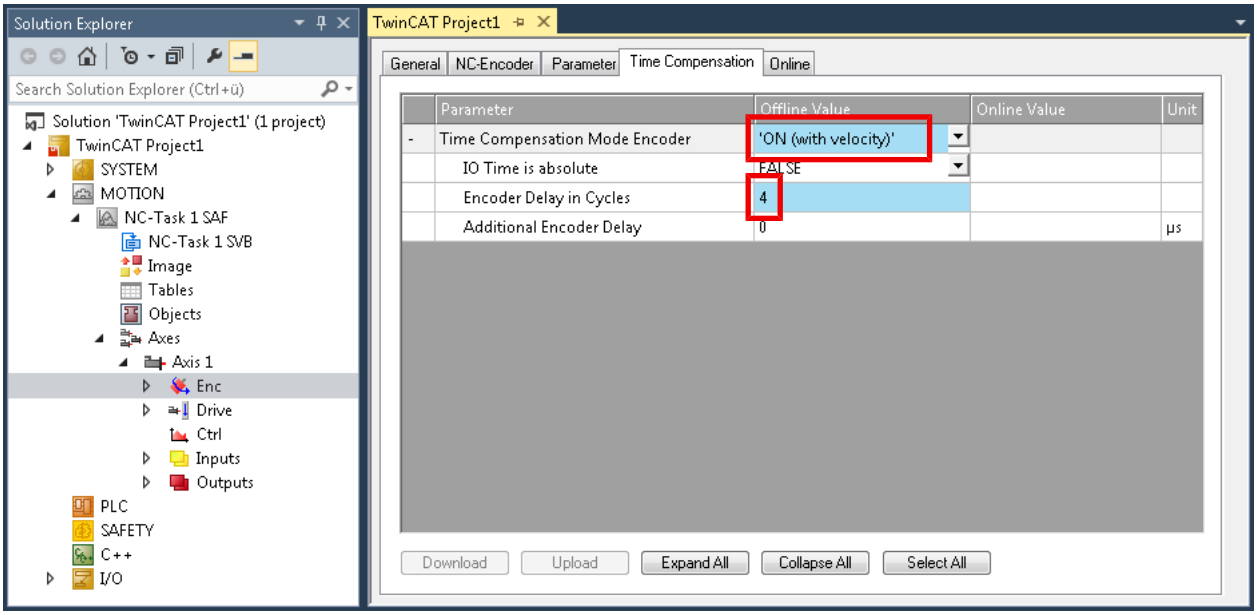
In the event of an overvoltage in the DC link, a protective mechanism switches off the motor output stage. The "Error" status bit in the process data object STM status [▶ 18] is set.

- Check whether impermissibly high voltages occur in the DC link during braking.
- If necessary, use EP9576-1032 to prevent overvoltages in the DC link.
  EP9576-1032 contains a brake resistor to dissipate drive-related overvoltages.

5.4.1 Parameterizing the encoder

Dead time compensation

The dead time compensation of the axis can be set in the *Time Compensation* tab of the *Axis1_ENC* encoder settings. It should, in theory, be 3 cycles of the NC cycle time, although in practice 4 cycles were found to be preferable. The parameter *Time Compensation Mode Encoder* should be set to 'ON (with velocity)', the parameter *Encoder Delay in Cycles* to 4.



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.

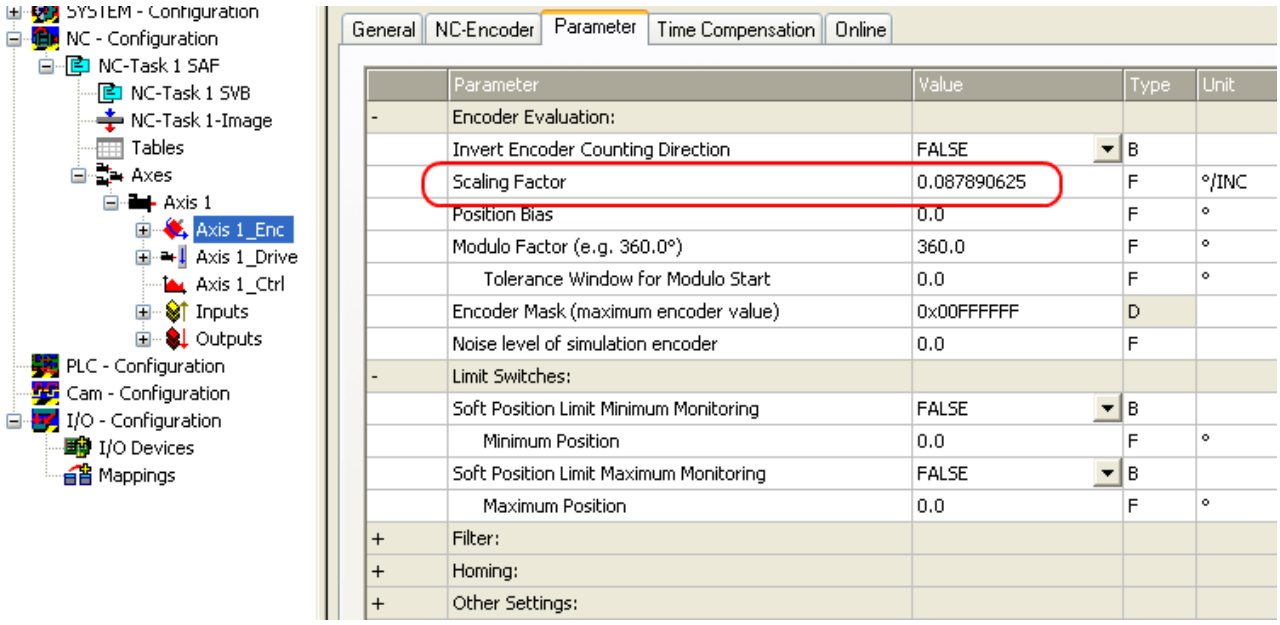


Fig. 4: Setting the Scaling Factor

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 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

5.4.2 Parameterizing the controller

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 a speed ramp with thresholds of 30%. The K_v factor for Standstill (t_1 and t_3) can be different than the K_v 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.

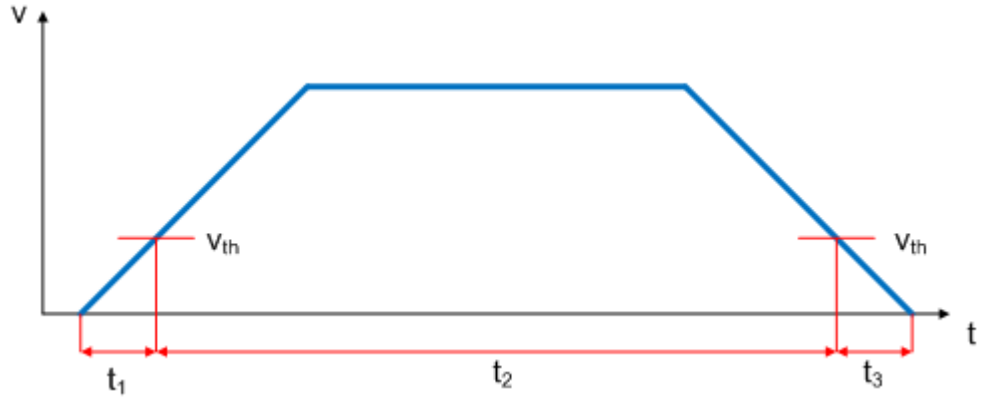
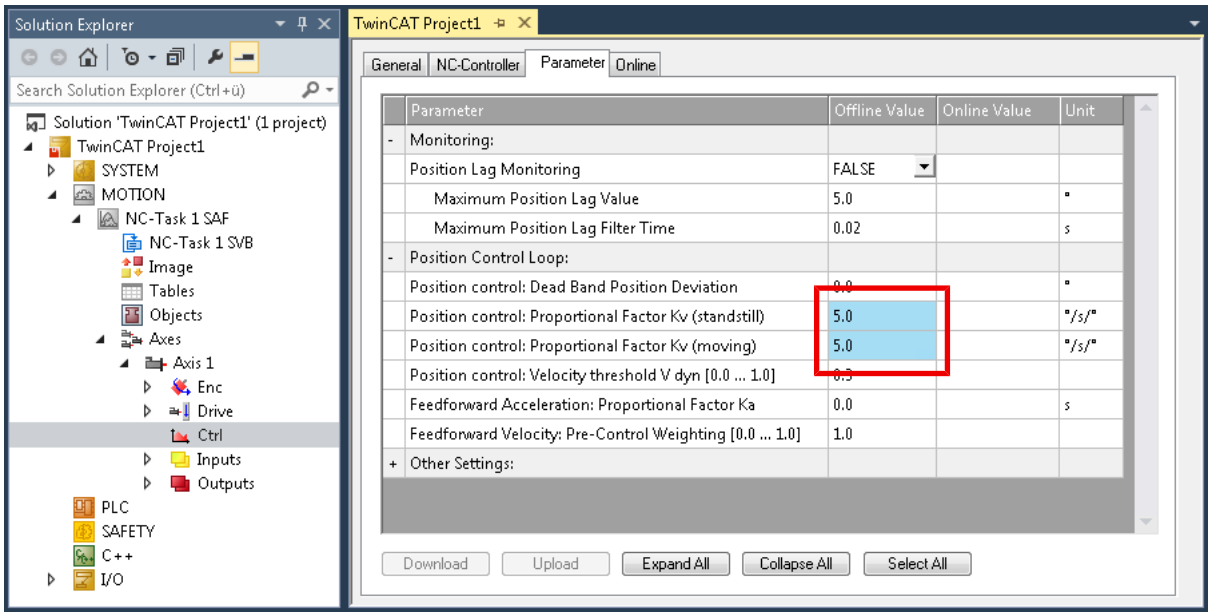


Fig. 5: *Speed ramp with K factor limit values*



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 controller 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.

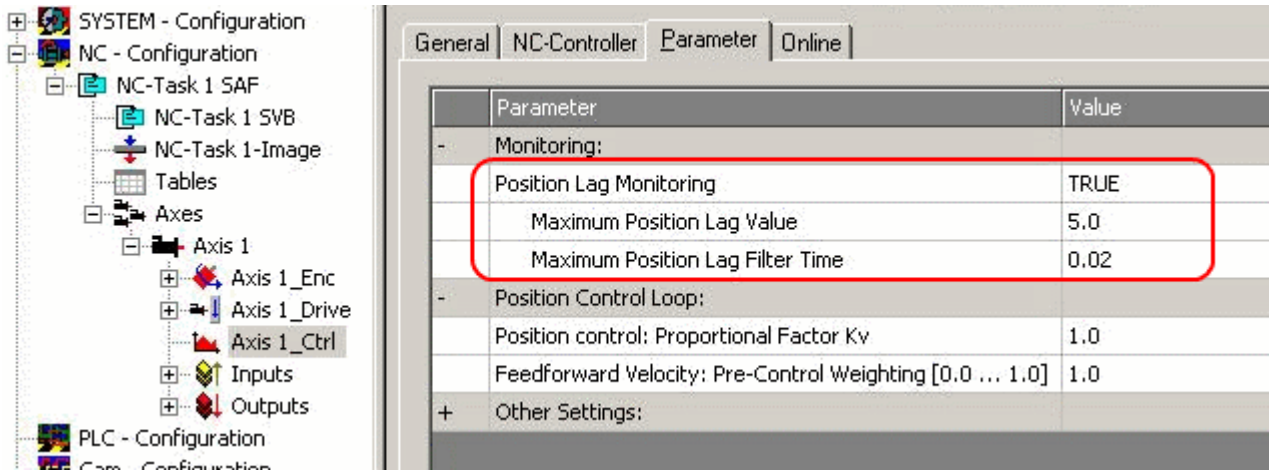


Fig. 6: *Position lag monitoring parameters*

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 controller 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*").

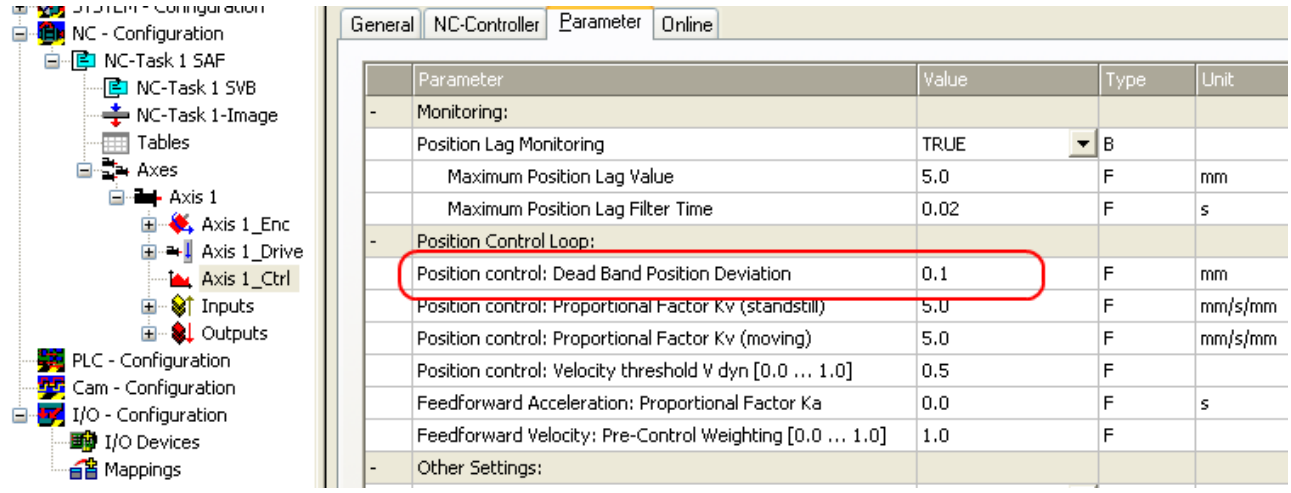


Fig. 7: *Dead band for position errors*

5.5 Performing a test run

NOTE

Important parameters must be set before the test run.

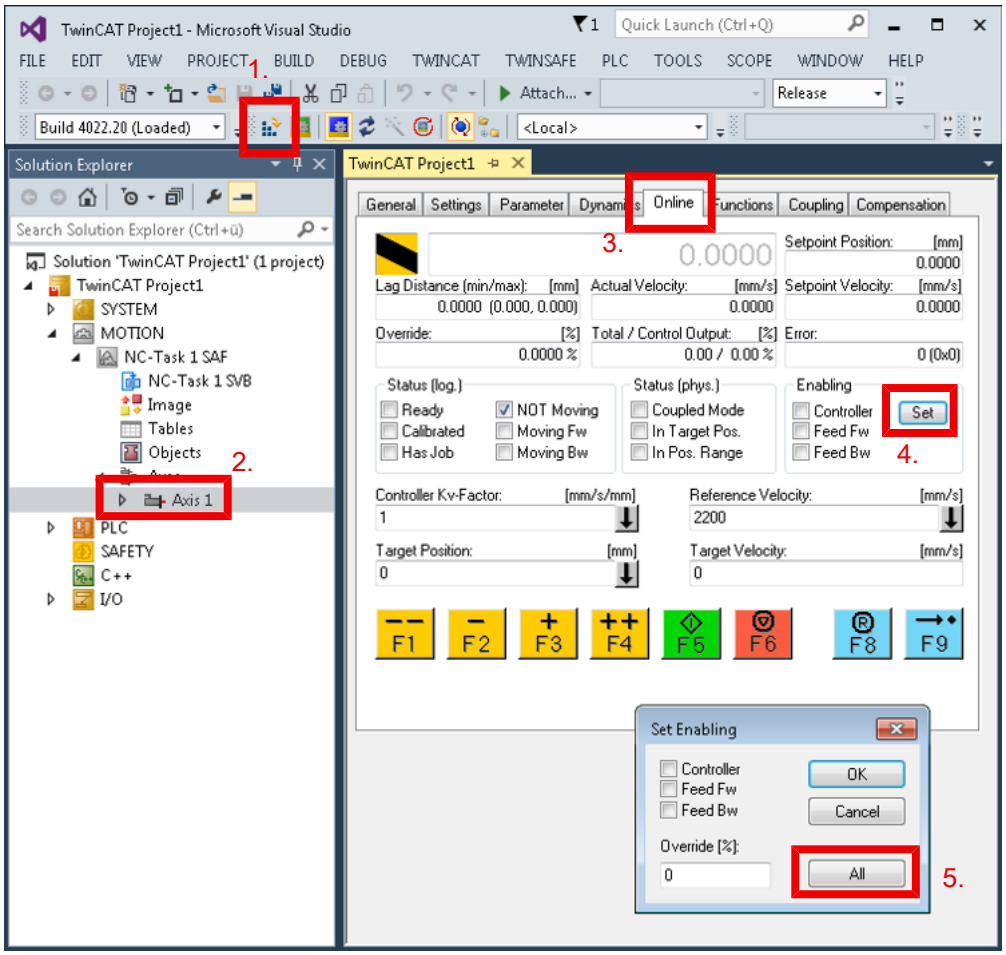
Risk of defect.

- Before the test run, carefully set the important motor parameters [▶ 40].

The procedure for a test run depends on whether you are using TwinCAT NC or not.

- Test run with TwinCAT NC [▶ 51]
- Test run without the TwinCAT NC [▶ 52]

5.5.1 Test run with TwinCAT NC



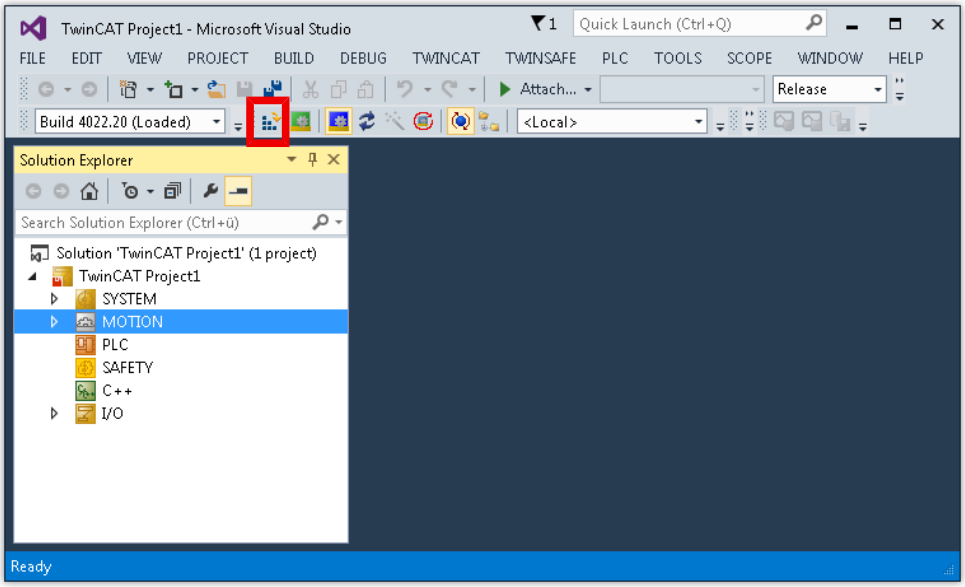
1. Activate the TwinCAT configuration.
2. Double-click the NC axis.
3. Click the "Online" tab.
4. Click the "Set" button in the "Enabling" field.
5. Click the "All" button in the window that appears.
   ⇨ The output stage is enabled.

⇨ You can use the colored buttons to move the axis for testing purposes.

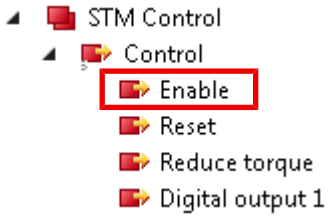
5.5.2 Test run without the TwinCAT NC

✓ Requirement: you are *not* using the "Positioning Interface".

1. Activate the TwinCAT configuration.



2. Set the variable "Enable" in the process data object "STM Control" to 1.



⇨ The output stage is enabled.

3. Specify a setpoint, depending on the operating mode:

Operation mode	Process data object for specifying a setpoint
Velocity direct	"STM Velocity" [▶ 20]
Velocity controller	"STM Velocity" [▶ 20]
Position controller	"STM Position" [▶ 20]

5.6 Further applications

5.6.1 Using the "Positioning Interface"

The "Positioning interface" can be used to execute motion commands without TwinCAT NC.

5.6.1.1 Basic principles: "Positioning interface"

Predefined PDO Assignment

The "Predefined PDO Assignment" 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.

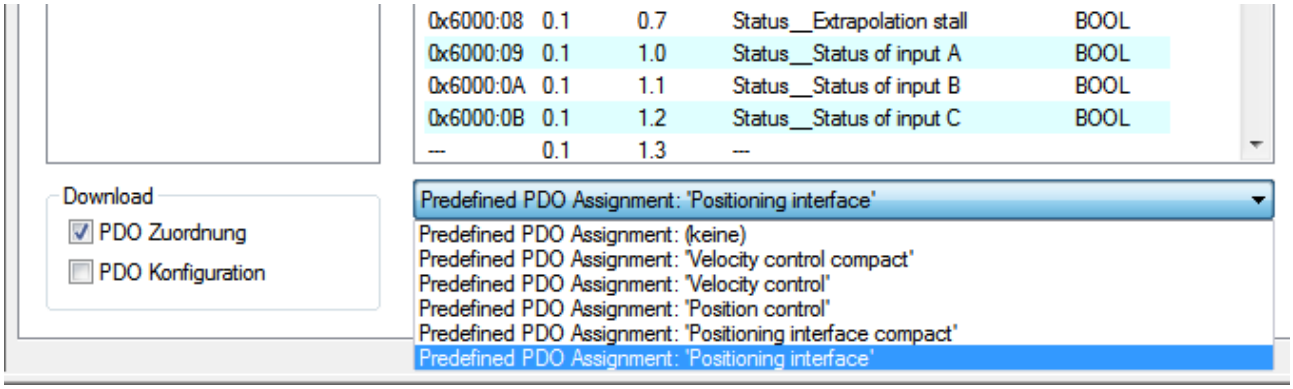


Fig. 8: Predefined PDO Assignment

Parameter set

Two objects are at the user’s disposal in the CoE for the configuration – the “POS Settings” (Index 0x8020) and the “POS Features” (Index 0x8021).

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. 9: Settings objects in the CoE

POS Settings

Velocity min.:

For reasons of performance when ramping down to the target position, EP7047-1032 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, EP7047-1032 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)

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 motor is accelerated 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 recognized 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).

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

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. 10: Diagnostic objects in the CoE

POS Info data

Status word:

The “*Status word*” reflects the status bits used in *Index 0xA020* 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 aborted
Bit 2: Target overrun

State (drive controller):

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

POS Diag data:

Command rejected:

A dynamic change of the target position is not accepted each time, 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

The state (drive controller) (Index 0x9020:03) 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	Initialization/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)

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:

Monitoring of the internal status of EPP7041-x002 and dynamic changing of the target position, if applicable.

Error handling:

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

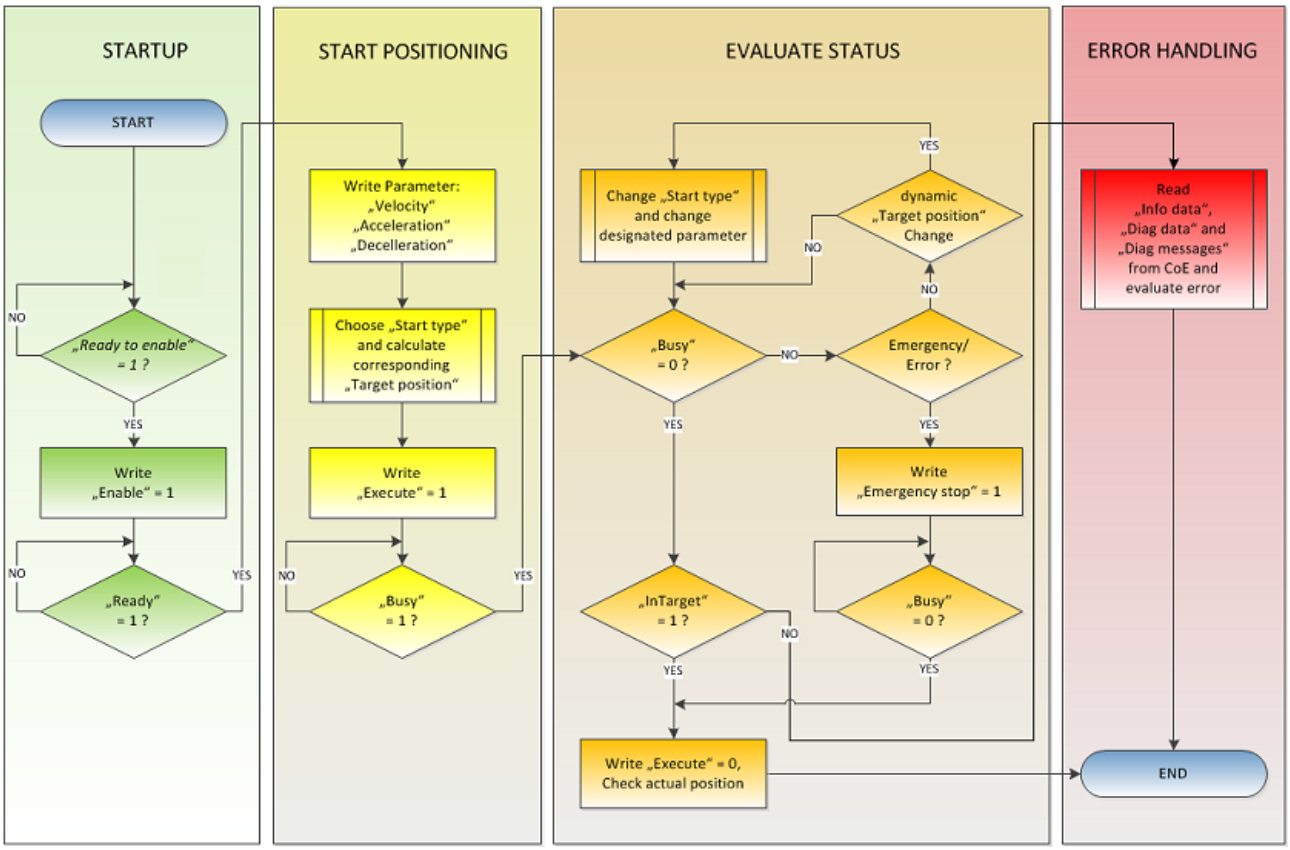


Fig. 11: Flow diagram for a travel command

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 [▶ 59]	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. [▶ 61]	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 [▶ 62]	Modulo positioning along the shortest path to the modulo position (positive or negative), calculated by the “*Modulo factor*” (Index 8020:0E)
MODULO_SHORT_EXT	0x0115		Modulo positioning along the shortest path to the modulo position; the “*Modulo tolerance window*” (Index 8020:0F) 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 [▶ 61]	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 flag “*Calibrated*”
SET_CALIBRATION_AUTO	0x6E01		Automatically set the flag “*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.

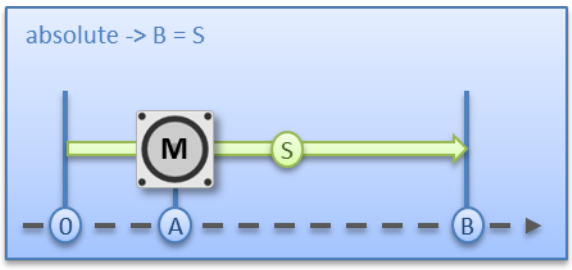


Fig. 12: Absolute positioning

RELATIVE:

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

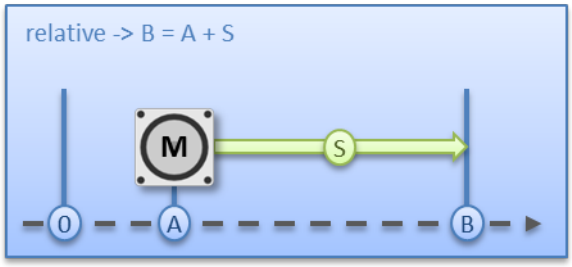


Fig. 13: Relative positioning

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.

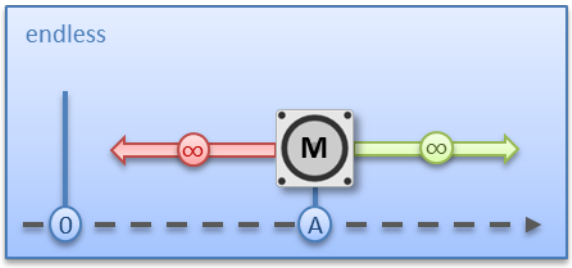


Fig. 14: Endless travel

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.

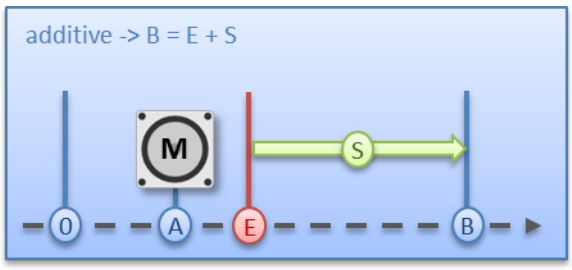


Fig. 15: Additive positioning

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 and the “Command rejected” bit is set. This is the case, for example, at standstill (since a standard positioning is expected 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) in direction 1 (Index 0x8021:13) towards the cam. Subsequently, in the 2nd step, it travels with velocity 2 (Index 0x8020:0A) in direction 2 (Index 8021:14) away from the cam. After the "*In-Target timeout*" (Index 8020:0C) has elapsed, the calibration position (Index 0x8020:08) is taken on 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.

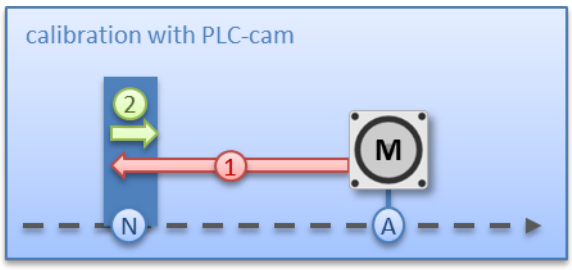


Fig. 16: *Calibration with cam*

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 as the current position.

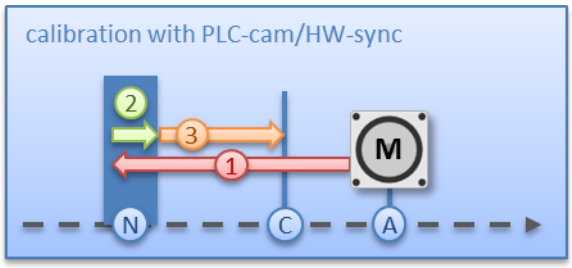


Fig. 17: *Calibration with cam and C-track*

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) 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), 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) 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 behavior.

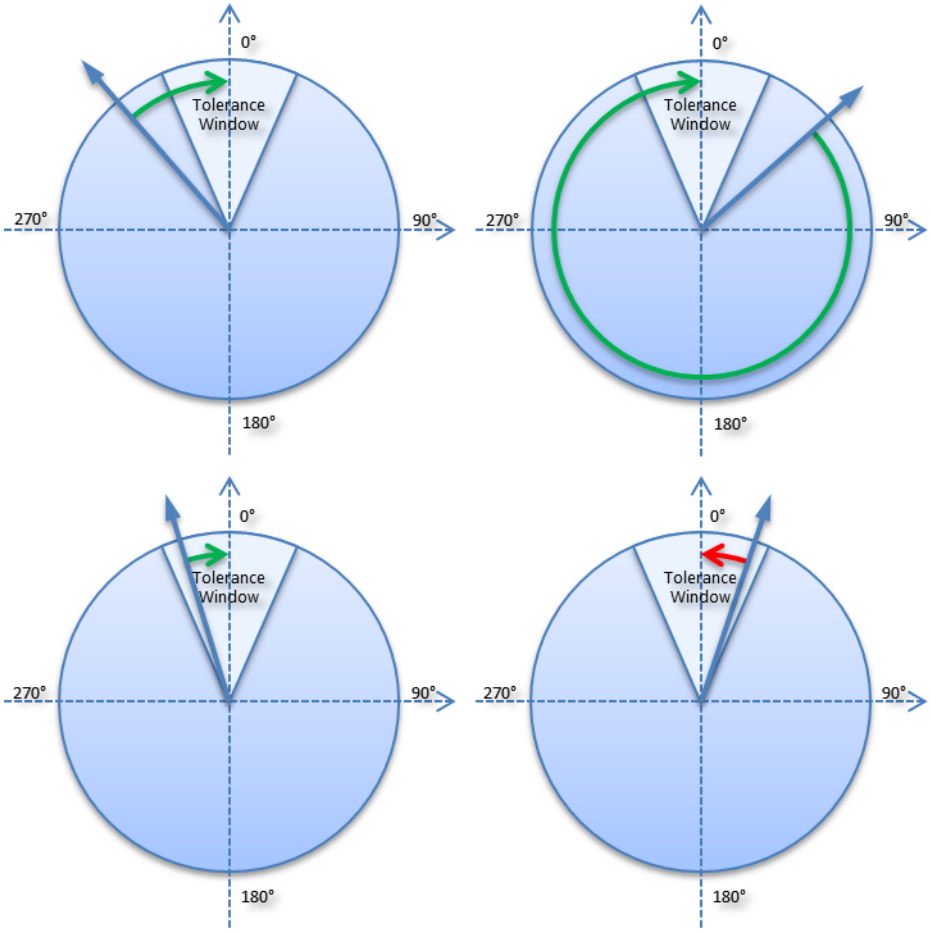


Fig. 18: Effect of the modulo tolerance window - modulo target position 0° in positive direction

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. above, 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) 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) 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.

Examples of modulo positioning with less than one revolution

Modulo start type: MODULO_PLUS				
Absolute start position	**Modulo target position**	**Relative travel path**	**Absolute end position**	**Modulo end position**
90°	0°	270°	360°	0°
90°	360°	630°	720°	0°
90°	720°	990°	1080°	0°

Modulo start type: MODULO_MINUS				
Absolute start position	**Modulo target position**	**Relative travel path**	**Absolute end position**	**Modulo end position**
90°	0°	-90°	0°	0°
90°	360°	-450°	-360°	0°
90°	720°	-810°	-720°	0°

Modulo start type: MODULO_SHORT				
Absolute start position	**Modulo target position**	**Relative travel path**	**Absolute end position**	**Modulo end position**
90°	0°	-90°	0°	0°

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 behavior described in the example [▶ 63] 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 (TF) is set to 1° here. Special cases for which the starting position is outside this window are identified.

Examples of modulo positioning with whole revolutions

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

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

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

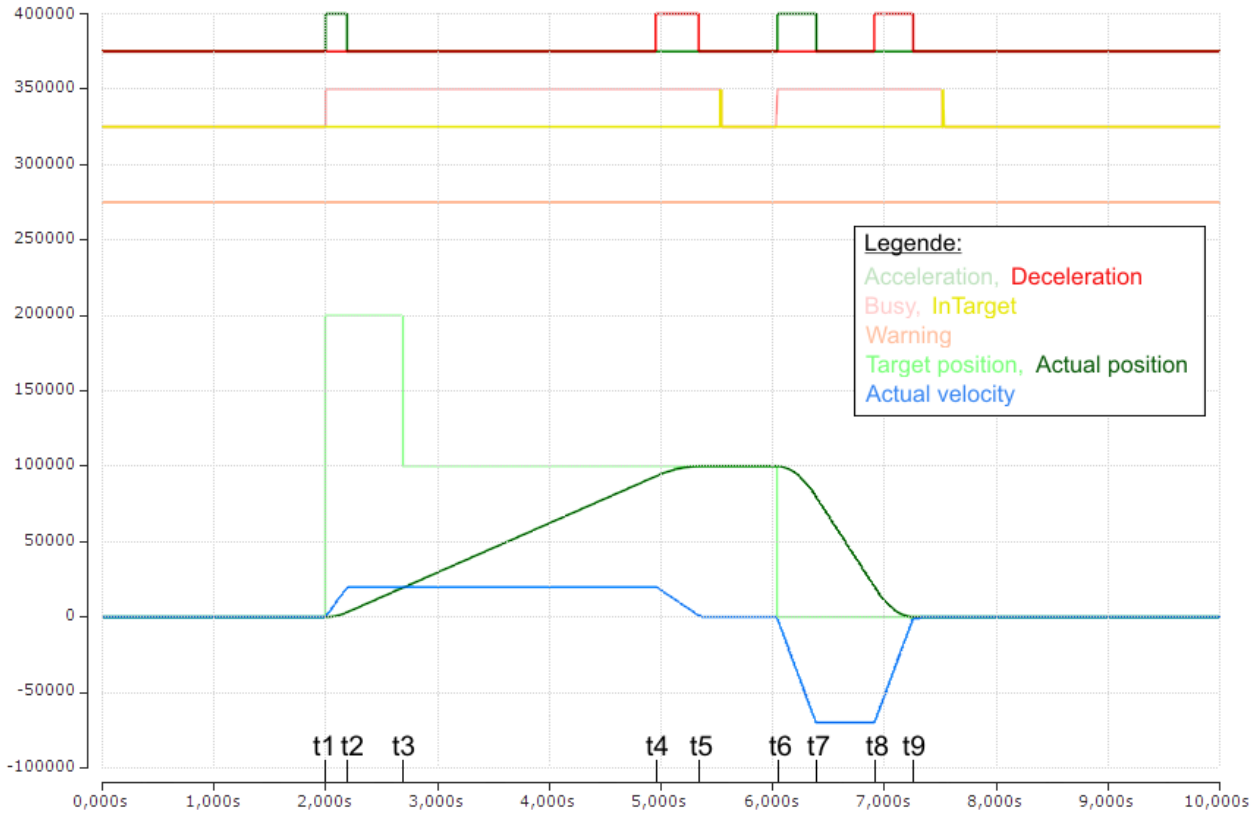


Fig. 19: Scope recording of a travel command with a dynamic change of the target position, without overrunning the target position
(The axis scaling refers only to the positions, not to the speed or the status bits)

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

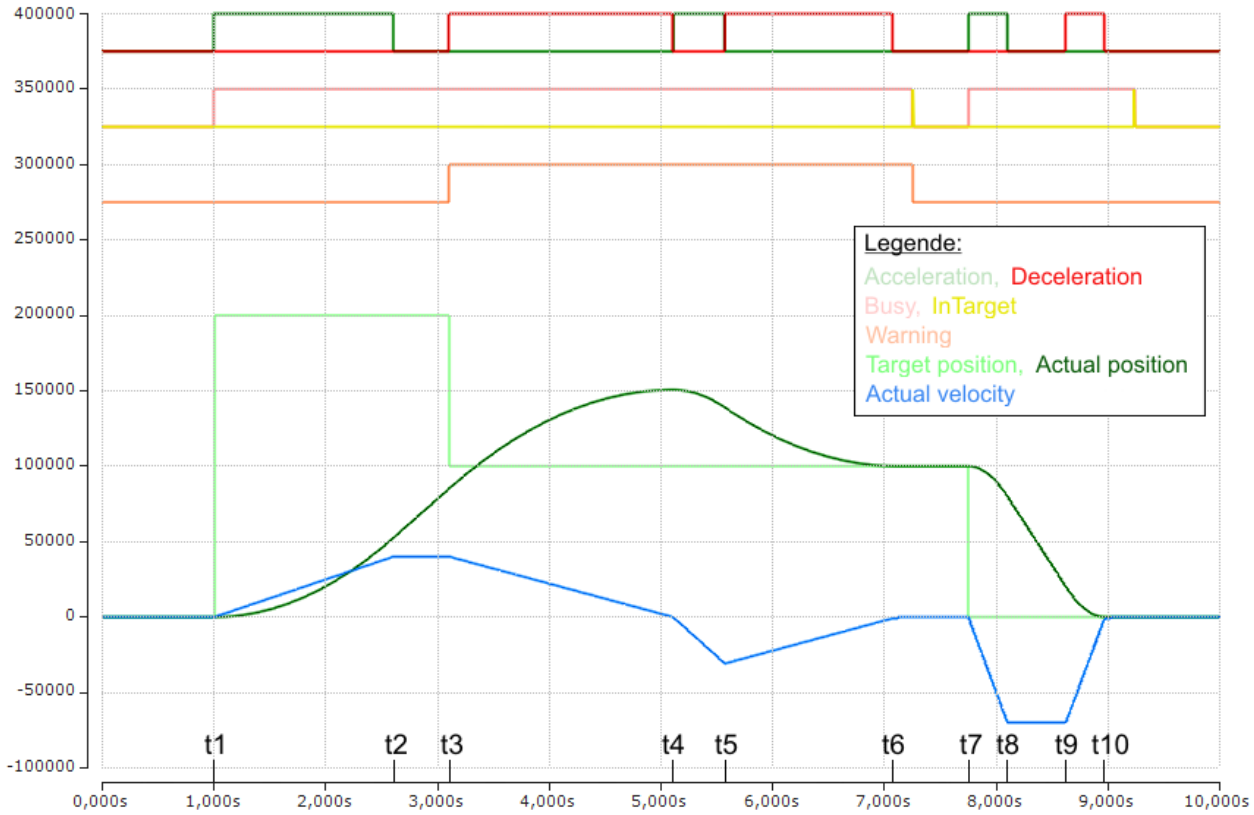
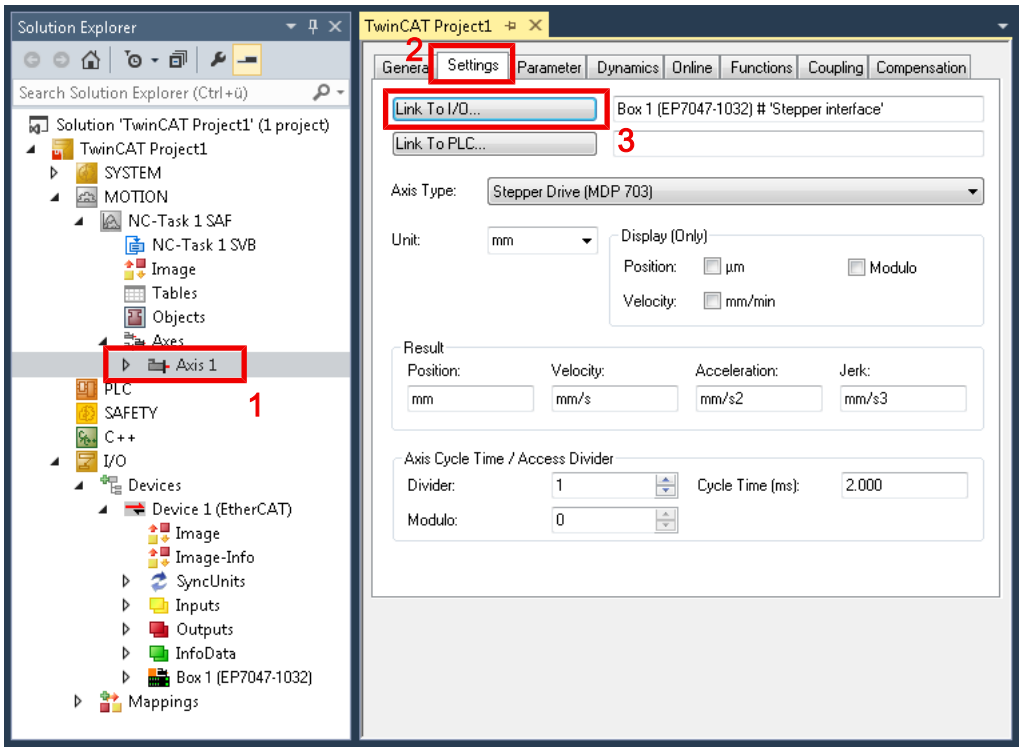


Fig. 20: Scope recording of a travel command with a dynamic change of the target position, with overrunning of the final target position
(The axis scaling refers only to the positions, not to the speed or the status bits)

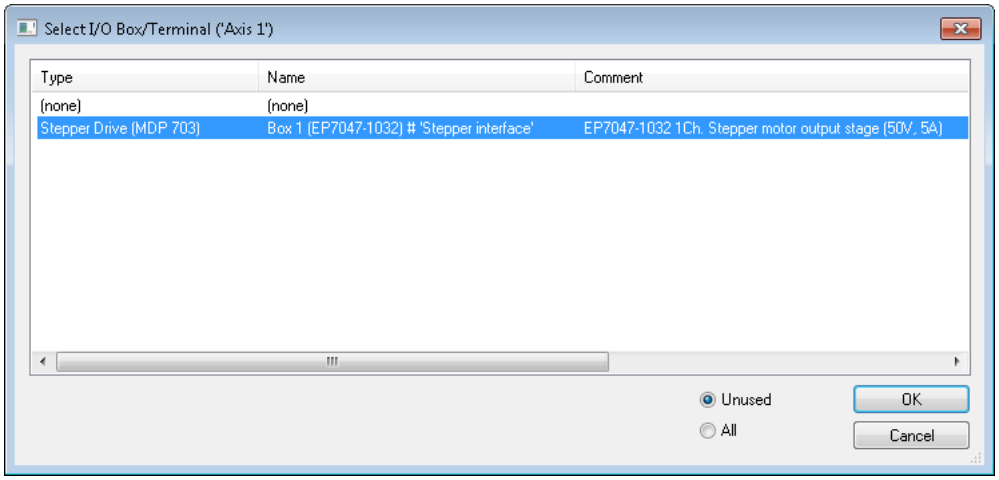
5.6.2 Linking an NC axis with EPP7041-x002

This step can usually be skipped

If you have carried out the commissioning in accordance with this documentation, an NC axis has already been linked to EPP7041-x002. See chapter Integrating EPP7041 into a TwinCAT project [▶ 38].



1. In the Solution Explorer: Double-click "Axis *n*".
2. Click on the "Settings" tab.
3. Click "Link to I/O".
   ⇨ A dialog box opens.



4. Select EPP7041-x002 and click "OK".
   Note: If EPP7041-x002 is not available for selection here, please check:
   - Is EPP7041-x002 included in the "I/O" section?
   - Is a predefined "Positioning interface ..." process image selected?

⇨ The process data from EPP7041-x002 are linked to the axis.

5.6.3 Restoring the delivery state

To restore the delivery state for backup objects in ELxxxx terminals / EPxxxx- and EPPxxxx box modules, the CoE object *Restore default parameters, SubIndex 001* can be selected in the TwinCAT System Manager (Config mode).

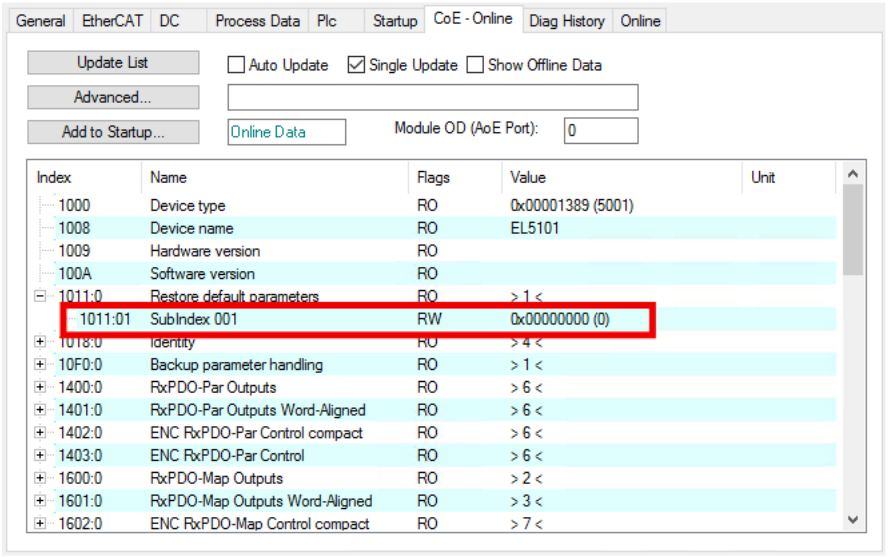


Fig. 21: Selecting the Restore default parameters PDO

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.

All backup objects are reset to the delivery state.

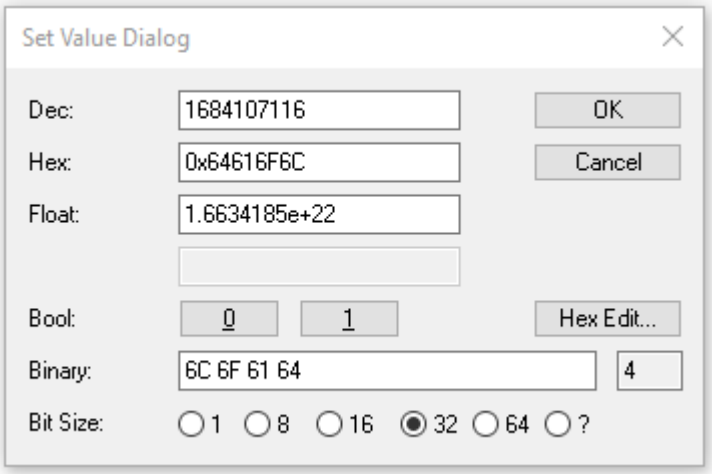


Fig. 22: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals / boxes the backup objects can be switched with an alternative restore value:
Decimal value: 1819238756
Hexadecimal value: 0x6C6F6164

An incorrect entry for the restore value has no effect.

5.7 Decommissioning

⚠ WARNING

Risk of electric shock!

Bring the bus system into a safe, de-energized state before starting disassembly of the devices!

6 Diagnosis

6.1 Diagnostics – basic principles of diag messages

DiagMessages designates a system for the transmission of messages from an EtherCAT device to the EtherCAT Master/TwinCAT. The messages are stored by the EtherCAT 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 EtherCAT 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 an EtherCAT device 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 time in the EtherCAT device 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 EtherCAT 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:

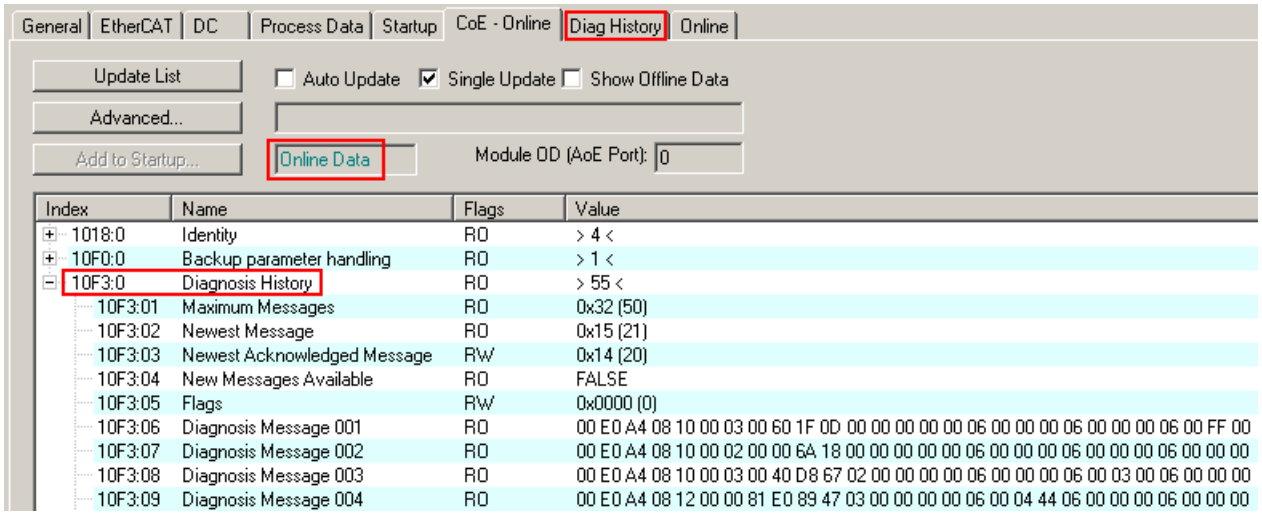


Fig. 23: DiagMessages in the CoE

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 EtherCAT 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 EtherCAT device's own interface. Operation (collection, confirmation) also takes place via this interface.

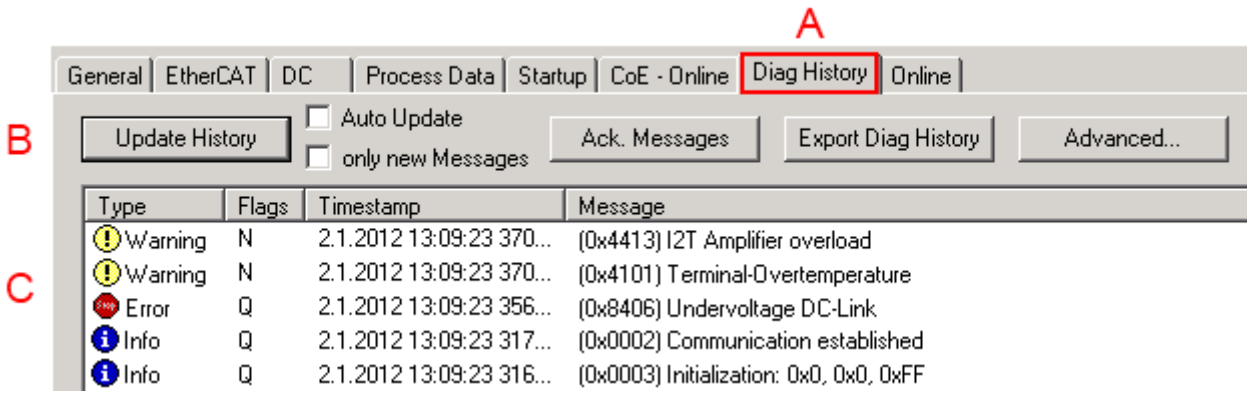


Fig. 24: Implementation of the DiagMessage system in the TwinCAT System Manager

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 EtherCAT device 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 EtherCAT device and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT devices 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 EtherCAT device via 0x10F3:05, so such EtherCAT devices have the following entry in the StartUp list by default:

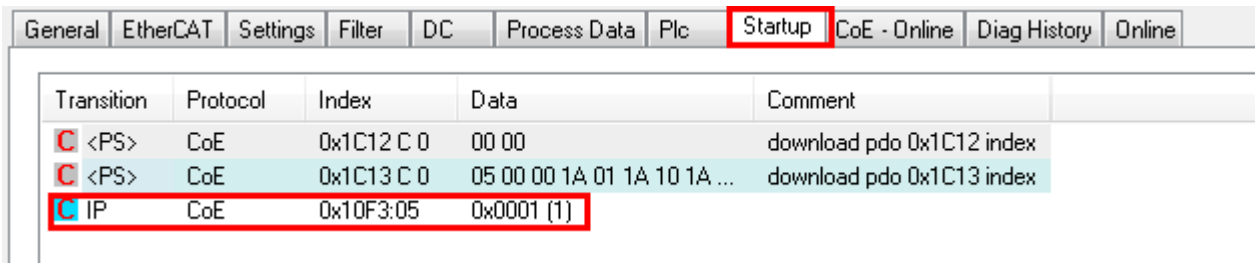


Fig. 25: Startup List

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 EtherCAT device 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 EtherCAT device 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 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 is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values.

Text ID	Type	Place	Text Message	Additional comment
				• The I2T-model of the motor is incorrectly parameterized.
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x%X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the EtherCAT device exceeds the parameterized error threshold. Activation of the EtherCAT device 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 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 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

6.2 Diag Messages of EtherCAT devices for drive technology

„Ack. Message“ Button

The ‚Ack. Message’ button has no effect on the Drive State Machine, 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.

7 CoE parameters

7.1 Object directory

CoE parameters are grouped into logical groups called "objects".

Object index (hex)	Name
1000	Device type [▶ 88]
1008	Device name [▶ 88]
1009	Hardware version [▶ 88]
100A	Software version [▶ 88]
1011	Restore default parameters [▶ 88]
1018	Identity [▶ 89]
10F0	Backup parameter handling
10F3	Diagnosis History
10F8	Actual Time Stamp
1400	ENC RxPDO-Par Control compact
1401	ENC RxPDO-Par Control
1403	STM RxPDO-Par Position
1404	STM RxPDO-Par Velocity
1405	POS RxPDO-Par Control compact
1406	POS RxPDO-Par Control
1407	POS RxPDO-Par Control 2
1600	ENC RxPDO-Map Control compact
1601	ENC RxPDO-Map Control
1602	STM RxPDO-Map Control
1603	STM RxPDO-Map Position
1604	STM RxPDO-Map Velocity
1605	POS RxPDO-Map Control compact
1606	POS RxPDO-Map Control
1607	POS RxPDO-Map Control 2
1800	ENC TxPDO-Par Status compact
1801	ENC TxPDO-Par Status
1806	POS TxPDO-Par Status compact
1807	POS TxPDO-Par Status
1A00	ENC TxPDO-Map Status compact
1A01	ENC TxPDO-Map Status
1A02	ENC TxPDO-Map Timest. compact
1A03	STM TxPDO-Map Status
1A04	STM TxPDO-Map Synchron info data
1A05	STM TxPDO-Map Motor load
1A06	POS TxPDO-Map Status compact
1A07	POS TxPDO-Map Status
1A08	STM TxPDO-Map Internal position
1A09	STM TxPDO-Map External position
1A0A	POS TxPDO-Map Actual position lag

Index (hex)	Name
1C00	Sync manager type
1C12	RxPDO assign
1C13	TxPDO assign
1C32	SM output parameter
1C33	SM input parameter
6000	ENC Inputs Ch.1
6010	STM Inputs Ch.1
6020	POS Inputs Ch.1
7000	ENC Outputs Ch.1
7010	STM Outputs Ch.1
7020	POS Outputs Ch.1
7021	POS Outputs 2 Ch.1
8000	ENC Settings Ch.1 [▶ 84]
8010	STM Motor Settings Ch.1 [▶ 85]
8012	STM Features Ch.1 [▶ 86]
8013	STM Controller Settings 2 Ch.1 [▶ 87]
8014	STM Motor Features Ch.1
8020	POS Settings Ch.1 [▶ 87]
8021	POS Features Ch.1 [▶ 87]
9010	STM Info data Ch.1
9020	POS Info data Ch.1
A010	STM Diag data Ch.1
A020	POS Diag data Ch.1
F000	Modular device profile
F008	Code word
F010	Module list
F081	Download revision
F083	BTN
F80F	STM Vendor data
F900	STM Info data
FB00	STM Command
FB40	Memory interface

7.2 Data format of CoE parameters

CoE parameters have different data formats.

The data format of the CoE parameters is specified by data type identifiers in the chapter Object description [▶ 84]:

Data type identifier	Format	Size
BOOL	True / false	8-bit
SINT	Short integer	8-bit
USINT	Unsigned short integer	8-bit
INT	Integer	16-bit
UINT	Unsigned integer	16-bit
DINT	Double integer	32-bit
UDINT	Unsigned double integer	32-bit
STRING	String	max. 255 characters, 1 byte per character

The data type identifiers correspond to the data types that can also be used in TwinCAT in a PLC program.

7.3 Object description

7.3.1 Objects for parameterization

Index 8000: ENC Settings Ch.1

Access rights: read/write

Index (hex)	Name	Description	Unit	Data type	Default value
8000:08	Disable filter	Deactivates the input filter.	-	BOOL	FALSE
8000:0A	Enable micro increments	Enables extrapolation of the "Counter value". The lower 8 bits of the counter value are extrapolated.	-	BOOL	FALSE
8000:0E	Reversion of rotation	Inverts the counting direction of the encoder.	-	BOOL	FALSE

Index 8010: STM Motor Settings Ch.1

Access rights: read/write

Index (hex)	Name	Description	Unit	Data type	Default value
8010:01	Maximal current	The maximum current that the current controller outputs per motor winding. The maximum value that should be entered here is the nominal motor current.	mA	UINT	1000_{dec}
8010:02	Reduced current	Setpoint for the winding current when the motor is at standstill.	mA	UINT	1000_{dec}
8010:03	Nominal voltage	The DC link voltage U_P	1 mV	UINT	50000_{dec}
8010:06	Motor fullsteps	Number of full steps per motor revolution.	-	UINT	200_{dec}
8010:07	Encoder increments (4-fold)	Number of encoder increments per revolution with 4-fold evaluation. Usually this is the resolution (ppr) of the encoder multiplied by 4.	-	UINT	4096_{dec}
8010:09	Start velocity	Maximum possible start velocity of the motor.		UINT	0
8010:10	Drive on delay time	Delay between enabling of the driver stage (variable "Enable") and setting the "Ready" status bit to 1.	ms	UINT	100_{dec}
8010:11	Drive off delay time	Delay between setting the "Ready" status bit to 0 and disabling the driver stage.	ms	UINT	150_{dec}

Index 8012: STM Features Ch.1

Access rights: read/write

Subindex (hex)	Name	Description	Unit	Data type	Default value
01	Operation mode	Operation mode [▶ 43] 0_{dec}: Automatic 1_{dec}: Velocity direct 2_{dec}: Velocity controller 3_{dec}: Position controller	-	USINT	0_{dec}
05	Speed range	The maximum step frequency [▶ 42] that EPP7041-x002 outputs.	Full steps / s	USINT	1_{dec}
08	Feedback type	Possible values: 0: "Encoder" 1: "Internal counter"	-	USINT	1_{dec}
09	Invert motor polarity	Reverses the direction of rotation of the motor.	-	BOOL	FALSE
11	Select info data 1	This value determines the content of the variable "Info data 1" in the process data object "STM Synchron info data" [▶ 18]. Possible values: 0_{dec}: Status word 7_{dec}: Motor velocity 11_{dec}: Motor load 13_{dec}: Motor dc current 101_{dec}: Internal temperature 103_{dec}: control voltage 104_{dec}: Motor supply voltage 150_{dec}: Drive – Status word 151_{dec}: Drive – State 152_{dec}: Drive - Position lag (low word) 153_{dec}: Drive - Position lag (high word)	-	USINT	11_{hex}
19	Select info data 2	This value determines the content of the variable "Info data 2" in the process data object "STM Synchron info data" [▶ 18]. Possible values: see subindex 11 "Select info data 1"	-	USINT	13_{dec}
30	Invert digital input 1	Invert digital input 1.	-	BOOL	FALSE
31	Invert digital input 2	Invert digital input 2.	-	BOOL	FALSE
32	Function for input 1	0_{dec}: Normal input (*factory setting*) 1_{dec}: Hardware enable 2_{dec}: PLC cam 3_{dec}: Auto start	-	BOOL	FALSE
36	Function for input 2	0_{dec}: Normal input (*factory setting*) 1_{dec}: Hardware enable 2_{dec}: PLC cam 3_{dec}: Auto start	-	BOOL	FALSE
3A	Function for output 1	0_{dec}: Normal output 1_{dec}: Break (linked with driver enable) 15_{dec}: Disabled (*factory setting*)	-	BOOL	FALSE

Index 8013: STM Controller Settings 2 Ch.1

Access rights: read/write

Index (hex)	Name	Description	Unit	Data type	Default value
8013:01	Kp factor (velo./ pos.)	Proportional component of the velocity controller / position controller.	0.001	UINT	1000_{dec}
8013:02	Ki factor (velo./ pos.)	Integral component of the velocity controller / position controller.	0.001	UINT	0
8013:03	Inner window (velo./pos.)	Inner window of the integral component.	1%	UINT	0
8013:05	Outer window (velo./pos.)	Outer window of the integral component.	1%	UINT	0
8013:06	Filter cut off frequency (velo./ pos.)	Filter limit frequency of the velocity controller / position controller.	1 Hz	UINT	0
8013:07	Ka factor (velo./ pos.)	Acceleration component of the velocity controller / position controller.	0.001	UINT	0
8013:08	Kd factor (velo./ pos.)	Braking component of the velocity controller / position controller.	0.001	UINT	0

Index 8020: POS Settings Ch.1

Access rights: read/write

See Positioning Interface [▶ 53].

Index 8021: POS Features Ch.1

Access rights: read/write

See Positioning Interface [▶ 53].

7.3.2 Standard objects

Index 1000 Device type

Access rights: read only

Subindex (hex)	Name	Description	Data type	Value
-	Device type	Bit 0 .. 15: Device profile number Bit 16 .. 31: Module profile number (Device profile number 5001: Modular Device Profile MDP)	UDINT	5001_{dec}

Index 1008 Device name

Access rights: read only

Subindex (hex)	Name	Description	Data type	Value
-	Device name	Name of the EtherCAT device	STRING	EPP7041-x002

Index 1009 Hardware version

Access rights: read only

Subindex (hex)	Name	Description	Unit	Data type	Value
-	Hardware version	Hardware version [▶ 7] of the EtherCAT device	-	STRING	1)

Index 100A Software version

Access rights: read only

Subindex (hex)	Name	Description	Unit	Data type	Value
-	Software version	Firmware version [▶ 7] of the EtherCAT device	-	STRING	1)

Index 1011 Restore default parameters

Access rights: read/write

Subindex (hex)	Name	Description	Data type	Default
1	Subindex 001	Resets the CoE parameters to the factory settings. To do this, write the value 0x64616F6C in this parameter.	UDINT	0

Index 1018 Identity

Access rights: read only

Subindex (hex)	Name	Description	Data type	Value
01	Vendor ID	Vendor identifier (2: Beckhoff Automation)	UDINT	2_{dec}
02	Product code	Product code	UDINT	$64777E18_{hex}$
03	Revision	Bit 0...15: Index number of the product version Bit 16...31: Revision of the device description (ESI)	UDINT	Bit 0...15: $x002_{dec}$
04	Serial number	Reserved	UDINT	0

8 Appendix

8.1 General operating conditions

Protection degrees (IP-Code)

The standard IEC 60529 (DIN EN 60529) defines the degrees of protection in different classes.

1. Number: dust protection and touch guard	Definition
0	Non-protected
1	Protected against access to hazardous parts with the back of a hand. Protected against solid foreign objects of Ø 50 mm
2	Protected against access to hazardous parts with a finger. Protected against solid foreign objects of Ø 12.5 mm.
3	Protected against access to hazardous parts with a tool. Protected against solid foreign objects Ø 2.5 mm.
4	Protected against access to hazardous parts with a wire. Protected against solid foreign objects Ø 1 mm.
5	Protected against access to hazardous parts with a wire. Dust-protected. Intrusion of dust is not totally prevented, but dust shall not penetrate in a quantity to interfere with satisfactory operation of the device or to impair safety.
6	Protected against access to hazardous parts with a wire. Dust-tight. No intrusion of dust.
2. Number: water* protection	Definition
0	Non-protected
1	Protected against water drops
2	Protected against water drops when enclosure tilted up to 15°.
3	Protected against spraying water. Water sprayed at an angle up to 60° on either side of the vertical shall have no harmful effects.
4	Protected against splashing water. Water splashed against the disclosure from any direction shall have no harmful effects
5	Protected against water jets
6	Protected against powerful water jets
7	Protected against the effects of temporary immersion in water. Intrusion of water in quantities causing harmful effects shall not be possible when the enclosure is temporarily immersed in water for 30 min. in 1 m depth.

*) These protection classes define only protection against water.

Chemical Resistance

The Resistance relates to the Housing of the IP67 modules and the used metal parts. In the table below you will find some typical resistance.

Character	Resistance
Steam	at temperatures >100°C: not resistant
Sodium base liquor (ph-Value > 12)	at room temperature: resistant > 40°C: not resistant
Acetic acid	not resistant
Argon (technical clean)	resistant

Key

- resistant: Lifetime several months
- non inherently resistant: Lifetime several weeks
- not resistant: Lifetime several hours resp. early decomposition

8.2 Accessories

Protective caps for connectors

Ordering information	Description
ZS5000-0010	Protective cap for M8 sockets, IP67 (50 pieces)
ZS5000-0020	Protective cap for M12 sockets, IP67 (50 pcs.)

Labelling material

Ordering information	Description
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10
ZS5000-xxxx	Printed inscription labels on enquiry

Cables

A complete overview of pre-assembled cables for fieldbus components can be found here.

Ordering information	Description	Link
ZK2000-6xxx-xxxx	Sensor cable M12, 4-pin	Website
ZK2000-5xxx-xxxx	Sensor cable M12, 5-pin	Website
ZK4000-51xx-xxxx	Encoder cable, shielded	Website
ZK4000-6768-0xxx	Motor cable, shielded	Website
ZK700x-xxxx-xxxx	EtherCAT P cable M8	Website

Tools

Ordering information	Description
ZB8801-0000	Torque wrench for plugs, 0.4…1.0 Nm
ZB8801-0001	Torque cable key for M8 / wrench size 9 for ZB8801-0000
ZB8801-0002	Torque cable key for M12 / wrench size 13 for ZB8801-0000

Further accessories

Further accessories can be found in the price list for fieldbus components from Beckhoff and online at https://www.beckhoff.com.

8.3 Version identification of EtherCAT devices

8.3.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

8.3.2 Version identification of EP/EPI/EPP/ER/ERI boxes

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

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

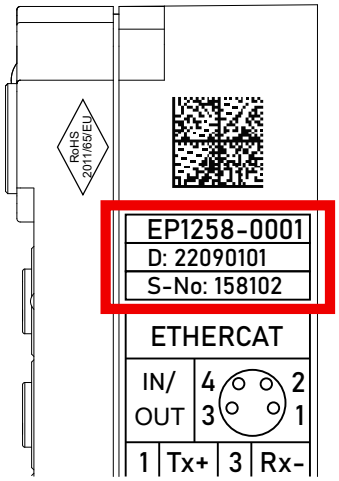


Fig. 26: EP1258-00001 IP67 EtherCAT Box with batch number/DateCode 22090101 and unique serial number 158102

8.3.3 Beckhoff Identification Code (BIC)

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

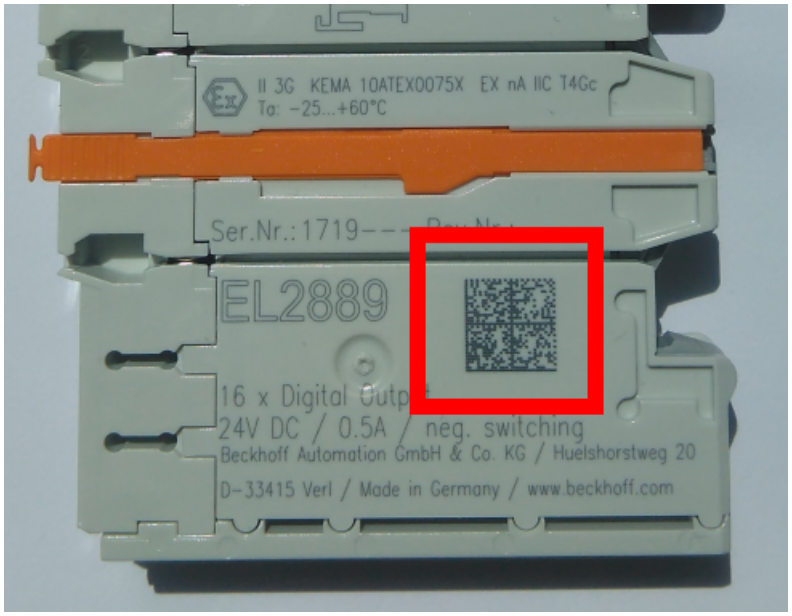


Fig. 27: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	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	51S678294
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 positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 28: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

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

8.3.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

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

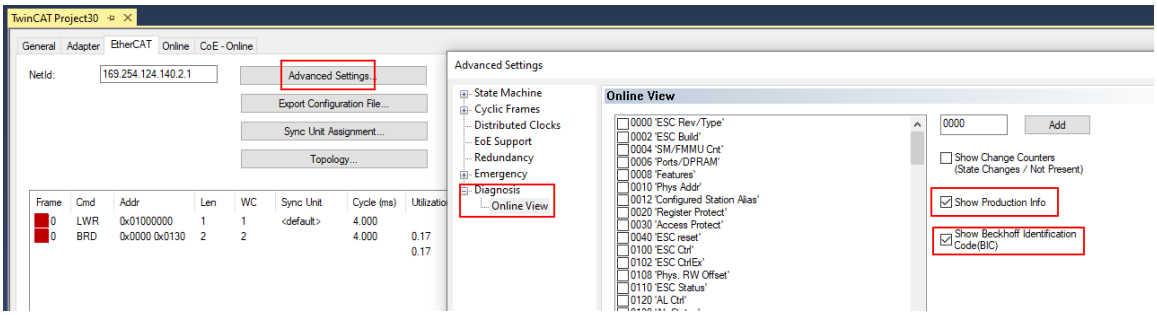
EtherCAT devices (IP20, IP67)

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

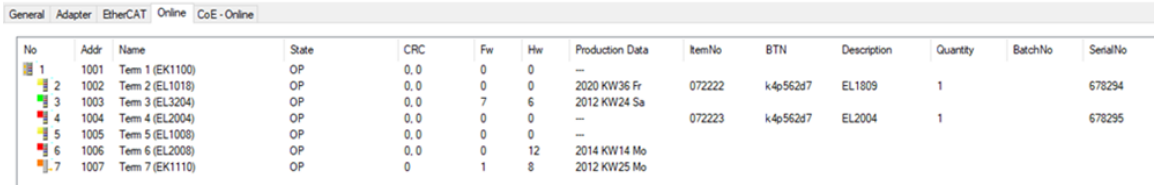
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

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

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



  - The BTN and its contents are then displayed:



  - Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
  - From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.
- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:

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

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

- the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.

- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
  - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
  - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
  - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

Profibus/Profinet/DeviceNet... Devices

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

8.4 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

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