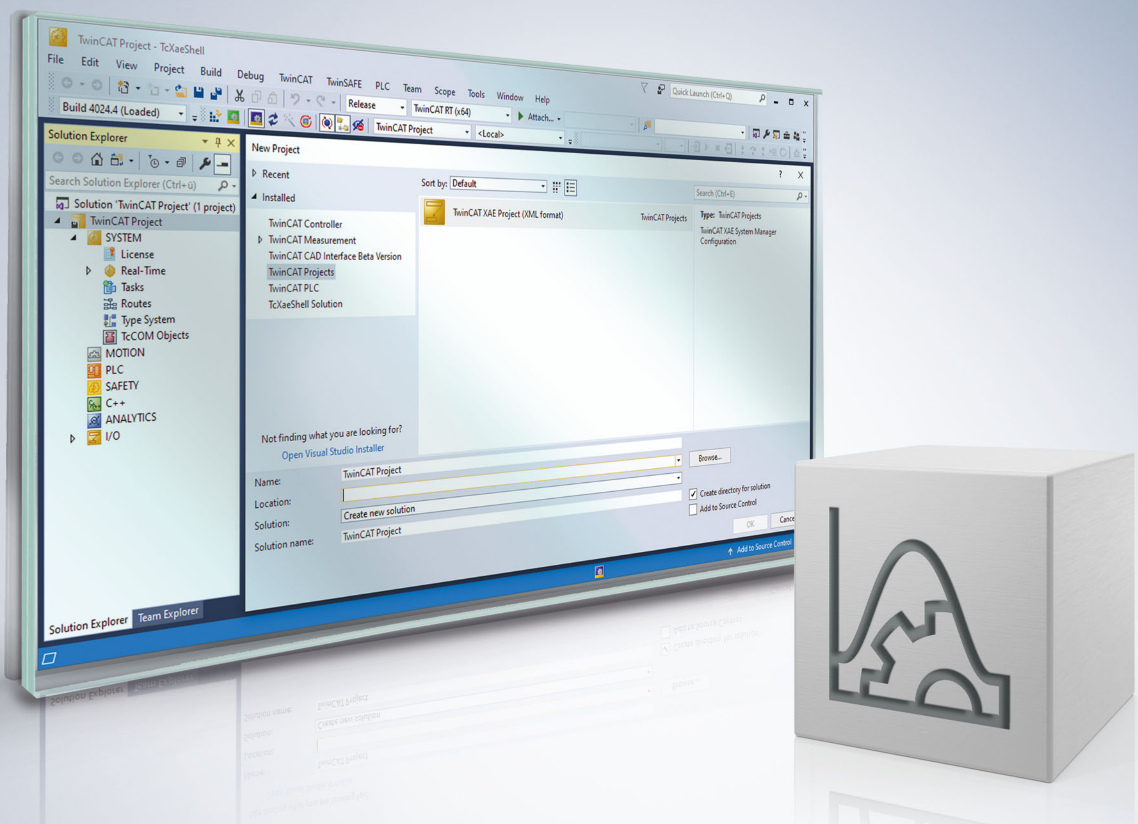
BECKHOFF New Automation Technology

Manual | EN

# TF50x0

## TwinCAT 3 | NC PTP




2024-04-11 | Version: 2.1.1

# 1 Foreword

## 1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all 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 notice.
No claims to modify 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 and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

**Patents**

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

EtherCAT

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

**Copyright**



## 1.2 For your safety

**Safety regulations**

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

### Exclusion of liability

All the components are supplied in particular hardware and software configurations which are 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 technology who are familiar with the applicable national standards.

### Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

### Personal injury warnings

| ⚠ DANGER |
| --- |
| Hazard with high risk of death or serious injury. |

| ⚠ WARNING |
| --- |
| Hazard with medium risk of death or serious injury. |

| ⚠ CAUTION |
| --- |
| There is a low-risk hazard that could result in medium or minor injury. |

### Warning of damage to property or environment

| *NOTICE* |
| --- |
| The environment, equipment, or data may be damaged. |

### Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

## 1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

# Table of contents


**1 Foreword ... 3**
- 1.1 Notes on the documentation ... 3
- 1.2 For your safety ... 3
- 1.3 Notes on information security ... 5

**2 Overview ... 9**

**3 Safety measures for axis commissioning ... 11**

**4 NC safety functions ... 13**

**5 NC/PTP NCI Configuration ... 15**
- 5.1 SAF task ... 15
- 5.2 SVB task ... 17
- 5.3 Image ... 18
- 5.4 Tables ... 18
- 5.5 Objects ... 18

**6 Axes ... 20**
- 6.1 Axis types ... 20
  - 6.1.1 Continuous axis ... 20
  - 6.1.2 Encoder Axis ... 20
  - 6.1.3 Discrete Axis (two speed) ... 20
- 6.2 NC axis ... 21
  - 6.2.1 Axis dialog box ... 21
  - 6.2.2 Axis component encoder ... 43
  - 6.2.3 Axis component Drive ... 57
  - 6.2.4 Axis component cyclic interface ... 71

**7 Parameters for Motion ... 73**
- 7.1 Axes | Axis 1 ... 73
  - 7.1.1 Maximum Dynamics, Default Dynamics ... 73
  - 7.1.2 Manual Motion and Homing ... 75
  - 7.1.3 Fast Axis Stop ... 77
  - 7.1.4 Limit Switches ... 78
  - 7.1.5 Monitoring ... 79
  - 7.1.6 Setpoint Generator ... 83
  - 7.1.7 NCI Parameters ... 84
  - 7.1.8 Other Settings ... 85
- 7.2 Axes | Axis 1 | Enc ... 88
  - 7.2.1 Encoder Evaluation ... 88
  - 7.2.2 Limit Switches ... 91
  - 7.2.3 Filter ... 92
  - 7.2.4 Homing ... 93
  - 7.2.5 Other Settings ... 95
- 7.3 Axes | Axis 1 | Drive ... 96
  - 7.3.1 Output Settings ... 97
  - 7.3.2 Position and Velocity Scaling ... 98
  - 7.3.3 Torque and Acceleration Scaling ... 99

7.3.4 Valve Diagram ... 101
7.3.5 Optional Position Command Output Smoothing Filter ... 102
7.3.6 Sercos Behavior ... 103
7.3.7 Other Settings ... 103
7.4 Axes | Axis 1 | Ctrl ... 104
7.4.1 Monitoring ... 106
7.4.2 Position Control Loop ... 106
7.4.3 Velocity Control Loop ... 110
7.4.4 Observer ... 111
7.4.5 Other Settings ... 112
7.5 Technical Terms ... 113
7.5.1 Acceleration Feedforward ... 113
7.5.2 Automatic DAC Offset Adjustment ... 113
7.5.3 Axis Error ... 114
7.5.4 AXIS_REF ... 114
7.5.5 NC ... 114
7.5.6 Path Override (Interpreter Override Types) ... 114
7.5.7 PTP ... 116
7.5.8 PT1 Filter ... 117
7.5.9 Rapid Traverse ... 117
7.5.10 ReadStatus() ... 117
7.5.11 Tolerance Ball ... 118

**8 First steps ... 120**
8.1 Create project ... 120
8.2 Create axis ... 122
8.2.1 Create simulation axis ... 122
8.2.2 Create AX5000 and NC axis automatically ... 124
8.2.3 Create AX5000 and NC axis manually ... 129
8.3 Axis configuration ... 134
8.3.1 Limiting positions ... 134
8.4 Moving the axis ... 135
8.4.1 Activating a TwinCAT project ... 135
8.4.2 Manual movement via the commissioning dialog ... 136
8.4.3 Simple movement via the PLC ... 137
8.5 Scope integration ... 141

**9 Special NC functionalities ... 146**
9.1 Modulo positioning ... 146
9.2 External set value generation ... 149
9.3 NC Backlash Compensation ... 151
9.3.1 Mechanical backlash ... 151
9.3.2 NC Implementing of the TwinCAT Position Correction ... 153
9.3.3 NC Implementing the TwinCAT Backlash Compensation ... 153
9.4 Discrete Axis (two speed) ... 154

**10 Support and Service ... 162**

# 2 Overview

TF5000 TwinCAT NC PTP is the basis for controlling simple axes up to complex multi-axis systems.

- **TF5000**: Up to 10 axes (extendable up to a maximum of 255 axes).
- **TF5010**: Extension of TF5000 to up to 25 axes (extendable to a maximum of 255 axes).
- **TF5020**: Extension of TF5000 to up to 255 axes.

### Axis objects

The real axes are represented by axis objects. This results in a separation between physical and logical axis. Rotatory and linear, electric and hydraulic drives can thus be programmed in the same way. In addition, the axis object can simulate an axis. Initial program tests without physical hardware are thus possible. The control of the axes can be configured in various conformations (position or velocity interface) and various controllers. The axes are configured in TwinCAT *Engineering*.

- Supports electrical and hydraulic servo drives, frequency converter drives, stepper motor drives, DC drives, switched drives (two-speed axes [▶ 154]), simulation axes and encoder axes [▶ 20].
- Supports various encoders such as incremental encoder, absolute encoder, digital interface to the drives such as EtherCAT, SERCOS, SSI, Lightbus, PROFIBUS DP/MC, Pulse Train.
- Provides standard axis functions such as start, stop, reset, referencing, velocity override, master/slave couplings, electronic gearing, online position compensation.
- Provides convenient methods for axis commissioning.
- Contains online monitors for all axis state variables, such as actual/setpoint values, enables, control values, online axis tuning.
- Forcing of axis variables.
- Configuration of all axis parameters, such as measuring system, drive parameters and position controller.
- Configurable controller structures: P control, PID control, PID with velocity pre-control, PID with velocity and acceleration pre-control.

### Programming

For programming, the Tc2_MC2 library provides PLCopen-compliant administrative and motion-initiating function blocks. In addition to the pure movements of an axis from an initial position to a target position, movements of two axes can also be operated in a gear coupling. This is referred to as gearing.

A coordinated master-slave motion with non-linear relationships is called cam plate coupling. These can be added to the TwinCAT NC PTP by TF5050 | TwinCAT NC Camming.

TF5060 | TwinCAT NC FIFO AXES extends the TwinCAT NC PTP with a table coupling concept for arbitrary, usually not cyclically recurring processes. In this case, rather than having one master-slave table prescribed in advance, the latest axis positions are constantly "topped up" by a PLC program.

Another addition to the axis coupling is TF5055 | TwinCAT Flying Saw. Here, a slave axis is coupled 'flying' to a moving master axis at a specific position. As the name suggests, such couplings are often used when, for example, a board needs to be cut with a saw on a moving conveyor belt.

- Programming is carried out via PLCopen-compliant IEC 61131-3 function blocks of the Tc2_MC2 library.
- Online master/slave and slave/master conversion
- "Flying Saw" (diagonal saw) with TF5055 | TwinCAT Flying Saw
- Cam plates with TF5050 | TwinCAT NC Camming
  (supported by TwinCAT 3 CAM Design Editor [optional])
- FIFO axes with TF5060 | TwinCAT NC FIFO AXES
- External setpoint generators [▶ 149]
- Multi-master coupling

**Further information**

- PLC libraries
- ADS specification for NC

# 3 Safety measures for axis commissioning

**Axis commissioning includes:**

1. In particular, the knowledge and use of NC security functionalities,
2. taking the necessary safety precautions and
3. compliance with a specific sequence of commissioning steps.

**⚠ DANGER**

**Danger to life or risk of serious injury or damage to property due to unintentional movements of the axis**

When commissioning axes, there is a movement of them and the mechanics coupled to them, which creates a hazard for people and a risk of damage to the machine. The following safety measures provide guidelines for safe commissioning. The actual measures to be taken depend on the axis and its surroundings.

As a general rule, "Don't take an action whose consequence you can't estimate."

### General safety precautions

Inform yourself about the safety instructions and commissioning steps of your drive, such as: AX8000 and the NC safety functions [▶ 13] and use them.

Handling the emergency stop case is one of the most important and safety-relevant features of a machine. For this reason it is usual for an additional hardware safety circuit to be built on top of all the logical (logical axis enables) and software enables (PLC: controller and feed enables). Extensive statutory regulations exist governing the reaction and the implementation in hardware of the emergency stop system. Here we only make reference to these.

Within the limits prescribed by the statutory regulations, there remains often a degree of free choice as to how, for instance, NC axes are stopped in the event of an emergency stop. In some cases the mechanical properties of the machine mean that it is not always possible to bring axes that are moving to an abrupt halt (e.g. abruptly output of 0.0 V of an analog interface). It is therefore quite common for NC axes to be slowed following special braking ramps in an emergency stop situation, after which they are then electrically and mechanically locked.

One of the first procedures for commissioning a machine is to test the **emergency stop circuit**. We begin with safe, standard situations (the axes are stationary under emergency stop conditions, although special attention must be paid here to axes that move vertically). The testing then moves on to more complicated cases (axes moving slowly, then axes moving rapidly).

It is possible, in order to test and commission rapidly, to set the logical axis enables (controller and feed enables) and the velocity override of the axes from the TwinCAT development environment via the Axis online dialog [▶ 38] for the particular axis*. This is, however, only possible if the axis interface is not involved in cyclical exchange with the PLC (mapping between the NC and PLC tasks). In that case, the single write of information from the development environment would immediately be overwritten again by the cyclic data exchange. It must, however, be pointed out that this procedure can be very dangerous, because the usual safety monitoring and manipulation facilities of the PLC are disabled. Therefore, this feature can be safely used on an axis test rig with endless axes.
***It is extremely inadvisable to use this facility on a real machine.**

### Essential requirements

The description below covers all the steps of commissioning an axis. Several types of axis and a range of different situations are covered. In each step, all the settings which must be made are mentioned, even when they are identical with the settings of the previous step. In practice, only some of the listed (partial) steps are actually executed for an axis.

Before you can begin the commissioning itself, a number of preparations must be made:

- Check their completeness and correctness of all the electrical connections.
- All parts of the axis (encoder, drive, controller, PLC interface) must be of appropriate types, and must be provided with the correct resources (PLC variables, I/O hardware in the fieldbus etc.).

- The NC architecture must not only be created, but must also be written into the registry, and TwinCAT must be started with it.

**Safety precautions**

The following safety precautions must be taken:

- Ensure that no one can enter or reach into the machine: the machine can behave unpredictably during commissioning.
- Isolate those parts of the equipment on which you are currently working. Anything that is not needed must be safely shut down as it may interfere or distract.
- Inform all persons in the vicinity that there is an increased risk of accidents and that they must keep the necessary distance from the machine.
- Make sure you are alone in the hazardous area. Only necessary and instructed as well as competent persons should be present in case of need.

**Sequence**

It is necessary to observe a particular sequence, because the individual steps of axis commissioning are built logically on each other. The sequence, however, does depend on the particular axis composition, i.e. on the combination of the encoder, controller and drive types.

# 4 NC safety functions

A monitoring function (watchdog) is provided in order to monitor the operation and regular updating of the cyclical interface between the PLC and the NC. In addition, there is task runtime overrun monitoring for each task and position lag monitoring and end position monitoring for the NC axes.

**NOTICE**

**No safety functions in the sense of safety technology**

The NC safety functionalities are purely functional settings that are not safety functions in the sense of safety technology.

There are further monitoring options on the hardware side.
Also find out about other safety measures for axis commissioning [▸ 11].

### Watchdog for cyclic axis interface between PLC and NC

The watchdog (function monitoring) functionality between the cyclic axis interface of the PLC and NC should always be activated. This is the case if any value other than zero is entered for the watchdog. The value specifies the number of sequential task cycles following which the watchdog will trigger if no new information has been transferred between the NC and the PLC. If the watchdog is triggered, the corresponding axis interface (PlcToNc or NcToPlc) is cleared, i.e. zeroed.

Example: When the watchdog is active, it is ensured in the following cases that the NC axes are stopped by means of the removal of the controller and feed enable caused by the watchdog. When:

- the PLC is stopped
- an infinite loop has been programmed within the PLC
- an FPU exception occurs

### Task time-out monitoring

For purposes of diagnosis and analysis the task time-out monitoring should be activated. This is true both for the SAF task (I/O task of the NC) and for the SVB task of the NC. As regards content this monitoring has no effect, but should there be an occasion where an unexpected task time-out occurs, the response is in the form of a message box and an additional entry in the event display.

### Position lag monitoring, end position monitoring, target position control

Right from the start of any operations each axis should be driven with both "position lag monitoring" and "end position monitoring" active. Even if an axis that has not yet been optimized travels with sometimes very large lag errors, these elementary monitoring mechanisms should not be switched off, but should be parameterized accordingly (Position Lag Monitoring [▸ 29]).
Furthermore, there is the option of automatically checking the target position (Target Position Monitoring).

### Direction inversion, direction monitoring

There are functionalities, e.g. position compensation on a master axis, which can cause an inversion of the direction of movement. To avoid an unwanted direction of travel, there is a direction-dependent feed enable that stops the axis instantaneously if it travels in the wrong direction.

### Maximum velocity

Furthermore, there is the possibility to define the maximum allowed velocity of an axis in the axis parameters or to limit the output of the drive in percent.

For example, it can happen that, by mistake, the direction of actuation of the axis control loop (positive feedback) is altered as a result of changeover of drive or encoder polarity, and the axis, with full logical consistency, drives towards the mechanical end position at maximum output value.

In the following situations the maximum velocity can be exceeded:

- by position compensation of the master or slave axes

- by setting or changing the coupling factor of a slave axis or (indirectly) of the flying saw
- by externally generated data in the FIFO or the table slave axes

### Stop

All master axes can be stopped at any time.

*Notice* **The flying saw is the only slave axis that has a stop function. However, there are situations when a flying saw cannot be stopped.**

Slave axes can be stopped by disconnecting them online and converting them into master axes, which can then be stopped. The FIFO axes and the NCI group can be stopped.

### Hardware monitoring

It may be that a facility in an emergency situation (emergency stop, watchdog, etc.) must not be allowed under any circumstances, for mechanical or other reasons, to halt abruptly its axes in the next I/O cycle (e.g. to output 0 V suddenly). Such behavior can only be ensured via the drive hardware that is present. To this end most manufacturers offer simple digital circuitry options that ensure that an axis is brought to a halt in a defined way (braking ramp, standstill window for electrical deactivation of the control system and activation of the brakes, etc.).

# 5 NC/PTP NCI Configuration

A NC/PTP NCI Configuration is divided into the following sections:

| | |
|---|---|
| NC-Task 1 SAF [▸ 15] | • SAF task<br>• Task for the block execution<br>• Task in which the setpoint generation takes place.<br>• Task that serves the fieldbus IO of the NC. |
| NC-Task 1 SVB [▸ 17] | • SVB task<br>• Task for the block preparation<br>• Linking and "Look-ahead" function of NCI segments<br>• no effect on single-axis movements (PTP)<br>• not responsible for the fieldbus IO of the NC |
| Image [▸ 18] | • NC process image |
| Tables [▸ 18] | • Tables, e.g. for cam plates |
| Objects [▸ 18] | • additional TcCOM objects |
| Axes [▸ 20] | • NC axis configuration. |

Additionally, channels of the following types can be added to the configuration:

| | |
|---|---|
| **NC Channel (for Interpolation)** | • TF5100 \| TwinCAT 3 NC I |
| **NC Channel (for FIFO Axes)** | • TF5055 \| TwinCAT 3 NC FIFO Axes |
| **NC Channel (for Kinematic Transformation)** | • TF511x \| TwinCAT 3 Kinematic Transformation |

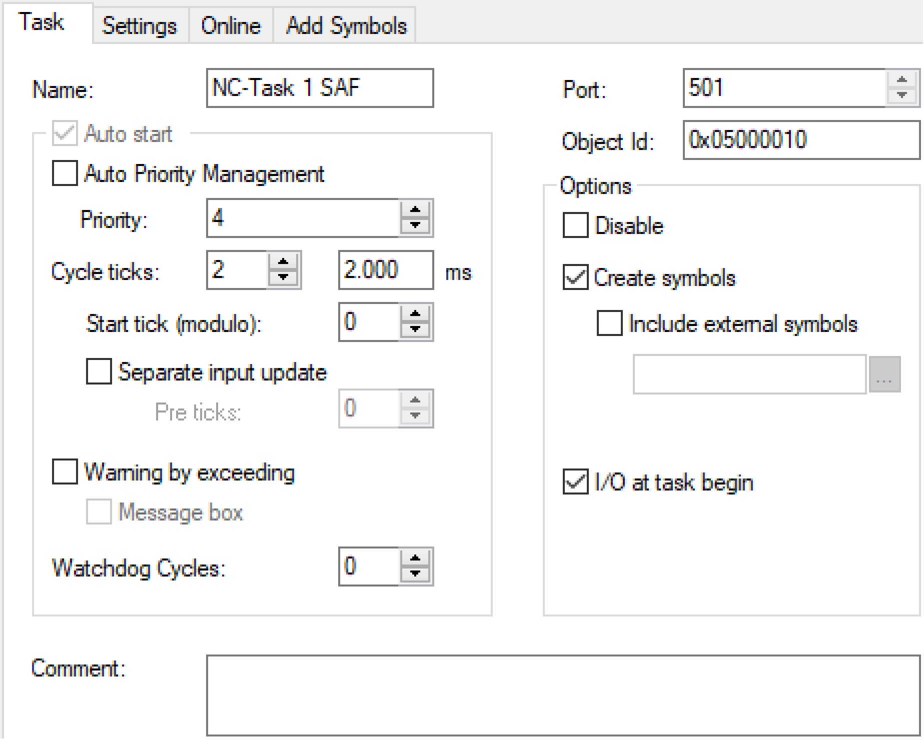
## 5.1 SAF task

The block execution task (SAF task) directly executes commands that do not require preprocessing or have been prepared by the SVB task [▸ 17] and handles cyclic communication with the drive devices.

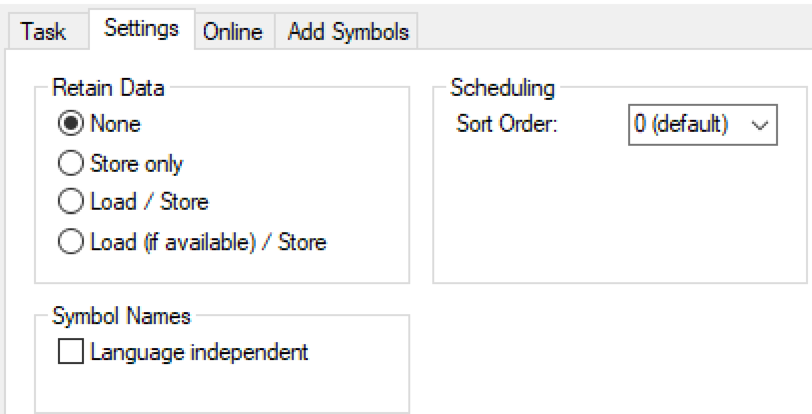
- motion commands of the Tc2_MC2 library, like MC_MoveAbsolute, MC_MoveRelative, MC_MoveVeloctiy etc.
- coupling axes
- cyclic setpoint generation for all axes and output to drives
- cyclic acquisition of the actual position and position control, if not directly controlled by the drive
- I/O communication, e.g. for the evaluation of latch positions

### Task tab

The SAF task is configured via the **NC/PTP NCI Configuration** node below the **MOTION** node, which is called **NC-Task 1 SAF** by default. Details about the task dialog, see TE1000 | The TwinCAT Project.




### Settings tab




### Retain data

TwinCAT-NC uses retain data to restore the position of individual axes with absolute measured value system at system startup. The Retain setting determines globally whether data required for this purpose is saved when the system is stopped and loaded when the system is started. In addition, the storage must be parameterized for each axis that requires such data (see Data Persistence [▶ 37]).

**NOTICE**

**Prevent data loss**

If retain data is used, it is recommended that the system is backed up with a UPS so that the data can be stored safely even if the supply voltage fails.

- *None:* No retain data is saved or loaded.
- *Store only:* No retain data is loaded at system startup. Retain data is saved when the system is stopped. This setting is only used for compatibility with old configurations.
- *Load/Store:* Retain data is stored at system stop and loaded at system start. If there is no data or only corrupt data at system startup, the system aborts the startup process with an error.
  The system cannot be started in this mode when the retain data is configured for the first time, so the *Mode Load (if available)/Store* should be set first and then reset to *Load/Store* after a successful system start.

- *Load (if available)/Store:* Retain data is stored at system stop and loaded at system start. If there is no data or only corrupt data at system startup, the system starts without retain data.
  Axes that rely on retain data are in the "not referenced" state. The application should check this state and take action.

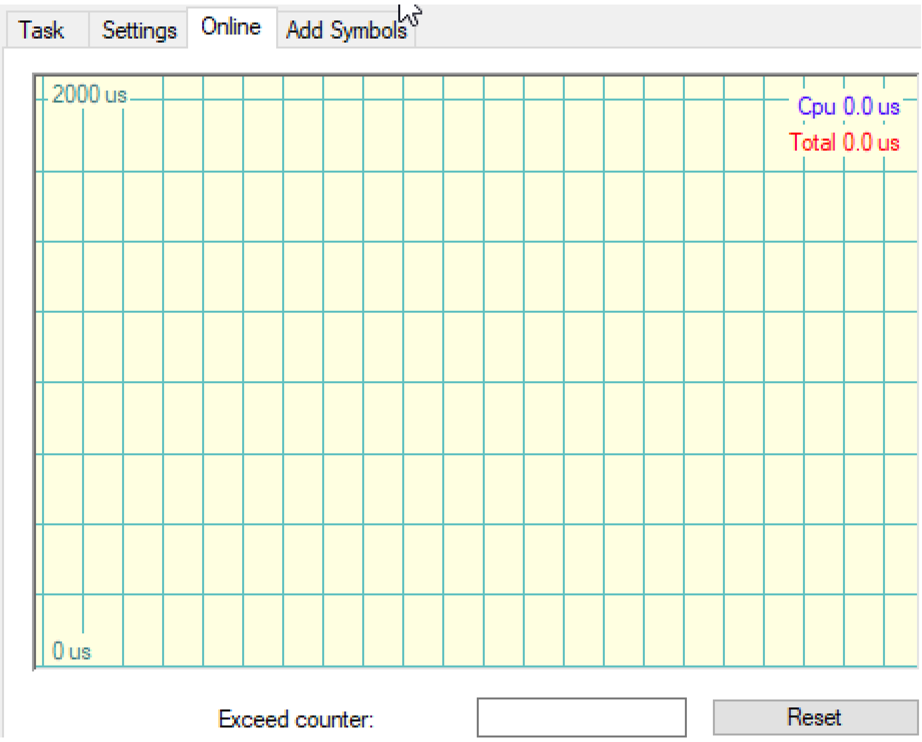
**Symbol Names**

The setting *Symbol Names, Language independent* specifies that the generic part of the symbol name is not changed. This is then always held in English.

For example, without this setting *Axes.Axis 1.SetPos* would change to *Achsen.Axis 1.SetPos* when switching language from English to German, while with the setting it would remain English.

**Online tab**

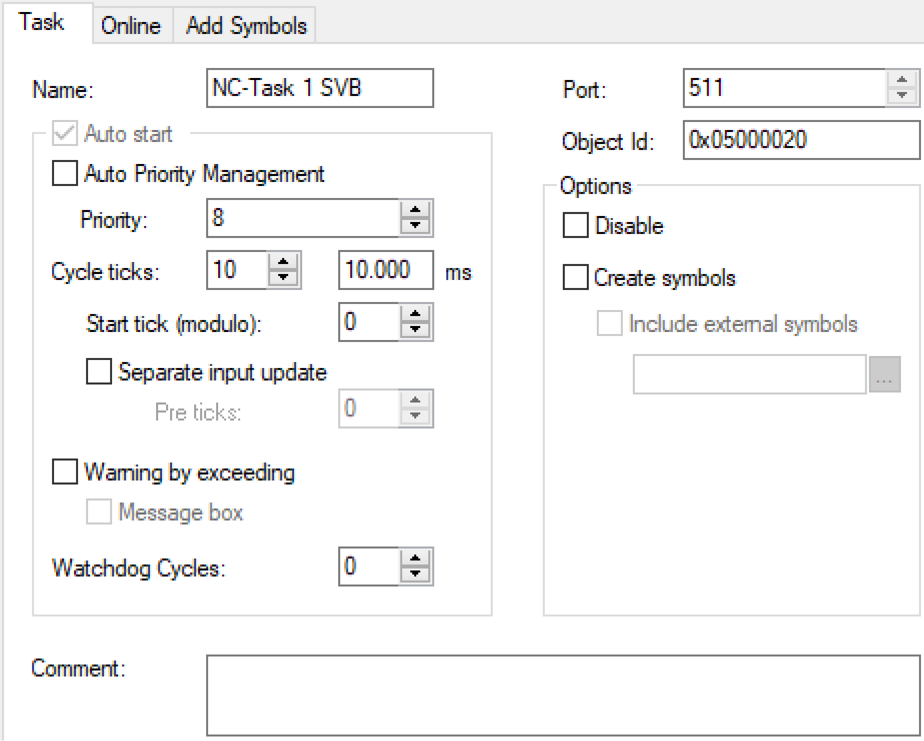
The Online tab shows the utilization of the task over time and indicates the number of cycle time overruns. Details can be found in the TE1000 | TwinCAT 3 XAE documentation.




## 5.2 SVB task

The block preparation task (SVB task) prepares selected commands so that they can subsequently be executed quickly in the SAF task [▶ 15].
These are for example:

- Operation of the axes in the development environment via the Axis online dialog [▶ 38].
- Homing sequence, which can be started e.g. with MC_Home.
- Group commands for kinematic, FIFO and NCI groups




**Settings in relation to the SAF task**

In relation to the SAF task, a higher cycle time and a lower priority should be selected for the SVB task, as is also the case by default.

## 5.3 Image

Under Image the process image of the SAF task is displayed.

General information on the process image of a task, see SAF task [▶ 15].

## 5.4 Tables

Electronic cam plates describe a nonlinear relationship between a master and a slave axis in a coupled axis system.

Valve characteristic curves are required for the control of hydraulic axes. They describe a non-linear relationship between the target velocity of an NC axis and the variable output to the controlled device.

Electronic cam plates and valve characteristic curves can be managed under Tables.

Further information:

- Motion Diagrams
  TE1510 | TwinCAT 3 CAM Design Tool
  TF5050 | TwinCAT 3 NC Camming
- Valve Diagrams
  TE1500 | TwinCAT 3 Valve Diagram Editor

## 5.5 Objects

TwinCAT 3 has a modular structure.

This modular structure is also used for Motion Control.

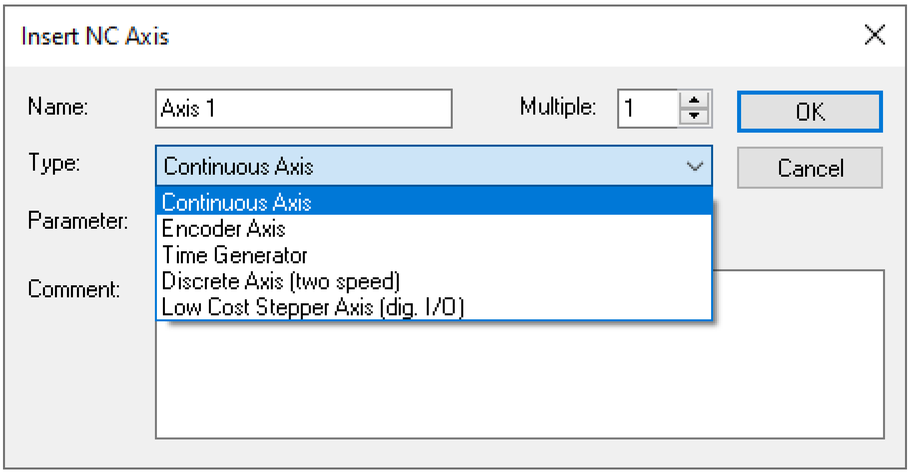
The individual motion objects can be channel-dependent, e.g. axes in the axis channel or kinematic transformations from the "NC Channel (for Kinematic Transformation)" or channel-independent.

The independent objects are managed via the Objects node. These are, for example, the Collision Avoidance and Coordinated Motion groups.

# 6 Axes

## 6.1 Axis types

TwinCAT supports different axis types. The axis type must be set when creating an axis at **Motion > Axes**:




The most important axis types are described below with their characteristics.

### 6.1.1 Continuous axis

A continuous axis in TwinCAT means that a drive hardware is to be operated that can process continuous setpoint specifications.

It is irrelevant in which form or with which hardware these specifications are ultimately made. This can be a drive controller that follows continuous position or even continuous velocity setpoints. This group includes digital servo drive controllers. But this can also be a hardware that follows an analog value.

Usually, an actual value is also provided by this hardware.

### 6.1.2 Encoder Axis

In the TwinCAT context, one speaks of an encoder axis if a pure encoder system is to be integrated into the system as an axis object.

Encoder axes do not have "Drive" or "Ctrl" as sub-elements of the axis.

An encoder axis can be used in the system as master of couplings, but never be slave.

A special feature of the encoder axis is that the setpoint corresponds to the actual value, since this axis type does not have its own setpoint generator. If couplings are set up with an encoder axis, they are therefore indirectly coupled to the actual value (couplings in the TwinCAT system are always setpoint-based).

This leads to the fact that a noise component is to be seen in the setpoint and is also transferred to slave axes.

### 6.1.3 Discrete Axis (two speed)

This TwinCAT axis type (Two-Speed) enables the positioning of a so-called rapid/creep axis. Such an axis can be physically made up of a motor with two speeds (switching of the pole pair numbers), or alternatively by a motor that can be driven at two speeds with the help of a frequency inverter.

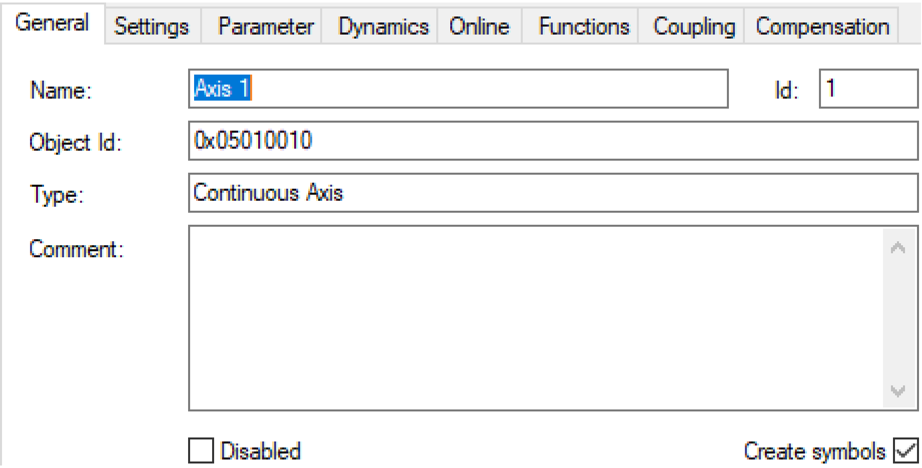
The TwinCAT axis type Two-Speed represents a special form of an axis. The following pages do not deal with this TwinCAT axis type in particular, but assume a Continuous axis [▸ 20] or Encoder axis [▸ 20].

Detailed information for this special type can be found in the chapter Discrete Axis (two speed) [▶ 154] under the special NC functionalities [▶ 146].

# 6.2 NC axis

## 6.2.1 Axis dialog box

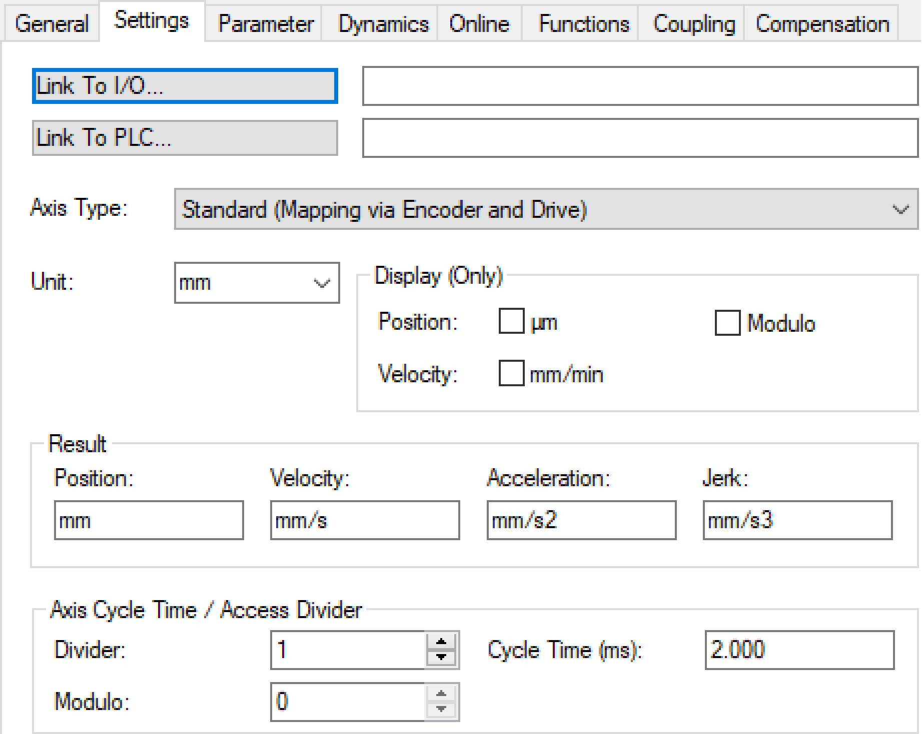
### 6.2.1.1 General




| Property | Description |
|---|---|
| **Name** | Selected name of the object.<br>**Do not assign an object name more than once!** |
| **Object Id** | 32-bit identification number that is automatically assigned and unique within a project. |
| **Type** | Axis type [▶ 20]<br>The axis type is defined when the axis is created and cannot be changed later. |
| **Comment** | Freely editable field for user notes. |
| **Id** | Identification number of the axis, which is assigned consecutively. When an axis is deleted, the number becomes free again and can be assigned to a new axis. |
| **Disabled** | Option to disable an axis |
| **Create symbols** | This option generates typical symbol names that are used in TwinCAT Scope, for example. |

### 6.2.1.2 Settings

In the **Settings** tab essential settings like linking the NC axis with hardware and PLC as well as type and unit can be made.




| Setting | | Description |
|---|---|---|
| **Link To I/O…** | | The Link button opens a dialog for linking the NC axis with the drive hardware under I/O.<br>The link is displayed in the field to the right. |
| **Link To PLC…** | | The Link button opens a dialog for linking the NC axis with the PLC instance of the axis.<br>The link is displayed in the field to the right. |
| **Axis Type** | | Type of connected drive hardware and protocol used. |
| **Unit** | | Physical unit of the position of the axis. The unit can be chosen arbitrarily and edited in the Unit input field. It should be noted that the scaling factor of the axis must be set accordingly (see Encoder parameters).<br>Standard: Millimeter (mm) |
| **Display (only)** | | Adjustments to the display in the online axis dialog. These settings do not affect data in the process image. |
| | Position | Changes the position display by one thousand digit (mm/µm) |
| | Velocity | Display of velocity in mm/min instead of mm/s.<br>The default time reference in NC and PLC remains the second regardless of the display setting. |
| | Modulo | Display of the modulo position instead of the absolute position. |
| **Result** | | |
| | Position | Position |
| | Velocity | Velocity |
| | Acceleration | Acceleration |
| | Jerk | Jerk |
| **Axis Cycle Time / Access Divider** | | |
| | Divider | The axis is executed in every nth cycle of the NC-SAF task with the cycle time divider. The divider can be set to a value greater than 1 to reduce the system load for low-priority axes. |
| | Modulo | With a cycle time divider greater than 1, the modulo value determines in which NC-SAF cycle the axis is processed. Axes with the same modulo are processed in the same cycle. For |

| Setting | | Description |
|---|---|---|
| | | even distribution of the system load, the axes should be distributed to different modulo values.<br>Example: Divider=4, Modulo 0..3 |
| | Cycle Time (ms) | Axis cycle time |

### 6.2.1.3 Parameter

Various axis settings can be made via the **Parameter** tab, which are described below.

General | Settings | Parameter | Dynamics | Online | Functions | Coupling | Compensation

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Maximum Dynamics: | | | | |
| | Reference Velocity | 2200.0 | 2200.0 | F | mm/s |
| | Maximum Velocity | 2000.0 | 2000.0 | F | mm/s |
| | Maximum Acceleration | 15000.0 | 15000.0 | F | mm/s2 |
| | Maximum Deceleration | 15000.0 | 15000.0 | F | mm/s2 |
| - | Default Dynamics: | | | | |
| | Default Acceleration | 1500.0 | 1500.0 | F | mm/s2 |
| | Default Deceleration | 1500.0 | 1500.0 | F | mm/s2 |
| | Default Jerk | 2250.0 | 2250.0 | F | mm/s3 |
| + | Manual Motion and Homing: | | | | |
| + | Fast Axis Stop: | | | | |
| + | Limit Switches: | | | | |
| + | Monitoring: | | | | |
| + | Setpoint Generator: | | | | |
| + | NCI Parameter: | | | | |
| + | Other Settings: | | | | |

Download | Upload | Expand All | Collapse All | Select All

#### 6.2.1.3.1 Maximum Dynamics, Default Dynamics

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Maximum Dynamics: | | | | |
| | Reference Velocity | 2200.0 | | F | mm/s |
| | Maximum Velocity | 2000.0 | | F | mm/s |
| | Maximum Acceleration | 15000.0 | | F | mm/s2 |
| | Maximum Deceleration | 15000.0 | | F | mm/s2 |

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Default Dynamics: | | | | |
| | Default Acceleration | 1500.0 | | F | mm/s2 |
| | Default Deceleration | 1500.0 | | F | mm/s2 |
| | Default Jerk | 2250.0 | | F | mm/s3 |

**Dynamic-Parameters**

- Velocity *Vel*,
- Acceleration *Acc*,
- Deceleration *Dec*,
- Jerk.

The *jerk* is the derivative of acceleration or deceleration with respect to time. Thus, it describes how quickly acceleration or deceleration change.

**Reference Velocity**

For drives that are not directly controlled by a digital velocity value, e.g. a voltage or current interface, the Reference Velocity is used to scale the drive output. The Reference Velocity is at the same time an upper velocity limit that cannot be exceeded in addition to the maximum velocity. For all drive types, the Reference Velocity must be set greater than or equal to the maximum velocity.

(Strictly speaking, the velocity upper limit is the Reference Velocity divided by the Output Ratio, if an Output Ratio smaller than 1.0 is parameterized)

For details, see the Drive Parameters > Reference Velocity [▶ 58].

**"Maximum Dynamics" and "Default Dynamics"**

The dynamic parameters are absolute, unsigned values. The default values are used if the user has not specified any values, e.g. for a motion command. The maximum values limit the axis dynamics and must be parameterized greater than or equal to the default dynamics.

The maximum values are observed by newer products like the Tc3_McCoordinatedMotion Library. However, for some products, such as the Tc2_MC2 library, the maximum acceleration and the maximum jerk are not taken into account.

**Tc2_MC2 library**

| | |
|---|---|
| Default Dynamics | • If the input value "0.0" is assigned to one of the dynamic parameters "Acceleration, Deceleration, Jerk" at a motion function block or this input is left empty, then a default value is used instead. |
| Maximum Dynamics | • Velocity values that exceed the maximum velocity will not be accepted and will result in an error.<br>• Values for Acceleration, Deceleration and Jerk are not checked for exceeding the maximum parameters, but are accepted. |
| Coupled axes | • In the case of a coupled slave axis, its dynamics depend exclusively on the master movement and maximum values are not checked.<br>• When decoupling a slave axis, various measures are taken to prevent the maximum velocity from being exceeded or the direction of movement from being reversed.<br>• Examples of such measures are increasing the jerk or increasing the acceleration or deceleration to the maximum value. |

**Tc3_McCoordinatedMotion Library, Tc3_McCollisionAvoidance Library**

| | |
|---|---|
| *Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Default Values* | • If for one of the dynamic-parameters “Acc, Dec, jerk” the input value “0.0” is assigned to a motion function block, this assignment leads to an error that means that this value is not allowed.<br>• If for one of the dynamic-parameters “Acc, Dec, jerk” you would like to refer to a default value at a motion function block, this parameter has to be set to the constant value “*MC_Default*”. |

| | |
|---|---|
| *Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Maximum Dynamics* | Vel, Acc, Dec<br>• For the dynamic-parameters “Vel, Acc, Dec” the parameterized values are used.<br>• For the dynamic-parameters “Vel, Acc, Dec” maximum values can be parameterized at a motion function block using the constant value “*MC_Maximum*”.<br>Jerk<br>• There is no maximum value for the jerk.<br>• The jerk is set to the value “unlimited”. Simultaneously, a three-phase-profile or a three-phase-acceleration-setter is applied for motion.<br>Default Values<br>• It is allowed to parameterize default values that exceed their corresponding maximum values.<br>• If a default value is parameterized that exceeds ist corresponding maximum value, a warning will be given, but no error is thrown.<br>• At a Tc3_McCoordinatedMotion-function block or a Tc3_McCollisionAvoidance-function block parameterized default values using the constant value *MC_Default* will be mutually limited to the corresponding maximum values without giving an error message. |

### 6.2.1.3.2 Manual Motion and Homing

**Homing Velocity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Homing Velocity (towards plc cam) | 30.0 | | F | mm/s |
| | Homing Velocity (off plc cam) | 30.0 | | F | mm/s |

**bCalibrationCam**

A boolean input of `MC_Home`. It evaluates the signal of a referencing cam. This reference signal can be coupled into the control unit via a digital input.

**Homing Velocity (towards plc cam)**

Velocity used by a `MC_Home` function block when homing to a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

**Homing Velocity (off plc cam)**

Velocity used by a function block `MC_Home` when moving away from a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

**Manual Velocity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Manual Velocity (Fast) | 600.0 | | F | mm/s |
| | Manual Velocity (Slow) | 100.0 | | F | mm/s |

**Manual Velocity (Fast)**

Online Dialog:

- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | -- F1.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | ++ F4.
- Similar for other identifiers

MC_Jog:

- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards TRUE` and selected as its `Mode MC_JOGMODE_STANDARD_FAST`.

**Manual Velocity (Slow)**

Online Dialog:

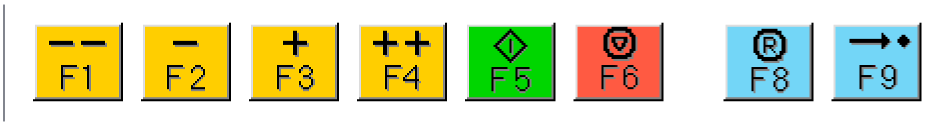
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | - F2.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | + F3.
- Similar for other identifiers.

MC_Jog:

- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards TRUE` and selected as its `Mode MC_JOGMODE_STANDARD_SLOW`.

**Buttons in the Online dialog**

In the "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online" dialog, there are the buttons -- F1, - F2, + F3 and ++ F4.




**Jog Increment**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Jog Increment (Forward) | 5.0 | | F | mm |
| | Jog Increment (Backward) | 5.0 | | F | mm |

**Jog Increment (Forward)**

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

**Jog Increment (Backward)**

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

**MC_JOGMODE_INCHING**

The function block `MC_Jog` enables an axis to be moved via manual keys. The key signal can be connected directly to the `JogForward` or `JogBackwards` input. The desired operation mode is specified by input `Mode`. When using the `MC_JOGMODE_INCHING` mode, a rising edge at one of the jog inputs moves the axis over a certain distance that is assigned at input `Position`.

**More Information:**

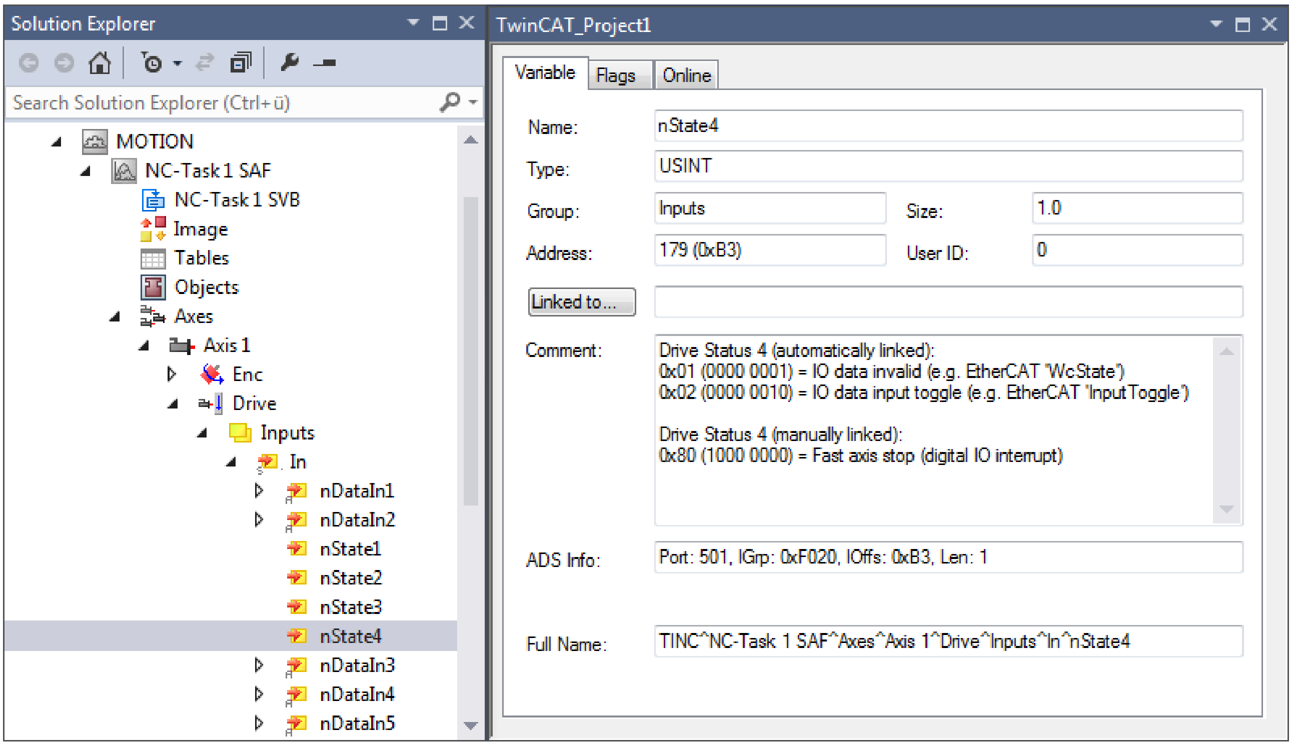
- MC_Jog (PLC library Tc2_MC2)

### 6.2.1.3.3 Fast Axis Stop

**Fast Axis Stop**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Fast Axis Stop: | | | | |
| | Fast Axis Stop Signal Type (optional) | 'OFF (default)' | | E | |
| | Fast Acceleration (optional) | 0.0 | | F | mm/s2 |
| | Fast Deceleration (optional) | 0.0 | | F | mm/s2 |
| | Fast Jerk (optional) | 0.0 | | F | mm/s3 |

Usually a stop is triggered by PLC code using `MC_Stop`. However, there are special applications within that the time delay of stop has to be as small as possible. Within this situation the input Drive.Inputs.In.nState4 comes into play triggering a stop directly without being mapped via the PLC process image.




Drive Status 4 (manually linked):
0x80 (1000 0000) = Fast Axis Stop (digital IO interrupt)

**Variable nState4**

The variable Drive.Inputs.In.nState4 can be mapped to any event source. Note its data type `USINT` and the bit nState4.7 being responsible for the Fast Axis Stop.

**Fast Axis Stop**

For an axis the Fast Axis Stop is performed when its Drive.Inputs.In.nState4.7 variable exhibits the signal type that is selected within the drop-down list "Fast Axis Stop Signal Type (optional)" and differs from “OFF (default)”.

**Signal Type**

The “Fast Axis Stop Signal Type (optional)” enumeration specifies six elements:

- OFF (default)
  For any signal type that the Drive.Inputs.In.nState4.7 variable exhibits no Fast Axis Stop is performed.
- Rising Edge
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a rising edge.

- Falling Edge
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a falling edge.
- Both Edges
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a rising edge and a falling edge, respectively. Alternatively, a Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a falling edge and a rising edge, respectively.
- High Active
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit appears high active.
- Low Active
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit appears low active.

**Optional**

An optional fast axis parameter has to differ from zero to be applied when a Fast Axis Stop is performed.

**Acceleration, Deceleration, Jerk**

When a Fast Axis Stop is performed on the corresponding axis within the boundaries prescribed by the “Maximum Dynamics” the “Fast Acceleration (optional)” float value accelerates the axis, the “Fast Deceleration (optional)” float value decelerates the axis and the “Fast Jerk (optional)” float value is applied. When a Fast Axis Stop is not performed, not any fast axis parameter is applied.

**Further Information:**

- MC_Stop (PLC library Tc2_MC2)

### 6.2.1.3.4 Limit Switches

The Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

**Soft Position Limit Minimum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Minimum Monitoring | FALSE | | B | |
| | Minimum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Minimum Monitoring is not enabled.

`TRUE`: Soft Position Limit Minimum Monitoring is enabled.

**Minimum Position**

Position lower limit for the axis, which must not be violated when Soft Position Limit Minimum Monitoring is enabled. Commands that violate this lower limit will be rejected.

**Soft Position Limit Maximum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Soft Position Limit Maximum Monitoring | FALSE | | B | |
| | Maximum Position | 0.0 | | F | mm |

FALSE: Soft Position Limit Maximum Monitoring is not enabled.

TRUE: Soft Position Limit Maximum Monitoring is enabled.

**Maximum Position**

Position upper limit for the axis, which must not be violated when Soft Position Limit Maximum Monitoring is enabled. Commands that violate this upper limit will be rejected.

## 6.2.1.3.5 Monitoring

**Position Lag Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Position Lag Monitoring | TRUE | | B | |
| | Maximum Position Lag Value | 5.0 | | F | mm |
| | Maximum Position Lag Filter Time | 0.02 | | F | s |

The position lag monitoring monitors the position lag value. If the parameterized limits for position and time are exceeded, a runtime error is output.

Position lag value = actual position - current set position.

TRUE: Position Lag Monitoring is enabled.

FALSE: Position Lag Monitoring is not enabled.

**Maximum Position Lag Value and Maximum Position Lag Filter Time**

The Maximum Position Lag Value is the upper limit, which must not be exceeded for longer than the Maximum Position Lag Filter Time. Otherwise the NC axis is stopped immediately by direct shutdown and set to the logical state "Error", whereby the error 0x4550 is output.

**Position Range Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Position Range Monitoring | TRUE | | B | |
| | Position Range Window | 5.0 | | F | mm |

Position Range Monitoring monitors whether the actual position of the NC axis reaches a window around the target position. Once the window is reached, the status flag Axis.Status.InPositionArea is set to TRUE.

TRUE: Position Range Monitoring is enabled.

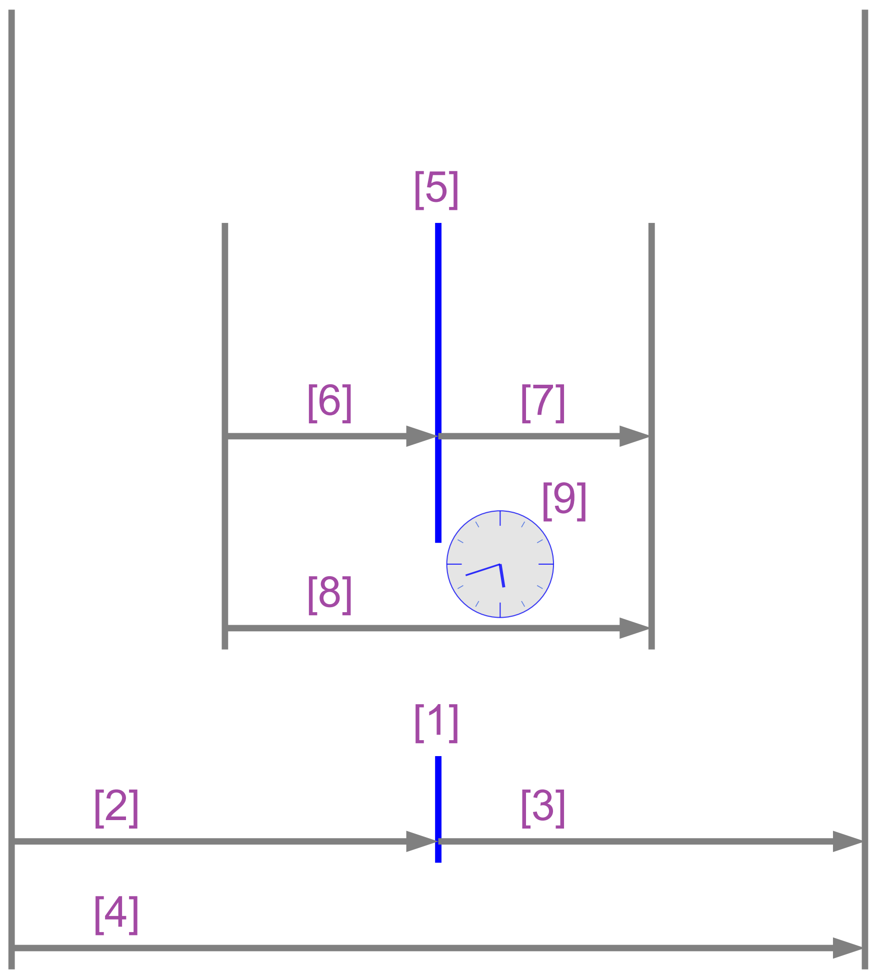
FALSE: Position Range Monitoring is not enabled.

**Position Range Window**

Specifies the tolerance of the actual position of the NC axis with respect to the target position so that the status flag Axis.Status.InPositionArea is set to TRUE.

> **NC-Online: "In Pos. Range" – `Axis.Status.InPositionArea`**
>
> The value of variable `Axis.Status.InPositionArea` corresponds to the state of the checkbox "In Pos. Range" within the group box "Status (phys.)" of the NC-Online dialog. If the variable `Axis.Status.InPositionArea` is set on `TRUE`, the checkbox "In Pos. Range" is checked.

**Graphic example**




[1]
- Nominal value of the target position.

[2]
- Position Range Window.

[3]
- Position Range Window.

[4]
Variable `Axis.Status.InPositionArea`:

*Position Range Monitoring*
- If the parameter "Position Range Monitoring" is set to `TRUE` and ...
- ... if the actual position is in this range [4],
- then the variable `Axis.Status.InPositionArea` is set to `TRUE`.

**Target Position Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Target Position Monitoring | TRUE | | B | |
| | Target Position Window | 2.0 | | F | mm |
| | Target Position Monitoring Time | 0.02 | | F | s |

Target Position Monitoring monitors whether the actual position of the NC axis reaches a window around the target position and also remains in this window for a minimum time. After that the status `Flag Axis.Status.InTargetPosition` is set to `TRUE`.

`TRUE`: Target Position Monitoring is enabled.

`FALSE`: Target Position Monitoring is not enabled.

**Target Position Window**

The Target Position Window specifies the tolerance of the actual position of the NC axis in relation to the target position, which is to be taken into account in Target Position Monitoring.

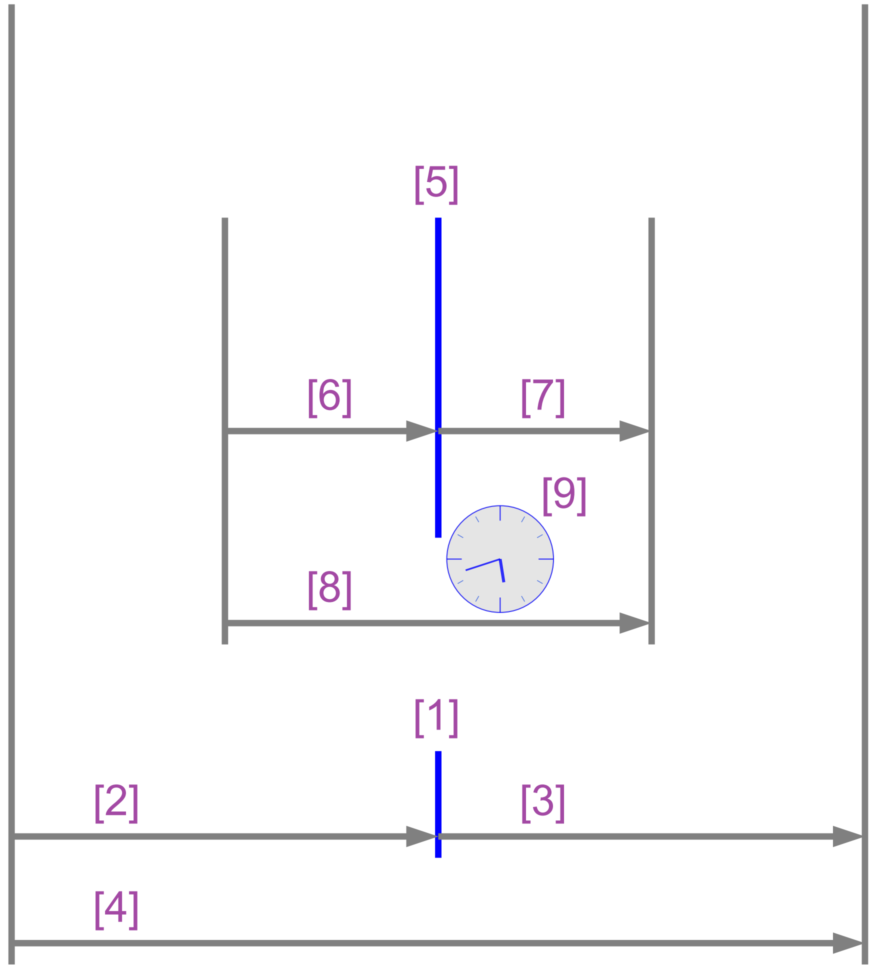
**Target Position Monitoring Time**

The Target Position Monitoring Time specifies the time in which the actual position of the NC axis must be at least within the tolerance range of the target position (Target Position Window) so that the status flag `Axis.Status.InTargetPosition` is set to `TRUE`.

**NC-Online: “In Target Pos.” – `Axis.Status.InTargetPosition`**

The value of the variable `Axis.Status.InTargetPosition` corresponds to the state of the checkbox “In Target Pos.” within the group box “Status (phys.)” of the NC-Online dialog. If the variable `Axis.Status.InTargetPosition` is set on `TRUE`, the checkbox “In Target Pos.” is checked.

**Graphic example**




| | |
|---|---|
| [5] | • Nominal value of the target position. |
| [6] | • Target Position Window. |
| [7] | • Target Position Window. |
| [8], [9] | Target position: |
| *Target Position Monitoring* | • If the parameter "Target Position Monitoring" is set to `TRUE` and ...<br>• ... if the actual position is in this range [8] at least for the duration"Target Position Monitoring Time" [9] without interruption up to the actual time,<br>• then the variable `Axis.Status.InTargetPosition` is set to `TRUE`. |

**In-Target Alarm**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | In-Target Alarm | FALSE | | B | |
| | In-Target Timeout | 5.0 | | F | s |

The In-Target Alarm monitors whether the axis reaches the Target Position Window within the In-Target Timeout.

`TRUE`: The In-Target alarm is enabled.

FALSE: The In-Target alarm is not enabled.

**In-Target Timeout**

If the NC axis does not reach the Target Position Window within the In-Target Timeout, the Nc axis reports the error 0x435C. The time measurement is started when the set position of the axis has reached its nominal position.

### Motion Monitoring

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Motion Monitoring | FALSE | | B | |
| | Motion Monitoring Window | 0.1 | | F | mm |
| | Motion Monitoring Time | 0.5 | | F | s |

Motion Monitoring checks whether an axis is actually moving while it is executing a motion command. This makes it possible, for example, to detect the mechanical blocking of an axis at an early stage.

TRUE: Motion Monitoring is enabled.

FALSE: Motion Monitoring is not enabled.

**Motion Monitoring Window**

The Motion Monitoring Window defines the distance that the encoder (actual position) should be expected to travel during one cycle of the NC SAF task. Here a value/distance/length of some encoder increments can be set.

**Motion Monitoring Time**

Monitoring starts as soon as the axis executes a motion command and ends when the axis comes to a logical standstill. If its actual position does not change by more than the Motion Monitoring Window in at least one NC cycle during the Motion Monitoring Time, the NC axis outputs the error 0x435D.

## 6.2.1.3.6 Setpoint Generator

### Setpoint Generator Type

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Setpoint Generator: | | | | |
| | Setpoint Generator Type | 7 Phases (optimized) | | E | |

7 Phase (optimized)

Only an optimized 7-phase setpoint generator is supported.

### Velocity Override Type

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Setpoint Generator: | | | | |
| | Velocity Override Type | Reduced (iterated) | | E | |

The NC axis supports a speed override. This means that changing the override creates a new velocity, but does not affect the ramps (acceleration or jerk). The used override types only differ in terms of reference velocity.

More information about override types can also be found in Path Override (Interpreter Override Types) [▸ 114].

`Reduced (iterated)`

The override refers to the maximum velocity of the profile calculated by the setpoint generator.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. With a smaller override value, the actual travel velocity is further reduced.

`Original (iterated)`

The override refers to the parameterized velocity of the executed motion command.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. Since the override refers to the velocity of the motion command, the actual velocity is only reduced here below an override value of 70 %.

## 6.2.1.3.7 NCI Parameters

### Rapid Traverse Velocity (G0)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Rapid Traverse Velocity (G0) | 2000.0 | | F | mm/s |

The Rapid Traverse Velocity is used when an interpreter command `G0` is executed. See section Rapid traverse velocity [▸ 117] for a brief description of the interpreter command `G0`.

### Velo Jump Factor

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Velo Jump Factor | 0.0 | | F | |

The reduction factor `C0[i]` is the Velo Jump Factor.

### Background information

#### Segment transitions

Segments are geometric objects. We consider them as curves in the sense of differential geometry, parameterized by means of their length `arc`.

A segment transition from a segment `S_in` to a segment `S_out` is called of geometric type `Ck`, where `k` is a natural number (including `0`) describing `k` continuous `arc` length differentials for each segment and the corresponding `k^th` derivatives at the transition point.

`C0` transitions: Have a knee-point at the transition point.

`C1` transitions: Appear smooth, but are not smooth in dynamic terms. At the segment transition point, there is a jump in acceleration.

`C2` transitions: Are dynamically smooth and their smoothness is only jerk-limited.

`Ck` transitions: are dynamically smooth.

#### Segment dynamics

Velocity `v`: The segment set velocity `v` changes at the segment transition from `v_in` to `v_out`. At the segment transition the set velocity is always reduced to the lower of the two values.

Acceleration `a`: At the segment transition, the current path acceleration is always reduced to zero.

Jerk `j`: At the segment transition, the jerk changes according to the geometry of the segment transition. This jerk change can cause a noticeable dynamic jump.

**Velocity reduction modes for C0 transitions**

Several reduction methods are available for `C0` transitions. One of them is the reduction method VELOJUMP. The reduction method VELOJUMP reduces the velocity after permitted velocity jumps for each axis.

**The VELOJUMP reduction method for C0 transitions**

Basically `v_link = min(v_in, v_out)` applies. For the axis `[i]`, the allowed absolute velocity jump is `v_jump[i] = C0[i] * min(A+[i], -A-[i]) * T`, where `C0[i]` is the reduction factor, `A+[i], A-[i]` is the acceleration or deceleration limits for the axis `[i]`, and `T` is the cycle time. The VELOJUMP reduction method ensures that the path velocity is reduced at the segment transition `v_link` until the absolute step change in the set axis velocity of axis `[i]` is at most `v_jump[i]`. However, `v_min` has priority: if `v_link` is smaller than `v_min`, then `v_link` is set to `v_min`. In the case of motion reversal with no programmed stop, there will be a step change in axis velocity.

**Tolerance ball auxiliary axis**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Tolerance ball auxiliary axis | 0.0 | | F | |

See section Tolerance Ball [▶ 118] for further information.

**Max. position deviation, aux. axis**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Max. position deviation, aux. axis | 0.0 | | F | |

Introduced for future extensions.

## 6.2.1.3.8 Other Settings

**Position Correction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Position Correction | FALSE | | B | |
| | Filter Time Position Correction (P-T1) | 0.0 | | F | s |

The Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

`FALSE`: The Position Correction is disabled.

`TRUE`: The Position Correction is enabled.

The variable `axis.PlcToNc.PositionCorrection` is of data type `LREAL` and belongs to the structure `PLCTONC_AXIS_REF`. If Position Correction is enabled, this variable adds an additional offset to the target position. It should be noted that this correction does not affect the software end positions.

**Filter Time Position Correction (P-T1)**

The filter time for the PT-1 filter, which filters fluctuations within the Actual Position Correction with the filter time set here. See section PT1 Filter [▶ 117] for further information on the PT1 filter.

**See also:**

**`MC_PositionCorrectionLimiter`**

- TwinCAT 3 PLC Lib: Tc2_MC2

The function block `MC_PositionCorrectionLimiter` adds the correction value `PositionCorrectionValue` to the actual position value of the axis. Depending on the `CorrectionMode` the position correction value is either written directly or filtered.

> To use the `MC_PositionCorrectionLimiter` function block successfully the Position Correction has to be enabled by setting the parameter Position Correction `TRUE`.

### Backlash

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Backlash | 0.0 | | F | mm |

This parameter is only present for compatibility reasons. For more information, visit NC Backlash Compensation.

### Error Propagation Mode

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Error Propagation Mode | 'INSTANTANEOUS' | | E | |
| | Error Propagation Delay | 0.0 | | F | s |

For the slave axis the error transmission can be delayed.

`'INSTANTANEOUS'`: The error transmission is not delayed.

`'DELAYED'`: The error transmission is delayed by the Error Propagation Delay.

**Error Propagation Delay**

The delay time by which error propagation for the slave axis is delayed when Error Propagation Mode `'DELAYED'` is selected.

If an error occurs at a slave axis during runtime, the corresponding master axis is not set to error state until the time assigned here has elapsed. A state of interest of the slave axis, especially its error state, can be observed by PLC code. In this way, the faulty slave axis can be safely decoupled to safely prevent the entire axis combination from entering the error state.

### Couple slave to actual values if not enabled

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Couple slave to actual values if not enabled | FALSE | | B | |
| | Velocity Window | 1.0 | | F | mm/s |
| | Filter Time for Velocity Window | 0.01 | | F | s |

`FALSE`: Not coupled.

`TRUE`: Coupled. The slave axis follows the master actual position while and also when the master is disabled.

### Velocity Window und Filter Time for Velocity Window

The coupled slave axis follows the master axis within the `Velocity Window`. If velocity deviations exceed `Filter Time for Velocity Window` beyond `Velocity Window`, an error is output.

### Allow motion commands

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Allow motion commands to slave axis | TRUE | | B | |
| | Allow motion commands to external setpoint axis | FALSE | | B | |

### Allow motion commands to slave axis

In general terms, an axis is in PTP mode all the time. This is about indirectly converting a slave axis into a master axis. Thus it is implicitly decoupled without the need to call on `MC_GearOut` from the PLC code.

`TRUE`: A PTP command can be triggered to the slave axis without first setting the axis to PTP mode.

`FALSE`: Before a PTP command can be triggered to the slave axis, the slave axis must be set to PTP mode.

### Allow motion commands to external setpoint axis

`FALSE`: Before a PTP command can be triggered to the external setpoint axis, the external setpoint axis must be set to PTP mode.

`TRUE`: A PTP command can be triggered to the external setpoint axis without first setting the axis to PTP mode.

### Dead Time Compensation (Delay Velo and Position)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Dead Time Compensation (Delay Velo and Position) | 0.0 | | F | s |

This parameter is only present for compatibility reasons. Do not use it on new projects.

### Data Persistence

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Data Persistence | FALSE | | B | |

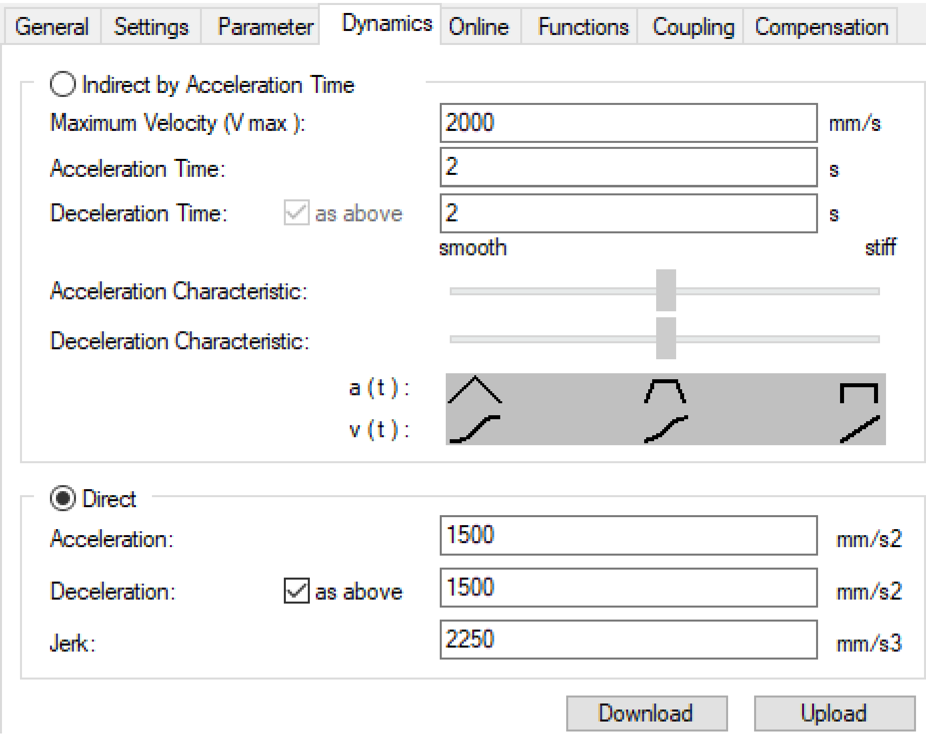
Data Persistence is used for special encoder problems.

FALSE: Data Persistence is not enabled.

TRUE: Data Persistence is enabled.

### 6.2.1.4 Dynamics

The axis dynamics can be configured via the **Parameter** tab as well as via the **Dynamics** tab.




In this dialog the default axis dynamics is set. The default values are used if no explicit dynamics data are given to a travel command.

The dynamics data Acceleration, Deceleration and Jerk can be entered here either directly or they are determined indirectly via an acceleration time to the maximum velocity. In addition to the acceleration time, the slider is used to determine the ratio of jerk and acceleration, thus selecting a rather stiff or smooth setting.

### 6.2.1.5 Online

The **Online** tab is the main dialog for online axis operation. Here the enable can be set and after this the corresponding axis can be moved manually. In addition, the most important axis states are displayed when the configuration is active.

**⚠ DANGER**

**Risk of injury due to movement of axes!**

The commissioning results in a movement of axes.

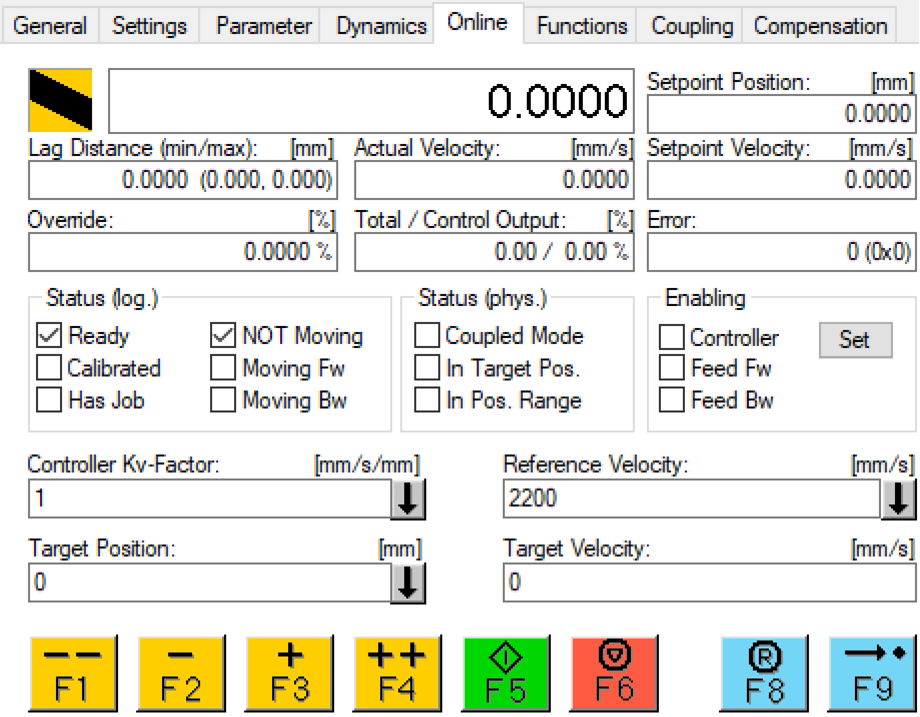
- Make sure that neither you nor others are harmed by the movement, e.g. by maintaining a suitable safety distance.
- Do not perform any action whose consequences you cannot estimate

**⚠ WARNING**

**Incorrect axis position during initial commissioning**

Without referencing / calibrating the axis position, the displayed axis position may deviate from the actual axis position.

- Perform a homing to determine the correct actual position using a reference signal.




The dialog is divided into the display of the most important axis states, the setting of the axis enable, input fields as well as function keys for controller settings and travel commands. The units of the values depend on the set "Base" unit.

### Display

The actual position of the axis is determined from the feedback of the encoder system and displayed in the large unlabeled field.

| | |
|---|---|
| Setpoint Position | The target position is specified by the NC during a movement. |
| Lag Distance | The lag error is the difference between the nominal and actual position. |
| Actual Velocity | The actual velocity of the axis is determined by deriving the actual position. |
| Setpoint Velocity | The target velocity is calculated by the NC during a movement. |
| Override | The override set by the user (0..100%) is displayed here. |
| Total/Control Output | Total output of the NC axis to the drive in % and position control portion of the output in %. |
| Error | Axle error code. An axis error can be deleted with the reset key. |

### Set enables

Via the button **Set** the controller enable, the feed enables and the override of the axis can be set.

### Input fields for position controller setting

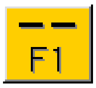
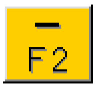
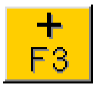
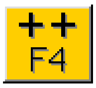
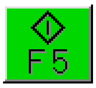
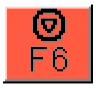
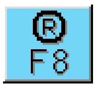
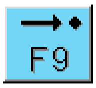
If the NC axis is operated in velocity mode (CSV), the position is controlled by the NC. At this point the two most important parameters, the gain factor Kv and the reference velocity, can be set. Further setting options can be found in the parameters of the axis.

### Input fields for travel commands

Target Position: Target position for a subsequent travel command (F5)

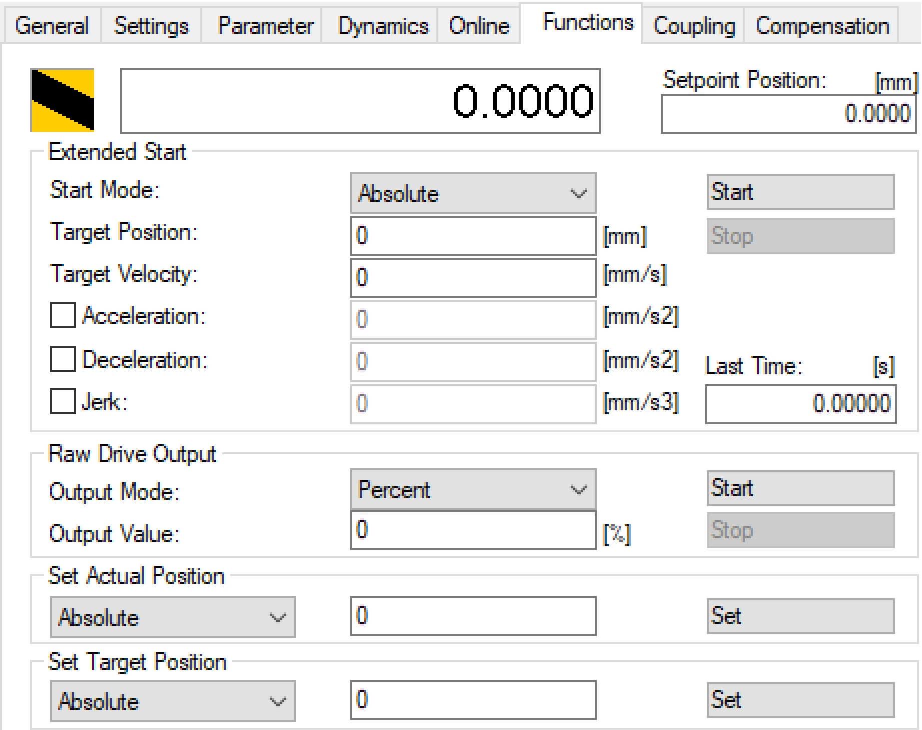
Target Velocity: Velocity of a subsequent travel command (F5)

**Function button**

| Button | Key | Description |
|---|---|---|
|  | F1 | Reverse travel with Manual Velocity (Fast) |
|  | F2 | Reverse travel with "Manual Velocity (Slow)" |
|  | F3 | Forward travel with "Manual Velocity (Slow)" |
|  | F4 | Forward travel with "Manual Velocity (Fast)" |
|  | F5 | Start, with the values set in the input fields and the set dynamics. |
|  | F6 | Stop |
| | | |
|  | F8 | Reset |
|  | F9 | Calibrate with the values set in the "Global" menu.<br>***Notice*** **The signal source of the referencing cam can be set in the encoder parameters (Homing Sensor Source). In the default setting, the referencing cam signal must be mapped into the axis data structure (Axis.PlcToNc.ControlDword.5) by the PLC so that the sequence triggered with F9 can react to the cam.** |

## 6.2.1.6 Functions

In the **Functions** tab the most important commands for commissioning can be given to the axis.




Different travel commands like absolute start, relative start or endless travel can be selected in the *Extended Start* frame . The necessary parameters below will adjust according to the selection of the start type. Thus, in reverse mode, there are two target positions between which the axis moves back and forth.

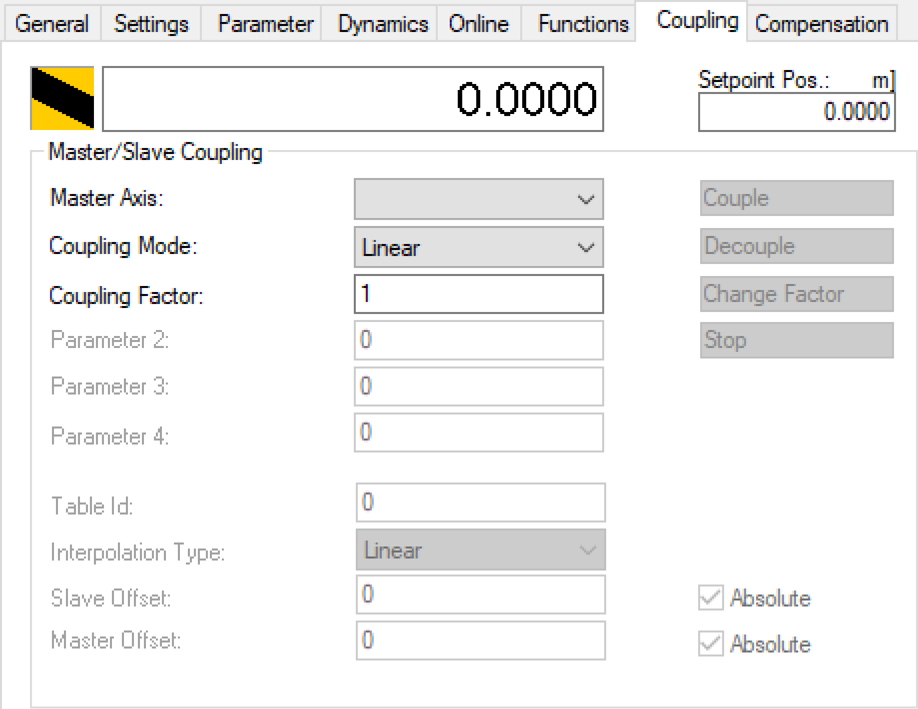
The drive output can be set to a fixed value for commissioning in the *Raw Drive Output* frame. This setting must be used very carefully.

The axis position can be set to a new value in the *Set Actual Position* frame.

*Set Target Position* changes the target position of the axis during a travel command.

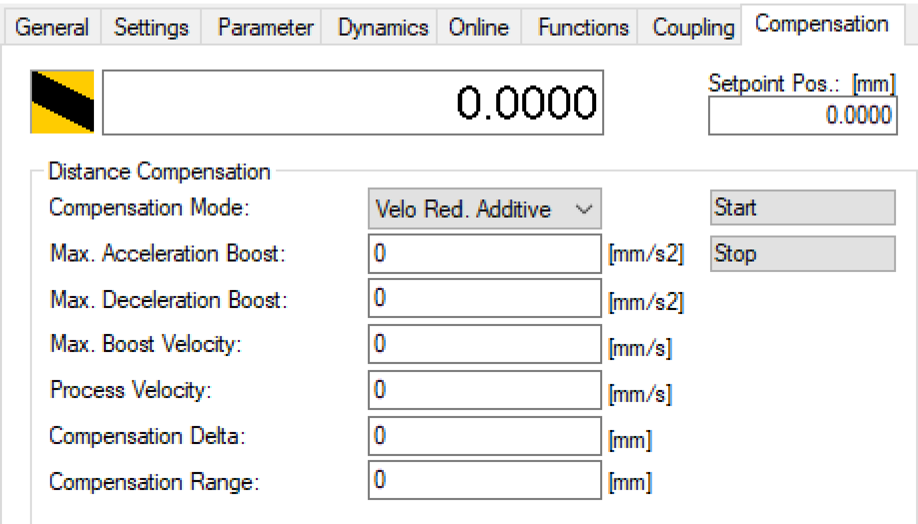
### 6.2.1.7 Coupling

Two axes can be coupled with each other in the **Coupling** tab. To do this, first select the master axis and the desired coupling method, e.g. linear coupling. Afterwards, necessary parameters, such as the gear ratio, can be entered. The *Couple* button executes the coupling, *Decouple* decouples the axes again.



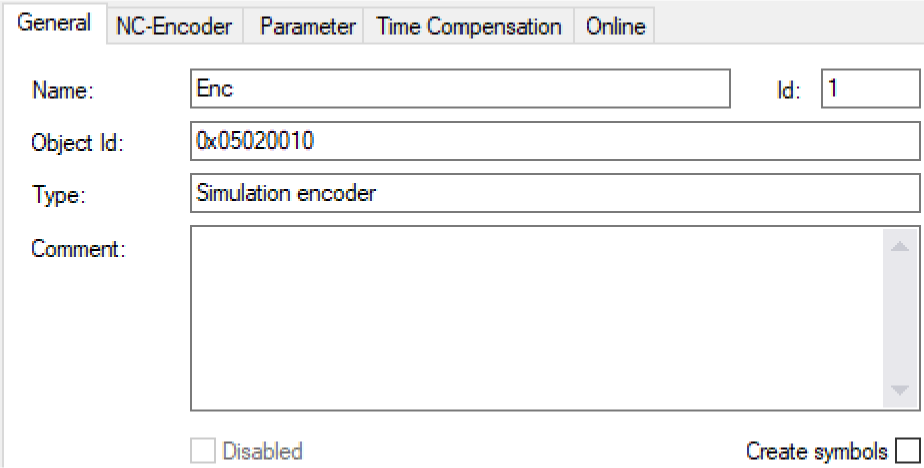

### 6.2.1.8 Compensation

A superimposed travel command can be triggered during the travel of an axis in the **Compensation** tab (position compensation or superposition).



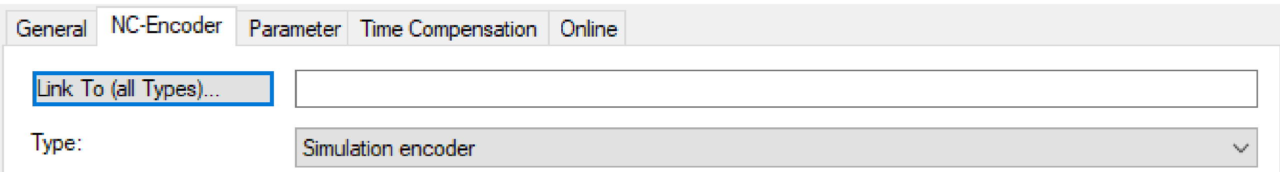
## 6.2.2 Axis component encoder

### 6.2.2.1 General




| Property | Description |
|---|---|
| **Name** | Selected name of the object. |
| **Object Id** | 32-bit identification number that is automatically assigned and unique within a project. |
| **Type** | Encoder type<br>The selected encoder type is displayed at this point. Changes of the encoder type can be done via the NC-Encoder [▶ 43] dialog. |
| **Comment** | Freely editable field for user notes. |
| **Id** | Identification number of the encoder, which is assigned consecutively. When an encoder is deleted, the number becomes free again and can be assigned to a new encoder. |
| **Disabled** | An encoder cannot be disabled on its own, but only by disabling the entire axis. |
| **Create symbols** | This option generates typical symbol names that are used in TwinCAT Scope, for example. |

### 6.2.2.2 NC-Encoder




| Property | Description |
|---|---|
| Link To (all Types) | Every encoder that is not of the "Simulation encoder" type must be associated with an actual value acquisition module. In the simplest case, this is done automatically with the linking of the axis to the drive. In individual cases, in particular when unknown hardware components are integrated in the system, these links must be established manually , see Process image [▶ 53]. |
| Type | Encoder type selection |

### 6.2.2.3 Parameter

**Encoder**

Encoders generally record a physical quantity and provide parameters for e.g. scaling. In most cases, a position is read in and processed here, and velocity, acceleration and jerk are determined by derivation from the position.

Since actual values often fluctuate strongly, a filter is also available for each individual variable.

A wide variety of encoders versions is supported. Both absolute and incremental encoders are available. Furthermore, there are simulation encoders, as well as special encoders for force determination, for example.

If you use an encoder axis, you can also use it to read in a different physical quantity and use the existing parameters for scaling, for example.

The encoder parameters include the scaling, zero offset shift and the modulo factor. There are also encoder parameters for software end positions as well as for homing.

#### 6.2.2.3.1 Encoder Evaluation

**Invert Encoder Counting Direction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Invert Encoder Counting Direction | FALSE | | B | |

If set to `TRUE`, the Invert Encoder Counting Direction parameter reverses the encoder counting direction.

- `FALSE`: The polarity of the axis movement agrees with the counting direction of the acquisition hardware.
- `TRUE`: The polarity of the axis movement is in the opposite sense to that of the counting direction of the acquisition hardware.

**⚠ WARNING**

**Risk of unexpected movements**

If the counting direction of the encoder and the motor polarity do not match, the axis will perform unexpected movements.

**Scaling Factor Numerator and Scaling Factor Denominator (default: 1.0)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Scaling Factor Numerator | 0.0001 | | F | mm/INC |
| | Scaling Factor Denominator (default: 1.0) | 1.0 | | F | |

The scaling factor is divided into a numerator and a denominator. It is used to convert the path increments into axis positions or to calculate a user unit from encoder increments.

**Position Bias**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Position Bias | 0.0 | | F | mm |

The zero offset shift is an offset that absolute encoders use to align their position in the machine coordinate system and is thus used to define the machine-related zero point. The offset value of the zero offset shift is added to the encoder position to determine the axis position.

**Modulo Factor (e.g. 360.0°)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Modulo Factor (e.g. 360.0°) | 360.0 | | F | mm |
| | Tolerance Window for Modulo Start | 0.0 | | F | mm |

The value of the Modulo Factor (e.g. 360.0°) is the value for calculating modulo turns and modulo positions from the absolute axis position. For rotary axes, the modulo factor is the "path distance" that forms one revolution. If the actual value is acquired, for instance, in degrees, 360.0° should be entered here.

**Tolerance Window for Modulo Start**

The Tolerance Window for Modulo Start should be large enough to allow backward movement when a modulo forward command is executed in the window area. Conversely, the Tolerance Window for Modulo Start should be large enough to allow forward movement when a modulo backward command is executed in the window area.

**Further Information:**

- Notes on modulo positioning (PLC library Tc2_MC2)

**Encoder Mask (maximum encoder value)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Encoder Mask (maximum encoder value) | 0x0036EE7F | | D | |

The encoder mask defines the number of allowed increments until the feedback value overflows. In most cases, the increments are not the real encoder increments due to internal encoder overflow calculation on the connected hardware.

**Encoder Sub Mask (absolute range maximum value)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Encoder Sub Mask (absolute range maximum value) | 0x000FFFFF | | D | |

The encoder sub mask defines the number of increments per motor revolution.

**Noise level of simulation encoder**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Noise level of simulation encoder | 0.0 | | F | |

This parameter creates artificial noise for the simulation axis by setting a level to make it appear more realistic.

**Reference System**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Reference System | 'INCREMENTAL' | | E | |

Use the Reference System parameter to select how the encoder values should be interpreted:

`INCREMENTAL`

INCREMENTAL mode does not use absolute characteristics of the physical feedback system, but can be used with both incremental and absolute encoder systems. After starting the control, homing is required to initialize the actual position of the axis.

`INCREMENTAL (singleturn absolute)`

This mode extends the INCREMENTAL mode and can be used instead.

If in this mode the retain data of the NC-SAF task and the axis parameter *Data Persistence* are activated, the system starts up with the last stored position. If a single-turn absolute encoder is used at the same time, a deviation of up to half an encoder revolution is also taken into account and the axis can be operated within these limits without homing. The absolute range of the encoder system is defined by the parameter *Encoder Sub Mask*.

`ABSOLUTE`

The ABSOLUTE mode can be used with multi-turn absolute encoder systems. The prerequisite is that the encoder system or the motor is installed in such a way that there can't be overflow of the position feedback within the travel path. An overflow would be acknowledged with a runtime error.

`ABSOLUTE MULTITURN RANGE (with single overflow)`

The ABSOLUTE MULTITURN RANGE mode can also be used with multi-turn absolute encoder systems. The encoder system can be installed in any way, so that a maximum of one overflow may occur in the travel path. To ensure correct evaluation, the two end positions must be parameterized.

The total travel path must be less than the absolute range of the encoder. The absolute range of the encoder system is defined by the parameter *Encoder Mask*.

`ABSOLUTE SINGLETURN RANGE (with single overflow)`

The ABSOLUTE SINGLETURN RANGE mode can be used with single-turn absolute encoder systems. The encoder system can be installed in such a way that an overflow is located in the travel path. To ensure correct evaluation, the two end positions must be parameterized.

The total travel path must be less than the absolute range of the encoder, i.e. less than one encoder revolution. The absolute range of the encoder system is defined by the parameter *Encoder Sub Mask*.

`ABSOLUTE (modulo)`

The ABSOLUTE (modulo) mode can be used when the travel path of an axis exceeds the absolute range of the encoder system. For example, in the case of an endlessly running conveyor belt.

To initialize the position of the axis correctly at system start, the retain data of the NC-SAF task and the parameter *Data Persistence* of the axis must be activated. Furthermore, the parameter *Modulo Factor* of the encoder of the NC axis must be parameterized correctly. The axis position is initialized to this module range after the system startup.

When switched off, the axis may be mechanically displaced by a maximum of half of the absolute range. This maximum distance is defined by the *Encoder Sub Mask* parameter. The *Encoder Sub Mask* is typically set to the increments of one motor revolution, but can also be set to the entire absolute range here.

### 6.2.2.3.2 Limit Switches

The Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

### Soft Position Limit Minimum Monitoring

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Minimum Monitoring | FALSE | | B | |
| | Minimum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Minimum Monitoring is not enabled.

`TRUE`: Soft Position Limit Minimum Monitoring is enabled.

**Minimum Position**

Position lower limit for the axis, which must not be violated when Soft Position Limit Minimum Monitoring is enabled. Commands that violate this lower limit will be rejected.

### Soft Position Limit Maximum Monitoring

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Maximum Monitoring | FALSE | | B | |
| | Maximum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Maximum Monitoring is not enabled.

`TRUE`: Soft Position Limit Maximum Monitoring is enabled.

**Maximum Position**

Position upper limit for the axis, which must not be violated when Soft Position Limit Maximum Monitoring is enabled. Commands that violate this upper limit will be rejected.

## 6.2.2.3.3 Filter

### Filter Time for Actual Position (P-T1)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Position (P-T1) | 0.0 | | F | s |

Filter time for the PT1 filtering of the actual position.

In the case of very noisy encoder signals, it may therefore be necessary to filter the position value with a low filter time.

Filtering the position leads to a position distortion.

**Filter Time for Actual Velocity (P-T1)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Velocity (P-T1) | 0.01 | | F | s |

Filter time for the PT1 filtering of the actual velocity.

**Filter Time for Actual Acceleration (P-T1)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Acceleration (P-T1) | 0.1 | | F | s |

Filter time for the PT1 filtering of the actual acceleration.

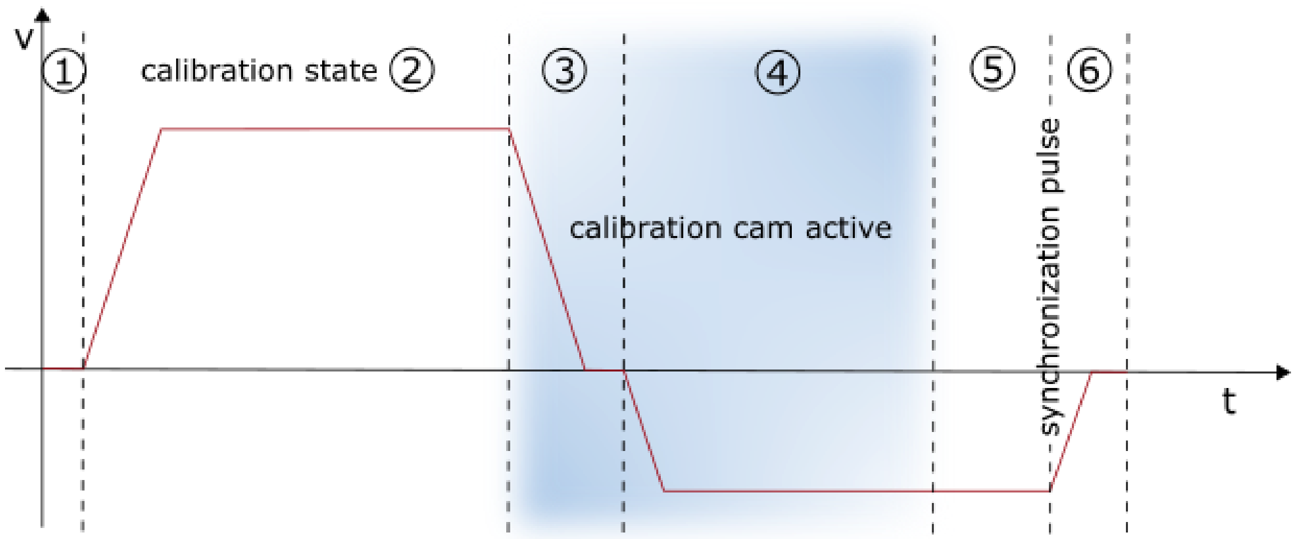
**Further information: PT1 filter**

See section PT1 Filter [▶ 117] for further information on the PT1 filter.

### 6.2.2.3.4 Homing

Homing refers to an axis initialization run during which the correct actual position is determined by means of a reference signal. This can be NC-guided, as described here:

1. The axis is at a random position.
2. The axis moves in the parameterized direction and searches for the reference switch (referencing cam/calibration cam).
3. The axis reverses as soon as the referencing cam is detected.
4. The axis moves away from the referencing cam and detects the falling edge of the cam signal.
5. The axis continues to move and searches for the sync pulse (latch signal or zero pulse of the encoder).
6. The reference position is set and the axis stops. The standstill position of the axis deviates slightly from the reference position.



Alternatively, there is also drive-guided homing, which is performed independently by a suitable drive without the influence of the control system. For details, please refer to the documentation of the drive system used.

### Invert Direction for Homing Sensor Search

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Invert Direction for Homing Sensor Search | FALSE | | B | |

For the default homing sequence the direction for calibration cam search can be inverted.

- `FALSE`: cam is looked for in the direction of positive movement.
- `TRUE`: cam is looked for in the direction of negative movement.

### Invert Direction for Sync Impuls Search

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Invert Direction for Sync Impuls Search | TRUE | | B | |

For the default homing sequence the direction for sync pulse search can be inverted.

- `FALSE`: sync pulse is looked for in the direction of positive movement.
- `TRUE`: sync pulse is looked for in the direction of negative movement.

### Home Position (Calibration Value)

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Home Position (Calibration Value) | 0.0 | | F | mm |

The Home Position is the position value that will be assigned to the axis when the sync pulse appears within the homing. This is done exactly with the sync pulse while in motion. Since the axis then stops, the standstill position deviates from the parameterized value. The referencing is nevertheless exact.

### Reference Mode (Sync condition)

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Reference Mode (Sync condition) | 'Default' | | E | |

The parameter Reference Mode offers the opportunity to select which signal shall be used for sync impulse search:

- `Default / Homing Sensor Only (PLC cam or digital input 1..8)`
  - The simplest way of referencing an axis where only the cam signal is evaluated. Another sync signal is not used, so this setting should only be selected if the axis does not offer more extensive options.
- `Hardware Sync (Feedback reference pulse)`
  - Some encoder systems provide a sync pulse per revolution (zero track) in addition to the count. The homing precision can be enhanced by selecting this mode, if the encoder evaluation logic is able to pick up the sync pulse. The precision is comparable with Software Sync. Hardware Sync mode may require parameterization or special wiring of the drive or encoder system.
- `Hardware Latch 1 (pos. edge), Hardware Latch 1 (neg. edge)`

- Hardware latch responds to an external latch signal in addition to the cam signal. For this purpose, the latch mechanism must be configured in the drive and the latch input must be wired.

- `Software Sync`
  - Software Sync mode enhances the homing precision by additionally detecting the encoder count overflow after an encoder or motor revolution, after the reference cam signal has been detected. This mode requires a part-absolute encoder (e.g. resolver) with constant overflow interval relative to the referencing cam. Overflow detection is parameterized via the Reference Mask parameter.
- `Application (PLC code)`
  - In this case, the entire homing procedure must be programmed in the PLC:
    In this case, the NC sets bit 23 in the "NcApplicationRequest" status, which is polled by the PLC. Via MC_ReadApplicationRequest the request of the NC can be determined in the PLC. So far, only 0 = NONE and 1 = HOMING are supported (see 0x33 at "Index offset" specification for axis parameters). In the case of homing, the PLC can perform the sequence itself by moving the axis.

**Homing Sensor Source**

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Homing Sensor Source | 'Default: PLC Cam (MC_Home)' | | E | |

- `Default PLC Cam (MC_Home)`
  - By default the homing sensor signal comes from the PLC with the function block MC_Home with the parameter bCalibrationCam, which is automatically linked to the PlcToNc.ControlWordD.7.
- `Digital Input [1-8]`
  - Alternatively, a digital input linked to the NC process can be used. For this purpose, a general Drive Status Byte with 8 digital inputs is defined in the process image (Drive->Inputs->nState8), which can serve as a signal source for the homing sensor. A digital input to be used must therefore be mapped manually to the desired position in this byte.
  - ***Notice*** **The digital inputs 1 and 2 may differ depending on the hardware used. For the MDP703/733 hardware (e.g. EL7031, EL7041, EL7201, EL7411) the direct digital inputs E1 and E2 of the terminal are used instead, which are located in the Drive.nState2 byte of the terminal at bit position 3 (E1) and 4 (E2). The lower two bits of Drive.nState8 are not assigned in this case.**

**Further information**

- Homing on the AX5000
- MC_Home in the Tc2_MC2

## 6.2.2.3.5 Other Settings

**Encoder Mode**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Encoder Mode | 'POSVELO' | | E | |

The parameter Encoder Mode offers the possibility to select which values should be calculated from the encoder position:

- `'POS'`: The actual position is determined.
- `'POSVELO'`: The actual position and the actual velocity are determined.
- `'POSVELOACC'`: The actual position, the actual velocity and the actual acceleration are determined.

**Position Correction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Position Correction | FALSE | | B | |
| | Filter Time Position Correction (P-T1) | 0.0 | | F | s |

The Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

`FALSE`: The Position Correction is disabled.

`TRUE`: The Position Correction is enabled.

The variable `axis.PlcToNc.PositionCorrection` is of data type `LREAL` and belongs to the structure `PLCTONC_AXIS_REF`. If Position Correction is enabled, this variable adds an additional offset to the target position. It should be noted that this correction does not affect the software end positions.

**Filter Time Position Correction (P-T1)**

The filter time for the PT-1 filter, which filters fluctuations within the Actual Position Correction with the filter time set here. See section PT1 Filter [▶ 117] for further information on the PT1 filter.

**See also:**

**`MC_PositionCorrectionLimiter`**

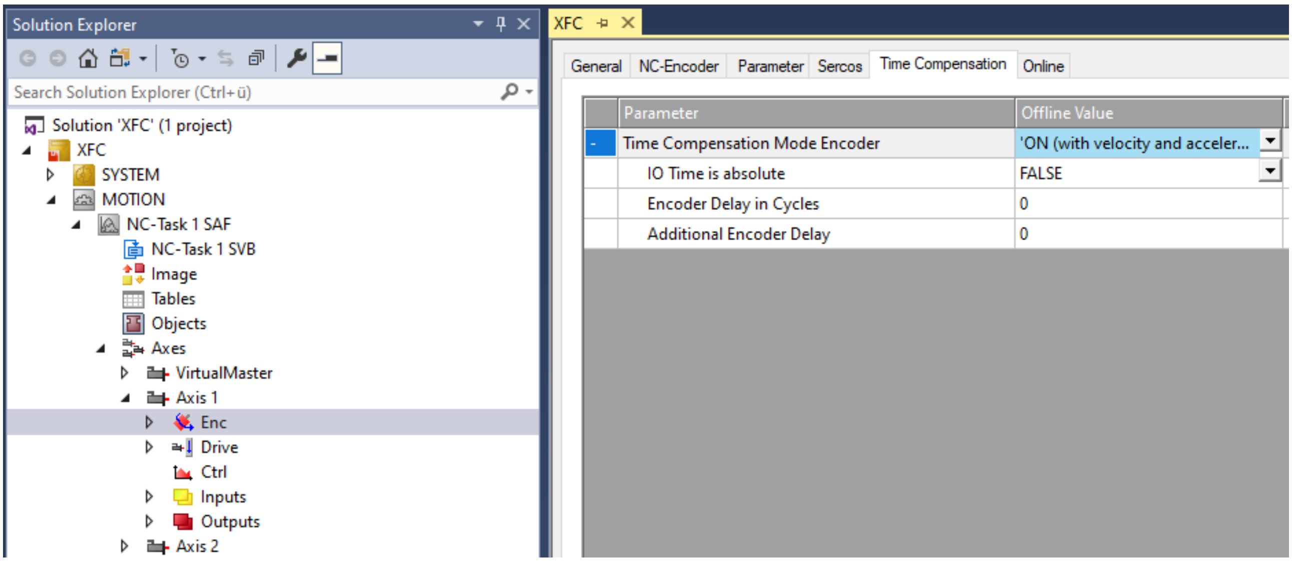
- TwinCAT 3 PLC Lib: Tc2_MC2

The function block `MC_PositionCorrectionLimiter` adds the correction value `PositionCorrectionValue` to the actual position value of the axis. Depending on the `CorrectionMode` the position correction value is either written directly or filtered.

> **i** To use the `MC_PositionCorrectionLimiter` function block successfully the Position Correction has to be enabled by setting the parameter Position Correction `TRUE`.

### 6.2.2.4 Time Compensation

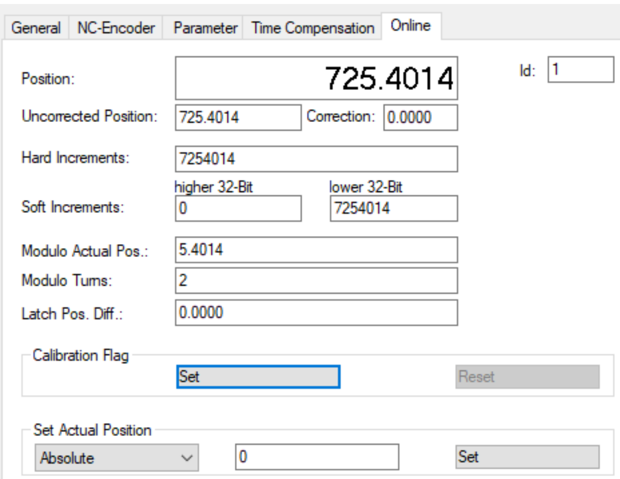
A requirement for high-precision conversion of positions into times and vice versa is precise dead time compensation of the axes. For data acquisition, this can be enabled and configured via the **Time Compensation** tab of the axis encoder (e.g. **Axis 1 > Enc**).




| Parameter | Description | Type | Unit |
|---|---|---|---|
| Time Compensation Mode Encoder | Enabling/disabling the dead time compensation of the axis encoder incl. selection of the mode:<br>• `OFF`<br>• `ON (with velocity)` (recommended)<br>• `ON (with velocity and acceleration)` | enum | |
| IO Time is absolute | Default `FALSE`<br>Must be set to `TRUE` if the encoder supplies an absolute timestamp (special case). | bool | |
| Encoder Delay in Cycles | Additional delays by entire I/O cycles. This time is not a fixed value, but changes with the cycle time. | double | |
| Additional Encoder Delay | Fixed time value in µs caused by the hardware used. | double | µs |

Further details on dead time compensation can be found in the documentation TF5065 TwinCAT 3 | PLC Motion Control XFC/XFC NC I.

### 6.2.2.5 Online




| Property | Description |
|---|---|
| Position | Actual position of the encoder |
| Id | Automatically assigned encoder ID |
| Uncorrected Position | Actual position without corrections, such as position correction or position filtering |

| Property | Description |
|---|---|
| Correction | Position correction |
| Hard Increments | Incremental counter value from the encoder hardware |
| Soft Increments | Evaluated incremental counter value of the NC |
| Modulo Actual Pos | Modulo actual position (calculated with the parameterized modulo factor) |
| Modulo Turns | Integer modulo turns (calculated from the actual position) |
| Latch Pos. Diff. | Position difference between the referencing cam and the latch event, which is determined during homing. |
| Calibration Flag | If an absolute encoder system is configured, it is calibrated from the start. An incremental encoder system is only calibrated after homing.<br>The calibration state can be changed manually via the Set and Reset buttons. |
| Set Actual Position | The actual position can be set manually here. The position is set directly as entered in the "Absolute" mode. The position is shifted by this amount in the "Relative" mode. |

## 6.2.2.6 Process image

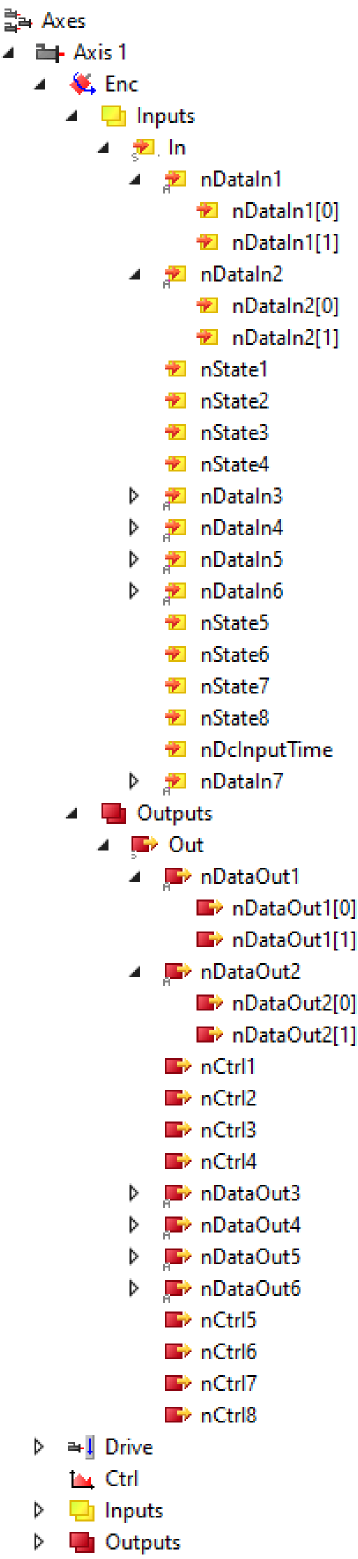
The process image (cyclic interface) of an axis is used to connect it to different drive components. In the simplest case, a link is established between the axis and the drive and the necessary links between the process images are made automatically. In individual cases, in particular when unknown hardware components are integrated in the system, these links must be established manually.

***Notice*** **The data structures described here form an internal interface between NC drivers and connected drive hardware. This interface is constantly under development and can change in the future.**

### Encoder process image of an axis

Various encoder hardware or the corresponding Bus Terminals for detecting the position of an axis are connected via the encoder process image (cyclic data exchange). Insofar as this hardware is directly supported by the system, no manual configuration of individual variables is necessary.

Depending on the encoder/drive, the NC variables can be interpreted as `signed` or `unsigned`. In addition, the `nDataIn` variables can be linked with 2 or 4 byte references. (In the special case not discussed here also with 8 byte references)




### Input data to the encoder process image of an axis

| NC-In variable | Data type | Description |
| --- | --- | --- |
| `nDataIn1`<br>or<br>`nDataIn1[0]` | `UDINT` / `DINT` (4 byte)<br>or<br>`UINT` or `INT` (2 byte) | Current actual position of the encoder or the drive in *increments*. The number of allowed increments is defined in the encoder parameter Encoder Mask (maximum encoder value) [▶ 45]. |

| NC-In variable | Data type | Description |
|---|---|---|
| | | The incremental value is processed by the NC and converted to the actual position in physical units, e.g. mm or degrees. Overflows of the incremental values are also counted by the NC. The system normally makes no distinction between incremental and absolute encoders, see encoder parameter Reference System [▶ 45].<br>Mapping with:<br>• SoE: S-0-0051 Position feedback 1 value<br>• CoE: 0x6064 Position actual value |
| nDataIn2<br>or<br>nDataIn2[0] | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Optional actual latch position of the encoder or the drive in *increments*.<br>The incremental value is processed by the NC and converted to the actual position in physical units, e.g. mm or degrees.<br>(See function block MC_TouchProbe.) |
| nState1 | USINT | Optional status information<br>For example *Participant in Data Exchange*, *Encoder Error*, communication to *Position Latch* or *Register Communication*. |
| nState2 | USINT | Additional optional status information<br>For example, communication to *Position Latch*, communication *Encoder Reset*, communication to *Read an Absolute Position*. |
| nState3 | USINT | reserved |
| nState4 | USINT | Optional fieldbus-dependent I/O status such as *WcState* (Working Counter) in the case of EtherCAT or *CdlState* in the case of Beckhoff Lightbus.<br>Meaning:<br>0 = I/O data valid<br>1 = I/O data invalid |
| nDataIn3 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0130 Touch probe position 1 (positive edge)<br>• CoE: 0x60BA Touch probe position 1 (positive edge) |
| nDataIn4 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE (S-0-0131): Touch probe position 1 (negative edge)<br>• CoE (0x60BB): Touch probe position 1 (negative edge) |
| nDataIn5 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0132 Touch probe position 2 (positive edge)<br>• CoE: 0x60BC Touch probe position 2 (positive edge) |
| nDataIn6 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0133 Touch probe position 2 (negative edge)<br>• CoE: 0x60BD Touch probe position 2 (negative edge) |
| nState5 | USINT | Mapping with:<br>• SoE: S-0-0409 Probe 1 positive edge latched<br>• CoE: 0x60B9 (low byte) Status touch probe 1 + 2 |
| nState6 | USINT | Mapping with: |

| NC-In variable | Data type | Description |
|---|---|---|
| | | • SoE: S-0-0410 Probe 1 negative edge latched<br>• CoE (0x60B9, high byte): Status touch probe 1 + 2 |
| nState7 | USINT | Mapping with:<br>• SoE: S-0-0411 Probe 2 positive edge latched |
| nState8 | USINT | Mapping with:<br>• SoE: S-0-0412 Probe 2 negative edge latched |
| nDcInputTime | DINT | Optional: Used for NC dead time compensation of the encoder. |
| nDataIn7 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Actual drive velocity from the drive device or from the encoder/feedback.<br>Mapping with:<br>• SoE: S-0-0040 Velocity feedback 1 value<br>• CoE:<br>◦ 0x606C Actual velocity value (4 Byte)<br>◦ 0x6044 Actual velocity for velocity mode (2 bytes), alternative if 0x606C is not available. |

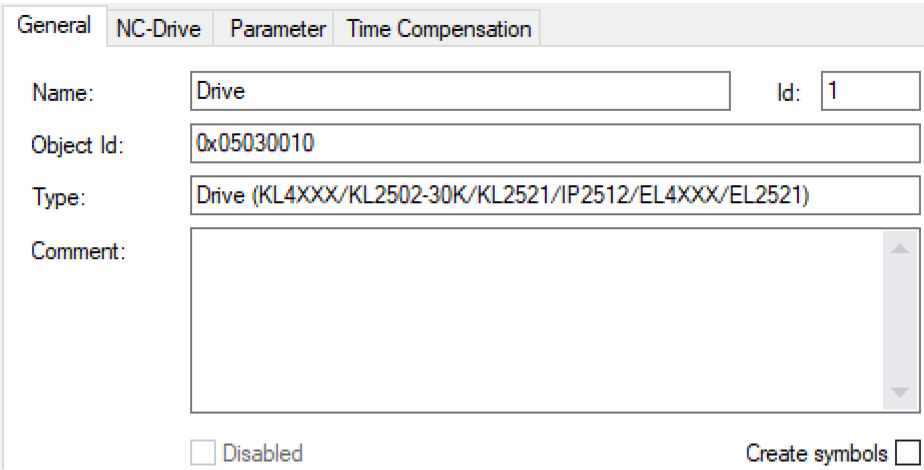
**Output data from the encoder process image of an axis**

| I/O variable | Data type | Description |
|---|---|---|
| nOutData1 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Current actual position of the encoder or drive in *increments*, which the NC copies directly from the input variable nInData1 to the output variable nOutData1. |
| nOutData2 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | reserved |
| nCtrl1 | USINT | Optional control information<br>For example, communication to *Position Latch*, communication *Encoder Reset*, *Register Communication* |
| nCtrl2 | USINT | Additional optional control information<br>For example, communication to *Position Latch*, communication *Encoder Reset*, communication to *Read an Absolute Position* |
| nCtrl3 | USINT | reserved |
| nCtrl4 | USINT | reserved |
| nOutData3 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | reserved |
| nOutData4 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | reserved |
| nOutData5 | UDINT / DINT (4 byte) | reserved |

| I/O variable | Data type | Description |
|---|---|---|
| | or<br>UINT / INT (2 byte) | |
| nOutData6 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | reserved |
| nCtrl5 | USINT | Mapping with:<br>• SoE: S-0-0405 (low byte) Probe 1 enable<br>• CoE: 0x60B8 (low byte) Control Touch probe 1+2 |
| nCtrl6 | USINT | Mapping with:<br>• SoE: S-0-0406 (low byte) Probe 2 enable<br>• CoE: 0x60B8 (high byte) Control Touch probe 1+2 |
| nCtrl7 | USINT | reserved |
| nCtrl8 | USINT | reserved |

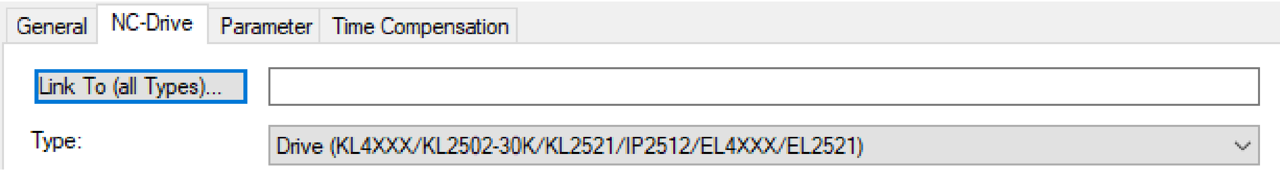
## 6.2.3 Axis component Drive

### 6.2.3.1 General




| Property | Description |
|---|---|
| **Name** | Selected name of the object. |
| **Object Id** | 32-bit identification number that is automatically assigned and unique within a project. |
| **Type** | Drive type<br>The selected drive type is displayed at this point. Changes to the drive type can be made via the NC-Drive [▶ 58] [▶ 43] dialog. |
| **Comment** | Freely editable field for user notes. |
| **Id** | Identification number of the drive, which is assigned consecutively. When a drive is deleted, the number becomes free again and can be assigned to a new encoder. |
| **Disabled** | A drive cannot be disabled on its own, but only by disabling the entire axis. |
| **Create symbols** | This option generates typical symbol names that are used in TwinCAT Scope, for example. |

## 6.2.3.2 NC-Drive




| Property | Description |
|---|---|
| Link To (all Types) | The button is used to establish the connection to the drive hardware. Note that drive devices containing both the motor and the encoder are linked via the link button of the higher-level NC axis. Only with separate devices (e.g. encoder terminal/analog output terminal) is the drive output linked at this point. |
| Type | After a configured drive device or output terminal has been selected, the drive type sets itself to match it. |

## 6.2.3.3 Parameter

**Drive**

The drive transfers the output voltage to the motor's power section. There is a wide range of supported drive variants: For example, servo drives, high speed drives, low speed drives, stepper motor drives. The drive parameters include the motor polarity and the reference velocity.

**Analog drive**

Within this context "analog" does not mean that the velocity is represented by a voltage (e.g. ±10 V) or a current (e.g. ±20 mA), but rather that the axis can be adjusted over an effectively continuous range of values. This kind of adjustment is also possible for drives with digital interfaces such as the BISSI terminals of type KL5051. When using such a digital interface, a velocity can be set in the same way as when controlling an analog value, even though it is transported as digital information.

### 6.2.3.3.1 Output Settings

**Invert Motor Polarity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Output Settings: | | | | |
| | Invert Motor Polarity | FALSE | | B | |

If set to `TRUE`, the Invert Motor Polarity parameter reverses the polarity of the motor and thus the direction of motor rotation.

- `FALSE`: The axis moves in the direction of larger positions when the drive is positively controlled.
- `TRUE`: The axis moves in the direction of smaller positions when the drive is positively controlled.

**⚠ WARNING**

**Risk of unexpected movements**

If the counting direction of the encoder and the motor polarity do not match, the axis will perform unexpected movements.

**Reference Velocity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Output Settings: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Reference Velocity | 2200.0 | | F | mm/s |
| | at Output Ratio [0.0 ... 1.0] | 1.0 | | F | |

For drives that are not directly controlled by a digital velocity value, e.g. a voltage or current interface, the Reference Velocity is used to scale the drive output. The Reference Velocity is at the same time an upper velocity limit that cannot be exceeded in addition to the maximum velocity. For all drive types, the Reference Velocity must be set greater than or equal to the maximum velocity.

(Strictly speaking, the velocity upper limit is the Reference Velocity divided by the Output Ratio, if an Output Ratio smaller than 1.0 is parameterized)

**at Output Ratio [0.0..1.0]**

*Proportional ratio for analog control*

The Reference Velocity normally refers to the maximum possible output, e.g. 10 V. The Output Ratio in this case is 1.0 or 100 %. With the Output Ratio factor it is possible to parameterize the reference velocity for another reference point, e.g. at 80 %.

*Reference Velocity and Maximum Velocity*

The Reference Velocity divided by the Output Ratio determines the maximum achievable velocity. A control component is required for controlling the axis, therefore the parameterized Maximum Velocity must be set 10% smaller, for example. For drives that are controlled with a digital velocity value (CoE, SoE), no reserve is necessary and both velocities can be equal.

### 6.2.3.3.2 Position and Velocity Scaling

**Output Scaling Factor**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Output Scaling Factor (Position) | 1.0 | | F | |
| | Output Scaling Factor (Velocity) | 6.0 | | F | |

**Output Scaling Factor (Position)**

Usually the drive parameter Output Scaling Factor (Position) does not cause any effect. To avoid that a future use of the parameter changes the behavior, the user should basically not change the default value `1.0` of this parameter.

The position output scaling is already set by the encoder input scaling. Regularly, the position input scaling corresponds to the position output scaling for a drive. For this reason, the drive parameter Output Scaling Factor (Position) is currently not evaluated. Instead, scaling is performed exclusively by the encoder parameters Scaling Factor Numerator and Scaling Factor Denominator, which must be adjusted for accurate scaling.

**Output Scaling Factor (Velocity)**

If a drive controller operates in velocity pre-control mode, the NC output value must be scaled. There are two ways for this scaling, depending on the type of drive controller used.

1. Analog drive controllers, e.g. supplied by a ±10 V terminal:

   This type of drive controller is scaled by applying the parameter Reference Velocity.
2. Digital drive controllers to which an absolute digital velocity command value is transferred, e.g. *CANopen DS402*:

   This type of drive controller is scaled by applying the parameter Output Scaling Factor (Velocity).

Below is a tabular comparison of when to apply the parameter Reference Velocity or the parameter Output Scaling Factor (Velocity).

| Drive type | Scaling with Reference Velocity | Scaling with Output Scaling Factor (Velocity) |
|---|---|---|
| M2400_DAC1 | x | |
| M2400_DAC2 | x | |
| M2400_DAC3 | x | |
| M2400_DAC4 | x | |
| KL4XXX | x | |
| KL4XXX_NONLINEAR | x | |
| TWOSPEED | x | |
| STEPPER | x | |
| SERCOS | | x |
| KL5051 | x | |
| AX2000_B200 | | x |
| SIMO611U | | x |
| UNIVERSAL | x | |
| NCBACKPLANE | x | |
| CANOPEN_LENZE | | x |
| DS402_MDP742 | | x |
| AX2000_B900 | x | |
| AX2000_B310 | x | |
| AX2000_B100 | x | |
| KL2531 | | x |
| KL2532 | | x |
| TCOM_DRV | | x |
| MDP_733 | | x |
| MDP_703 | x | (x)* |

*Also possible, but should be left at the value `1.0`.

**Output Delay (Velocity)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Output Delay (Velocity) | 0.0 | | F | s |

The output of the velocity value can be delayed by the time Output Delay (Velocity).

**Minimum /Maximum Drive Output Limitation [-1.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Minimum Drive Output Limitation [-1.0 ... 1.0] | -1.0 | | F | |
| | Maximum Drive Output Limitation [-1.0 ... 1.0] | 1.0 | | F | |

**Minimum Drive Output Limitation [-1.0 … 1.0]**

To limit the velocity and thus protect the hardware, a lower limit can be set for driving the axis. If only part of the output data type is valid, the minimum output value must be limited. Minimum Drive Output Limitation is a directional limitation of the total output. The value `1.0` corresponds to an unlimited output of 100 %. As a rule, the use of this parameter concerns a velocity output signal for the drive in connection with position control. In exceptional cases, the application of this parameter may concern a torque value or a current value.

**Maximum Drive Output Limitation [-1.0 … 1.0]**

To limit the velocity and thus protect the hardware, an upper output limit can be set for driving the axis. If only part of the output data type is valid, the maximum output value must be limited. The Maximum Drive Output Limitation is a direction-dependent limitation of the total output. The value `1.0` corresponds to an unlimited output of 100 %. As a rule, the use of this parameter concerns a velocity output signal for the drive in connection with position control. In exceptional cases, the application of this parameter may concern a torque value or a current value.

### 6.2.3.3.3 Torque and Acceleration Scaling

This parameter group generates an optional acceleration as servo control to intervene before a lag error occurs. It can, for example, define a torque value.

**Input Scaling Factor (Actual Torque)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input Scaling Factor (Actual Torque) | 0.0 | | F | |

The gain factor for the optional servo control.

**Input P-T1 Filter Time (Actual Torque)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input P-T1 Filter Time (Actual Torque) | 0.0 | | F | s |

Time for the P-T1 filter. This time comes as input to the P-T1 filter.

**Input P-T1 Filter (Actual Torque Derivative)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input P-T1 Filter (Actual Torque Derivative) | 0.0 | | F | s |

Derivative of the actual torque to be scaled. This derivative comes as input to the P-T1 filter.

See section PT1 Filter [▶ 117] for further information on the PT1 filter.

**Output Scaling Factor (Torque)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Scaling Factor (Torque) | 0.0 | | F | |

Sometimes an optional output scaling for the torque is necessary.

**Output Delay (Torque)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Delay (Torque) | 0.0 | | F | s |

The output of the torque value can be delayed by the time Output Delay (Torque).

**Output Scaling Factor (Acceleration)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Scaling Factor (Acceleration) | 0.0 | | F | |

Sometimes an optional output scaling is necessary for acceleration. (e.g. for the NC acceleration pre-control. Further information about acceleration pre-control can be found in the section Acceleration Feedforward [▶ 113].)

**Output Delay (Acceleration)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Delay (Acceleration) | 0.0 | | F | s |

The output of the acceleration value can be delayed by the time Output Delay (Acceleration).

### 6.2.3.3.4 Valve Diagram

Hydraulic axes have a non-linear characteristic curve. Valve characteristic curves are used to linearize the controlled system, which are constructed in a similar way to cam plates. For numerical control, for example, a hydraulic axis can be treated similarly to a servo axis.

**Valve Diagram: Table Id**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Table Id | 0 | | D | |

Each valve characteristic table has its own ID. This is assigned to each table consecutively. The table ID uniquely identifies the cam plate in the TwinCAT system.

**Valve Diagram: Interpolation type**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Interpolation type | 'LINEAR' | | E | |

Two types of interpolation

- 'LINEAR' or
- 'SPLINE'

can be selected to connect the discrete number pairs `(x, y)` in the valve characteristic table.

**Valve Diagram: Output offset [-1.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Output offset [-1.0 ... 1.0] | 0.0 | | F | |

This parameter allows you to adjust the zero crossing of the valve characteristic curve through the origin.

**More Information**

TE1500 Valve Diagram Editor.
Valve characteristic curves can be created with the TE1500 Valve Diagram Editor.
https://infosys.beckhoff.com/content/1033/te1500_tc3_valve_diagram_editor/index.html?id=5755459939328551364

## 6.2.3.3.5 Optional Position Command Output Smoothing Filter

In some applications, smoothing of the position output to the drive is used to reduce vibration of machine parts. This smoothing forms an additional filter that should be used with caution. This filter cannot be compensated by a static dead time.

**Smoothing Filter Type and Smoothing Filter Time**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Optional Position Command Output Smoothing Filter: | | | | |
| | Smoothing Filter Type | 'OFF (default)' | | E | |
| | Smoothing Filter Time | 0.01 | | F | s |

If

- 'OFF (default)'

is selected, smoothing of the position output is disabled.

As filter types

- `'Moving Average'` or

- `'P-Tn'`

can be selected. Both modes refer to the position setpoint.

`'Moving Average'`: As an output, the Moving Average Filter generates an average value of a series of position setpoints that comes to the filter as an input. Here, the Smoothing Filter Time prescribes the time interval over which the average extends. The values of the input are generated by position setpoint generation. Thus, the frequency of value generation is determined by the cycle time of the task for generating the setpoints.
For example, if a new value is generated every 1 ms and the Smoothing Filter Time is set to e.g. 20 ms, an average of 20 values is determined. The influence of a setpoint is perceptible for 20 ms.

`'P-Tn'`: As an output, the P-Tn filter generates an average value of a series of position setpoints that comes to the filter as an input. Here the Smoothing Filter Time prescribes the time constant of the P-Tn filter. The values of the input are generated by position setpoint generation. Thus, the frequency of value generation is determined by the cycle time of the task for generating the setpoints.

**Smoothing Filter Order (P-Tn only)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Optional Position Command Output Smoothing Filter: | | | | |
| | Smoothing Filter Order (P-Tn only) | 2 | | D | |

The order of the P-Tn smoothing filter used.

## 6.2.3.3.6 Sercos Behavior

**C1D Error Tolerance**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Sercos Behavior: | | | | |
| | C1D Error Tolerance | 'STANDARD' | | E | |

Some hardware drive errors can be parameterized on the drive side so that an error is signaled but the drive continues to follow the setpoints.

If the parameter *C1D Error Tolerance* is set to 'IGNORE SELECTED ERRORS', these errors do not lead to an immediate stop of the axis by the NC.

The user recognizes the drive error by the flag *DriveDeviceError* in the *StateDWord* of the NC axis and must stop the axis using the suitable PLC code.

## 6.2.3.3.7 Other Settings

**Drive Mode**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Drive Mode | 'STANDARD' | | E | |

`'STANDARD'`: There is currently only one drive mode, and that is the `'STANDARD'` mode. In principle, other modes are conceivable in the execution of a more open system.

**Drift Compensation (DAC-Offset)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Drift Compensation (DAC-Offset) | 0.0 | | F | mm/s |

This value is added to the drive control level. In this way, a constant offset can be added to the output, for example to compensate for zero point deviations in analog drives.
See section Automatic DAC offset adjustment [▶ 113].

**Following Error Calculation (Following Error)**

The following error corresponds to the position lag, also referred to as position lag value.

Position lag value = actual position - current set position.

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Following Error Calculation | 'Intern' | | E | |

The following error can be calculated with the NC or in the hardware drive. For all hardware drives working in position control mode, the calculation should be done externally in the hardware drive.

`'Intern'`: The following error is calculated internally in TwinCAT.

`'Extern'`: The following error is calculated externally in the drive controller.

**Error Tolerance**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Error Tolerance (NC error handling) | 'STANDARD' | | E | |

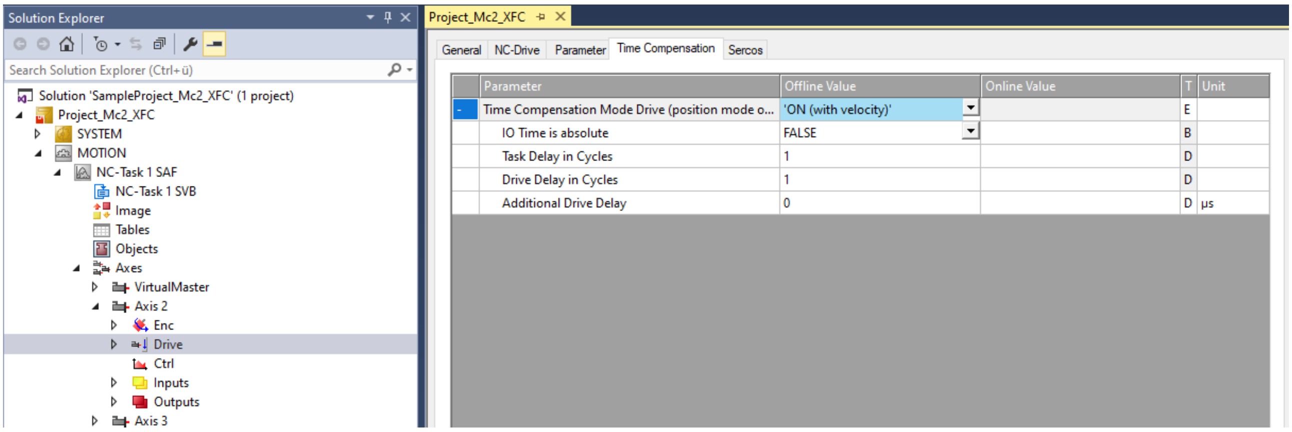
Some hardware drive errors can be parameterized on the drive side so that an error is signaled but the drive continues to follow the setpoints.

If the parameter **Error Tolerance** is set to `'IGNORE SELECTED ERRORS'`, these errors do not lead to an immediate stop of the axis by the NC.

The user recognizes the drive error by the flag `DriveDeviceError` in `StateDWord` of the NC axis and must stop the axis using the suitable PLC code.

## 6.2.3.4 Time Compensation

A requirement for high-precision conversion of positions into times and vice versa is precise dead time compensation of the axes. The dead time compensation can be enabled and configured via Drive > Tab **Time Compensation** in the output direction.




| Parameter | Description | Type | Unit |
|---|---|---|---|
| Time Compensation Mode Drive (position mode only) | Enabling/disabling the dead time compensation of the axis drive incl. selection of the mode:<br>• `OFF`<br>• `ON (with velocity)` (recommended)<br>• `ON (with velocity and acceleration)` | enum | |
| IO Time is absolute | Default `FALSE`<br>Must be set to `TRUE` if an absolute timestamp is present (special case). | bool | |
| Task Delay in Cycles | Delay due to NC SAF cycle time.<br>Default 1 | double | |
| Drive Delay in Cycles | Additional delays by entire I/O cycles. This time is not a fixed value, but changes with the cycle time. | double | |
| Additional Drive Delay | Fixed time value in µs caused by the hardware used. | double | µs |

Further details on dead time compensation can be found in the documentation TF5065 TwinCAT 3 \| PLC Motion Control XFC/XFC NCI.

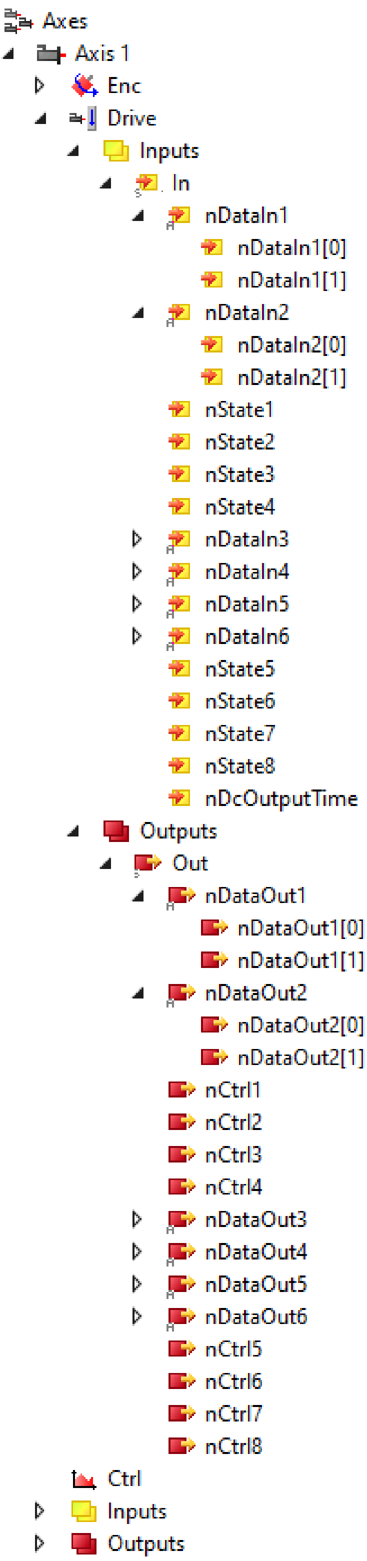
## 6.2.3.5 Process image

The process image (cyclic interface) of an axis is used to connect it to different drive components. In the simplest case, a link is established between the axis and the drive and the necessary links between the process images are made automatically. In individual cases, in particular when unknown hardware components are integrated in the system, these links must be established manually.

***Notice*** **The data structures described here form an internal interface between NC drivers and connected drive hardware. This interface is constantly under development and can change in the future.**

### Drive process image of an axis

Various drive hardware or the corresponding Bus Terminals (+/- 10V, PWM etc.) are connected via the drive process image. Insofar as this hardware is directly supported by the system, no manual configuration is necessary.




**Input data to the drive process image of an axis**

| NC-In variable | Data type | Description |
|---|---|---|
| `nDataIn1` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | Optional lag error (position control deviation, i.e. target position minus actual position) of the I/O Drive in *increments*. Exclusively for the operation mode *Cyclic Position Mode* (e.g. for EtherCAT, SERCOS, CANopen).<br>This quantity is processed by the NC accordingly and processed mathematically as an "external" lag distance in physical units (e.g. in mm or degrees). |

| NC-In variable | Data type | Description |
|---|---|---|
| | | Mapping with:<br>• SoE: S-0-0189 Following error<br>• CoE: 0x60F4 Following error actual value |
| `nDataIn2` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | reserved |
| `nState1` | `USINT` | Optional status information<br>e.g.: *Drive Error*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.<br>Mapping with<br>• SoE: S-0-0135 Drive Statusword<br>• CoE: 0x6041 Statusword |
| `nState2` | `USINT` | Additional optional status information.<br>e.g.: *Drive Error*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.<br>Mapping with:<br>• SoE: S-0-0135 Drive statusword<br>• CoE: 0x6041 Statusword |
| `nState3` | `USINT` | reserved |
| `nState4` | `USINT` | Optional fieldbus-dependent I/O status such as *WcState* (Working Counter) in the case of EtherCAT or *CdlState* in the case of Beckhoff Lightbus. |
| `nDataIn3` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | Optional actual torque/force value of the I/O Drive in increments.<br>This variable is processed by the NC accordingly and processed mathematically as "ActTorque" in physical units (e.g. % or A).<br>nDataIn3[0] mapping with:<br>• SoE: S-0-0084 Torque feedback value (2 byte)<br>• CoE: 0x6077 Torque actual value (2 byte) |
| `nDataIn4` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | reserved |
| `nDataIn5` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | reserved |
| `nDataIn6` | `UDINT / DINT` (4 byte)<br>or<br>`UINT / INT` (2 byte) | reserved |
| `nState5` | `USINT` | Mapping with:<br>• CoE: 0x6061 Modes of operation display |
| `nState6` | `USINT` | reserved |
| `nState7` | `USINT` | reserved |

| NC-In variable | Data type | Description |
|---|---|---|
| nState8 | USINT | reserved |
| nDcOutputTime | DINT | Optional: Used for NC dead time compensation of the drive. |

**Output data from the drive process image of an axis**

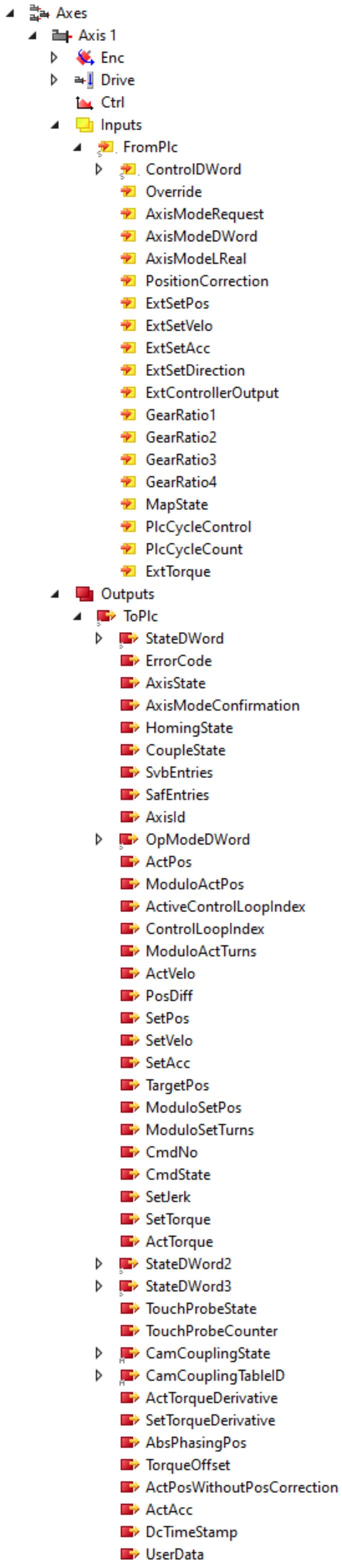
| NC-Out variable | Data type | Description |
|---|---|---|
| nDataOut1 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Current target velocity or current target position in *increments*<br>The target velocity or target position of the NC in physical units, e.g. mm or degrees, is mathematically converted back to an incremental value by the NC and transferred to the drive. Overflows of the incremental value are hereby taken into account by the NC in the target position.<br>Depending on the drive type, a standardization of the target velocity or target position is carried out in *increments*. If the drive type *Universal Drive* has been selected, nOutData1 contains the overall velocity (incl. position control part) with sign.<br>Mapping with:<br>• SoE: S-0-0047 Target Position<br>• CoE:<br>  ◦ 0x607A Target position<br>  ◦ 0x6062 Nominal position,<br>Alternative if 0x607A is not available |
| nDataOut2 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Current target velocity or current target position in *increments*<br>The target velocity or target position of the NC in physical units, e.g. mm or degrees, is mathematically converted back to an incremental value by the NC and transferred to the drive. Overflows of the incremental value are hereby taken into account by the NC in the target position.<br>Depending on the drive type, a standardization of the target velocity or target position is carried out in *increments*. If the *Universal Drive* type has been selected, nOutData2 contains the value of the overall velocity (incl. position control part, without sign).<br>Mapping with:<br>• SoE: S-0-0036 Target velocity (4 byte)<br>• CoE:<br>  ◦ 0x60FF Target velocity (4 byte)<br>  ◦ 0x606B Nominal velocity (4byte),<br>Alternative if 0x60FF is not available.<br>  ◦ 0x6042 Target velocity for velocity mode (for frequency inverter) (2 byte),<br>Alternative if 0x60FF and 0x606B are not available. |
| nCtrl1 | USINT | Optional control information<br>e.g.: *Drive Reset*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.<br>Mapping with:<br>• SoE: S-0-0134 Master Controlword<br>• CoE: 0x6040 Controlword |
| nCtrl2 | USINT | Additional optional control information<br>Digital direction output of the setpoint generation (corresponds to the sign of the target velocity, hence without position controller) |

| NC-Out variable | Data type | Description |
|---|---|---|
| | | Digital Outputs Setpoint Generator:<br>0x41 (0100 0001) = Minus<br>0x42 (0100 0010) = Plus<br>0x80 (1000 0000) = Stop<br>The only exception is the AX2xxx-B200/B900 drive. In this case, communication takes place with the Drive State Machine.<br>Mapping with<br>• SoE: S-0-0134 Master Controlword<br>• CoE: 0x6040 Controlword |
| nCtrl3 | USINT | Additional optional control information<br>Digital direction output or drive stages of the total output (sum of setpoint generation and position controller) Digital Outputs (Setpoint Generator + Position Controller):<br>0x41 (0100 0001) = Minus<br>0x42 (0100 0010) = Plus<br>0x80 (1000 0000) = Stop |
| nCtrl4 | USINT | reserved |
| nDataOut3 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Current target acceleration (2nd time derivative of target position) in *increments.* Optionally, it can also contain the current target torque in increments (e.g. from an extended transformation with dynamic model).<br>The target acceleration and the target torque can be available either individually or as a sum of the individual variables (depending on the parameterization of the output scaling).<br>The output size is signed.<br>Mapping with:<br>• SoE: S-0-0081 Torque offset (2 byte)<br>• CoE: 0x60B2 Torque offset (2 byte) |
| nDataOut4 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0080 Torque command value (corresponds to SetTorque multiplied by the output scaling)<br>• CoE:<br>  ◦ 0x6071 Target torque/force (2 byte) (corresponds to SetTorque multiplied by the output scaling)<br>  ◦ 0x6074 Nominal Torque (2 byte), Alternative if 0x6071 is not available. |
| nDataOut5 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0485 High velocity limit value<br>• CoE: 0x60C9 High velocity limit value |
| nDataOut6 | UDINT / DINT (4 byte)<br>or<br>UINT / INT (2 byte) | Mapping with:<br>• SoE: S-0-0484 Low velocity limit value<br>• CoE: 0x60C8 Low velocity limit value |
| nCtrl5 | USINT | Mapping with:<br>• CoE: 0x6060 Mode of operation |
| nCtrl6 | USINT | reserved |
| nCtrl7 | USINT | reserved |
| nCtrl8 | USINT | reserved |

## 6.2.4 Axis component cyclic interface

### Cyclic process image of an axis

The PLC is connected via the cyclic process image of an axis. For this purpose, an instance of the AXIS_REF structure is created in the PLC, which is then linked to the cyclic process image.



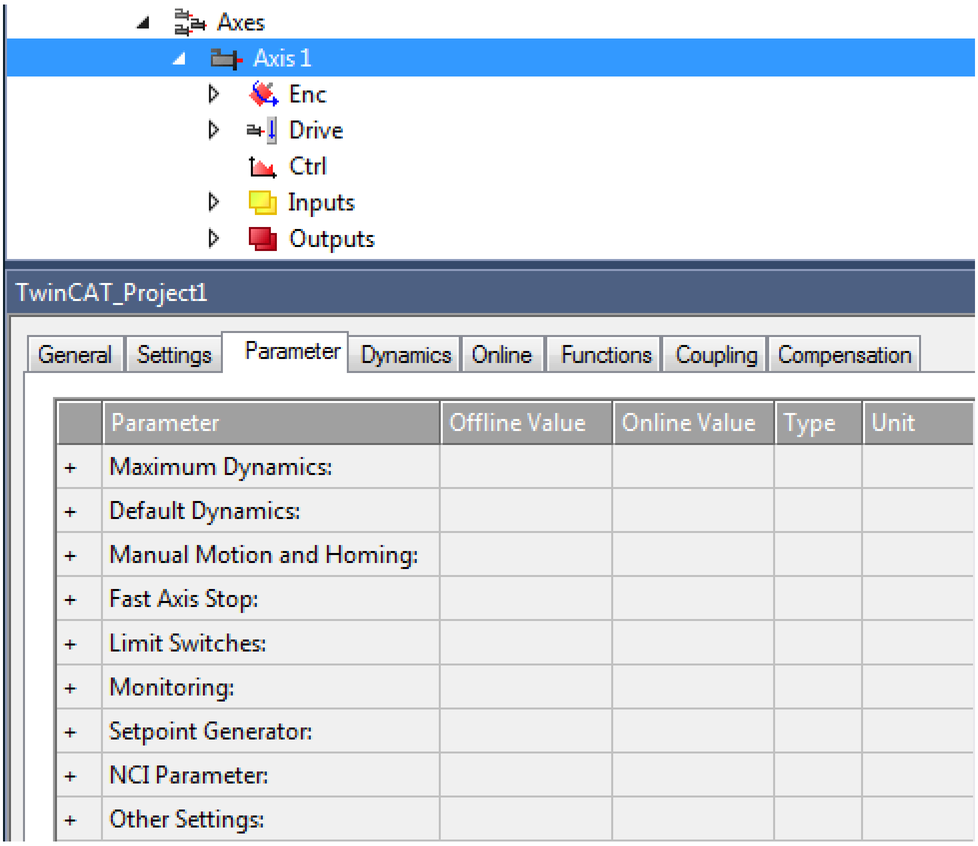

The structures are described in the documentation of the library Tc2_MC2:

PLCTONC_AXIS_REF

NCTOPLC_AXIS_REF

# 7 Parameters for Motion

## 7.1 Axes | Axis 1




Under "MOTION | NC-Task 1 SAF | Axes | Axis 1" the Parameter tab shows the parameter groups

- Maximum Dynamics,
- Default Dynamics,
- Manual Motion and Homing,
- Fast Axis Stop,
- Limit Switches,
- Monitoring,
- Setpoint Generator,
- NCI Parameter,
- Other Settings.

### 7.1.1 Maximum Dynamics, Default Dynamics

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Maximum Dynamics: | | | | |
| | Reference Velocity | 2200.0 | | F | mm/s |
| | Maximum Velocity | 2000.0 | | F | mm/s |
| | Maximum Acceleration | 15000.0 | | F | mm/s2 |
| | Maximum Deceleration | 15000.0 | | F | mm/s2 |
| | Parameter | Offline Value | Online Value | Type | Unit |

| | | | | | |
|---|---|---|---|---|---|
| - | Default Dynamics: | | | | |
| | Default Acceleration | 1500.0 | | F | mm/s2 |
| | Default Deceleration | 1500.0 | | F | mm/s2 |
| | Default Jerk | 2250.0 | | F | mm/s3 |

**Dynamic-Parameters**

- Velocity *Vel*,
- Acceleration *Acc*,
- Deceleration *Dec*,
- Jerk.

The *jerk* is the derivative of acceleration or deceleration with respect to time. Thus, it describes how quickly acceleration or deceleration change.

### Reference Velocity

For drives that are not directly controlled by a digital velocity value, e.g. a voltage or current interface, the Reference Velocity is used to scale the drive output. The Reference Velocity is at the same time an upper velocity limit that cannot be exceeded in addition to the maximum velocity. For all drive types, the Reference Velocity must be set greater than or equal to the maximum velocity.

(Strictly speaking, the velocity upper limit is the Reference Velocity divided by the Output Ratio, if an Output Ratio smaller than 1.0 is parameterized)

For details, see the Drive Parameters > Reference Velocity [▶ 97].

### "Maximum Dynamics" and "Default Dynamics"

The dynamic parameters are absolute, unsigned values. The default values are used if the user has not specified any values, e.g. for a motion command. The maximum values limit the axis dynamics and must be parameterized greater than or equal to the default dynamics.

The maximum values are observed by newer products like the Tc3_McCoordinatedMotion Library. However, for some products, such as the Tc2_MC2 library, the maximum acceleration and the maximum jerk are not taken into account.

### Tc2_MC2 library

| | |
|---|---|
| Default Dynamics | • If the input value "0.0" is assigned to one of the dynamic parameters "Acceleration, Deceleration, Jerk" at a motion function block or this input is left empty, then a default value is used instead. |
| Maximum Dynamics | • Velocity values that exceed the maximum velocity will not be accepted and will result in an error.<br>• Values for Acceleration, Deceleration and Jerk are not checked for exceeding the maximum parameters, but are accepted. |
| Coupled axes | • In the case of a coupled slave axis, its dynamics depend exclusively on the master movement and maximum values are not checked.<br>• When decoupling a slave axis, various measures are taken to prevent the maximum velocity from being exceeded or the direction of movement from being reversed.<br>• Examples of such measures are increasing the jerk or increasing the acceleration or deceleration to the maximum value. |

**Tc3_McCoordinatedMotion Library, Tc3_McCollisionAvoidance Library**

| | |
|---|---|
| *Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Default Values* | • If for one of the dynamic-parameters “Acc, Dec, jerk” the input value “0.0” is assigned to a motion function block, this assignment leads to an error that means that this value is not allowed.<br>• If for one of the dynamic-parameters “Acc, Dec, jerk” you would like to refer to a default value at a motion function block, this parameter has to be set to the constant value “*MC_Default*”. |
| *Tc3_Mc CoordinatedMotion Tc3_Mc CollisionAvoidance Maximum Dynamics* | Vel, Acc, Dec<br>• For the dynamic-parameters “Vel, Acc, Dec” the parameterized values are used.<br>• For the dynamic-parameters “Vel, Acc, Dec” maximum values can be parameterized at a motion function block using the constant value “*MC_Maximum*”.<br>Jerk<br>• There is no maximum value for the jerk.<br>• The jerk is set to the value “unlimited”. Simultaneously, a three-phase-profile or a three-phase-acceleration-setter is applied for motion.<br>Default Values<br>• It is allowed to parameterize default values that exceed their corresponding maximum values.<br>• If a default value is parameterized that exceeds ist corresponding maximum value, a warning will be given, but no error is thrown.<br>• At a Tc3_McCoordinatedMotion-function block or a Tc3_McCollisionAvoidance-function block parameterized default values using the constant value *MC_Default* will be mutually limited to the corresponding maximum values without giving an error message. |

## 7.1.2 Manual Motion and Homing

### Homing Velocity

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Homing Velocity (towards plc cam) | 30.0 | | F | mm/s |
| | Homing Velocity (off plc cam) | 30.0 | | F | mm/s |

### bCalibrationCam

A boolean input of `MC_Home`. It evaluates the signal of a referencing cam. This reference signal can be coupled into the control unit via a digital input.

### Homing Velocity (towards plc cam)

Velocity used by a `MC_Home` function block when homing to a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

### Homing Velocity (off plc cam)

Velocity used by a function block `MC_Home` when moving away from a referencing cam in the standard homing sequence when the `HomingMode MC_DefaultHoming` is selected and the `bCalibrationCam` input is evaluated.

### Manual Velocity

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Manual Velocity (Fast) | 600.0 | | F | mm/s |
| | Manual Velocity (Slow) | 100.0 | | F | mm/s |

### Manual Velocity (Fast)

Online Dialog:

- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | -- F1.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | ++ F4.
- Similar for other identifiers

MC_Jog:

- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards TRUE` and selected as its `Mode MC_JOGMODE_STANDARD_FAST`.

### Manual Velocity (Slow)

Online Dialog:

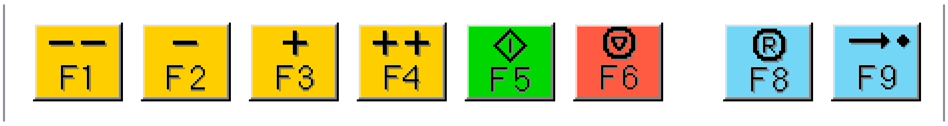
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | - F2.
- Used velocity for MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online | + F3.
- Similar for other identifiers.

MC_Jog:

- Velocity used by a function block `MC_Jog` applied to the axis when its input is `JogForward` or its input is `JogBackwards TRUE` and selected as its `Mode MC_JOGMODE_STANDARD_SLOW`.

### Buttons in the Online dialog

In the "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Online" dialog, there are the buttons -- F1, - F2, + F3 and ++ F4.




### Jog Increment

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Manual Motion and Homing: | | | | |
| | Jog Increment (Forward) | 5.0 | | F | mm |
| | Jog Increment (Backward) | 5.0 | | F | mm |

### Jog Increment (Forward)

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

### Jog Increment (Backward)

Not used.

This parameter is not currently used explicitly in TC3 Motion libraries. However, the parameter itself can be read or written or inserted indirectly by the user, e.g. in a function block created by the user or in an HMI.

**MC_JOGMODE_INCHING**

The function block `MC_Jog` enables an axis to be moved via manual keys. The key signal can be connected directly to the `JogForward` or `JogBackwards` input. The desired operation mode is specified by input `Mode`. When using the `MC_JOGMODE_INCHING` mode, a rising edge at one of the jog inputs moves the axis over a certain distance that is assigned at input `Position`.

**More Information:**

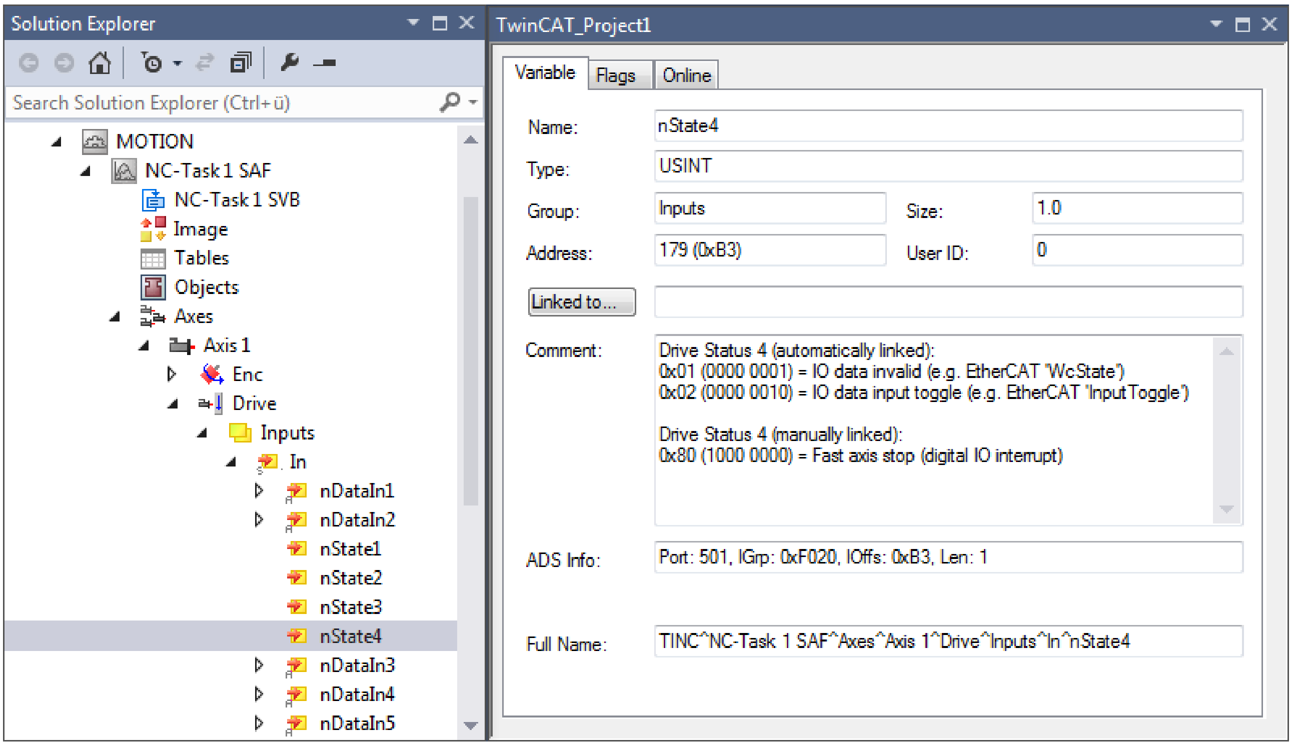
- MC_Jog (PLC library Tc2_MC2)

## 7.1.3 Fast Axis Stop

**Fast Axis Stop**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Fast Axis Stop: | | | | |
| | Fast Axis Stop Signal Type (optional) | 'OFF (default)' | | E | |
| | Fast Acceleration (optional) | 0.0 | | F | mm/s2 |
| | Fast Deceleration (optional) | 0.0 | | F | mm/s2 |
| | Fast Jerk (optional) | 0.0 | | F | mm/s3 |

Usually a stop is triggered by PLC code using `MC_Stop`. However, there are special applications within that the time delay of stop has to be as small as possible. Within this situation the input Drive.Inputs.In.nState4 comes into play triggering a stop directly without being mapped via the PLC process image.




Drive Status 4 (manually linked):
0x80 (1000 0000) = Fast Axis Stop (digital IO interrupt)

**Variable nState4**

The variable Drive.Inputs.In.nState4 can be mapped to any event source. Note its data type `USINT` and the bit nState4.7 being responsible for the Fast Axis Stop.

**Fast Axis Stop**

For an axis the Fast Axis Stop is performed when its Drive.Inputs.In.nState4.7 variable exhibits the signal type that is selected within the drop-down list "Fast Axis Stop Signal Type (optional)" and differs from “OFF (default)”.

**Signal Type**

The “Fast Axis Stop Signal Type (optional)” enumeration specifies six elements:

- OFF (default)
  For any signal type that the Drive.Inputs.In.nState4.7 variable exhibits no Fast Axis Stop is performed.
- Rising Edge
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a rising edge.
- Falling Edge
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a falling edge.
- Both Edges
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a rising edge and a falling edge, respectively. Alternatively, a Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit exhibits a falling edge and a rising edge, respectively.
- High Active
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit appears high active.
- Low Active
  A Fast Axis Stop is performed when the Drive.Inputs.In.nState4.7 bit appears low active.

**Optional**

An optional fast axis parameter has to differ from zero to be applied when a Fast Axis Stop is performed.

**Acceleration, Deceleration, Jerk**

When a Fast Axis Stop is performed on the corresponding axis within the boundaries prescribed by the “Maximum Dynamics” the “Fast Acceleration (optional)” float value accelerates the axis, the “Fast Deceleration (optional)” float value decelerates the axis and the “Fast Jerk (optional)” float value is applied. When a Fast Axis Stop is not performed, not any fast axis parameter is applied.

**Further Information:**

- MC_Stop (PLC library Tc2_MC2)

## 7.1.4 Limit Switches

The Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

**Soft Position Limit Minimum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Minimum Monitoring | FALSE | | B | |
| | Minimum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Minimum Monitoring is not enabled.

`TRUE`: Soft Position Limit Minimum Monitoring is enabled.

**Minimum Position**

Position lower limit for the axis, which must not be violated when Soft Position Limit Minimum Monitoring is enabled. Commands that violate this lower limit will be rejected.

**Soft Position Limit Maximum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Maximum Monitoring | FALSE | | B | |
| | Maximum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Maximum Monitoring is not enabled.

`TRUE`: Soft Position Limit Maximum Monitoring is enabled.

**Maximum Position**

Position upper limit for the axis, which must not be violated when Soft Position Limit Maximum Monitoring is enabled. Commands that violate this upper limit will be rejected.

## 7.1.5 Monitoring

**Position Lag Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Position Lag Monitoring | TRUE | | B | |
| | Maximum Position Lag Value | 5.0 | | F | mm |
| | Maximum Position Lag Filter Time | 0.02 | | F | s |

The position lag monitoring monitors the position lag value. If the parameterized limits for position and time are exceeded, a runtime error is output.

Position lag value = actual position - current set position.

`TRUE`: Position Lag Monitoring is enabled.

`FALSE`: Position Lag Monitoring is not enabled.

**Maximum Position Lag Value and Maximum Position Lag Filter Time**

The Maximum Position Lag Value is the upper limit, which must not be exceeded for longer than the Maximum Position Lag Filter Time. Otherwise the NC axis is stopped immediately by direct shutdown and set to the logical state "Error", whereby the error `0x4550` is output.

**Position Range Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Position Range Monitoring | TRUE | | B | |
| | Position Range Window | 5.0 | | F | mm |

Position Range Monitoring monitors whether the actual position of the NC axis reaches a window around the target position. Once the window is reached, the status flag `Axis.Status.InPositionArea` is set to `TRUE`.

`TRUE`: Position Range Monitoring is enabled.

`FALSE`: Position Range Monitoring is not enabled.

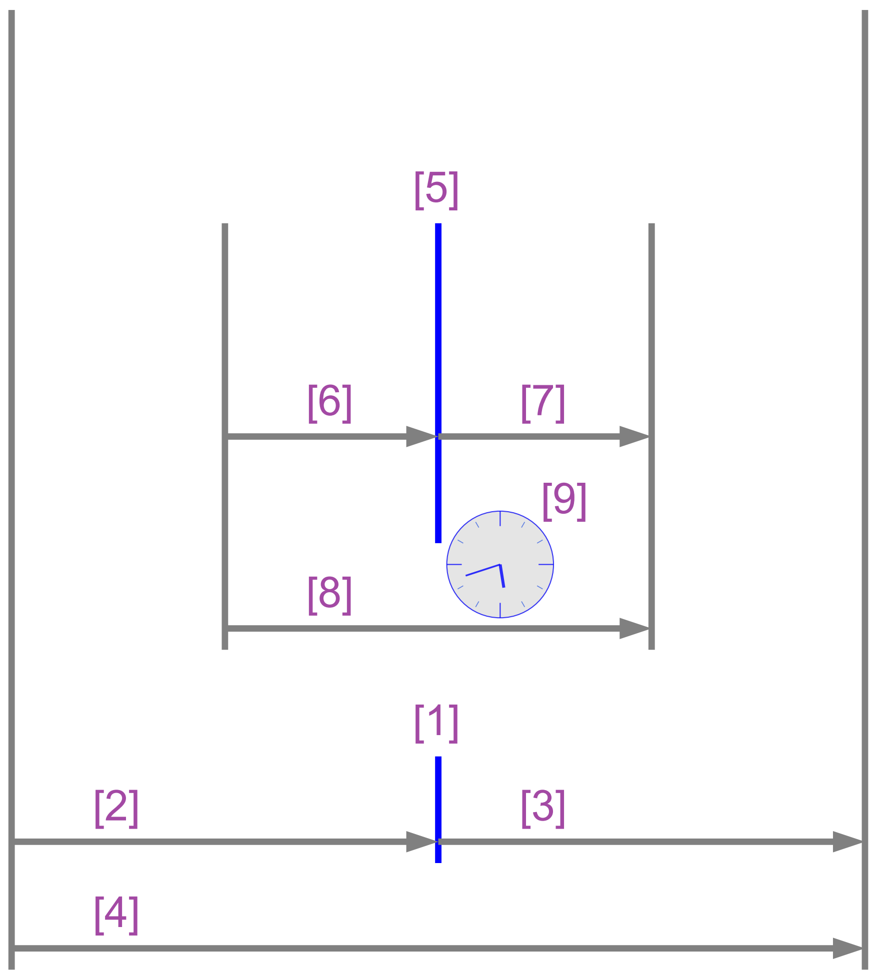
**Position Range Window**

Specifies the tolerance of the actual position of the NC axis with respect to the target position so that the status flag `Axis.Status.InPositionArea` is set to `TRUE`.

**NC-Online: "In Pos. Range" – `Axis.Status.InPositionArea`**

The value of variable `Axis.Status.InPositionArea` corresponds to the state of the checkbox "In Pos. Range" within the group box "Status (phys.)" of the NC-Online dialog. If the variable `Axis.Status.InPositionArea` is set on `TRUE`, the checkbox "In Pos. Range" is checked.

**Graphic example**




| | |
|---|---|
| [1] | • Nominal value of the target position. |
| [2] | • Position Range Window. |
| [3] | • Position Range Window. |
| [4] | Variable `Axis.Status.InPositionArea`: |

| | |
|---|---|
| *Position Range Monitoring* | • If the parameter "Position Range Monitoring" is set to `TRUE` and ...<br>• ... if the actual position is in this range [4],<br>• then the variable `Axis.Status.InPositionArea` is set to `TRUE`. |

### Target Position Monitoring

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Target Position Monitoring | TRUE | | B | |
| | Target Position Window | 2.0 | | F | mm |
| | Target Position Monitoring Time | 0.02 | | F | s |

Target Position Monitoring monitors whether the actual position of the NC axis reaches a window around the target position and also remains in this window for a minimum time. After that the status `Flag Axis.Status.InTargetPosition` is set to `TRUE`.

`TRUE`: Target Position Monitoring is enabled.

`FALSE`: Target Position Monitoring is not enabled.

**Target Position Window**

The Target Position Window specifies the tolerance of the actual position of the NC axis in relation to the target position, which is to be taken into account in Target Position Monitoring.

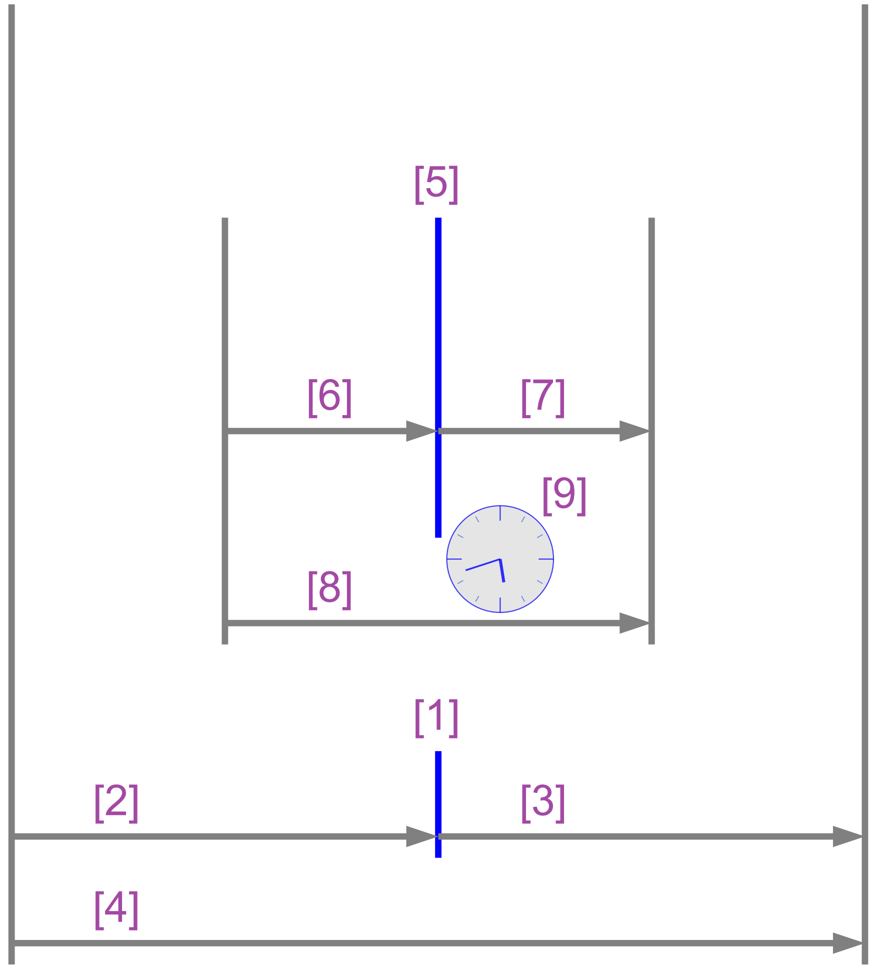
**Target Position Monitoring Time**

The Target Position Monitoring Time specifies the time in which the actual position of the NC axis must be at least within the tolerance range of the target position (Target Position Window) so that the status flag `Axis.Status.InTargetPosition` is set to `TRUE`.

**NC-Online: "In Target Pos." – `Axis.Status.InTargetPosition`**

The value of the variable `Axis.Status.InTargetPosition` corresponds to the state of the checkbox "In Target Pos." within the group box "Status (phys.)" of the NC-Online dialog. If the variable `Axis.Status.InTargetPosition` is set on `TRUE`, the checkbox "In Target Pos." is checked.

**Graphic example**




| | |
|---|---|
| [5] | • Nominal value of the target position. |
| [6] | • Target Position Window. |
| [7] | • Target Position Window. |
| [8], [9] | Target position: |
| *Target Position Monitoring* | • If the parameter "Target Position Monitoring" is set to `TRUE` and ...<br>• ... if the actual position is in this range [8] at least for the duration"Target Position Monitoring Time" [9] without interruption up to the actual time,<br>• then the variable `Axis.Status.InTargetPosition` is set to `TRUE`. |

### In-Target Alarm

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | In-Target Alarm | FALSE | | B | |
| | In-Target Timeout | 5.0 | | F | s |

The In-Target Alarm monitors whether the axis reaches the Target Position Window within the In-Target Timeout.

`TRUE`: The In-Target alarm is enabled.

FALSE: The In-Target alarm is not enabled.

**In-Target Timeout**

If the NC axis does not reach the Target Position Window within the In-Target Timeout, the Nc axis reports the error 0x435C. The time measurement is started when the set position of the axis has reached its nominal position.

**Motion Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Motion Monitoring | FALSE | | B | |
| | Motion Monitoring Window | 0.1 | | F | mm |
| | Motion Monitoring Time | 0.5 | | F | s |

Motion Monitoring checks whether an axis is actually moving while it is executing a motion command. This makes it possible, for example, to detect the mechanical blocking of an axis at an early stage.

TRUE: Motion Monitoring is enabled.

FALSE: Motion Monitoring is not enabled.

**Motion Monitoring Window**

The Motion Monitoring Window defines the distance that the encoder (actual position) should be expected to travel during one cycle of the NC SAF task. Here a value/distance/length of some encoder increments can be set.

**Motion Monitoring Time**

Monitoring starts as soon as the axis executes a motion command and ends when the axis comes to a logical standstill. If its actual position does not change by more than the Motion Monitoring Window in at least one NC cycle during the Motion Monitoring Time, the NC axis outputs the error 0x435D.

## 7.1.6 Setpoint Generator

**Setpoint Generator Type**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Setpoint Generator: | | | | |
| | Setpoint Generator Type | 7 Phases (optimized) | | E | |

7 Phase (optimized)

Only an optimized 7-phase setpoint generator is supported.

**Velocity Override Type**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Setpoint Generator: | | | | |
| | Velocity Override Type | Reduced (iterated) | | E | |

The NC axis supports a speed override. This means that changing the override creates a new velocity, but does not affect the ramps (acceleration or jerk). The used override types only differ in terms of reference velocity.

More information about override types can also be found in Path Override (Interpreter Override Types) [▸ 114].

`Reduced (iterated)`

The override refers to the maximum velocity of the profile calculated by the setpoint generator.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. With a smaller override value, the actual travel velocity is further reduced.

`Original (iterated)`

The override refers to the parameterized velocity of the executed motion command.

Example: A motion command with 1000 mm/s with a short travel distance is assigned. This velocity cannot be reached on this route and a travel profile with 700 mm/s at 100% override is calculated. Since the override refers to the velocity of the motion command, the actual velocity is only reduced here below an override value of 70 %.

## 7.1.7 NCI Parameters

### Rapid Traverse Velocity (G0)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Rapid Traverse Velocity (G0) | 2000.0 | | F | mm/s |

The Rapid Traverse Velocity is used when an interpreter command `G0` is executed. See section Rapid traverse velocity [▸ 117] for a brief description of the interpreter command `G0`.

### Velo Jump Factor

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Velo Jump Factor | 0.0 | | F | |

The reduction factor `C0[i]` is the Velo Jump Factor.

### Background information

**Segment transitions**

Segments are geometric objects. We consider them as curves in the sense of differential geometry, parameterized by means of their length `arc`.

A segment transition from a segment `S_in` to a segment `S_out` is called of geometric type `Ck`, where `k` is a natural number (including `0`) describing `k` continuous `arc` length differentials for each segment and the corresponding `k^th` derivatives at the transition point.

`C0` transitions: Have a knee-point at the transition point.

`C1` transitions: Appear smooth, but are not smooth in dynamic terms. At the segment transition point, there is a jump in acceleration.

`C2` transitions: Are dynamically smooth and their smoothness is only jerk-limited.

`Ck` transitions: are dynamically smooth.

**Segment dynamics**

Velocity `v`: The segment set velocity `v` changes at the segment transition from `v_in` to `v_out`. At the segment transition the set velocity is always reduced to the lower of the two values.

Acceleration `a`: At the segment transition, the current path acceleration is always reduced to zero.

Jerk `j`: At the segment transition, the jerk changes according to the geometry of the segment transition. This jerk change can cause a noticeable dynamic jump.

**Velocity reduction modes for C0 transitions**

Several reduction methods are available for `C0` transitions. One of them is the reduction method VELOJUMP. The reduction method VELOJUMP reduces the velocity after permitted velocity jumps for each axis.

**The VELOJUMP reduction method for C0 transitions**

Basically `v_link = min(v_in, v_out)` applies. For the axis `[i]`, the allowed absolute velocity jump is `v_jump[i] = C0[i] * min(A+[i], -A-[i]) * T`, where `C0[i]` is the reduction factor, `A+[i], A-[i]` is the acceleration or deceleration limits for the axis `[i]`, and `T` is the cycle time. The VELOJUMP reduction method ensures that the path velocity is reduced at the segment transition `v_link` until the absolute step change in the set axis velocity of axis `[i]` is at most `v_jump[i]`. However, `v_min` has priority: if `v_link` is smaller than `v_min`, then `v_link` is set to `v_min`. In the case of motion reversal with no programmed stop, there will be a step change in axis velocity.

**Tolerance ball auxiliary axis**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Tolerance ball auxiliary axis | 0.0 | | F | |

See section Tolerance Ball [▸ 118] for further information.

**Max. position deviation, aux. axis**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | NCI Parameter: | | | | |
| | Max. position deviation, aux. axis | 0.0 | | F | |

Introduced for future extensions.

## 7.1.8 Other Settings

**Position Correction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Position Correction | FALSE | | B | |
| | Filter Time Position Correction (P-T1) | 0.0 | | F | s |

The Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

FALSE: The Position Correction is disabled.

TRUE: The Position Correction is enabled.

The variable axis.PlcToNc.PositionCorrection is of data type LREAL and belongs to the structure PLCTONC_AXIS_REF. If Position Correction is enabled, this variable adds an additional offset to the target position. It should be noted that this correction does not affect the software end positions.

**Filter Time Position Correction (P-T1)**

The filter time for the PT-1 filter, which filters fluctuations within the Actual Position Correction with the filter time set here. See section PT1 Filter [▶ 117] for further information on the PT1 filter.

**See also:**

**MC_PositionCorrectionLimiter**

- TwinCAT 3 PLC Lib: Tc2_MC2

The function block MC_PositionCorrectionLimiter adds the correction value PositionCorrectionValue to the actual position value of the axis. Depending on the CorrectionMode the position correction value is either written directly or filtered.

> To use the MC_PositionCorrectionLimiter function block successfully the Position Correction has to be enabled by setting the parameter Position Correction TRUE.

**Backlash**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Backlash | 0.0 | | F | mm |

This parameter is only present for compatibility reasons. For more information, visit NC Backlash Compensation.

**Error Propagation Mode**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Error Propagation Mode | 'INSTANTANEOUS' | | E | |
| | Error Propagation Delay | 0.0 | | F | s |

For the slave axis the error transmission can be delayed.

'INSTANTANEOUS': The error transmission is not delayed.

'DELAYED': The error transmission is delayed by the Error Propagation Delay.

**Error Propagation Delay**

The delay time by which error propagation for the slave axis is delayed when Error Propagation Mode 'DELAYED' is selected.

If an error occurs at a slave axis during runtime, the corresponding master axis is not set to error state until the time assigned here has elapsed. A state of interest of the slave axis, especially its error state, can be observed by PLC code. In this way, the faulty slave axis can be safely decoupled to safely prevent the entire axis combination from entering the error state.

### Couple slave to actual values if not enabled

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Couple slave to actual values if not enabled | FALSE | | B | |
| | Velocity Window | 1.0 | | F | mm/s |
| | Filter Time for Velocity Window | 0.01 | | F | s |

`FALSE`: Not coupled.

`TRUE`: Coupled. The slave axis follows the master actual position while and also when the master is disabled.

### Velocity Window und Filter Time for Velocity Window

The coupled slave axis follows the master axis within the `Velocity Window`. If velocity deviations exceed `Filter Time for Velocity Window` beyond `Velocity Window`, an error is output.

### Allow motion commands

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Allow motion commands to slave axis | TRUE | | B | |
| | Allow motion commands to external setpoint axis | FALSE | | B | |

### Allow motion commands to slave axis

In general terms, an axis is in PTP mode all the time. This is about indirectly converting a slave axis into a master axis. Thus it is implicitly decoupled without the need to call on `MC_GearOut` from the PLC code.

`TRUE`: A PTP command can be triggered to the slave axis without first setting the axis to PTP mode.

`FALSE`: Before a PTP command can be triggered to the slave axis, the slave axis must be set to PTP mode.

### Allow motion commands to external setpoint axis

`FALSE`: Before a PTP command can be triggered to the external setpoint axis, the external setpoint axis must be set to PTP mode.

`TRUE`: A PTP command can be triggered to the external setpoint axis without first setting the axis to PTP mode.

### Dead Time Compensation (Delay Velo and Position)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Dead Time Compensation (Delay Velo and Position) | 0.0 | | F | s |

This parameter is only present for compatibility reasons. Do not use it on new projects.

### Data Persistence

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Data Persistence | FALSE | | B | |

Data Persistence is used for special encoder problems.

FALSE: Data Persistence is not enabled.

TRUE: Data Persistence is enabled.

## 7.2 Axes | Axis 1 | Enc

**Encoder**

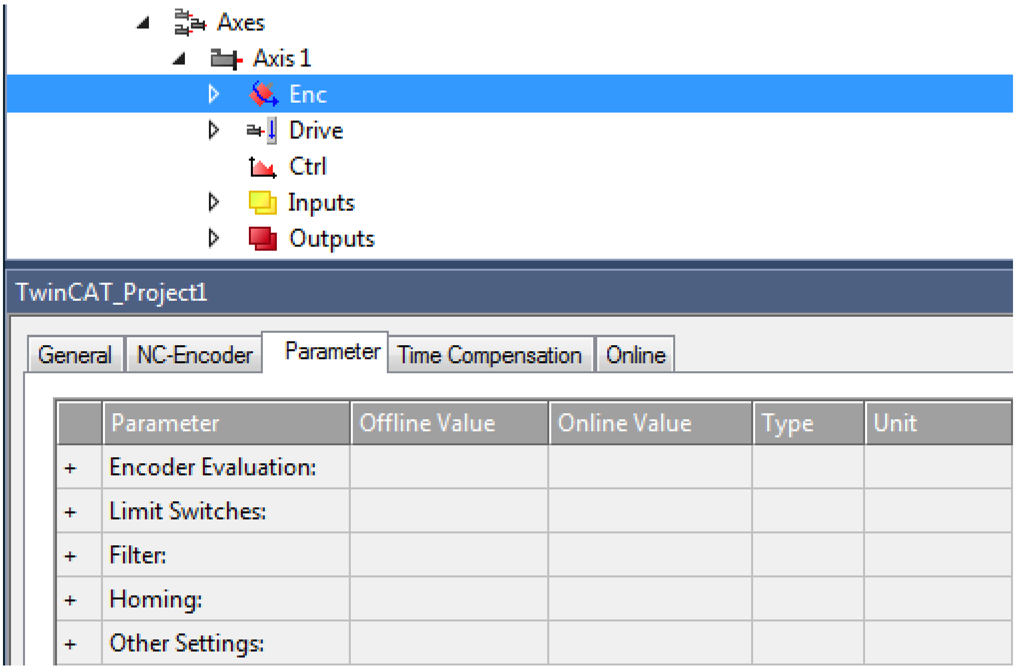
Encoders generally record a physical quantity and provide parameters for e.g. scaling. In most cases, a position is read in and processed here, and velocity, acceleration and jerk are determined by derivation from the position.

Since actual values often fluctuate strongly, a filter is also available for each individual variable.

A wide variety of encoders versions is supported. Both absolute and incremental encoders are available. Furthermore, there are simulation encoders, as well as special encoders for force determination, for example.

If you use an encoder axis, you can also use it to read in a different physical quantity and use the existing parameters for scaling, for example.

The encoder parameters include the scaling, zero offset shift and the modulo factor. There are also encoder parameters for software end positions as well as for homing.




Under "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc" the Parameter tab shows the parameter groups

- Encoder Evaluation,
- Limit Switches,
- Filter,
- Homing,
- Other Settings.

### 7.2.1 Encoder Evaluation

**Invert Encoder Counting Direction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Invert Encoder Counting Direction | FALSE | | B | |

If set to TRUE, the Invert Encoder Counting Direction parameter reverses the encoder counting direction.

- FALSE: The polarity of the axis movement agrees with the counting direction of the acquisition hardware.
- TRUE: The polarity of the axis movement is in the opposite sense to that of the counting direction of the acquisition hardware.

**⚠ WARNING**

**Risk of unexpected movements**

If the counting direction of the encoder and the motor polarity do not match, the axis will perform unexpected movements.

### Scaling Factor Numerator and Scaling Factor Denominator (default: 1.0)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Scaling Factor Numerator | 0.0001 | | F | mm/INC |
| | Scaling Factor Denominator (default: 1.0) | 1.0 | | F | |

The scaling factor is divided into a numerator and a denominator. It is used to convert the path increments into axis positions or to calculate a user unit from encoder increments.

### Position Bias

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Position Bias | 0.0 | | F | mm |

The zero offset shift is an offset that absolute encoders use to align their position in the machine coordinate system and is thus used to define the machine-related zero point. The offset value of the zero offset shift is added to the encoder position to determine the axis position.

### Modulo Factor (e.g. 360.0°)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Modulo Factor (e.g. 360.0°) | 360.0 | | F | mm |
| | Tolerance Window for Modulo Start | 0.0 | | F | mm |

The value of the Modulo Factor (e.g. 360.0°) is the value for calculating modulo turns and modulo positions from the absolute axis position. For rotary axes, the modulo factor is the "path distance" that forms one revolution. If the actual value is acquired, for instance, in degrees, 360.0° should be entered here.

### Tolerance Window for Modulo Start

The Tolerance Window for Modulo Start should be large enough to allow backward movement when a modulo forward command is executed in the window area. Conversely, the Tolerance Window for Modulo Start should be large enough to allow forward movement when a modulo backward command is executed in the window area.

### Further Information:

- Notes on modulo positioning (PLC library Tc2_MC2)

### Encoder Mask (maximum encoder value)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Encoder Mask (maximum encoder value) | 0x0036EE7F | | D | |

The encoder mask defines the number of allowed increments until the feedback value overflows. In most cases, the increments are not the real encoder increments due to internal encoder overflow calculation on the connected hardware.

### Encoder Sub Mask (absolute range maximum value)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Encoder Sub Mask (absolute range maximum value) | 0x000FFFFF | | D | |

The encoder sub mask defines the number of increments per motor revolution.

### Noise level of simulation encoder

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Noise level of simulation encoder | 0.0 | | F | |

This parameter creates artificial noise for the simulation axis by setting a level to make it appear more realistic.

### Reference System

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Encoder Evaluation: | | | | |
| | Reference System | 'INCREMENTAL' | | E | |

Use the Reference System parameter to select how the encoder values should be interpreted:

`INCREMENTAL`

INCREMENTAL mode does not use absolute characteristics of the physical feedback system, but can be used with both incremental and absolute encoder systems. After starting the control, homing is required to initialize the actual position of the axis.

`INCREMENTAL (singleturn absolute)`

This mode extends the INCREMENTAL mode and can be used instead.

If in this mode the retain data of the NC-SAF task and the axis parameter *Data Persistence* are activated, the system starts up with the last stored position. If a single-turn absolute encoder is used at the same time, a deviation of up to half an encoder revolution is also taken into account and the axis can be operated within these limits without homing. The absolute range of the encoder system is defined by the parameter *Encoder Sub Mask*.

`ABSOLUTE`

The ABSOLUTE mode can be used with multi-turn absolute encoder systems. The prerequisite is that the encoder system or the motor is installed in such a way that there can't be overflow of the position feedback within the travel path. An overflow would be acknowledged with a runtime error.

`ABSOLUTE MULTITURN RANGE (with single overflow)`

The ABSOLUTE MULTITURN RANGE mode can also be used with multi-turn absolute encoder systems. The encoder system can be installed in any way, so that a maximum of one overflow may occur in the travel path. To ensure correct evaluation, the two end positions must be parameterized.

The total travel path must be less than the absolute range of the encoder. The absolute range of the encoder system is defined by the parameter *Encoder Mask*.

`ABSOLUTE SINGLETURN RANGE (with single overflow)`

The ABSOLUTE SINGLETURN RANGE mode can be used with single-turn absolute encoder systems. The encoder system can be installed in such a way that an overflow is located in the travel path. To ensure correct evaluation, the two end positions must be parameterized.

The total travel path must be less than the absolute range of the encoder, i.e. less than one encoder revolution. The absolute range of the encoder system is defined by the parameter *Encoder Sub Mask*.

`ABSOLUTE (modulo)`

The ABSOLUTE (modulo) mode can be used when the travel path of an axis exceeds the absolute range of the encoder system. For example, in the case of an endlessly running conveyor belt.

To initialize the position of the axis correctly at system start, the retain data of the NC-SAF task and the parameter *Data Persistence* of the axis must be activated. Furthermore, the parameter *Modulo Factor* of the encoder of the NC axis must be parameterized correctly. The axis position is initialized to this module range after the system startup.

When switched off, the axis may be mechanically displaced by a maximum of half of the absolute range. This maximum distance is defined by the *Encoder Sub Mask* parameter. The *Encoder Sub Mask* is typically set to the increments of one motor revolution, but can also be set to the entire absolute range here.

## 7.2.2 Limit Switches

The Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Limit Switches parameters can be set under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

**Soft Position Limit Minimum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Minimum Monitoring | FALSE | | B | |
| | Minimum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Minimum Monitoring is not enabled.

`TRUE`: Soft Position Limit Minimum Monitoring is enabled.

**Minimum Position**

Position lower limit for the axis, which must not be violated when Soft Position Limit Minimum Monitoring is enabled. Commands that violate this lower limit will be rejected.

**Soft Position Limit Maximum Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Maximum Monitoring | FALSE | | B | |
| | Maximum Position | 0.0 | | F | mm |

`FALSE`: Soft Position Limit Maximum Monitoring is not enabled.

`TRUE`: Soft Position Limit Maximum Monitoring is enabled.

**Maximum Position**

Position upper limit for the axis, which must not be violated when Soft Position Limit Maximum Monitoring is enabled. Commands that violate this upper limit will be rejected.

## 7.2.3 Filter

**Filter Time for Actual Position (P-T1)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Position (P-T1) | 0.0 | | F | s |

Filter time for the PT1 filtering of the actual position.

In the case of very noisy encoder signals, it may therefore be necessary to filter the position value with a low filter time.

Filtering the position leads to a position distortion.

**Filter Time for Actual Velocity (P-T1)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Velocity (P-T1) | 0.01 | | F | s |

Filter time for the PT1 filtering of the actual velocity.

**Filter Time for Actual Acceleration (P-T1)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Filter: | | | | |
| | Filter Time for Actual Acceleration (P-T1) | 0.1 | | F | s |

Filter time for the PT1 filtering of the actual acceleration.

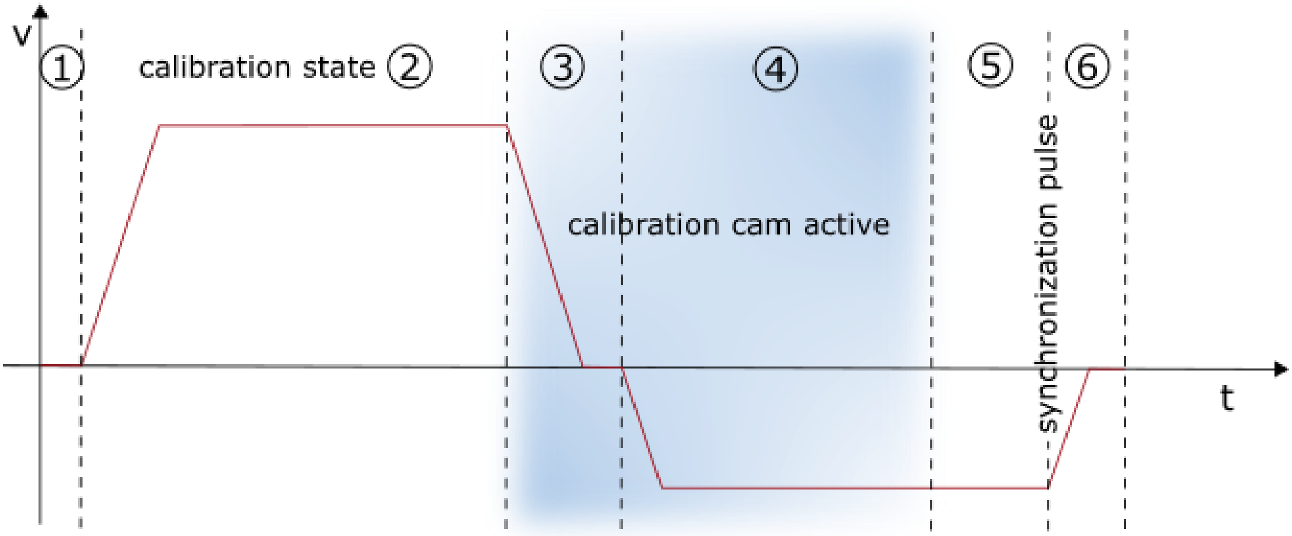
**Further information: PT1 filter**

See section PT1 Filter [▶ 117] for further information on the PT1 filter.

## 7.2.4 Homing

Homing refers to an axis initialization run during which the correct actual position is determined by means of a reference signal. This can be NC-guided, as described here:

1. The axis is at a random position.
2. The axis moves in the parameterized direction and searches for the reference switch (referencing cam/calibration cam).
3. The axis reverses as soon as the referencing cam is detected.
4. The axis moves away from the referencing cam and detects the falling edge of the cam signal.
5. The axis continues to move and searches for the sync pulse (latch signal or zero pulse of the encoder).
6. The reference position is set and the axis stops. The standstill position of the axis deviates slightly from the reference position.




Alternatively, there is also drive-guided homing, which is performed independently by a suitable drive without the influence of the control system. For details, please refer to the documentation of the drive system used.

**Invert Direction for Homing Sensor Search**

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Invert Direction for Homing Sensor Search | FALSE | | B | |

For the default homing sequence the direction for calibration cam search can be inverted.

- `FALSE`: cam is looked for in the direction of positive movement.
- `TRUE`: cam is looked for in the direction of negative movement.

**Invert Direction for Sync Impuls Search**

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Invert Direction for Sync Impuls Search | TRUE | | B | |

For the default homing sequence the direction for sync pulse search can be inverted.

- `FALSE`: sync pulse is looked for in the direction of positive movement.
- `TRUE`: sync pulse is looked for in the direction of negative movement.

### Home Position (Calibration Value)

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Home Position (Calibration Value) | 0.0 | | F | mm |

The Home Position is the position value that will be assigned to the axis when the sync pulse appears within the homing. This is done exactly with the sync pulse while in motion. Since the axis then stops, the standstill position deviates from the parameterized value. The referencing is nevertheless exact.

### Reference Mode (Sync condition)

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |
| | Reference Mode (Sync condition) | 'Default' | | E | |

The parameter Reference Mode offers the opportunity to select which signal shall be used for sync impulse search:

- `Default / Homing Sensor Only (PLC cam or digital input 1..8)`
  - The simplest way of referencing an axis where only the cam signal is evaluated. Another sync signal is not used, so this setting should only be selected if the axis does not offer more extensive options.
- `Hardware Sync (Feedback reference pulse)`
  - Some encoder systems provide a sync pulse per revolution (zero track) in addition to the count. The homing precision can be enhanced by selecting this mode, if the encoder evaluation logic is able to pick up the sync pulse. The precision is comparable with Software Sync. Hardware Sync mode may require parameterization or special wiring of the drive or encoder system.
- `Hardware Latch 1 (pos. edge), Hardware Latch 1 (neg. edge)`
  - Hardware latch responds to an external latch signal in addition to the cam signal. For this purpose, the latch mechanism must be configured in the drive and the latch input must be wired.
- `Software Sync`
  - Software Sync mode enhances the homing precision by additionally detecting the encoder count overflow after an encoder or motor revolution, after the reference cam signal has been detected. This mode requires a part-absolute encoder (e.g. resolver) with constant overflow interval relative to the referencing cam. Overflow detection is parameterized via the Reference Mask parameter.
- `Application (PLC code)`
  - In this case, the entire homing procedure must be programmed in the PLC:
    In this case, the NC sets bit 23 in the "NcApplicationRequest" status, which is polled by the PLC. Via MC_ReadApplicationRequest the request of the NC can be determined in the PLC. So far, only 0 = NONE and 1 = HOMING are supported (see 0x33 at "Index offset" specification for axis parameters). In the case of homing, the PLC can perform the sequence itself by moving the axis.

### Homing Sensor Source

| | Parameter | Offline Value | Online Val... | Type | Unit |
|---|---|---|---|---|---|
| - | Homing: | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Homing Sensor Source | 'Default: PLC Cam (MC_Home)' | | E | |

- `Default PLC Cam (MC_Home)`
  - By default the homing sensor signal comes from the PLC with the function block MC_Home with the parameter bCalibrationCam, which is automatically linked to the PlcToNc.ControlWordD.7.
- `Digital Input [1-8]`
  - Alternatively, a digital input linked to the NC process can be used. For this purpose, a general Drive Status Byte with 8 digital inputs is defined in the process image (Drive->Inputs->nState8), which can serve as a signal source for the homing sensor. A digital input to be used must therefore be mapped manually to the desired position in this byte.
  - ***Notice*** **The digital inputs 1 and 2 may differ depending on the hardware used. For the MDP703/733 hardware (e.g. EL7031, EL7041, EL7201, EL7411) the direct digital inputs E1 and E2 of the terminal are used instead, which are located in the Drive.nState2 byte of the terminal at bit position 3 (E1) and 4 (E2). The lower two bits of Drive.nState8 are not assigned in this case.**

**Further information**

- Homing on the AX5000
- MC_Home in the Tc2_MC2

## 7.2.5 Other Settings

**Encoder Mode**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Encoder Mode | 'POSVELO' | | E | |

The parameter Encoder Mode offers the possibility to select which values should be calculated from the encoder position:

- `'POS'`: The actual position is determined.
- `'POSVELO'`: The actual position and the actual velocity are determined.
- `'POSVELOACC'`: The actual position, the actual velocity and the actual acceleration are determined.

**Position Correction**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Position Correction | FALSE | | B | |
| | Filter Time Position Correction (P-T1) | 0.0 | | F | s |

The Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Parameter.

Alternatively, the Position Correction can be activated under MOTION | NC-Task 1 SAF | Axes | Axis 1 | Enc | Parameter.

Similar for other identifiers.

`FALSE`: The Position Correction is disabled.

`TRUE`: The Position Correction is enabled.

The variable `axis.PlcToNc.PositionCorrection` is of data type `LREAL` and belongs to the structure `PLCTONC_AXIS_REF`. If Position Correction is enabled, this variable adds an additional offset to the target position. It should be noted that this correction does not affect the software end positions.

**Filter Time Position Correction (P-T1)**

The filter time for the PT-1 filter, which filters fluctuations within the Actual Position Correction with the filter time set here. See section PT1 Filter [▶ 117] for further information on the PT1 filter.

**See also:**

**`MC_PositionCorrectionLimiter`**

- TwinCAT 3 PLC Lib: Tc2_MC2

The function block `MC_PositionCorrectionLimiter` adds the correction value `PositionCorrectionValue` to the actual position value of the axis. Depending on the `CorrectionMode` the position correction value is either written directly or filtered.

> To use the `MC_PositionCorrectionLimiter` function block successfully the Position Correction has to be enabled by setting the parameter Position Correction `TRUE`.

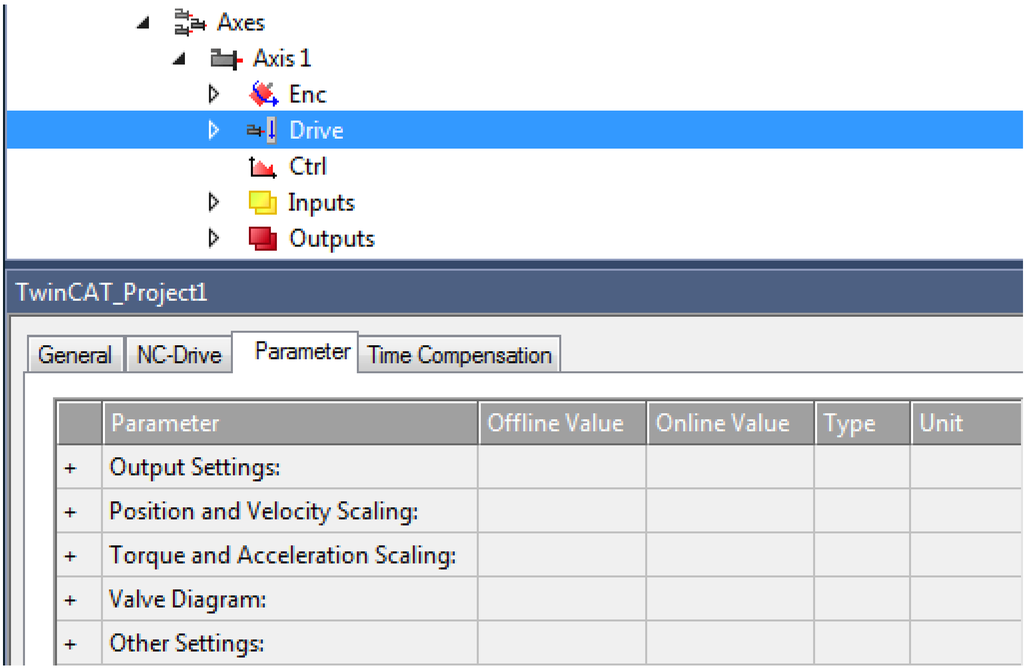
## 7.3 Axes | Axis 1 | Drive

**Drive**

The drive transfers the output voltage to the motor's power section. There is a wide range of supported drive variants: For example, servo drives, high speed drives, low speed drives, stepper motor drives. The drive parameters include the motor polarity and the reference velocity.

**Analog drive**

Within this context "analog" does not mean that the velocity is represented by a voltage (e.g. ±10 V) or a current (e.g. ±20 mA), but rather that the axis can be adjusted over an effectively continuous range of values. This kind of adjustment is also possible for drives with digital interfaces such as the BISSI terminals of type KL5051. When using such a digital interface, a velocity can be set in the same way as when controlling an analog value, even though it is transported as digital information.




Under "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Drive" the Parameter tab shows the parameter groups

- Output Settings,
- Position and Velocity Scaling,

- Torque and Acceleration Scaling,
- Valve Diagram,
- Other Settings.

Furthermore, the parameter groups

- Optional Position Command Output Smoothing Filter,
- Sercos Behavior

can be displayed.

## 7.3.1 Output Settings

**Invert Motor Polarity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Output Settings: | | | | |
| | Invert Motor Polarity | FALSE | | B | |

If set to TRUE, the Invert Motor Polarity parameter reverses the polarity of the motor and thus the direction of motor rotation.

- FALSE: The axis moves in the direction of larger positions when the drive is positively controlled.
- TRUE: The axis moves in the direction of smaller positions when the drive is positively controlled.

⚠ WARNING

**Risk of unexpected movements**

If the counting direction of the encoder and the motor polarity do not match, the axis will perform unexpected movements.

**Reference Velocity**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Output Settings: | | | | |
| | Reference Velocity | 2200.0 | | F | mm/s |
| | at Output Ratio [0.0 ... 1.0] | 1.0 | | F | |

For drives that are not directly controlled by a digital velocity value, e.g. a voltage or current interface, the Reference Velocity is used to scale the drive output. The Reference Velocity is at the same time an upper velocity limit that cannot be exceeded in addition to the maximum velocity. For all drive types, the Reference Velocity must be set greater than or equal to the maximum velocity.

(Strictly speaking, the velocity upper limit is the Reference Velocity divided by the Output Ratio, if an Output Ratio smaller than 1.0 is parameterized)

**at Output Ratio [0.0..1.0]**

*Proportional ratio for analog control*

The Reference Velocity normally refers to the maximum possible output, e.g. 10 V. The Output Ratio in this case is 1.0 or 100 %. With the Output Ratio factor it is possible to parameterize the reference velocity for another reference point, e.g. at 80 %.

*Reference Velocity and Maximum Velocity*

The Reference Velocity divided by the Output Ratio determines the maximum achievable velocity. A control component is required for controlling the axis, therefore the parameterized Maximum Velocity must be set 10% smaller, for example. For drives that are controlled with a digital velocity value (CoE, SoE), no reserve is necessary and both velocities can be equal.

## 7.3.2 Position and Velocity Scaling

**Output Scaling Factor**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Output Scaling Factor (Position) | 1.0 | | F | |
| | Output Scaling Factor (Velocity) | 6.0 | | F | |

**Output Scaling Factor (Position)**

Usually the drive parameter Output Scaling Factor (Position) does not cause any effect. To avoid that a future use of the parameter changes the behavior, the user should basically not change the default value `1.0` of this parameter.

The position output scaling is already set by the encoder input scaling. Regularly, the position input scaling corresponds to the position output scaling for a drive. For this reason, the drive parameter Output Scaling Factor (Position) is currently not evaluated. Instead, scaling is performed exclusively by the encoder parameters Scaling Factor Numerator and Scaling Factor Denominator, which must be adjusted for accurate scaling.

**Output Scaling Factor (Velocity)**

If a drive controller operates in velocity pre-control mode, the NC output value must be scaled. There are two ways for this scaling, depending on the type of drive controller used.

1. Analog drive controllers, e.g. supplied by a ±10 V terminal:

   This type of drive controller is scaled by applying the parameter Reference Velocity.
2. Digital drive controllers to which an absolute digital velocity command value is transferred, e.g. *CANopen DS402*:

   This type of drive controller is scaled by applying the parameter Output Scaling Factor (Velocity).

Below is a tabular comparison of when to apply the parameter Reference Velocity or the parameter Output Scaling Factor (Velocity).

| Drive type | Scaling with Reference Velocity | Scaling with Output Scaling Factor (Velocity) |
|---|---|---|
| M2400_DAC1 | x | |
| M2400_DAC2 | x | |
| M2400_DAC3 | x | |
| M2400_DAC4 | x | |
| KL4XXX | x | |
| KL4XXX_NONLINEAR | x | |
| TWOSPEED | x | |
| STEPPER | x | |
| SERCOS | | x |
| KL5051 | x | |
| AX2000_B200 | | x |
| SIMO611U | | x |
| UNIVERSAL | x | |
| NCBACKPLANE | x | |
| CANOPEN_LENZE | | x |
| DS402_MDP742 | | x |
| AX2000_B900 | x | |
| AX2000_B310 | x | |

| | | |
|---|---|---|
| AX2000_B100 | x | |
| KL2531 | | x |
| KL2532 | | x |
| TCOM_DRV | | x |
| MDP_733 | | x |
| MDP_703 | x | (x)* |

*Also possible, but should be left at the value `1.0`.

**Output Delay (Velocity)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Output Delay (Velocity) | 0.0 | | F | s |

The output of the velocity value can be delayed by the time Output Delay (Velocity).

**Minimum /Maximum Drive Output Limitation [-1.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position and Velocity Scaling: | | | | |
| | Minimum Drive Output Limitation [-1.0 ... 1.0] | -1.0 | | F | |
| | Maximum Drive Output Limitation [-1.0 ... 1.0] | 1.0 | | F | |

**Minimum Drive Output Limitation [-1.0 … 1.0]**

To limit the velocity and thus protect the hardware, a lower limit can be set for driving the axis. If only part of the output data type is valid, the minimum output value must be limited. Minimum Drive Output Limitation is a directional limitation of the total output. The value `1.0` corresponds to an unlimited output of 100 %. As a rule, the use of this parameter concerns a velocity output signal for the drive in connection with position control. In exceptional cases, the application of this parameter may concern a torque value or a current value.

**Maximum Drive Output Limitation [-1.0 … 1.0]**

To limit the velocity and thus protect the hardware, an upper output limit can be set for driving the axis. If only part of the output data type is valid, the maximum output value must be limited. The Maximum Drive Output Limitation is a direction-dependent limitation of the total output. The value `1.0` corresponds to an unlimited output of 100 %. As a rule, the use of this parameter concerns a velocity output signal for the drive in connection with position control. In exceptional cases, the application of this parameter may concern a torque value or a current value.

## 7.3.3 Torque and Acceleration Scaling

This parameter group generates an optional acceleration as servo control to intervene before a lag error occurs. It can, for example, define a torque value.

**Input Scaling Factor (Actual Torque)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input Scaling Factor (Actual Torque) | 0.0 | | F | |

The gain factor for the optional servo control.

### Input P-T1 Filter Time (Actual Torque)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input P-T1 Filter Time (Actual Torque) | 0.0 | | F | s |

Time for the P-T1 filter. This time comes as input to the P-T1 filter.

### Input P-T1 Filter (Actual Torque Derivative)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Input P-T1 Filter (Actual Torque Derivative) | 0.0 | | F | s |

Derivative of the actual torque to be scaled. This derivative comes as input to the P-T1 filter.

See section PT1 Filter [▸ 117] for further information on the PT1 filter.

### Output Scaling Factor (Torque)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Scaling Factor (Torque) | 0.0 | | F | |

Sometimes an optional output scaling for the torque is necessary.

### Output Delay (Torque)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Delay (Torque) | 0.0 | | F | s |

The output of the torque value can be delayed by the time Output Delay (Torque).

### Output Scaling Factor (Acceleration)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Scaling Factor (Acceleration) | 0.0 | | F | |

Sometimes an optional output scaling is necessary for acceleration. (e.g. for the NC acceleration pre-control. Further information about acceleration pre-control can be found in the section Acceleration Feedforward [▶ 113].)

**Output Delay (Acceleration)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Torque and Acceleration Scaling: | | | | |
| | Output Delay (Acceleration) | 0.0 | | F | s |

The output of the acceleration value can be delayed by the time Output Delay (Acceleration).

## 7.3.4 Valve Diagram

Hydraulic axes have a non-linear characteristic curve. Valve characteristic curves are used to linearize the controlled system, which are constructed in a similar way to cam plates. For numerical control, for example, a hydraulic axis can be treated similarly to a servo axis.

**Valve Diagram: Table Id**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Table Id | 0 | | D | |

Each valve characteristic table has its own ID. This is assigned to each table consecutively. The table ID uniquely identifies the cam plate in the TwinCAT system.

**Valve Diagram: Interpolation type**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Interpolation type | 'LINEAR' | | E | |

Two types of interpolation

- 'LINEAR' or
- 'SPLINE'

can be selected to connect the discrete number pairs `(x, y)` in the valve characteristic table.

**Valve Diagram: Output offset [-1.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Valve Diagram: | | | | |
| | Valve Diagram: Output offset [-1.0 ... 1.0] | 0.0 | | F | |

This parameter allows you to adjust the zero crossing of the valve characteristic curve through the origin.

**More Information**

TE1500 Valve Diagram Editor.
Valve characteristic curves can be created with the TE1500 Valve Diagram Editor.
https://infosys.beckhoff.com/content/1033/te1500_tc3_valve_diagram_editor/index.html?id=5755459939328551364

## 7.3.5 Optional Position Command Output Smoothing Filter

In some applications, smoothing of the position output to the drive is used to reduce vibration of machine parts. This smoothing forms an additional filter that should be used with caution. This filter cannot be compensated by a static dead time.

**Smoothing Filter Type and Smoothing Filter Time**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Optional Position Command Output Smoothing Filter: | | | | |
| | Smoothing Filter Type | 'OFF (default)' | | E | |
| | Smoothing Filter Time | 0.01 | | F | s |

If

- 'OFF (default)'

is selected, smoothing of the position output is disabled.

As filter types

- `'Moving Average'` or
- `'P-Tn'`

can be selected. Both modes refer to the position setpoint.

`'Moving Average'`: As an output, the Moving Average Filter generates an average value of a series of position setpoints that comes to the filter as an input. Here, the Smoothing Filter Time prescribes the time interval over which the average extends. The values of the input are generated by position setpoint generation. Thus, the frequency of value generation is determined by the cycle time of the task for generating the setpoints.
For example, if a new value is generated every 1 ms and the Smoothing Filter Time is set to e.g. 20 ms, an average of 20 values is determined. The influence of a setpoint is perceptible for 20 ms.

`'P-Tn'`: As an output, the P-Tn filter generates an average value of a series of position setpoints that comes to the filter as an input. Here the Smoothing Filter Time prescribes the time constant of the P-Tn filter. The values of the input are generated by position setpoint generation. Thus, the frequency of value generation is determined by the cycle time of the task for generating the setpoints.

**Smoothing Filter Order (P-Tn only)**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Optional Position Command Output Smoothing Filter: | | | | |
| | Smoothing Filter Order (P-Tn only) | 2 | | D | |

The order of the P-Tn smoothing filter used.

## 7.3.6 Sercos Behavior

### C1D Error Tolerance

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Sercos Behavior: | | | | |
| | C1D Error Tolerance | 'STANDARD' | | E | |

Some hardware drive errors can be parameterized on the drive side so that an error is signaled but the drive continues to follow the setpoints.

If the parameter *C1D Error Tolerance* is set to 'IGNORE SELECTED ERRORS', these errors do not lead to an immediate stop of the axis by the NC.

The user recognizes the drive error by the flag *DriveDeviceError* in the *StateDWord* of the NC axis and must stop the axis using the suitable PLC code.

## 7.3.7 Other Settings

### Drive Mode

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Drive Mode | 'STANDARD' | | E | |

`'STANDARD'`: There is currently only one drive mode, and that is the `'STANDARD'` mode. In principle, other modes are conceivable in the execution of a more open system.

### Drift Compensation (DAC-Offset)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Drift Compensation (DAC-Offset) | 0.0 | | F | mm/s |

This value is added to the drive control level. In this way, a constant offset can be added to the output, for example to compensate for zero point deviations in analog drives.
See section Automatic DAC offset adjustment [▶ 113].

### Following Error Calculation (Following Error)

The following error corresponds to the position lag, also referred to as position lag value.

Position lag value = actual position - current set position.

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Following Error Calculation | 'Intern' | | E | |

The following error can be calculated with the NC or in the hardware drive. For all hardware drives working in position control mode, the calculation should be done externally in the hardware drive.

`'Intern'`: The following error is calculated internally in TwinCAT.

`'Extern'`: The following error is calculated externally in the drive controller.

**Error Tolerance**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Error Tolerance (NC error handling) | 'STANDARD' | | E | |

Some hardware drive errors can be parameterized on the drive side so that an error is signaled but the drive continues to follow the setpoints.

If the parameter **Error Tolerance** is set to `'IGNORE SELECTED ERRORS'`, these errors do not lead to an immediate stop of the axis by the NC.

The user recognizes the drive error by the flag `DriveDeviceError` in `StateDWord` of the NC axis and must stop the axis using the suitable PLC code.

## 7.4 Axes | Axis 1 | Ctrl

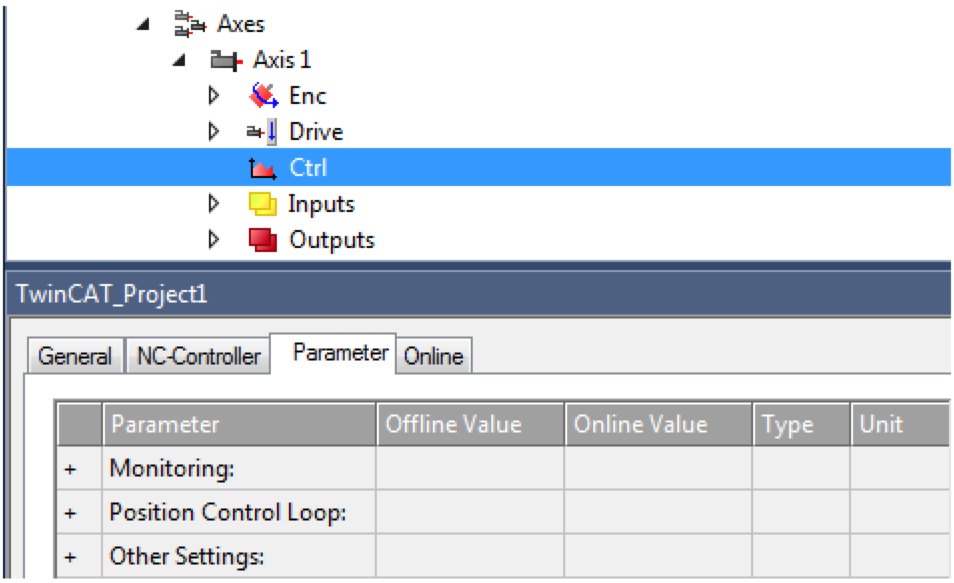
**Controller**

The purpose of the controller is to operate on the basis of setpoint velocities or other setpoint magnitudes (e.g. acceleration) and on the basis of setpoint magnitude differences as following errors. Controller operation is performed in a way to keep following errors as small as possible and to prevent any axis to undergo any overshots in position or velocity.

Supported is a wide variety of controller versions: Servo position controllers and special controllers for particular axis types.

**Controller Types**

- *Position Controllers*: Control the actual position to follow the setpoint position as precisely as possible. Position controller `P`, following error proportional controller, position controller with two `P` constants, following error proportional controller with different constants for the stationary state and for movement, position PID-T1 controller with proportional acceleration feed forward.
- *Controllers for Axes*: Servo axes, stepper motors, low-speed axes, high-speed axes.




Under "MOTION | NC-Task 1 SAF | Axes | Axis 1 | Ctrl" the Parameter tab shows the parameter groups

- Monitoring,
- Position Control Loop,
- Other Settings.

Furthermore, the parameter groups

- Velocity Control Loop,
- Observer

can be displayed.

## Controller Types and their Parameters

The following table lists the available controller types and shows what parameters configure (“x”) or do not configure (“-”) each one of them.

| | Position controller P | Position controller with two P constants (with Ka) | Position controller PID (with Ka) | Position P and velocity PID controller (Torque) | Position P and velocity PI controller with Observer (Torque) | Two speed controller | Stepper controller | SERCOS controller (Position by SERCOS) |
|---|---|---|---|---|---|---|---|---|
| **Monitoring:** | x | x | x | x | x | x | x | x |
| Position Lag Monitoring | x | x | x | x | x | x | x | x |
| Maximum Position Lag Value | x | x | x | x | x | x | x | x |
| Maximum Position Lag Filter Time | x | x | x | x | x | x | x | x |
| **Position Control Loop:** | x | x | x | x | x | x | x | x |
| Position control: Dead Band Position Deviation | - | x | x | - | x | - | - | - |
| Position control: Proportional Factor Kv | x | - | x | x | x | - | - | - |
| Position control: Proportional Factor Kv (standstill) | - | x | - | - | - | - | - | - |
| Position control: Proportional Factor Kv (moving) | - | x | - | - | - | - | - | - |
| Position control: Integral Action Time Tn | - | - | x | - | - | - | - | - |
| Position control: Derivative Action Time Tv | - | - | x | - | - | - | - | - |
| Position control: Damping Time Td | - | - | x | - | - | - | - | - |
| Position control: Min./max. limitation I-Part [0.0 … 1.0] | - | - | x | - | - | - | - | - |
| Position control: Min./max. limitation D-Part [0.0 … 1.0] | - | - | x | - | - | - | - | - |
| Disable I-Part during active positioning | - | - | x | - | - | - | - | - |
| Position control: Velocity threshold V dyn [0.0 … 1.0] | - | x | - | - | - | - | - | - |
| Feedforward Acceleration: Proportional Factor Ka | - | x | x | x | x | - | - | - |
| Feedforward Velocity: Pre-Control Weighting [0.0 … 1.0] | x | x | x | x | x | x | x | x |
| **Velocity Control Loop:** | - | - | - | x | x | - | - | - |
| Velocity control: Proportional Factor Kv | - | - | - | x | x | - | - | - |
| Velocity control: Integral Action Time Tn | - | - | - | x | x | - | - | - |
| Velocity control: Derivative Action Time Tv | - | - | - | x | - | - | - | - |
| Velocity control: Damping Time Td | - | - | - | x | - | - | - | - |
| Velocity control: Min./max. limitation D-Part [0.0 … 1.0] | - | - | - | x | - | - | - | - |
| Velocity control: Min./max. limitation I-Part [0.0 … 1.0] | - | - | - | x | x | - | - | - |
| **Observer:** | - | - | - | - | x | - | - | - |
| Velocity Filter: Time Constant T | - | - | - | - | x | - | - | - |
| Observer Mode | - | - | - | - | x | - | - | - |
| Motor: Torque Constant Kt | - | - | - | - | x | - | - | - |
| Motor: Moment of Inertia Jm | - | - | - | - | x | - | - | - |
| Bandwidth f0 | - | - | - | - | x | - | - | - |
| Correction Factor Kc | - | - | - | - | x | - | - | - |
| **Other Settings:** | x | x | x | x | x | x | x | x |
| Controller Mode | x | x | x | x | x | x | x | x |
| Auto Offset | x | x | - | - | - | - | - | - |
| Offset Timer | x | x | - | - | - | - | - | - |
| Offset Limit (of Calibration Velocity) | x | x | - | - | - | - | - | - |
| Slave coupling control: Proportional Factor Kcp | x | x | x | - | - | - | - | - |
| Controller Outputlimit [0.0 … 1.0] | x | x | x | x | x | - | - | - |

## 7.4.1 Monitoring

**Position Lag Monitoring**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Monitoring: | | | | |
| | Position Lag Monitoring | TRUE | | B | |
| | Maximum Position Lag Value | 5.0 | | F | mm |
| | Maximum Position Lag Filter Time | 0.02 | | F | s |

The position lag monitoring monitors the position lag value. If the parameterized limits for position and time are exceeded, a runtime error is output.

Position lag value = actual position - current set position.

`TRUE`: Position Lag Monitoring is enabled.

`FALSE`: Position Lag Monitoring is not enabled.

**Maximum Position Lag Value and Maximum Position Lag Filter Time**

The Maximum Position Lag Value is the upper limit, which must not be exceeded for longer than the Maximum Position Lag Filter Time. Otherwise the NC axis is stopped immediately by direct shutdown and set to the logical state "Error", whereby the error `0x4550` is output.

## 7.4.2 Position Control Loop

A set velocity is transmitted to the drive controller via the fieldbus. The actual position of the axis is transmitted to TwinCAT as feedback via the fieldbus, creating a position control loop. The position control can be carried out using this circuit.

The position controller of the TwinCAT axis is only active if the drive controller is operated via the set velocity. If the drive controller is controlled via the set position, the position control loop in the drive controller is closed and must be parameterized there. The control parameters described here are then not effective.

**Position control: Dead Band Position Deviation**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Dead Band Position Deviation | 0.0 | | F | mm |

The dead band position deviation defines a range in which the controller is inactive. This parameter is a symmetrical window based on the assumption that the position lag error with respect to the position control (position control deviation) within this window is zero. Position control is therefore disabled within the window.

This functionality is required because certain axes and mechanical systems may cause a stationary oscillation around the target position. The parameter can be used to enforce "calm". The parameterizable deviation around the exact target position is accepted.

There are axes that are not capable of position control or have insufficient holding torque around standstill, so that a certain inaccuracy around the target position is accepted.

For other real effects, such as pronounced static friction (with tear-off torque) or pronounced slack, dead band position deviation can also be used to enforce "calm" for an accepted accuracy.

### Position control: Proportional Factor Kv

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Proportional Factor Kv | 1.0 | | F | mm/s/mm |

The Proportional Factor Kv is the proportional gain factor of the P part of the P controller. `Output velocity = feed forward velocity + Kv * following error`.

### Position control: Proportional Factor Kv (standstill)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Proportional Factor Kv (standstill) | 1.0 | | F | mm/s/mm |

The Proportional Factor Kv (standstill) `Kvs` is the proportional gain factor of the P part of the P controller when the axis is stationary. `Output velocity when stationary = feedforward velocity + Kvs * following error`.

### Position control: Proportional Factor Kv (moving)

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Proportional Factor Kv (moving) | 0.1 | | F | mm/s/mm |

The Proportional Factor Kv (moving) `Kvf` is the proportional gain factor of the P part of the P controller when the axis is moving. `Output velocity when moving = feedforward velocity + Kvf * following error`.

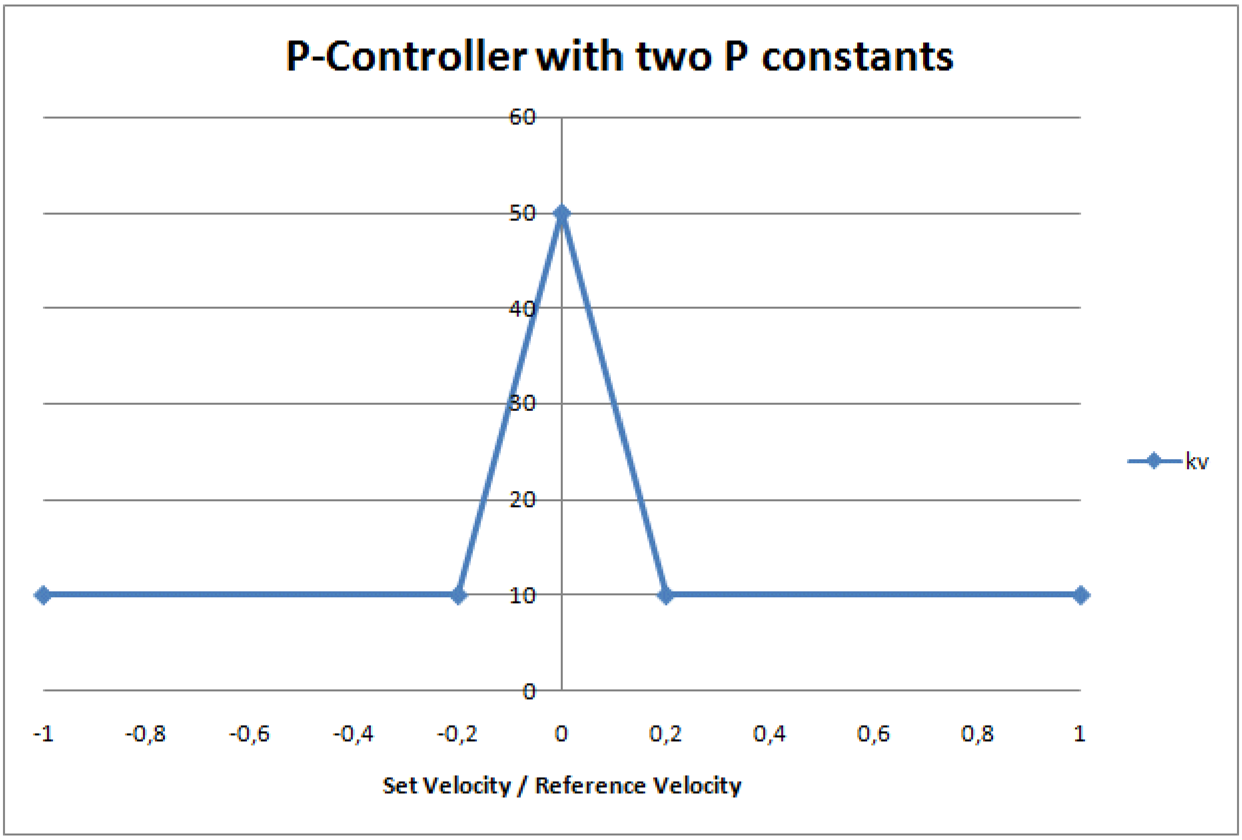
### Position control: Velocity threshold V dyn [0.0 … 1.0]

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Velocity threshold V dyn [0.0 ... 1.0] | 0.5 | | F | |

With a PP controller, two P constants

- Kv (standstill) `Kvs` and
- Kv (moving) `Kvf`

and a velocity threshold value $v_{dyn}$ define a function for a velocity-dependent control factor `kv`. If the value of the quotient "set velocity/reference velocity" is in the velocity threshold interval [- $v_{dyn}$ ... + $v_{dyn}$], the gain factor of the P controller is linearly interpolated from the proportional factor Kv (moving) `Kvf` to the gain factor Kv (standstill) `Kvs` with zero velocity. The relationships are shown in the figure below.




The example was parameterized with `Kvs = 50`, `Kvf = 10` and `v_dyn = 0.2`.

**Position control: Integral Action Time Tn**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Integral Action Time Tn | 0.0 | | F | s |

Integral action time `Tn` of the I part of the PID controller. Integration time.

**Position control: Derivative Action Time Tv**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Derivative Action Time Tv | 0.0 | | F | s |

Derivative action time `Tv` of the real D part (D-T1 component) of the PID controller.

**Position control: Damping Time Td**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Damping Time Td | 0.0 | | F | s |

Damping time `Td` of the real D part (D-T1 component) of the PID controller.

### Position control: Min./max. limitation I-Part [0.0 … 1.0]

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Min./max. limitation I-Part [0.0 ... 1.0] | 0.1 | | F | |

Limitation of the I part of the PID controller.

### Position control: Min./max. limitation D-Part [0.0 … 1.0]

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Position control: Min./max. limitation D-Part [0.0 ... 1.0] | 0.1 | | F | |

Limitation of the D part of the PID controller.

### Disable I-Part during active positioning

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Disable I-Part during active positioning | FALSE | | B | |

The I part of the PID controller can be disabled during active positioning.

### Feedforward Acceleration: Proportional Factor Ka

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Feedforward Acceleration: Proportional Factor Ka | 0.0 | | F | s |

The proportional factor `Ka` of the acceleration pre-control is the gain factor when a control with NC acceleration pre-control is used. `Output velocity component = Ka * setpoint acceleration`.

Further information about acceleration pre-control can be found in the section Acceleration Feedforward [▸ 113].

### Feedforward Velocity: Pre-Control Weighting [0.0 … 1.0]

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Position Control Loop: | | | | |
| | Feedforward Velocity: Pre-Control Weighting [0.0 ... 1.0] | 1.0 | | F | |

This parameter is a relative weighting of the pre-control. Here, the default parameter value `1.0` corresponds to a weighting of the pre-control of 100 %. The weighting of the velocity pre-control can be reduced to avoid overshooting in position.

## 7.4.3 Velocity Control Loop

Habitually, velocity control is performed within the drive controller by the velocity controller unit. This velocity controller is depicted within the controller overview of the Drive Manager. When velocity control is brought into the TwinCAT runtime this software velocity control unit has to be configured. The velocity control loop parameters configure the software velocity control within the TwinCAT runtime.

**Velocity control: Proportional Factor Kv**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Proportional Factor Kv | 0.1 | | F | |

The proportional factor Kv is the gain factor of the P controller.

**Velocity control: Integral Action Time Tn**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Integral Action Time Tn | 0.0 | | F | s |

Integral action time `Tn` of the I part of the PID controller. Integration time.

**Velocity control: Derivative Action Time Tv**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Derivative Action Time Tv | 0.0 | | F | s |

Derivative action time `Tv` of the real D part (D-T1 component) of the PID controller.

**Velocity control: Damping Time Td**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Damping Time Td | 0.0 | | F | s |

Damping time `Td` of the real D part (D-T1 component) of the PID controller.

**Velocity control: Min./max. limitation D-Part [0.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Min./max. limitation D-Part [0.0 ... 1.0] | 0.1 | | F | |

Limitation of the D part of the PID controller.

**Velocity control: Min./max. limitation I-Part [0.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Velocity Control Loop: | | | | |
| | Velocity control: Min./max. limitation I-Part [0.0 ... 1.0] | 0.1 | | F | |

Limitation of the I part of the PID controller.

## 7.4.4 Observer

An observer is part of the special controller type "Position P and velocity PI controller with Observer (Torque)" used in torque mode. The Observer is a mathematical model for alternative velocity calculation ("estimation"). This velocity is used as an alternative to the actual velocity (time derivative of the actual position). The model receives actual position and actual current as input variables and requires additional parameters.

**Velocity Filter: Time Constant T**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Observer: | | | | |
| | Velocity Filter: Time Constant T | 0.001 | | F | s |

Part of the Observer model for determining velocities.

**Observer Mode**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Observer: | | | | |
| | Observer Mode | 'OFF' | | E | |
| | Motor: Torque Constant Kt | 1.0 | | F | Nm/A |
| | Motor: Moment of Inertia Jm | 1.0 | | F | $kg\ cm^2$ |
| | Bandwidth f0 | 20.0 | | F | Hz |
| | Correction Factor Kc | 1.0 | | F | |

As Observer mode

- `'OFF'` or
- `'LUENBERGER'`

can be selected.

**Motor: Torque Constant Kt**

Part of the Observer model.

**Motor: Moment of Inertia Jm**

Part of the Observer model.

**Bandwidth f0**

Part of the Observer model.

**Correction Factor Kc**

Part of the Observer model.

## 7.4.5 Other Settings

### Controller Mode

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Controller Mode | 'STANDARD' | | E | |

'STANDARD': Currently only controller mode 'STANDARD' can be selected.

### Auto Offset

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Auto Offset | FALSE | | B | |
| | Offset Timer | 1.0 | | F | s |
| | Offset Limit (of Calibration Velocity) | 0.01 | | F | |

The Auto-Offset parameters only influence the behavior of the P-controller and the PP-controller. For the activation of

- Offset Timer

and

- Offset Limit (of Calibration Velocity)

the parameter Auto Offset must be set to TRUE.

The Auto Offset parameters are intended for an analog axis interface. A real axis may not be completely stopped at zero set velocity, but may drift slightly. This drift behavior may be due to offset problems, temperature problems or other reasons. The Auto-Offset is intended to act as an adaptive and automatic offset coordination to compensate for slight drifts and thereby prevent the axis from drifting further. For digital interfaces, the Auto-Offset has no effect.

Automatic offset adjustment calculates and activates a DAC offset that minimizes the following error in position control.
See section Automatic DAC offset adjustment [▶ 113].

### Offset Timer

The offset timer has a similar effect as an I part for the control.

### Offset Limit (of Calibration Velocity)

For a P-controller or for a PP-controller in the interval [- Offset limit ... + Offset limit] the offset is adjusted automatically. A relative control. At a limit, the offset remains constant and does not go beyond it.

### Slave coupling control: Proportional Factor Kcp

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Slave coupling control: Proportional Factor Kcp | 0.0 | | F | mm/s/mm |

This parameter only influences the control loop in TwinCAT NC. The proportional factor Kcp of the slave coupling differential control is the gain factor for an additional P controller. This P controller attempts to minimize the difference of the master and slave following error to improve accuracy.

**Controller Outputlimit [0.0 … 1.0]**

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Other Settings: | | | | |
| | Controller Outputlimit [0.0 ... 1.0] | 0.5 | | F | |

The output limit of the controller can be set in the interval [0.0 ... 1.0].

## 7.5 Technical Terms

### 7.5.1 Acceleration Feedforward

In addition to proportional feedback of the following error nearly all position controllers contain a proportional acceleration feedforward: The `K_a` factor. This acceleration feedforward should normally only be used in association with the proportional component of the position controller: The `K_v` factor. Acceleration feedforward control requires to adjust the axis for strict symmetry:

- When stationary, the following error is symmetrical about `0` (DAC offset).
  Confer section Automatic DAC Offset Adjustment [▶ 113].
- When moving steadily, the following error is symmetrical about `0` (reference velocity).
- Set `K_v`.
- Measure the extreme value of the acceleration `a_+max` and the associated following error `d_+max` in the middle of the acceleration phase. Measure the extreme value of the deceleration `a_-max` and the associated following error `d_-max` in the middle of the braking phase.
- `K_a+ = K_v * d_+max / a_+max,`
  `K_a- = K_v * d_-max / a_-max,`
  `K_a = (K_a+ + K_a-) / 2.`

### 7.5.2 Automatic DAC Offset Adjustment

Any controller with no I component has automatic DAC offset adjustment as an option. This adjustment is only active when the velocity feedforward of the axis falls below a certain magnitude. This limited activity prevents the DAC offset adjustment from being affected by the dynamic behavior of the axis. If the axis is subject to position control or is moving at a suitably low velocity, an offset velocity is generated by integrating the control velocity. This offset velocity is added to the output. The negative feedback of the position control loop results in a PT1 behavior creating an exponential function.

**Offset Adjustment Parameters**

*Offset filter time*: Data type `Double`. Unit `sec`. Time constant for the offset adjustment.

*Offset limit*: Data type `Double`. A relative control. At a limit the offset is kept constant not going beyond it.

Quite possibly, the behavior of the offset adjustment could be affected at runtime. Suitably, a range of “switches” is available: For instance, at runtime the PLC or another ADS device can modify time constant parameters or feedforward limit parameters.

The offset adjustment could be switched off entirely. Doing so, it is not always possible to avoid a jump in the output voltage. Therefore, a soft deactivation can be achieved with the “fade out” mode. A fade out reduces the adjustment to zero over time while following its own curve. To keep the adjustment steady for a period of time the “hold” mode can be activated. E.g., the “hold” mode is suitable when stopping the power section of a drive temporarily. Performing such a stop it would be impossible for the offset not to run out of control if the offset adjustment was to remain active.

## 7.5.3 Axis Error

The variables `axis.Status.Error` and `axis.Status.ErrorID` belong to the `ST_AxisStatus` data structure.

The boolean variable `axis.Status.Error` displays the StateDWord bit 31 and refers to the axis error state. When the boolean variable `axis.Status.Error` is `TRUE`, it indicates that there is an error for the axis.

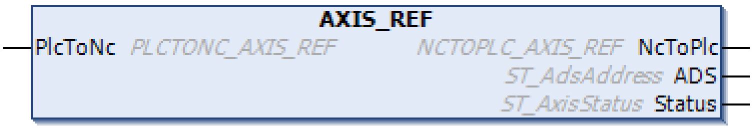
The variable `axis.Status.ErrorID` of data type `UDINT` refers to the error ID of the reported error and displays the axis error code.

## 7.5.4 AXIS_REF

```
PROGRAM MAIN
VAR
    axis: AXIS_REF;
END_VAR
```

```
axis.ReadStatus();
```

**MC axis variable**




The PLC axis variable `MAIN.axis` has the data type `AXIS_REF`.

The data type `AXIS_REF`:

- contains information about an axis,
- is an interface between the PLC and the NC and
- is given to the MC function blocks as a reference to an axis.

**Refreshing the status data structure in `AXIS_REF`**

The status data structure `Status` of the type `ST_AxisStatus` contains additional or prepared status and diagnostic information about an axis. The structure is not cyclically refreshed, but has to be updated by the PLC program

Calling the `ReadStatus()` action from `AXIS_REF` updates the status data structure and should be done once at the beginning of each PLC cycle.

The status information does not change within a PLC cycle. After calling `ReadStatus()`, the current status information can be accessed in `AXIS_REF` within the entire PLC program.
The nature of the status data structure is purely informative. Therefore, its use is not mandatory.

## 7.5.5 NC

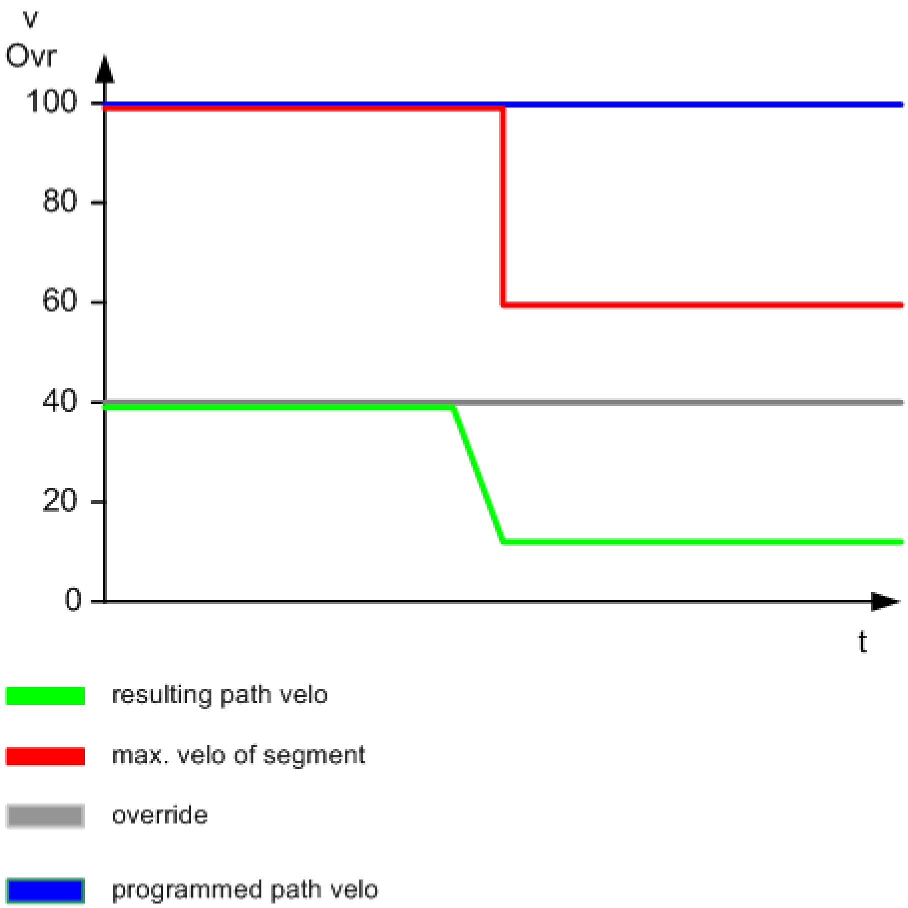
TwinCAT NC is an assembly of function groups used for the control and regulation of axes or of synchronized axis groups. An NC task consists of one or more channels of type PTP channel, FIFO channel or NCI channel, and their subsidiary parts. Generally, immediately after start-up the NC axes find themselves in one or more PTP channels. Particularly, they are moved to a different channel through a process of reconfiguration then, if necessary.

## 7.5.6 Path Override (Interpreter Override Types)

The path override is a velocity override. Consequently, changing the override creates a new velocity, but does not affect the ramps (acceleration or jerk). The applicable override types merely differ in terms of their underlying reference velocity.

The parameterization takes place in the interpolation channel under the group parameters.

**Option “Reduced (iterated)”**



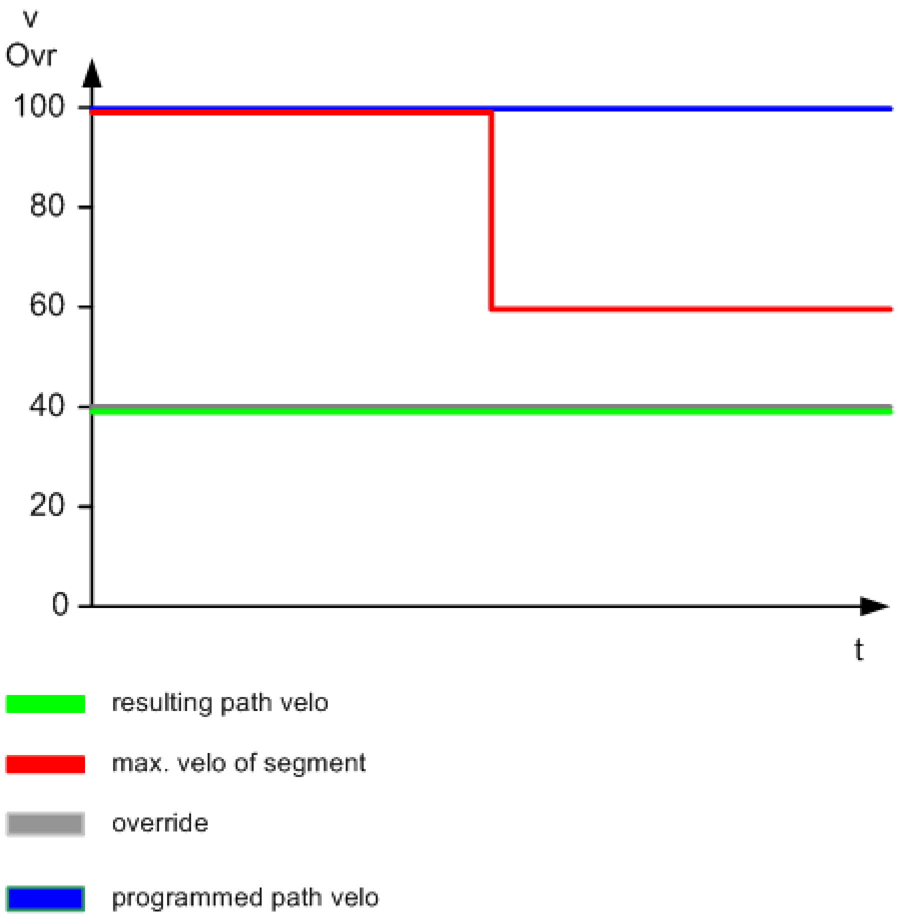

*Based on the reduced velocity (default).*

Because of the relevant dynamic parameters (braking distance, acceleration etc.) it is not possible for the programmed velocity, depicted by the blue line, to be achieved in every segment. For this reason, for each geometric segment a velocity, depicted by the red line, is calculated that may possibly be reduced compared to the programmed velocity. In the standard case, the override is made with reference to this possibly reduced segment velocity.

The advantage of this override type is that the machine operates with an approximately linear reduction in velocity when override values are small. Therefore, “Reduced (iterated)” is the correct setting for most applications:

$v_{res} = v_{max}$ * Override.

**Option “Original (iterated)”**




*Based on the programmed path velocity.*

The override value is based on the velocity programmed by the user. The maximum segment velocity has merely a limiting effect.

**Option “Reduced [0 … >100%]”**

*Based on internally reduced velocity with the option to specify a value greater than 100%.*

From TwinCAT V2.10, Build 1329.

Generally, the override type behaves like “Reduced (iterated)”. More specifically, with this override type it is possible to travel along the path more quickly than programmed in the G-Code. There is no limitation to 120%, for example. The maximum possible path velocity is limited by the maximum velocities of the axis components (`G0` velocity) and their dynamics.

If limitation to a particular value, e.g. 120%, is required, this limitation can be set within the PLC project.

## 7.5.7 PTP

PTP stands for point-to-point. PTP axis functionality is a control process for **one-dimensional positioning** of axes, in particular servo axes, but also other types of axes. One-dimensional does not necessarily mean linear. It simply means that one component is interpolated in some specified coordinate system (Cartesian coordinates, polar coordinates).

PTP is a licence level for basic axis numerical control. PTP forms the basis of the whole of TwinCAT NC, because at **system start-up** the axes are normally in PTP mode, and thus are position controlled. The extended TwinCAT NC functionalities are achieved on the basis of the PTP modes by reconfiguration (FIFO, NCI) or by coupling (all slave types).

### 7.5.8 PT1 Filter

A PT1 filter is a transfer function that performs convex interpolation between a new value `x_n` and an old value (from one cycle before) `x_a`. The filter time parameter, a nonnegative value in the unit of seconds, has to be entered into the calculation. If `I = SAF cycle time / (SAF cycle time + filter time)`, then `x = I * x_n + (1-I) * x_a`. The filter time should be finite. For the filter time being a positive value `I` resides within the open interval from `0` to `1`. If the filter time is close to `0.0`, the new value has a high weighting. If the filter time is long, the older value has a relatively high weighting.

### 7.5.9 Rapid Traverse

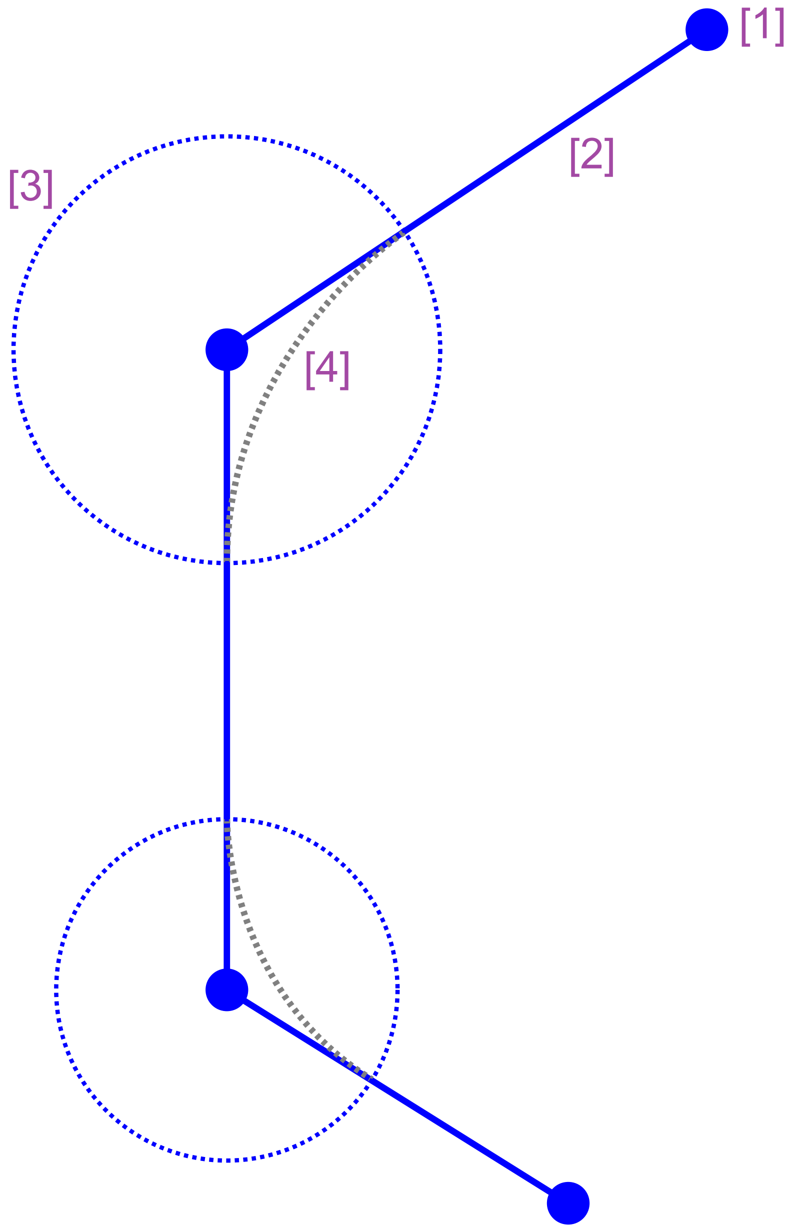
| | |
|---|---|
| Command | `G0` |
| Cancellation | `G1`, `G2` or `G3` |

- Rapid traverse is used to position the tool quickly, and is not to be used for machining the workpiece. With G0 the axes are moved with linear interpolation as fast as possible. The velocity is calculated with MIN (Rapid Traverse Velocity (G0), Reference Velocity, Maximum Velocity).
- If a number of axes are to be driven in rapid traverse the velocity is determined by that axis that requires the most time for its movement.
- An accurate stop (`G60`) is cancelled with `G0`.
- The rapid traverse velocity is set for each axis individually at “MOTION | NC-Task 1 SAF | Axes | Axis 1 | NCI Parameter | Rapid Traverse Velocity (`G0`)”.

### 7.5.10 ReadStatus()

See section AXIS_REF [▶ 114].

## 7.5.11 Tolerance Ball




[1] Glue dot.

[2] Segment, geometrical element, here: line.

[3] Tolerance ball.

[4] Transition, smooth path.

**Motivation**

| | |
|---|---|
| *Dynamical-Steadiness* | • Segment transitions that are not two times steadily differentiable with respect to their spatial coordinate lead to dynamic-unsteadinesses, if at this transition the path velocity is not reduced to *zero* value.<br>• Segment transitions can be smoothed out using Bézier-splines leading to the dynamics for the whole path to be steady at the segment transition, even though path velocity differs from *zero*. |
| *Smoothening* | • Tolerance balls are used to smooth out dynamic unsteadiness at segment transitions. |
| *Faster Dynamics* | • Smoothening permitts faster dynamics.<br>• The maximum segment transition velocity calculated by the system in advance has the value *VeloLink*.<br>• The user can change the system parameter *C2* for C2-velocity reduction online.<br>• The segment transition velocity has the value *C2 x VeloLink*. |

**Definition**

| | |
|---|---|
| *Segment Transition* | • For the smoothening a tolerance ball is placed around each segment transition. |
| *Permitted Path Deviation* | • Up to the extend that the path stays within the tolerance ball, this path may deviate from its predefined geometry within this tolerance ball. |

**Parameter**

| | |
|---|---|
| *Radius* | • The user adjusts the radius of the tolerance ball. |

**Range of Validity**

| | |
|---|---|
| *No Exact Halt, No Stop* | • The radius of the tolerance ball is valid modally for all segment transitions that at the segment transition do not imply an exact halt or a stop. |

**Algorithmic Behavior**

| | |
|---|---|
| *Adaptive* | • Automatically, the radii of the tolerance balls are set adaptively. |
| *Inhibit Overlap* | • Setting the radii adaptively prevents from overlap of the tolerance balls. Otherwise, especially for small segments an overlap of tolerance balls could occur. |
| *At the Segment Transition* | • Within the tolerance ball there is no override.<br>• Entering the tolerance ball the path acceleration obtains *zero* value.<br>• Entering the tolerance ball the path velocity obtains the value of the segement transition velocity.<br>• Within the tolerance ball the path acceleration stays at *zero* value.<br>• Within the tolerance ball the path velocity remains at the value of the segment transition velocity.<br>• The override induced change of velocity level is interrupted within the tolerance ball and is continued after leaving the tolerance ball. |

# 8 First steps

The necessary steps for creating a project, moving an axis and analyzing this movement are explained in the following.

**⚠ DANGER**

**Acute risk of injury when using real axes**

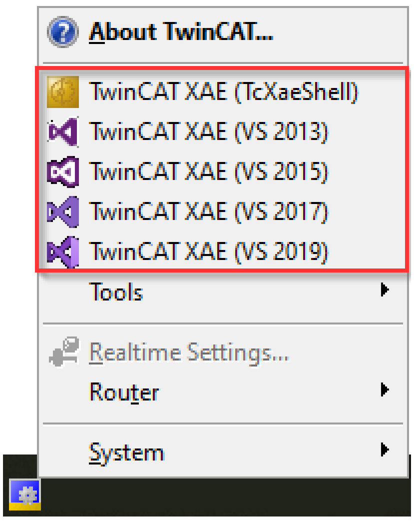
- Only specialist personnel with extensive knowledge in the areas of electrical engineering / drive technology are allowed to install and commission the equipment.
- Check that all live connection points are protected against accidental contact.
- Never undo the electrical connections to the motor when it is live.
- The surface temperature of the motor can exceed 100 °C in operation. Check (measure) the temperature of the motor. Wait until the motor has cooled down below 40 °C before touching it.
- Make sure that, even if the drive starts to move unintentionally, no danger can result for personnel or machinery.


1. Create project [▶ 120]
2. Create axis [▶ 122]
    - Create simulation axis [▶ 122]
    - Create AX5000 and NC axis manually [▶ 129]
    - Create AX5000 and NC axis automatically [▶ 124]
3. Axis configuration [▶ 134]
    - Setting dynamics
    - Limiting positions [▶ 134]
4. Moving the axis [▶ 135]
    - Manual movement via the commissioning dialog [▶ 136]
    - Simple movement via the PLC [▶ 137]
5. Scope integration [▶ 141]

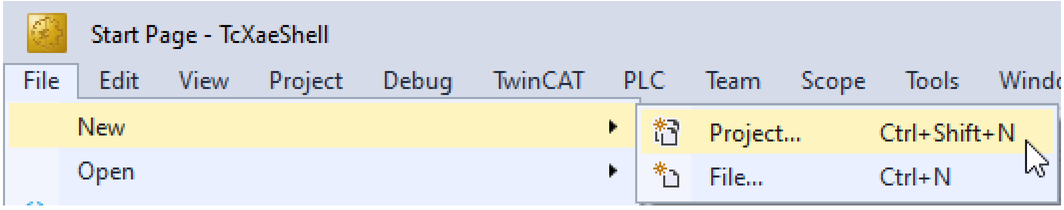

## 8.1 Create project

TwinCAT Motion is fully integrated in the TwinCAT system. A standard TwinCAT project is therefore the starting point for every TwinCAT Motion application:

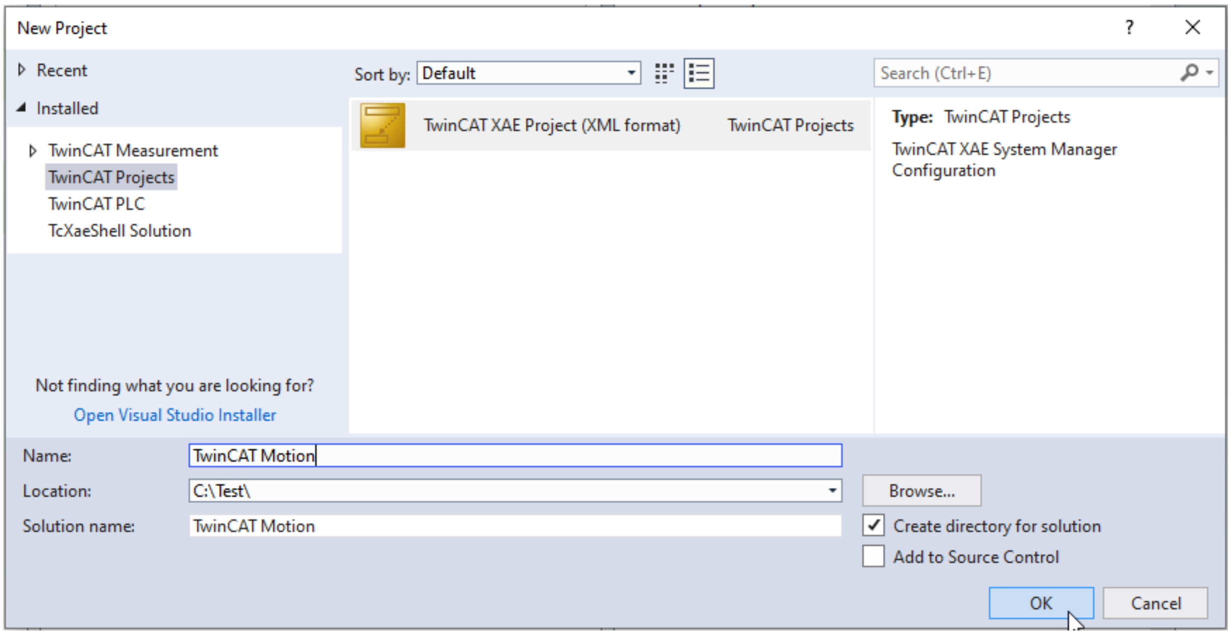
1. Open the TwinCAT XAE Shell or a Visual Studio with integrated TwinCAT.



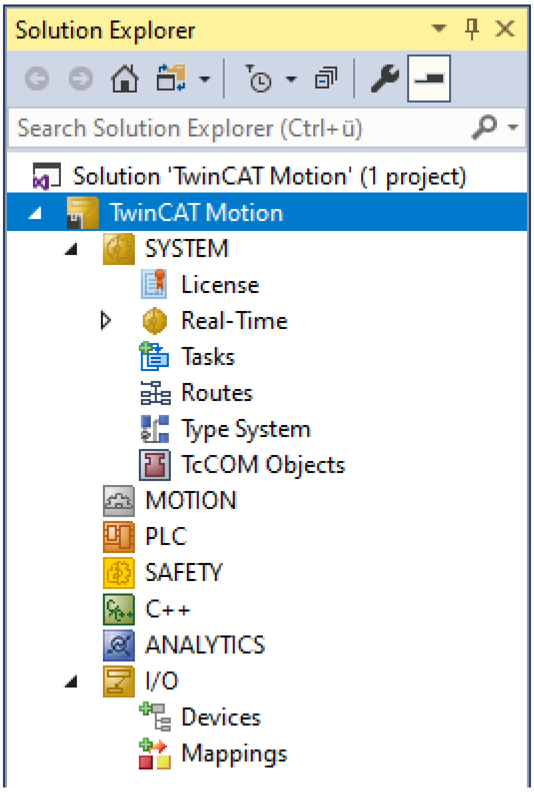
2. Open the dialog **New > Project**.




3. Select TwinCAT XAE Project, name and location.




⇨ A TwinCAT project with the selected name is created.



**Next step**

Create axis [▸ 122]

# 8.2 Create axis

An axis can be created both with and without a link to the real drive hardware. For the first steps we recommend to do without the real drive hardware and to create the axis as a simulation axis:

- Create simulation axis [▸ 122]

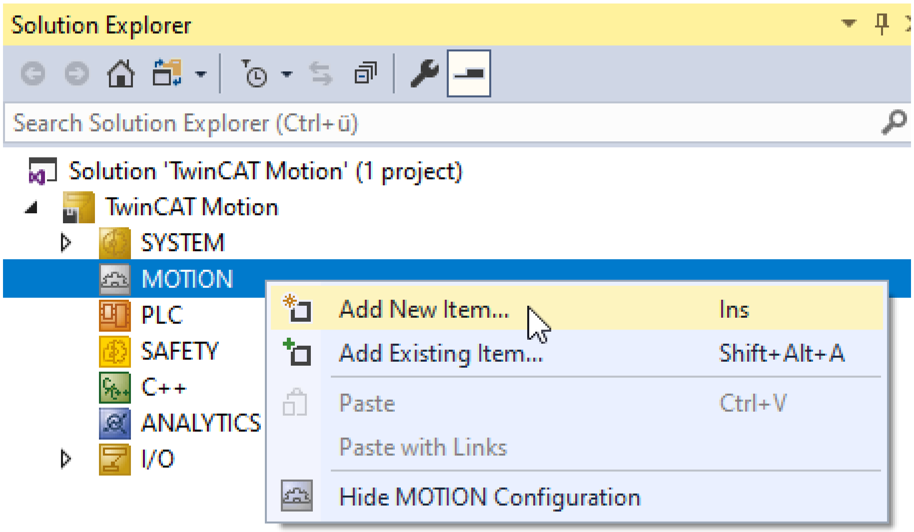
The creation of the axis with real drive hardware can be done manually as well as automatically and is explained below using the example of an AX5000. The steps can also be transferred to other servo drives:

- Create AX5000 and NC axis manually [▸ 129]
- Create AX5000 and NC axis automatically [▸ 124]

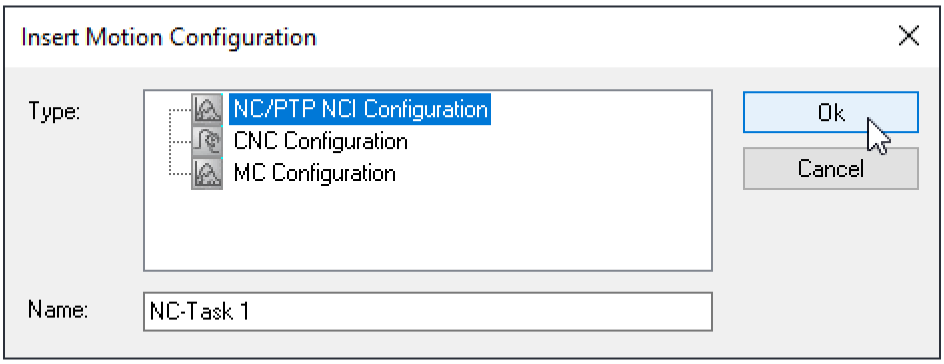
## 8.2.1 Create simulation axis

In order to familiarize yourself with TwinCAT Motion and to test new programs as well as functions, the use of simulation axes is recommended. In this section you will learn how to create such a simulation axis.

1. By default, the Motion node is shown within a TwinCAT project. If this is not the case, you can show it by right-clicking on the TwinCAT project in the **Solution Explorer > Show Hidden Configurations > Show MOTION Configuration**.
2. Below the Motion node you can insert various Motion configurations. To do this, right click **Motion > Add New Item**.

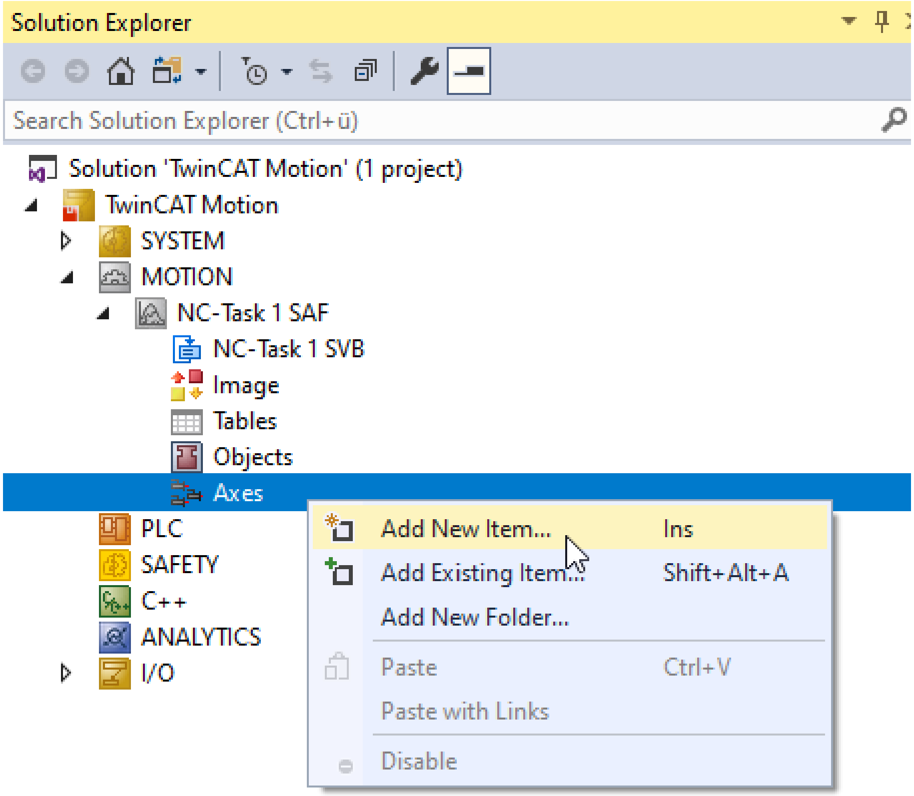



3. Select **NC/PTP NCI Configuration** and confirm with **OK**.

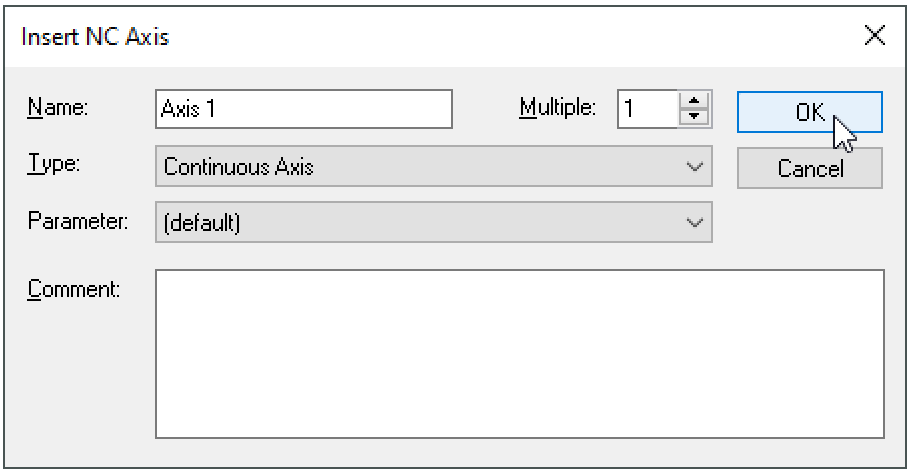



⇨ The NC/PTP NCI Configuration [▸ 15] is now below the Motion node.

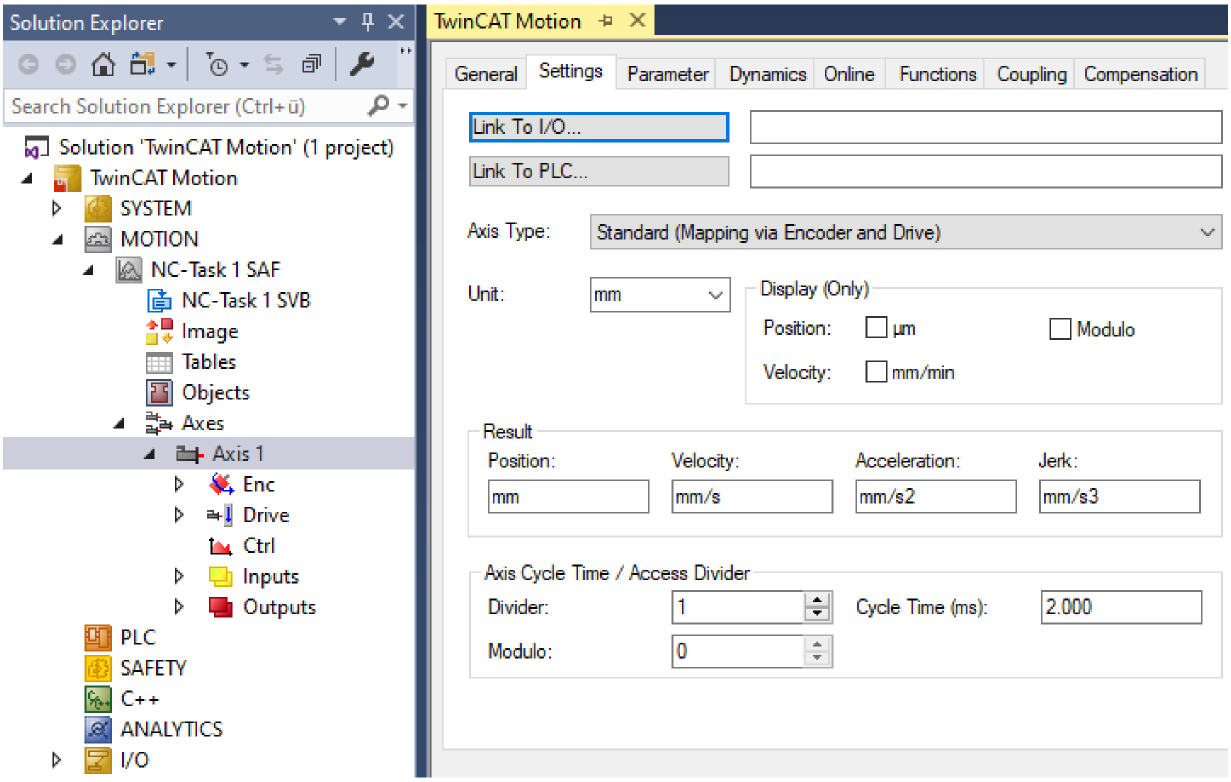
4. Axes can be added to the configuration with a right click on **Axes > Add New Item**.




5. For a simulation axis, select the type **Continuous Axis** and confirm your selection with **OK**.
   See details about the axis types [▶ 20].



⇨ A simulation axis is created. You can recognize it by the fact that no link to the I/O is listed in the axis dialog Settings [▶ 21] and the axis type **Standard (Mapping via Encoder and Drive)** is selected. See details about the axis dialogs [▶ 21].




### Alternative steps

Instead of a simulation axis, a real axis can alternatively be created:

- Create AX5000 and NC axis manually [▶ 129]
- Create AX5000 and NC axis automatically [▶ 124]

### Next steps

Before moving the axis [▶ 135], it should be configured:

- Setting dynamics
- Limiting positions [▶ 134]

## 8.2.2 Create AX5000 and NC axis automatically

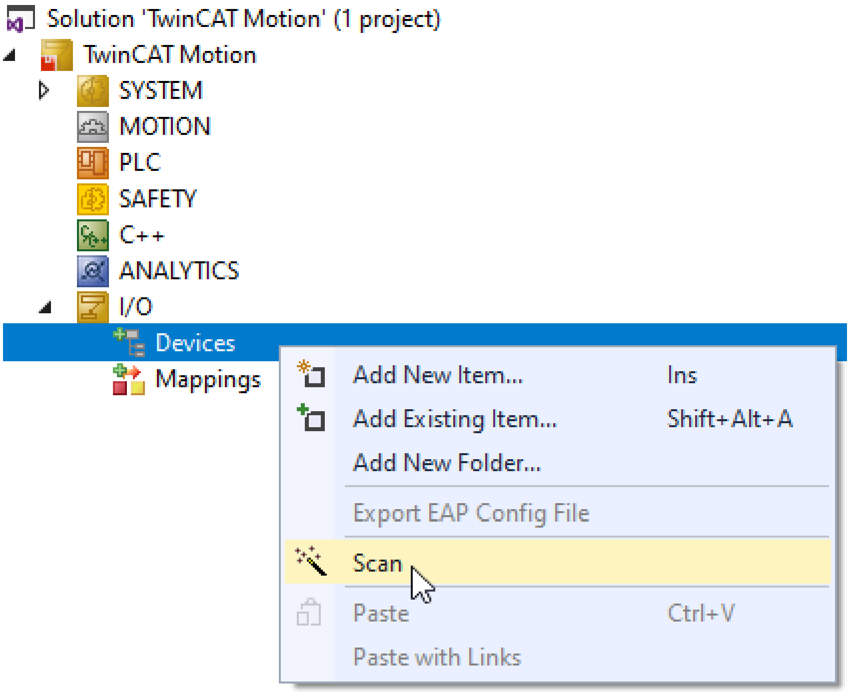
After you have familiarized yourself with a simulation axis (Create simulation axis [▶ 122]), you can familiarize yourself with an axis that is linked to real drive hardware. Here, the example of an AX5000 is used to explain how the drive hardware can be automatically created in TwinCAT Engineering and linked to an NC axis.

**Requirements:**

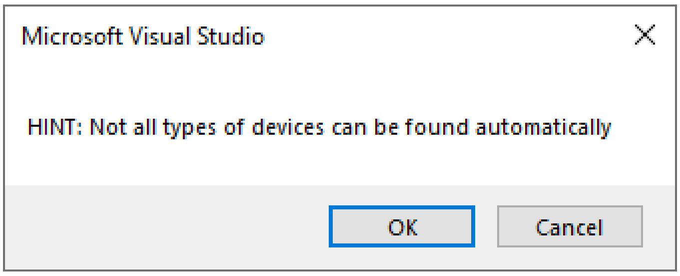
- Control voltage: 24 VDC
- EtherCAT connection with master card
- TwinCAT in Config Mode

**Scan Devices**

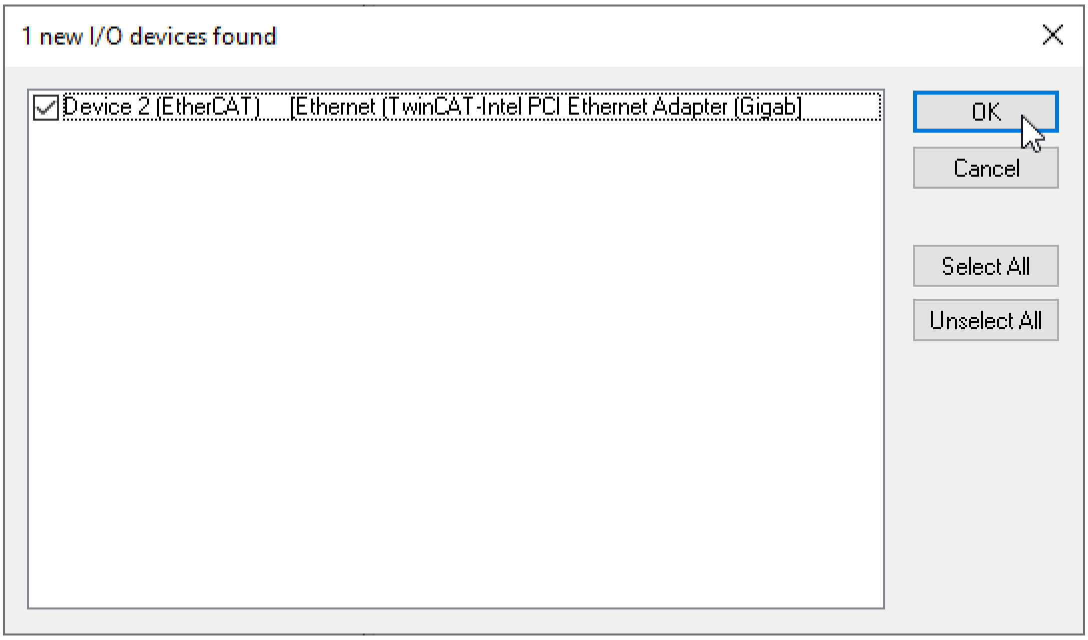
1. First the bus is scanned for connected EtherCAT devices.



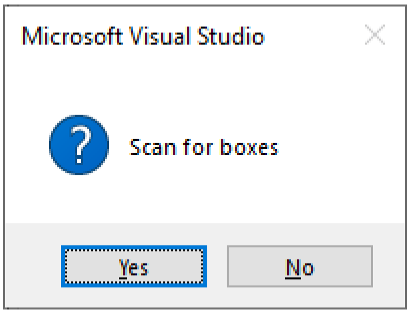

2. Confirm the hint that not all device types can be found automatically.



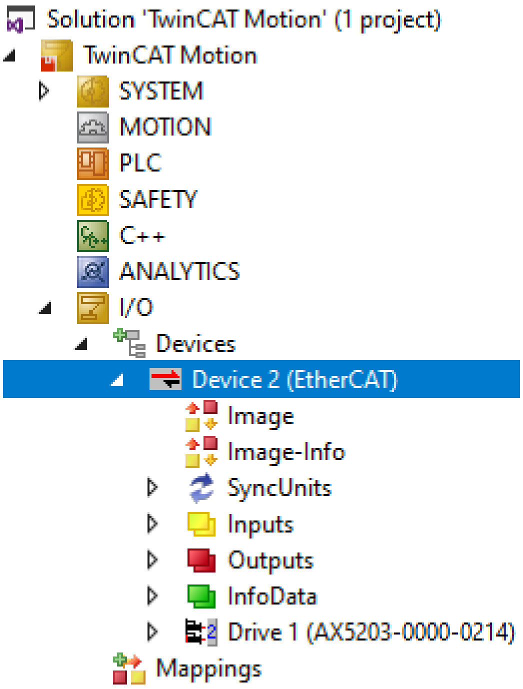

3. Select the EtherCAT interface card.



4. Scan for connected devices (**Scan for boxes**).

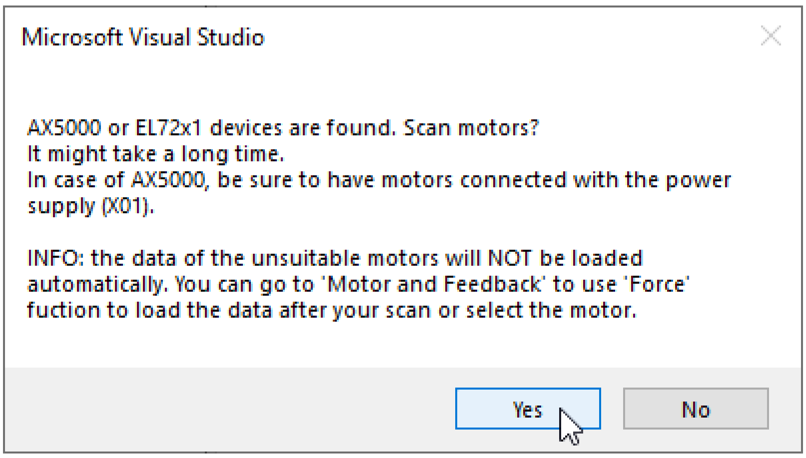



⇨ The AX5000 was found and is displayed below the device.



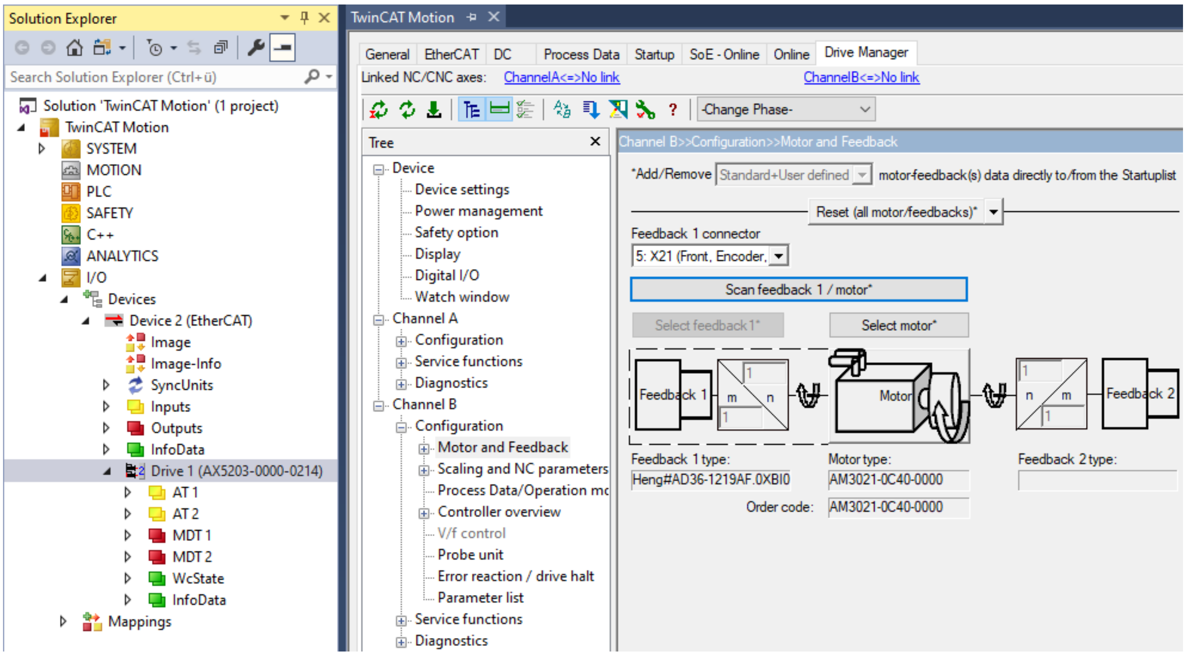

**Scan Motors**

5. Beckhoff motors with integrated name plate can be automatically detected and added to the configuration. To do this, confirm the "Scan Motors" message with **Yes**.
If you do not use a corresponding motor, please perform this step using the documentation for setting up an AX5000 manually.



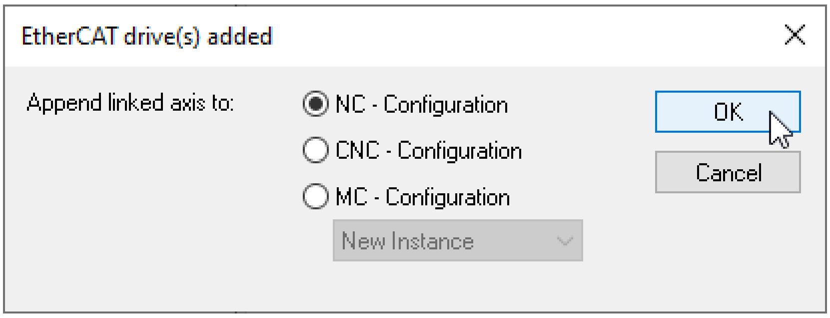

⇨ The detected feedback system and the detected motor are displayed at the following location: **Drive > Drive Manager > Channel > Configuration > Motor and Feedback**
The scan step can also be restarted via the button **Scan feedback 1 / motor***.
The TwinCAT 3 Drive Manger 2 (TE5950) also provides a commissioning tool for Beckhoff Drive

Technology. With this you can read the connected motors again. It also supports you during parameterization and configuration.




### Append link axis to NC

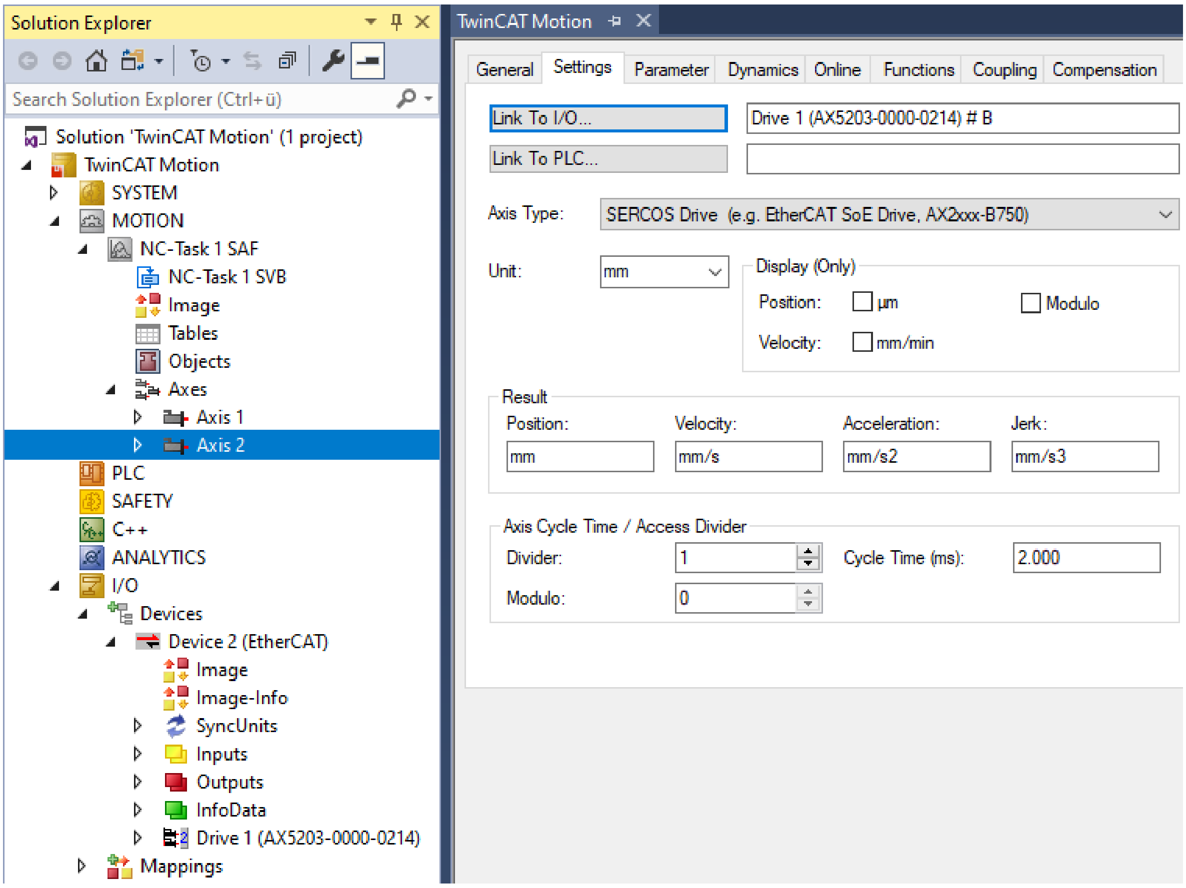
6. Append link axis to NC



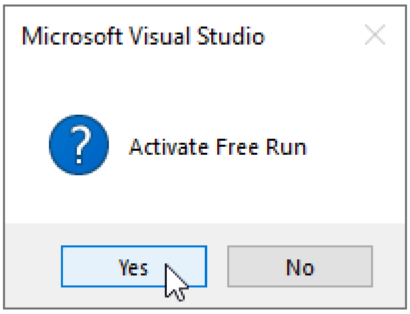

⇨ If there was no NC configuration in the project yet, it has now been created.

⇨ Axis objects corresponding to the drive were automatically created in the NC configuration and

⇨ the axis objects were automatically linked to the drive scanned under I/O.




7. Afterwards, you will be asked if you want to activate Free Run. This is not mandatory for the next steps, so the question may be answered as you wish with **Yes** or **No**.




⇨ An AX5000 with the associated standard NC configuration is now created.

### Alternative steps

- Create simulation axis [▸ 122]
- Create AX5000 and NC axis manually [▸ 129]

### Next steps

Before moving the axis [▸ 135], it should be configured:

- Setting dynamics
- Limiting positions [▸ 134]

## 8.2.3 Create AX5000 and NC axis manually

After you have familiarized yourself with a simulation axis (Create simulation axis [▸ 122]), you can now deal with an axis that is linked to real drive hardware. Here, the example of an AX5000 is used to explain how the drive hardware can be created manually in TwinCAT Engineering and linked to an NC axis.

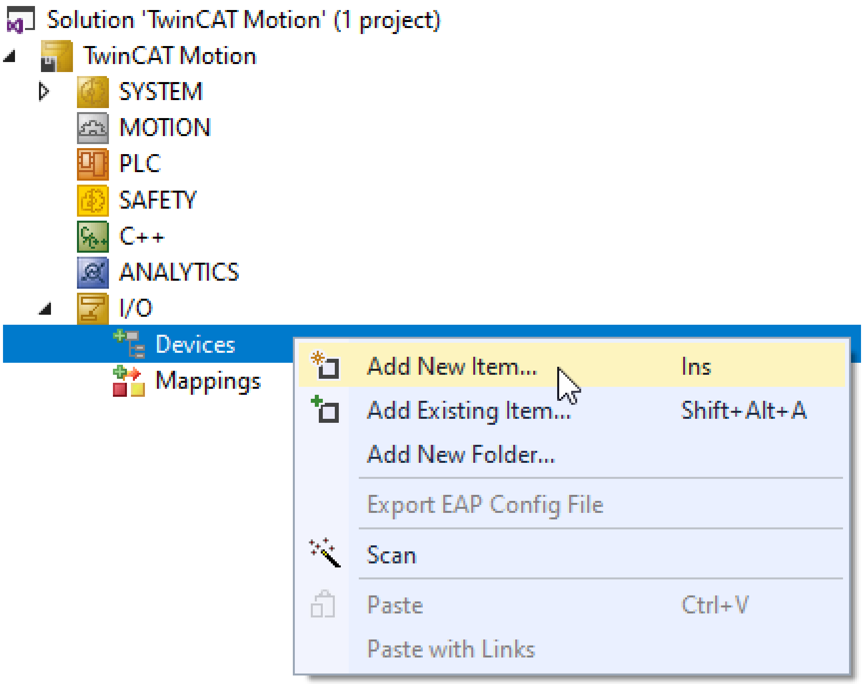
**Requirements**

- TwinCAT is in ConfigMode

**Create Devices**

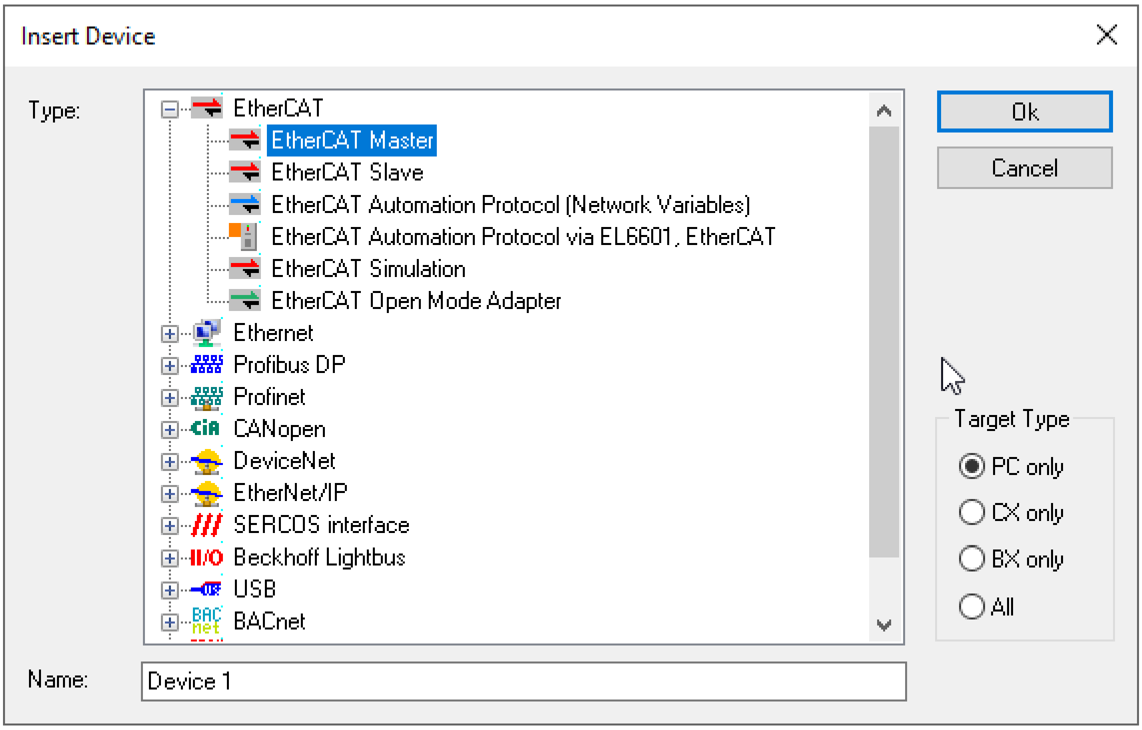
ℹ The AX5000 communicates via EtherCAT.

1. Click the path **I/O > Devices > Add New Item...**



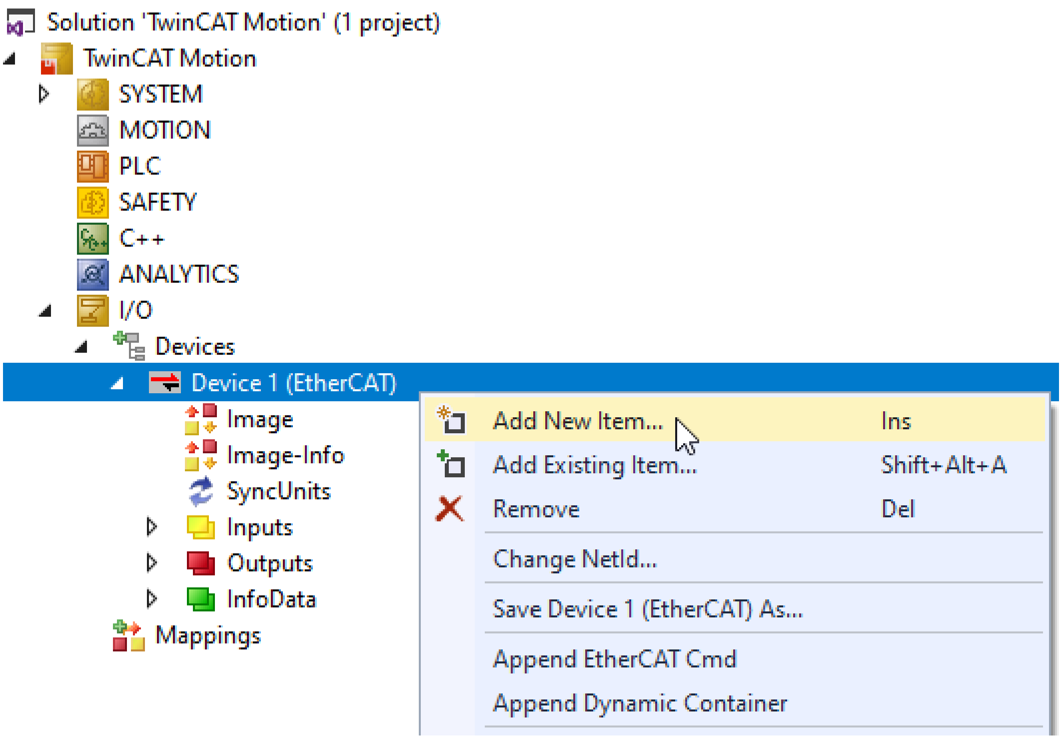

2. Open the dialog **Insert Device**.
3. Select an **EtherCAT Master**.

4. Confirm your selection with **OK**.



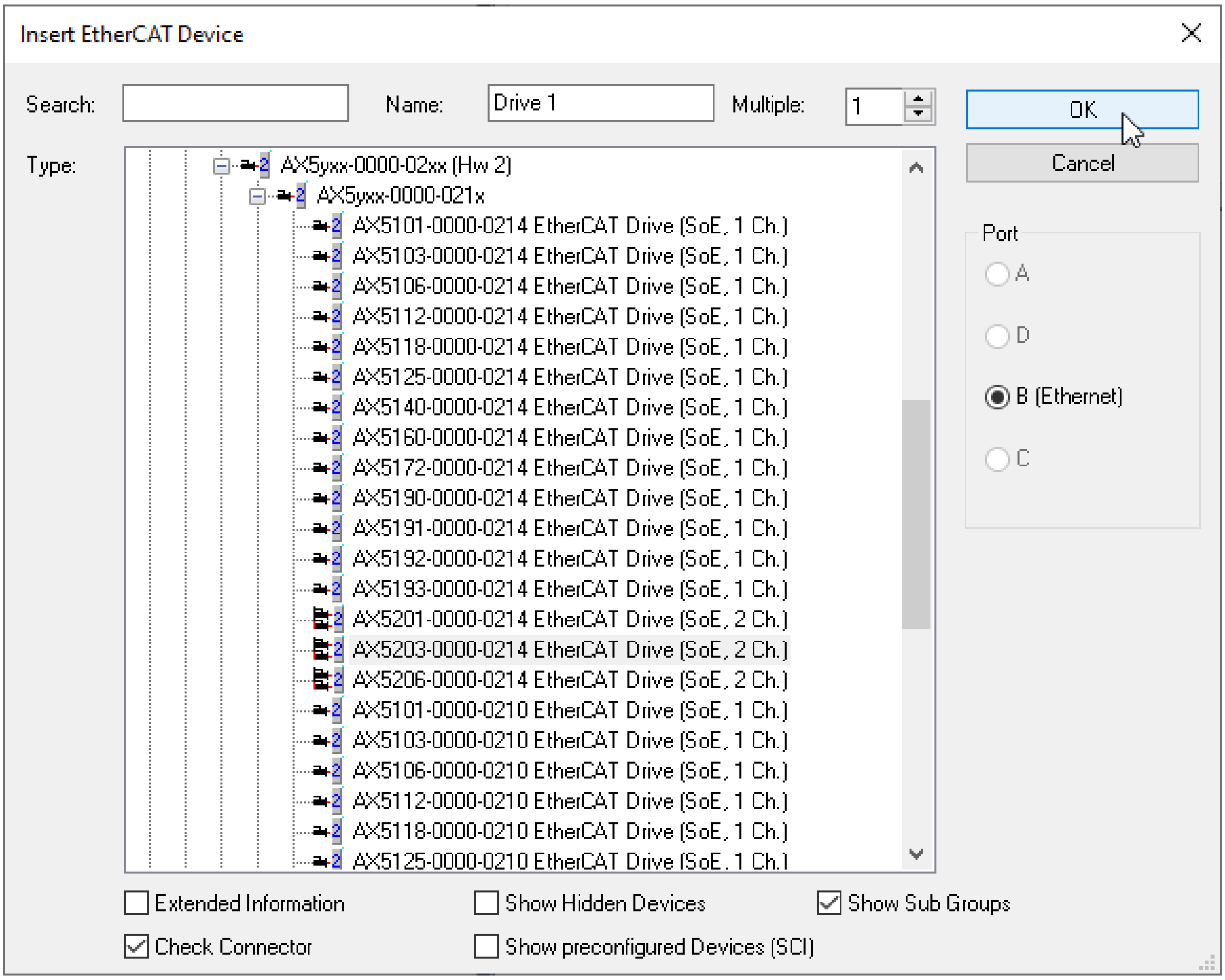

⇨ The EtherCAT master is added.

5. Click the path **Devices > Device 1 (EtherCAT) > Add New Item...**




⇨ The dialog **Insert EtherCAT Device** opens.

6. In the **Insert EtherCAT Device** dialog, select the AX5000 actually used and confirm with **OK**.

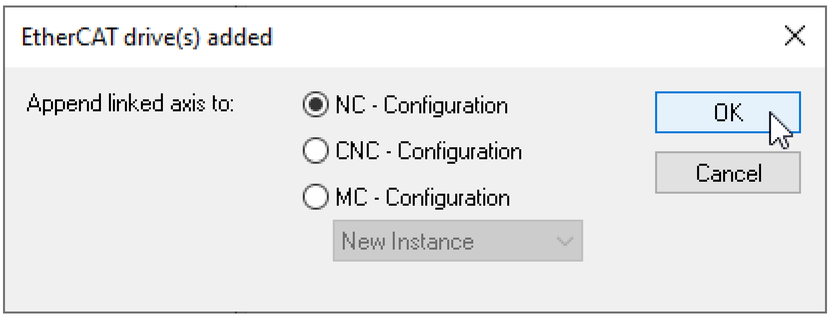



⇨ An EtherCAT master with the AX5000 used is created under **I/O**.

**Append link axis to NC automatically**

**Linking the axis automatically**

When the AX5000 is created, TwinCAT asks whether the axis should be linked to the NC configuration. By confirming this message with **OK** this step is done automatically. This has the advantage that the objects are correctly linked and settings such as the Axis Type are automatically set correctly.




**What happens after the confirmation of the NC configuration?**

- If there was no NC configuration in the project yet, it was created.
- Axis objects were automatically created in the NC configuration according to the drive.
- The axis objects were automatically linked to the drive scanned under **I/O**.

**Append link axis to NC manually**

If you have acknowledged the query with **Cancel**, you can create and link the NC configuration and axis objects manually.

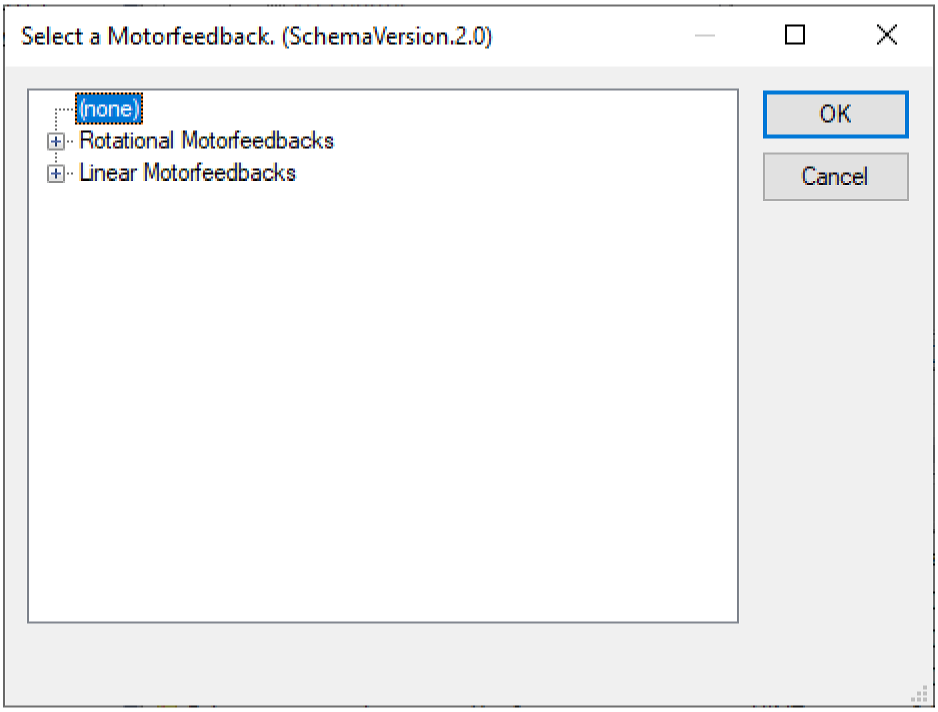
1. To do this, first carry out the steps as for Creating a simulation axis [▸ 122].
2. At **Axis > Settings > Axis Type**, select **SERCOS Drive**.
3. The NC axis must be linked to the channel of the AX5000 under **I/O**. To do this, click **Axis > Settings > Link To I/O...**.
   ⇨ The dialog **Select I/O Box/Terminal** opens.
4. Select the channel of your AX5000 to which you want to link the NC axis.

⇨ The NC axis is linked to the drive channel.

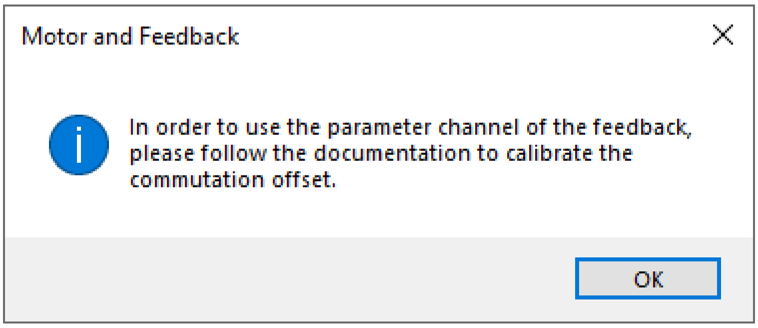
**Select feedback and motor**

✓ A digital name plate is required for this purpose.

1. To add a feedback system and motor, go to **I/O > Devices > Drive Manager Channel > Configuration > Motor and Feedback**.
2. Via Scan **feedback 1 / motor*** you can scan the feedback system and the motor as described at Create AX5000 and NC axis automatically [▸ 124] and thus add them automatically.
3. Alternatively, you can select the feedback system via **Select feedback 1*** using the following dialog and confirm with **OK**.

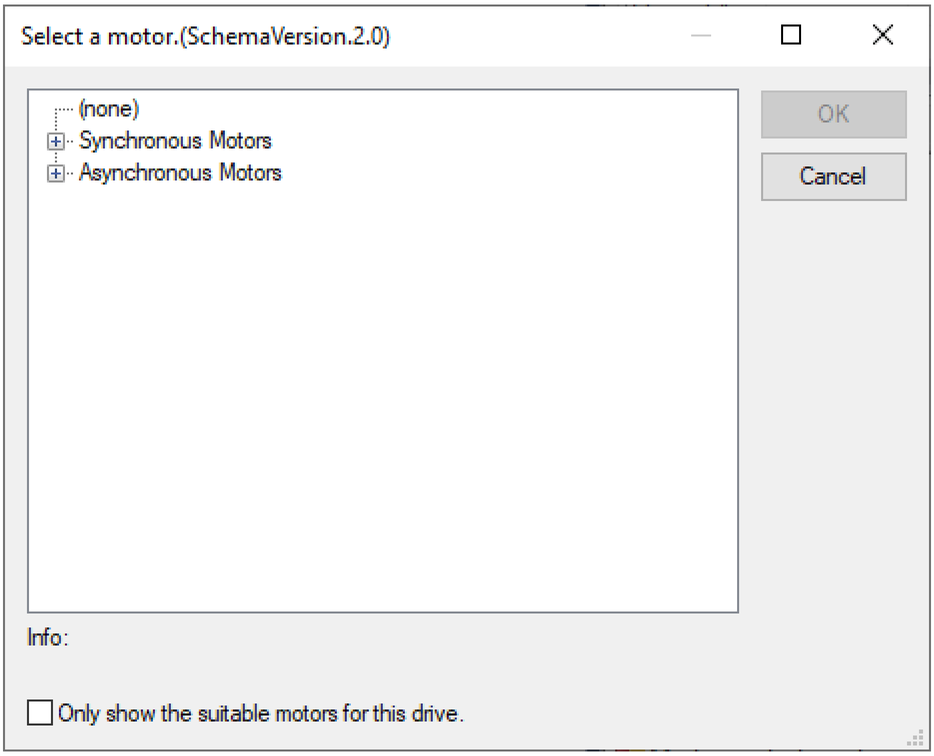



4. Confirm the following hint.

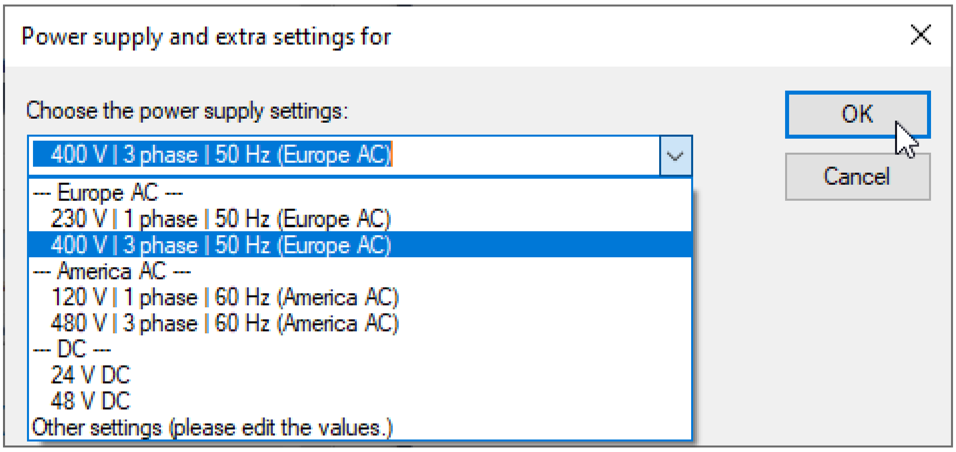



   ⇨ The feedback system has been selected.

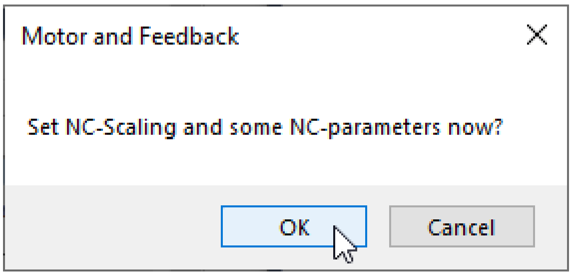
5. Via **Select motor*** you get to the following dialog. Select the motor and confirm with **OK**.




6. Select the power supply of the motor and confirm with **OK**.




7. Confirm the following query with **OK**.




⇨ Now adjustments can be made automatically to the NC axis. Please note the points mentioned at Axis configuration [▶ 134].

⇨ Feedback system and motor are configured.

**Alternative steps**

- Create simulation axis [▶ 122]
- Create AX5000 and NC axis automatically [▶ 124]

**Next steps**

Before moving the axis [▶ 135], it should be configured:

- Setting dynamics
- Limiting positions [▶ 134]

# 8.3 Axis configuration

## 8.3.1 Limiting positions

To prevent collisions, monitoring and limiting positions can be activated in the form of software limit switches and position lag monitoring.

**No safety functions in the sense of safety technology**

Purely functional settings are explained in this section; these are not safety functions in the sense of safety technology.

**Activating the software limit switches**

If the software limit switch monitoring is activated for the axis, the axis does not move beyond the set position (max./min. end position).

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|
| - | Limit Switches: | | | | |
| | Soft Position Limit Minimum Monitoring | FALSE | | B | |
| | Minimum Position | 0.0 | | F | mm |
| | Soft Position Limit Maximum Monitoring | FALSE | | B | |
| | Maximum Position | 0.0 | | F | mm |

**Activation steps:**

1. Select the NC axis for which you wish to activate the software limit switches.
2. Open the **Parameter** dialog for this NC axis.
3. Set the value TRUE for the parameter **Limit Switches: Soft Position Limit Minimum Monitoring**. Use the corresponding drop-down box in the table column **Offline Value**.
4. Set the value TRUE for the parameter **Limit Switches: Soft Position Limit Maximum Monitoring**. Use the corresponding drop-down box in the table column **Offline Value**.
5. Use the **Limit Switches: Minimum Position** parameter to set a value for the smallest position value that can be approached that allows sufficient freedom of movement and excludes the possibility of a collision.
6. Use the **Limit Switches: Maximum Position** parameter to set a value for the largest position value that can be approached that provides sufficient freedom of movement and eliminates the possibility of a collision.

⇨ With the next Activate configuration [▶ 135] you have activated the software limit switches for the selected NC axis.

**Activating the position lag monitoring**

The position lag monitoring function checks whether the current position lag of an axis has exceeded the limit value. The lag distance is the difference between the set value (control value) and the actual value reported back. Here, monitoring takes place both at standstill and during the movement of an axis.

| | Parameter | Offline Value | Online Value | Type | Unit |
|---|---|---|---|---|---|

| - | Monitoring: | | | | |
|---|---|---|---|---|---|
| | Position Lag Monitoring | TRUE | | B | |
| | Maximum Position Lag Value | 5.0 | | F | mm |
| | Maximum Position Lag Filter Time | 0.02 | | F | s |

**Activation steps:**

1. Select the NC axis for which you want to activate position lag monitoring.
2. Open the **Parameter** dialog for this NC axis.
3. Set the value TRUE for the parameter **Monitoring: Position Lag Monitoring**.
   Use the corresponding drop-down box in the Offline Value table column.
4. Use the **Monitoring: Maximum Position Lag Value** parameter to set a value for the maximum permissible lag distance that provides sufficient control freedom and eliminates the possibility of a collision.
   The default value is usually suitable.
5. Set a value for the parameter **Monitoring: Maximum Position Lag Filter Time** that provides sufficient control freedom and excludes the possibility of a collision.
   The default value is usually suitable.

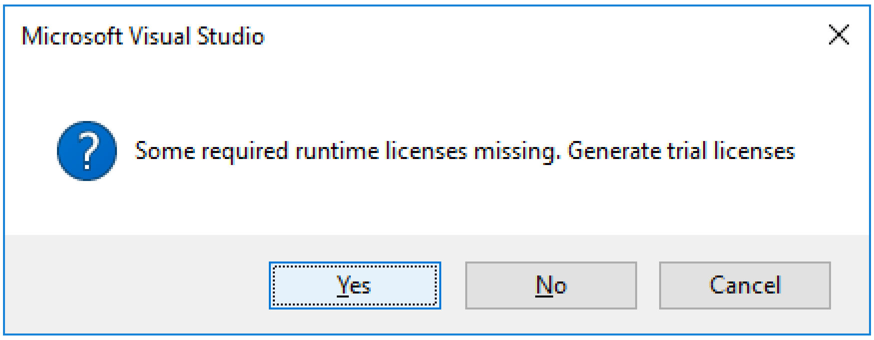
⇨ With the next Activate configuration [▶ 135] you have activated the position lag monitoring for the selected NC axis.

# 8.4 Moving the axis

## 8.4.1 Activating a TwinCAT project

✓ Execute the TwinCAT project with the following steps.

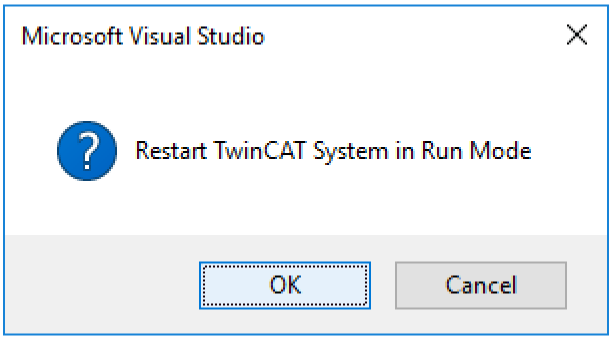
1. Activate the configuration via  or via the Visual Studio **Menu > TwinCAT > Activate Configuration**.
2. If no licenses have been activated yet on the target system, the following dialog appears. For an initial test you can create a trial license by selecting **Yes** and entering a security code.




⇨ You will then be asked whether you want to restart TwinCAT.

3. Confirm the restart with **Yes**.

   Alternatively, a restart can be triggered via or via the Visual Studio **Menu > TwinCAT > Restart TwinCAT System**.




⇨ The TwinCAT project is activated and the axes can be moved manually via the commissioning dialog [▶ 136].

⇨ If, on the other hand, you want to move an axis via the PLC, create a PLC in advance according to Simple movement via the PLC [▶ 137] and activate the TwinCAT project again.

4. Log in the PLC via or via Visual Studio **Menu > PLC > Login**.

5. Then start the PLC via or via the Visual Studio **Menu > PLC > Start**.

## 8.4.2 Manual movement via the commissioning dialog

After creating and configuring an axis, you can move it via the Commissioning dialog [▶ 38]. The prerequisite for this is that the TwinCAT project has been activated [▶ 135] and no link to the PLC exists or the PLC has not been started. Otherwise, the PLC has priority.

**⚠ DANGER**

**Risk of injury due to movement of axes!**

The commissioning results in a movement of axes.

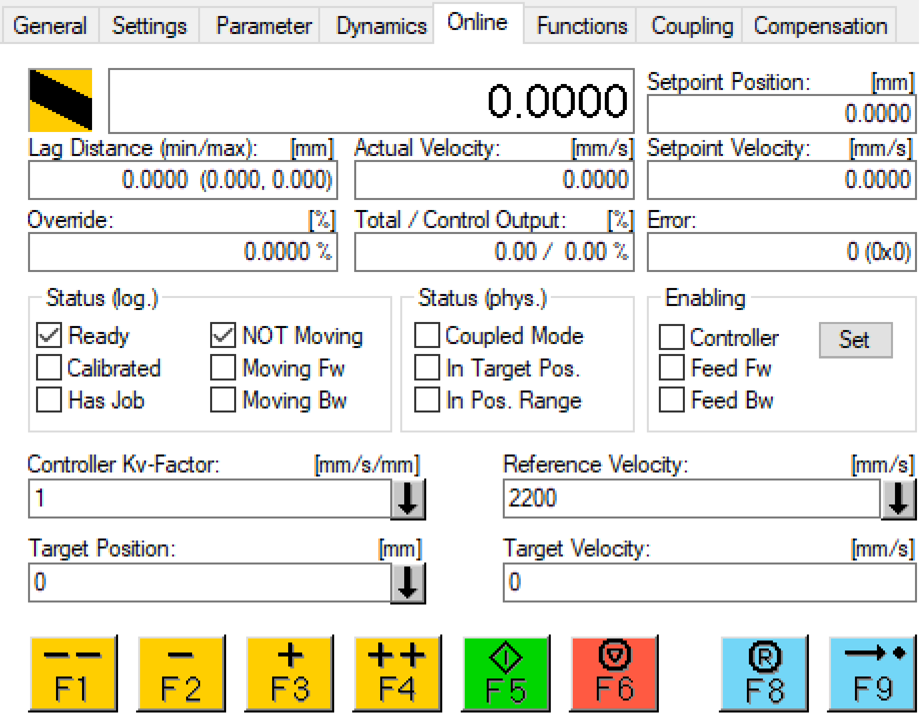
- Make sure that neither you nor others are harmed by the movement, e.g. by maintaining a suitable safety distance.
- Do not perform any action whose consequences you cannot estimate

**⚠ WARNING**

**Incorrect axis position during initial commissioning**

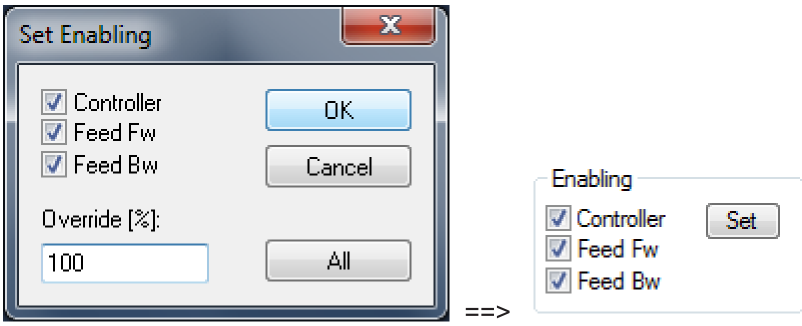
Without referencing / calibrating the axis position, the displayed axis position may deviate from the actual axis position.

- Perform a homing to determine the correct actual position using a reference signal.



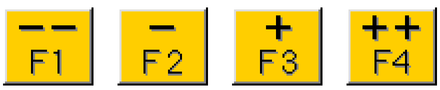

**Enable axis**

Via **Enabling > Set** the corresponding dialog opens, with which the axis and the movement directions can be enabled and the override can be set. Only when the axis and the direction of movement are enabled, it is possible to move the axis in the corresponding direction of movement.




**Manual movement of the axis**

The axis can be moved manually via the buttons or function keys F1-F4. The velocity set in the axis parameters [▸ 76] is used for "Manual Velocity (Fast)" or "Manual Velocity (Slow)".




## 8.4.3 Simple movement via the PLC

After creating and configuring an NC axis, you can create a PLC project and move the axis via this.

**⚠ DANGER**

**Risk of injury due to movement of axes!**

The commissioning results in a movement of axes.

- Make sure that neither you nor others are harmed by the movement, e.g. by maintaining a suitable safety distance.
- Do not perform any action whose consequences you cannot estimate

**⚠ WARNING**

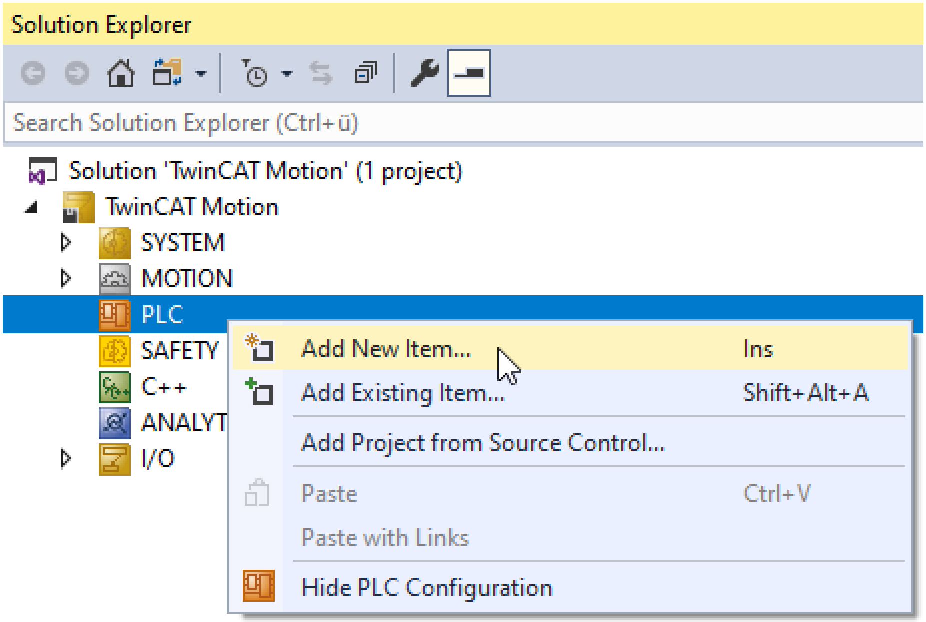
**Incorrect axis position during initial commissioning**

Without referencing / calibrating the axis position, the displayed axis position may deviate from the actual axis position.

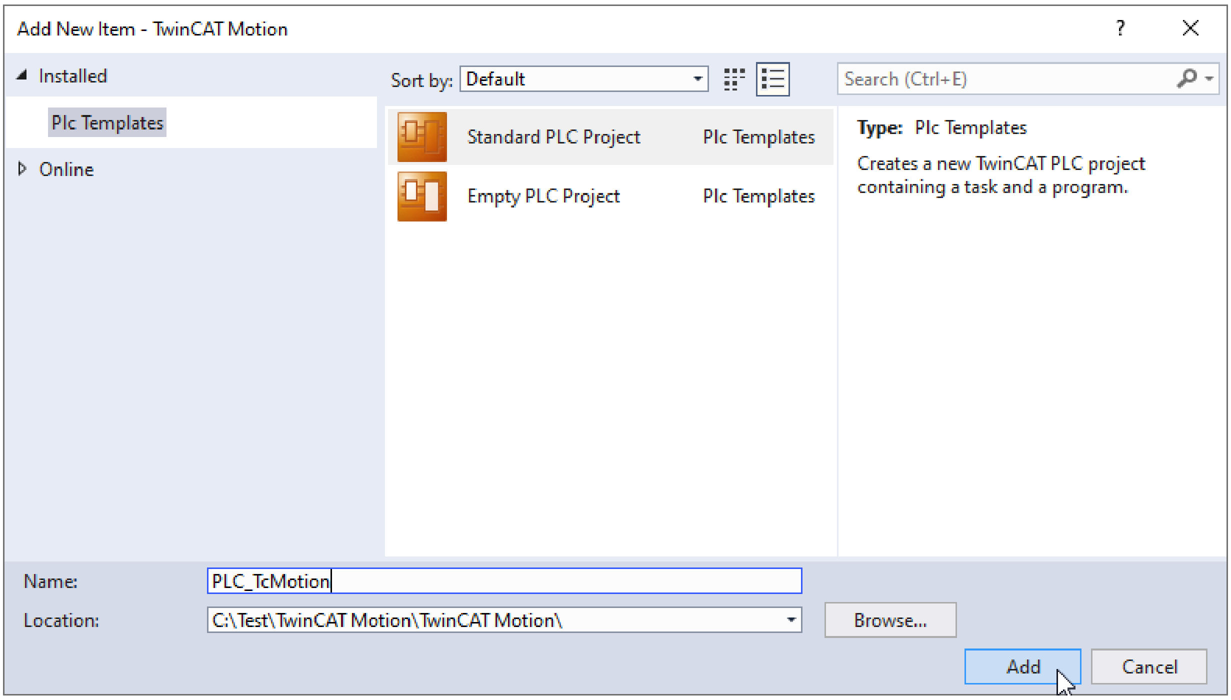
- Perform a homing to determine the correct actual position using a reference signal.

### Creating a PLC project

1. Right click in the Solution Explorer on **PLC > Add new Item...**



2. Select the name and location in the dialog **Add New Item**.

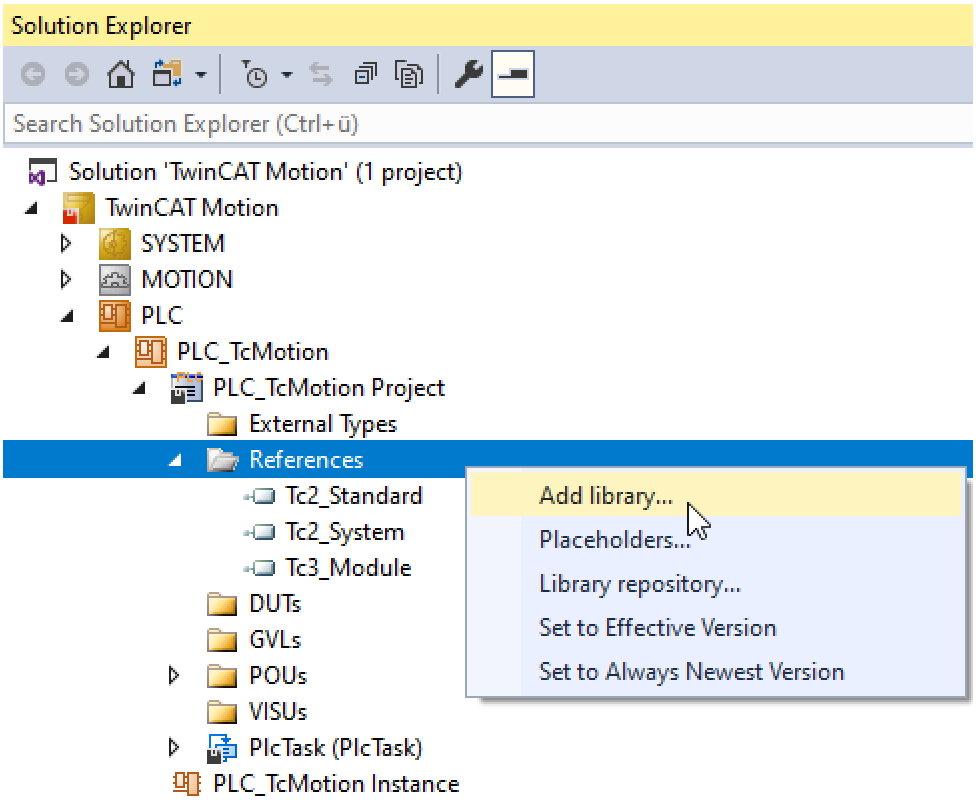



⇨ The PLC project was created.

### Integrating a PLC library

The Tc2_MC2 library contains the PLCopen specified Motion Control function blocks. These can be used to manage NC axes and program movements.

1. Right click on **References > Add library...**




2. Select the library Tc2_MC2 and confirm with **OK**.

⇨ The library is now added and can be viewed by double-clicking it.

### Writing a PLC program

All necessary data types and function blocks for a simple PLC Motion program are contained in the library Tc2_MC2.

### AXIS_REF

For each axis one instance of the data type AXIS_REF is required, which is the interface between PLC and NC. It contains all the information about the axis that is given to the MC function blocks as a reference.

### MC_Power

The function block MC_Power is used to enable an axis and its directions of movement.

### MC_Reset

The function block MC_Reset can be used to reset an error on an axis.

### MC_MoveAbsolute

MC_MoveAbsolute is a simple Motion function block that can be used to position an axis to an absolute target position.

### Simple programming

1. Open the `MAIN(PRG)` in the PLC project at POUs.
2. Add the following declarations to `MAIN(PRG)`.

```
PROGRAM MAIN
VAR
    axis            : AXIS_REF;
    fbPower         : MC_Power;
    fbStop          : MC_Stop;
    fbReset         : MC_Reset;
    fbMoveAbsolute  : MC_MoveAbsolute;
    enableAxis      : BOOL;
    executeStop     : BOOL;
    executeReset    : BOOL;
    executeMove     : BOOL;
    override        : LREAL := 100;
    position        : LREAL := ???; // ToDo: set to a reachable position
    velocity        : LREAL := ???; // ToDo: set velocity for move absolute
END_VAR
```

3. Insert the following program code into `MAIN(PRG)`.

```
fbPower(
        Axis            := axis,
        Enable          := enableAxis,
        Enable_Positive := enableAxis,
        Enable_Negative := enableAxis,
        Override        := override,
        BufferMode      := ,
        Options         := ,
        Status          => ,
        Busy            => ,
        Active          => ,
        Error           => ,
        ErrorID         =>);
```

```
fbStop(
        Axis            := axis,
        Execute         := executeStop, // The command is executed with a positive edge.
        Deceleration    := , // If the value is ≤ 0, the deceleration
                             // parameterized with the last Move command is used.
        Jerk            := , // If the value is ≤ 0, the jerk parameterized
                             // with the last Move command is used.
        Options         := ,
        Done            => ,
        Busy            => ,
        Active          => ,
        CommandAborted  => ,
        Error           => ,
        ErrorID         => );
```

```
fbReset(
        Axis            := axis,
        Execute         := executeReset, // The command is executed with a positive edge.
        Done            => ,
        Busy            => ,
        Error           => ,
        ErrorID         => );
```

```
fbMoveAbsolute(
        Axis            := axis,
        Execute         := executeMove, // The command is executed with a positive edge.
        Position        := position,
        Velocity        := velocity,
        Acceleration    := , // If the value is 0, the standard acceleration from
                             // the axis configuration in the System Manager is used.
        Deceleration    := , // If the value is 0, the standard deceleration from
                             // the axis configuration in the System Manager is used.
        Jerk            := , // If the value is 0, the standard jerk from the axis
                             // configuration in the System Manager is applied.
        BufferMode      := MC_BufferMode.MC_Buffered,
        Options         := ,
        Done            => ,
        Busy            => ,
        Active          => ,
        CommandAborted  => ,
        Error           => ,
        ErrorId         => );
```

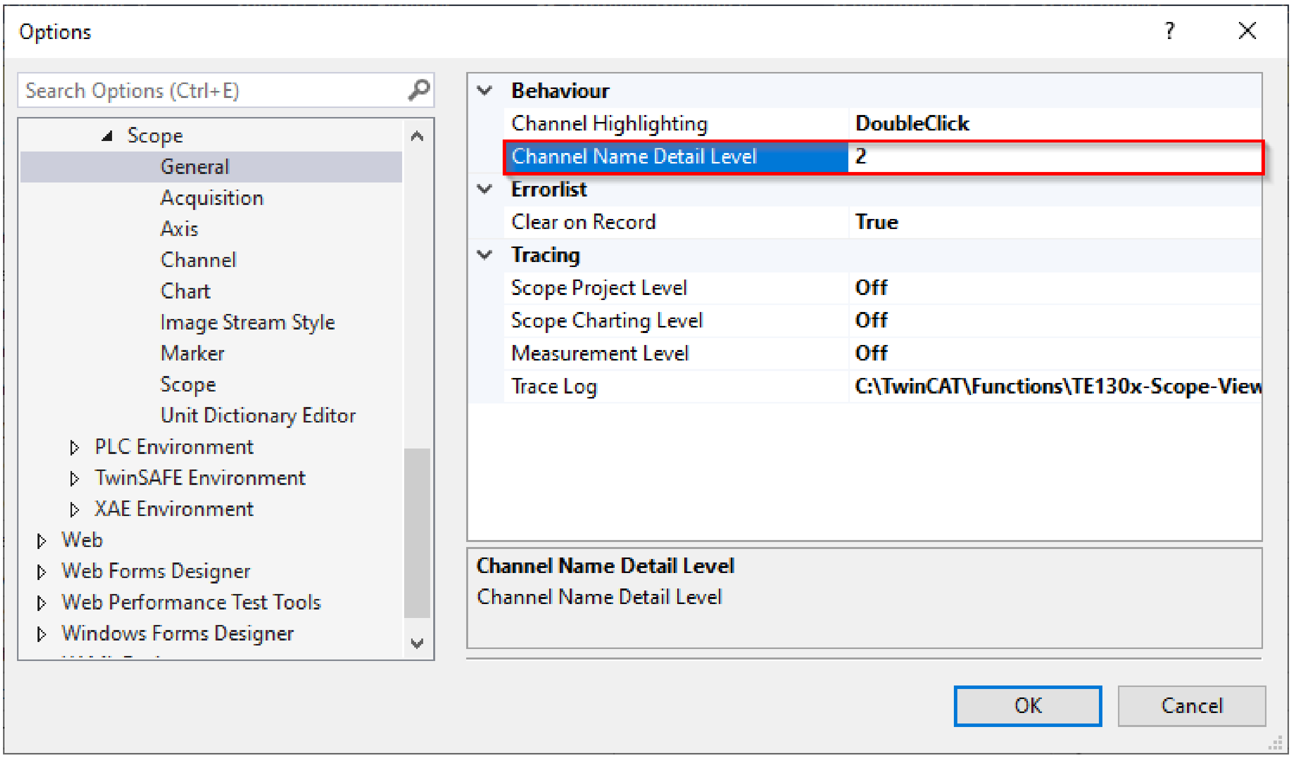
4. For the call from `fbMoveAbsolute`, adjust the target position and the dynamics according to your real axis.
5. Build the PLC project.
   ⇨ The `axis` instance of the `AXIS_REF` should now be displayed among the PLC instances in the Solution Explorer.
6. Link the PLC instance of `AXIS_REF` with the axis instance of the NC.
7. Activate the TwinCAT project  .
8. Log in the PLC  and start it  , see Activate TwinCAT project [▶ 135].
9. Check that the target position specified at `fbMoveAbsolute` can be approached safely with the specified dynamics. Adjust them as needed.
10. If you are sure that there is no danger from the movement specified at `fbMoveAbsolute`, you can enable the controller for the axis by setting the variable `enableAxis` online to `TRUE`.
11. If the controller enable was successful (`fbPower.Active = TRUE`), the motion command can then be activated with `executeMove`.
    ⇨ Via `fbStop.Execute = TRUE` the motion command can be stopped prematurely if required.
    ⇨ An axis error can be reset via `fbReset.Execute := TRUE`.

## 8.5 Scope integration

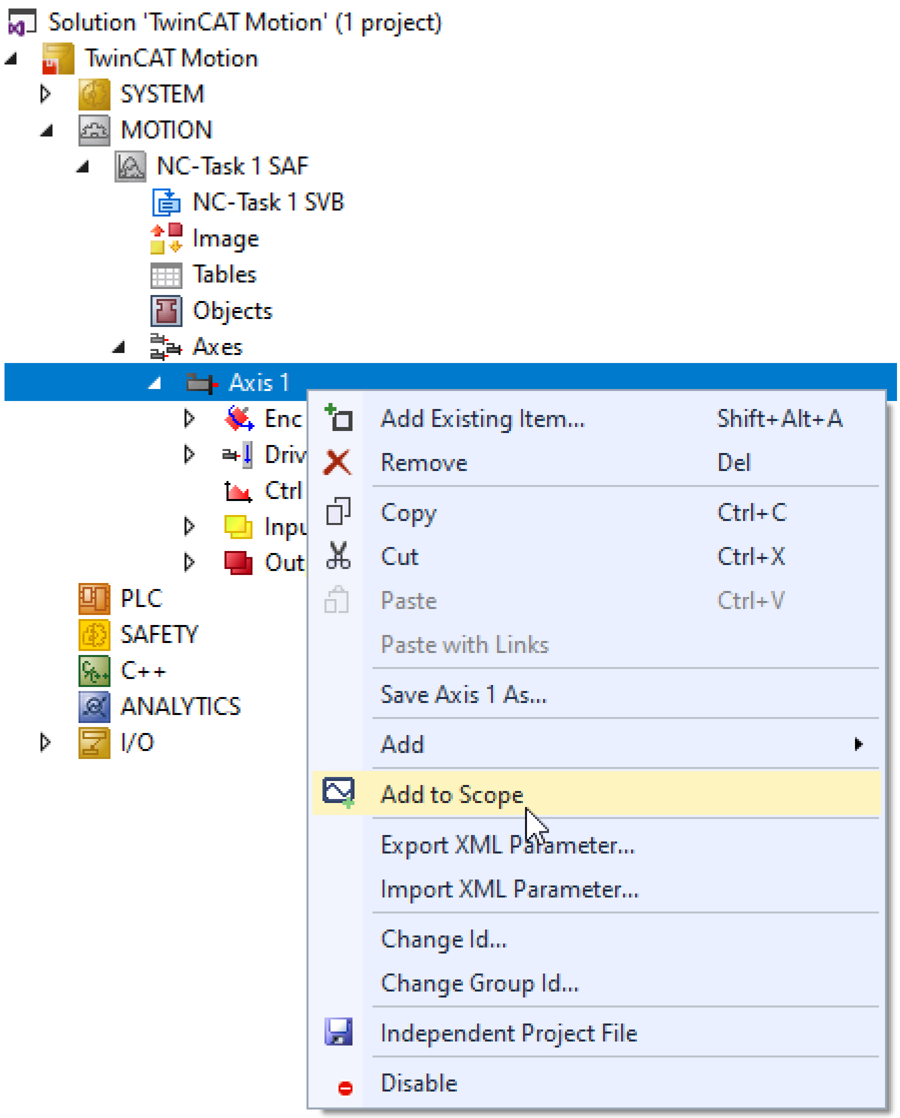
TwinCAT 3 Scope can be used to record variable values and display them in various diagrams. For a PTP axis, for example, position and dynamics in an XT diagram may be of interest. There is a simplified way to do this, which is described below. Further details can be found in the TE13xx | TwinCAT 3 Scope View documentation.

✓ Automatically name all variables according to the scheme AxisName.ParameterName (recommended procedure).

1. Open the Scope settings at **Menu > Scope > Options**.
2. At **Scope > General** set the **Channel Name Detail Level** to "2".

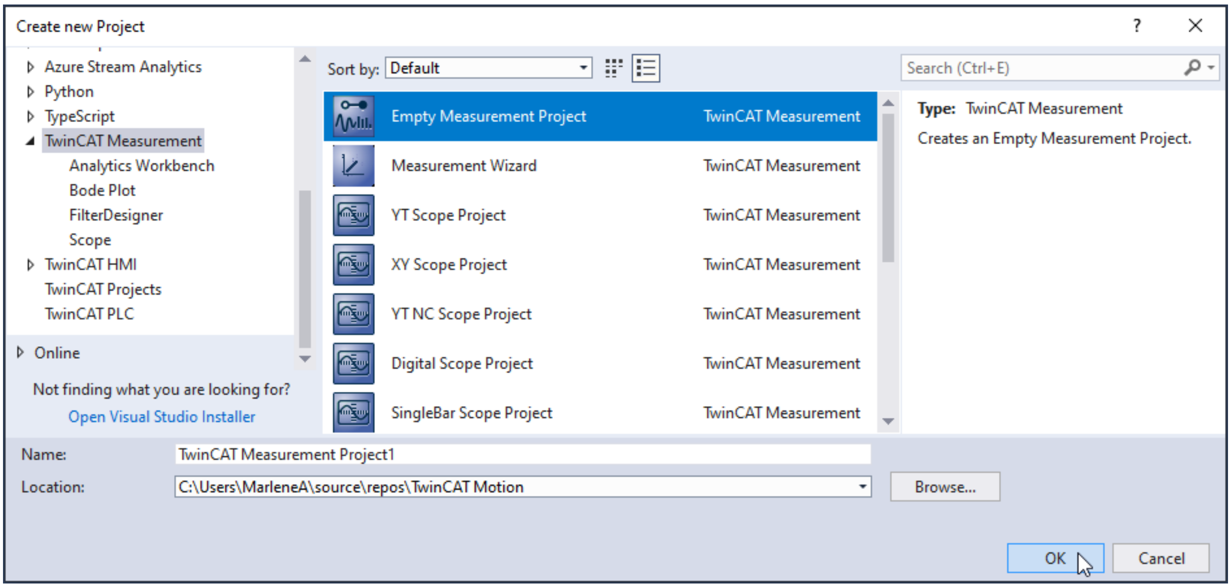



3. Right click on the **NC axis (Axis 1)** in the **Solution Explorer** and select **Add to Scope**.



⇨ If there is no Measurement Project in the Solution yet, the dialog **Create new Project** opens.

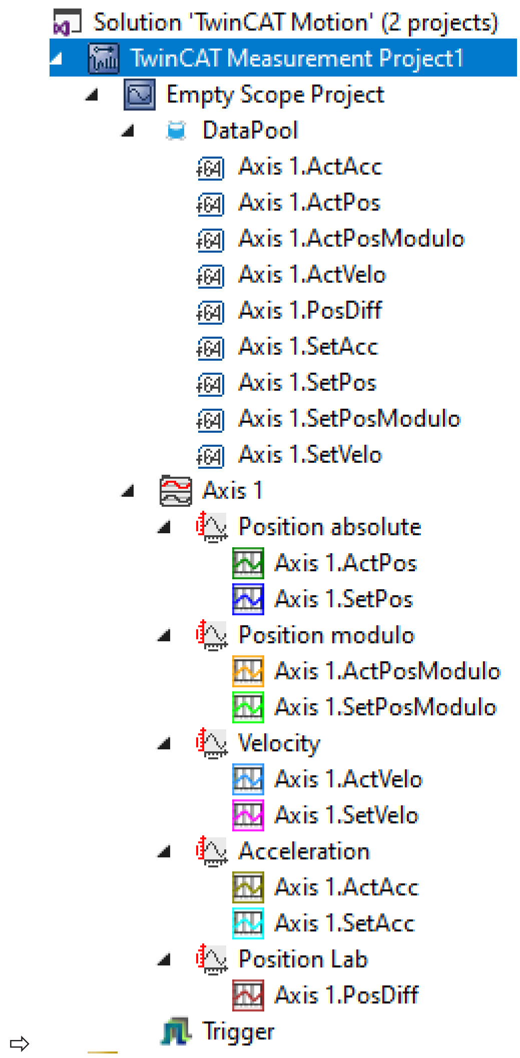
4. Select **Empty Measurement Project** as the project type and a name for your Measurement Project and confirm with **OK**.

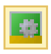



⇨ A Measurement Project has been created.

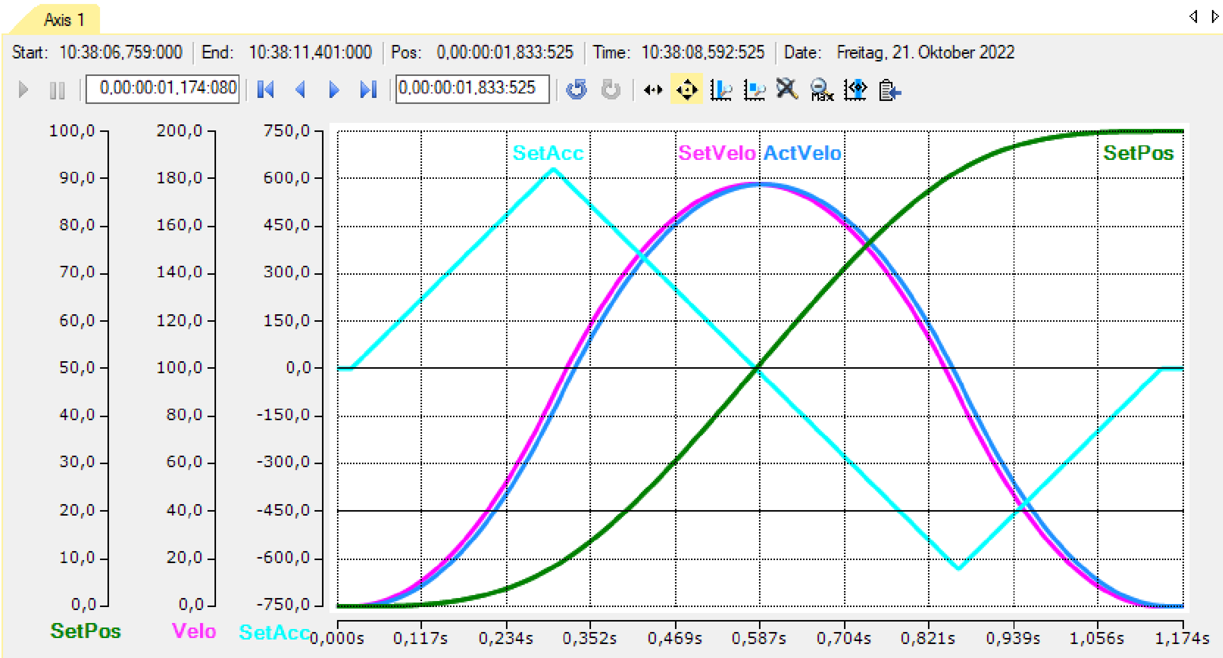
⇨ The Measurement Project contains a Scope Project.

⇨ Variables for position, velocity and acceleration of the selected axis are already added to the Scope **Data Pool** and to a YT diagram.




5. Activate the TwinCAT project  .
6. Switch TwinCAT to the Run mode  .
7. Start Scope recording  .
8. Move the axis (via the Commissioning dialog [▶ 136] or via the PLC [▶ 137]).
   **DANGER** **Observe the safety instructions!**
9. Stop Scope recording  .

⇨ Positions and dynamics of the axis have been recorded and can be displayed by the Scope View.




For further details, such as adding more variables to the **Data Pool**, and creating new/different diagrams, please refer to the TE13xx | TwinCAT 3 Scope View documentation.

# 9 Special NC functionalities

## 9.1 Modulo positioning

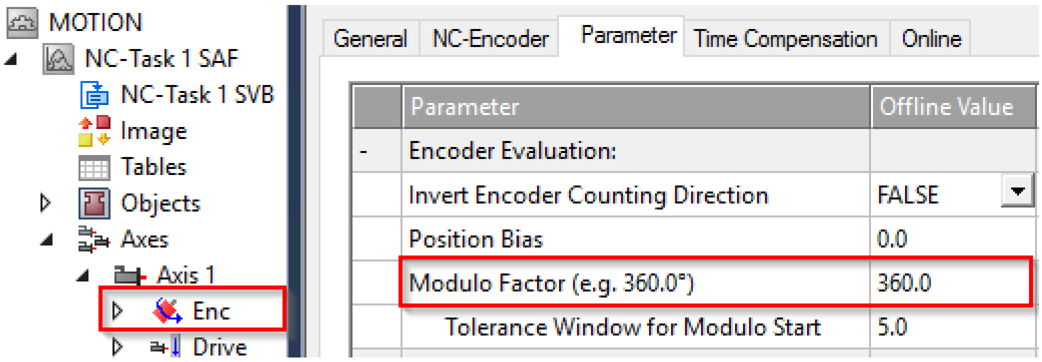
The modulo positioning can be applied to closed linear axes as well as to rotary axes. TwinCAT does not distinguish between these types. A modulo axis has a consecutive absolute position in the range ±∞. The modulo position of the axis is simply a piece of additional information about the absolute axis position. Modulo positioning represents the required target position in a different way. As opposed to absolute positioning, in which the user clearly specifies the target, the absolute target position is formed from the following parameters in modulo positioning:

- Modulo target position
- Modulo Factor
- Tolerance Window
- Direction, see MC_DIRECTION
- (Additional Turns, see ST_MoveAbsoluteCAOptions)

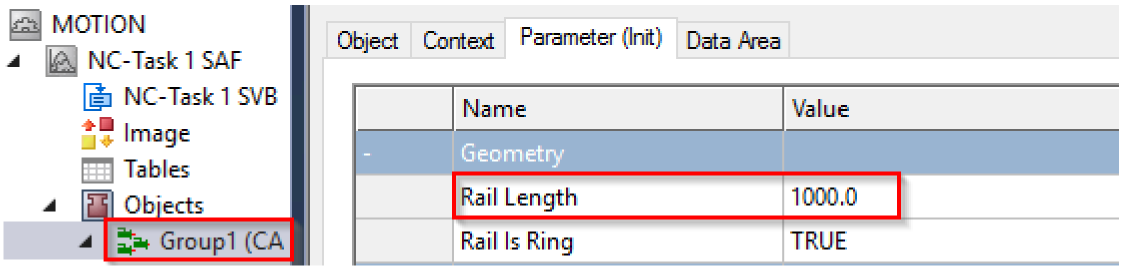
### Modulo Factor

The modulo positioning basically refers to an adjustable Modulo Factor, which is set in the TwinCAT Engineering. The axis and its use must be observed here, for example:

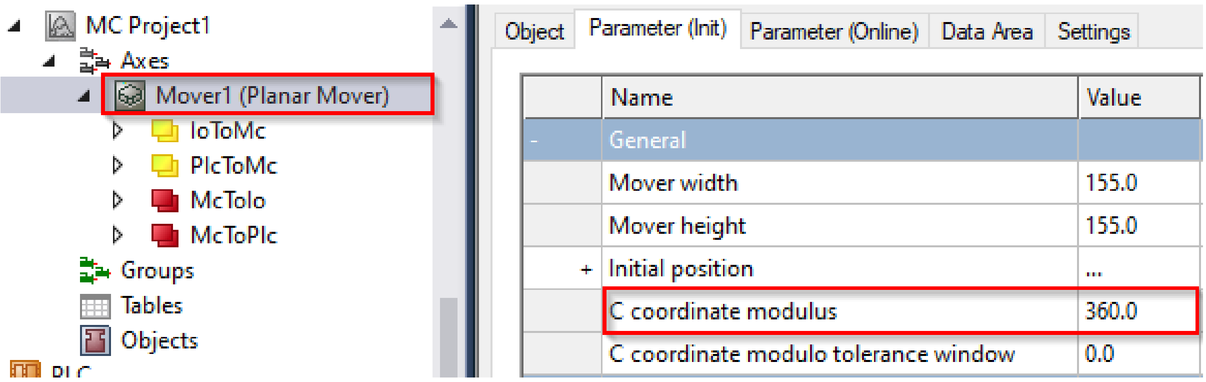
- If a PTP axis is used, the Modulo Factor of the axis encoder applies; details in the Notes on the modulo positioning of a PTP axis.




- If, for example, a mover is used on an XTS system in a CA group, the Rail Length set in the CA group applies.



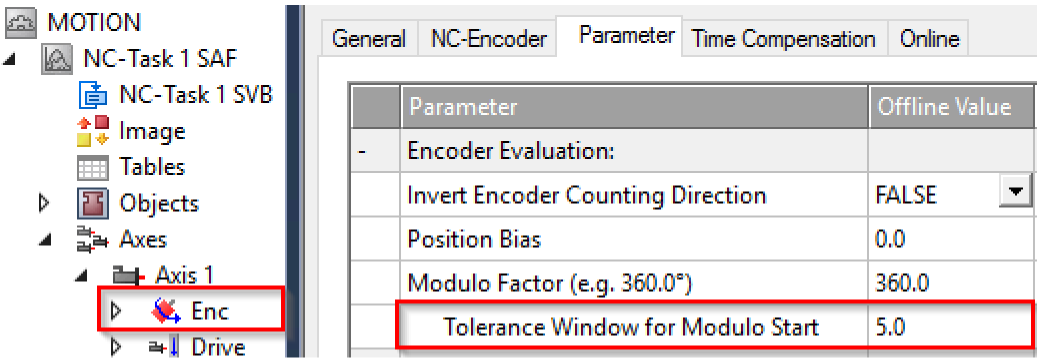
- If an XPlanar mover is used, its "C-axis" modulo can be positioned. Here, the Modulo Factor is set as "C coordinated modulus" in the Init parameters of the XPlanar mover.



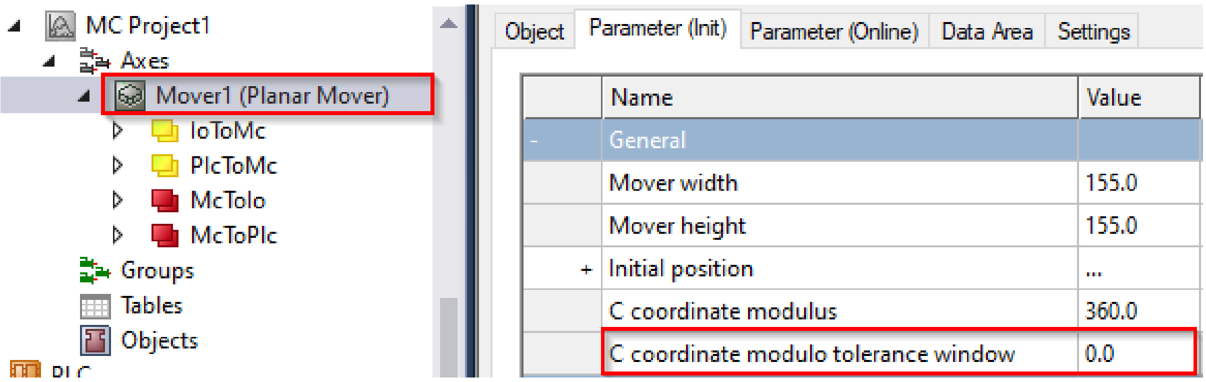

### Modulo Tolerance Window

The Modulo Tolerance Window defines a position window around the current modulo set position of the axis. The window width corresponds to twice the specified value (set position ± tolerance value) and is specified in the TwinCAT Engineering:

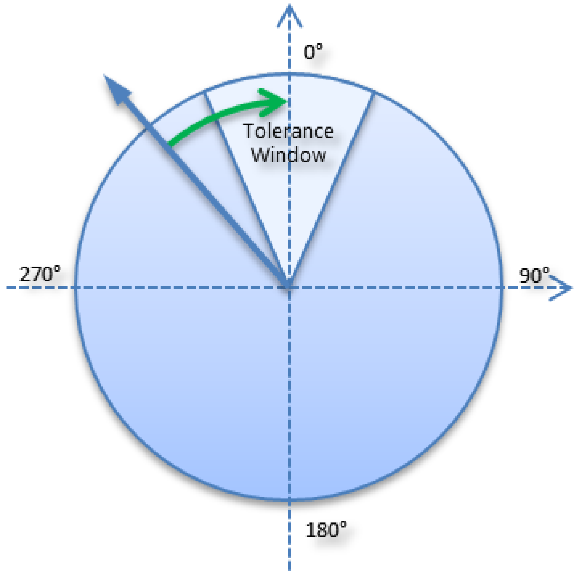
- In the case of a PTP axis or an axis in a CA group, the Tolerance Window is defined in the axis encoder

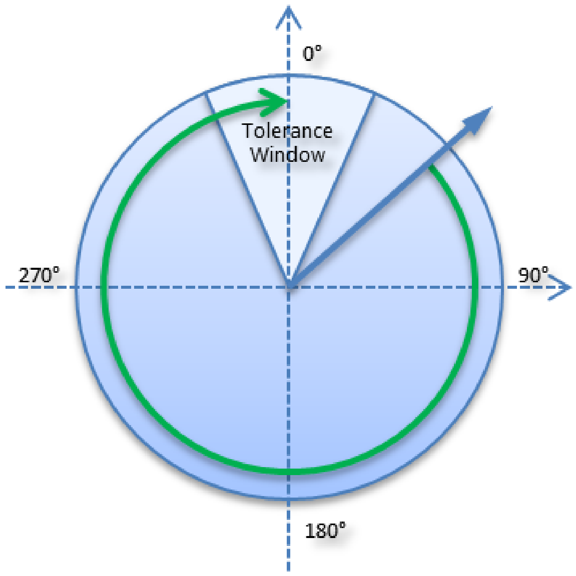


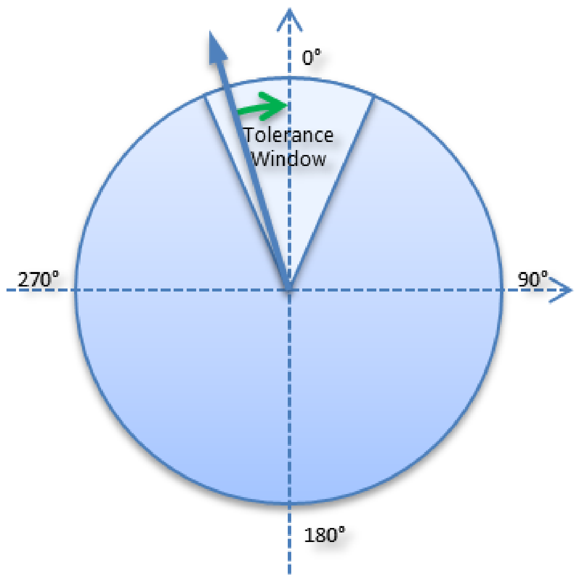

- In the case of the C-axis of an XPlanar mover, the Tolerance Window is defined in the Init parameters of the XPlanar mover.

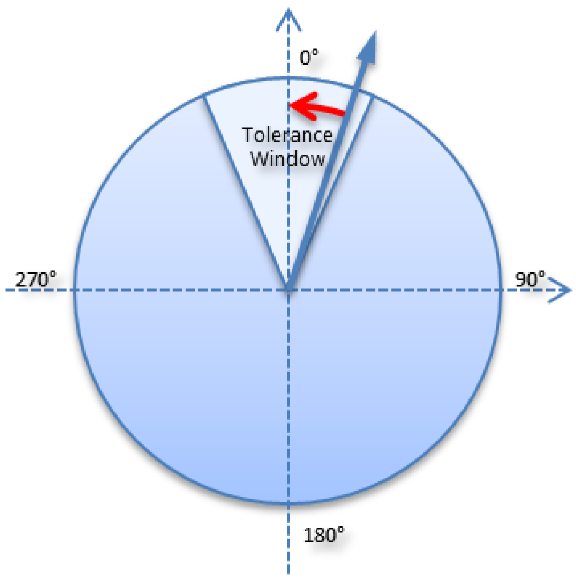



The positioning of an axis is always referenced to its current actual position. Unintentional revolutions may be performed if the actual position and the target position are very close to each other, for example, if the actual position is minimally greater than the target position and `Direction = mcDirectionPositive` was selected. This can occur in particular if the actual position is determined inaccurately (e.g. on account of incorrect positioning due to the axis stalling, or due to the finite resolution of the encoder). In order to avoid this, a tolerance window for modulo positioning can be set. If the distance between the start and target positions is less than or equal to the Tolerance Window, then the target position is driven to by the shortest route (as with `Direction = mcDirectionShortestWay`), i.e. also contrary to the `direction` specified.




**Examples**

- Modulo Factor = 100
- Tolerance Window = 1

| Parameter Direction | Absolute Startposi-tion | Target posi-tion | Parameter Additional Turns | Relative path | Absolute end posi-tion | Modulo end position |
|---|---|---|---|---|---|---|
| mcDirectionPositive | 110 | 10 | 0 | 0 | 110 | 10 |
| mcDirectionPositive | 110.9 | 10 | 0 | -0.9 | 110 | 10 |
| mcDirectionPositive | 112 | 10 | 0 | 98 | 110 | 10 |
| mcDirectionPositive | 95 | 10 | 0 | 15 | 110 | 10 |
| mcDirectionPositive | 110 | 110 | 0 | ERROR: INVALID TARGET POSITION | | |
| | | | | | | |
| mcDirectionPositive | 110 | 10 | 3 | 300 | 410 | 10 |
| mcDirectionPositive | 110.9 | 10 | 3 | 299.1 | 410 | 10 |
| mcDirectionPositive | 112 | 10 | 3 | 398 | 410 | 10 |

| Parameter Direction | Absolute Startposition | Target position | Parameter Additional Turns | Relative path | Absolute end position | Modulo end position |
|---|---|---|---|---|---|---|
| mcDirectionPositive | 95 | 10 | 3 | 315 | 410 | 10 |
| mcDirectionPositive | 110 | 110 | 3 | ERROR: INVALID TARGET POSITION | | |
| | | | | | | |
| mcDirectionNegative | 110 | 10 | 0 | 0 | 110 | 10 |
| mcDirectionNegative | 109.9 | 10 | 0 | 0.1 | 110 | 10 |
| mcDirectionNegative | 108 | 10 | 0 | -98 | 10 | 10 |
| mcDirectionNegative | 95 | 10 | 0 | -85 | 10 | 10 |
| mcDirectionNegative | 110 | 110 | 0 | ERROR: INVALID TARGET POSITION | | |
| | | | | | | |
| mcDirectionNegative | 410 | 10 | 3 | -300 | 110 | 10 |
| mcDirectionNegative | 409.9 | 10 | 3 | -299.9 | 110 | 10 |
| mcDirectionNegative | 408 | 10 | 3 | -398 | 10 | 10 |
| mcDirectionNegative | 495 | 10 | 3 | -385 | 10 | 10 |
| mcDirectionNegative | 410 | 110 | 3 | ERROR: INVALID TARGET POSITION | | |
| | | | | | | |
| mcDirectionShortestWay | 440 | 50 | 0 | 10 | 450 | 50 |
| mcDirectionShortestWay | 440 | 10 | 0 | -30 | 410 | 10 |
| | | | | | | |
| mcDirectionShortestWay | 440 | 50 | 1 | ERROR: INVALID ADDITIONAL TURN COUNT | | |

**Further samples**

Further examples without the Additional Turns parameter can be found in the Notes on the modulo positioning of a PTP axis.

## 9.2 External set value generation

External setpoint generation (external setpoint specification) enables the implementation of individual setpoint generators within the PLC, or appropriate connection of existing internal and external generators, and their simple and transparent integration into the TwinCAT system. Axes can be positioned either with an external setpoint generator alone, or superposition (mixing) of existing internal generators and external setpoint sources is possible.

These characteristics and the fact that an axis can change its different operation modes at runtime (e.g. interpolating path axis (NCI), master PTP axis, slave axis, external setpoint generation) further increase the flexibility of a TwinCAT axis.
The superimposition of internal and external setpoint sources generally enables a wide range of new applications and solutions.

**Examples of use:**

- External setpoint specification can be used for implementing an *individual new setpoint profile*, which is not available in TwinCAT NC. This profile can be mathematically very simple or as complex as necessary. Since the externally generated setpoint profile can also be superimposed on the standard profile, *corrective functions* can thus also be supplemented.
  In this way, a position-based setpoint profile can be implemented for systems that are difficult to control or non-linear, for example a position-based deceleration ramp for hydraulic axes.
- One application for the *superposition (mixing) of two internal setpoint sources* is the combination of interpolating path processing (NCI) and the 'flying saw' function (PTP slave). Multi-dimensional *machining*, e.g. a circle, *on a moving workpiece* is thus possible, which previously was only possible with stationary workpieces. This measure increases the efficiency of a machine.

- The external setpoint generation can also be used in order to enable a certain *lead time* (in the multi-digit second range) between the unit generating the setpoints (e.g. the execution of a DIN-NC program feeding virtual axes) and the real axes. *Collision monitoring and strategies for collision avoidance* can then be realized.
- A further important application are *coordinate transformations* (restricted to regular and unambiguous transformations without singularities), e.g. geometries such as a 'two-bar linkage' or conversion between rotary and translatory coordinates.

This new function is available with the PLC function blocks MC_ExtSetPointGenEnable and MC_ExtSetPointGenDisable in the *Tc2_MC2* library. Furthermore, there is an example for an external setpoint generator (3-phase profile with acceleration rectangle), see section TwinCAT NC-PTP Examples.

**Activation of the external setpoint generation:**

The adoption of the external setpoint specifications is initiated by an ADS axis function from the PLC with the function block *MC_ExtSetPointGenEnable*. The axis start types *absolute (1)* or *relative (2)* and, if sensible and required, the target position of the axis, are specified as parameters. This target position has either an absolute or relative effect, depending on the starting mode, and enables position and target position window monitoring (PEH). Thus, enabling the external setpoint generation is similar to a conventional axis start.
A status bit of the axis reference *Axis.Status.ExtSetPointGenEnabled* informs about the status, namely whether the external setpoint generation, is enabled or disabled.

So that enabling is accepted without errors:

- no axis error should be present
- the controller and feed enables must be active
- the drive hardware must be ready to operate
- the axis must not be coupled (no slave)

To be observed:

- In case of an axis error (runtime error of the axis) or an axis reset (including edge change of the software controller enable) the external setpoint specification is automatically disabled (similar to an abortion of PTP axis positioning in case of a runtime error or axis reset).
- If only external setpoint generation is carried out, both *absolute* and *relative* start types are permitted.
  If external setpoint generation is started in parallel or in addition to internal setpoint generation (PTP), only the *relative* starting mode is possible, since the internal positioning is ultimately always converted to an absolute start, and it is not possible to have two absolute positions simultaneously.
- Whilst external positioning can be activated during an active internal PTP positioning via the *Relative* start type, the opposite case is not permitted.
- If the external setpoint generation is already active and this is activated for the repeated time, then the start types must match, otherwise this is answered with an error.
  The repeated activation of the external setpoint generation can be useful insofar as here the target position is taken over each time and thus the target position window monitoring (PEH) is updated.

**Disabling the external setpoint generation:**

External setpoint specification is switched off from the PLC via the function block *MC_ExtSetPointGenDisable*. A feedback is given by the already mentioned status bit of the axis reference *Axis.Status.ExtSetPointGenEnabled.*

After disabling the external setpoint generation, the external target position is taken over for exactly one more cycle.

**Specification of setpoints via the cyclic axis interface:**

After faultless activation of the external setpoint generation, in each PLC cycle a setpoint packet can now be created at the axis interface. A setpoint package consists of the external set position, target velocity, set acceleration and set direction (-1, 0, +1).
The set direction is of particular importance here, because setpoints are only accepted and applied internally if this is not equal to zero (0: no movement). For the complete range of external setpoint generation, the direction flag therefore has to be set overlapping at the start and at the end.

To be observed:

- After resetting of the set direction to zero, the external target position is taken over for exactly one further SAF cycle.
- The set direction should already be set to a value not equal zero (i.e. *-1* or *+1*) one cycle before and for one more cycle after the motion. This provides a safe time frame that is guaranteed to prevent a value being ignored, either at the start or at the end.

**Settings of the PLC task:**

The PLC task that generates and specifies the setpoints must operate synchronously, i.e. with the same cycle time as the NC SAF task (port 501). This PLC task also has to have the same high priority, perhaps an even higher one, than the NC-SVB task (port 511).

**Optimization options:**

Dead time caused by transport via the cyclic axis interface can be compensated for an axis. This dead time compensation can be set by means of the dialogs on the encoder and drive of the NC axis (ON (with velocity)).

## 9.3 NC Backlash Compensation

### 9.3.1 Mechanical backlash

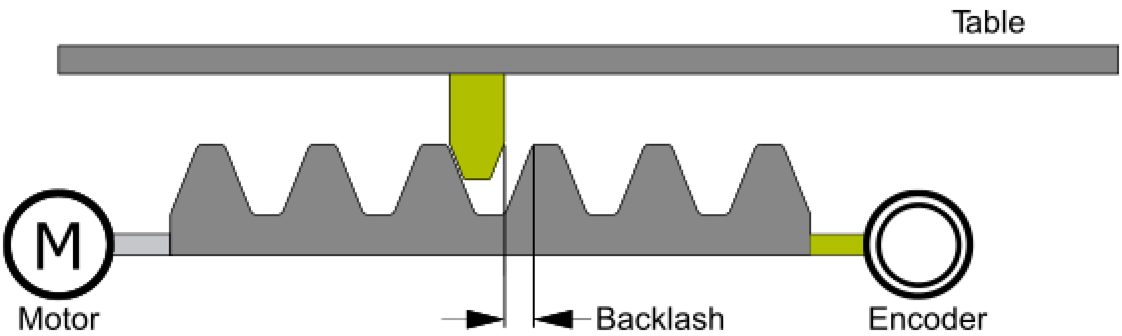
Mechanical backlash is the difference in position between a drive or an encoder and the load. Mechanical backlash arises due mechanical tolerances in the drivetrain. This causes a difference between the required and the actual position of the load. This is especially important when the direction of motion is reversed.

There are three types of mechanical backlash:

**Positive backlash**

Positive backlash occurs in systems where the measuring system is directly coupled to the drive. In this case the backlash exists between the drive and the load. When the direction of motion is reversed, the measuring system will detect a change in position before the load has moved. So the encoder, which here measures the position of the load indirectly, leads the actual position of the load. As a consequence, the load will not reach the set position, it will be shorten by the length of the backlash.

In the figure below, a movement from left to right is defined as a positive movement.

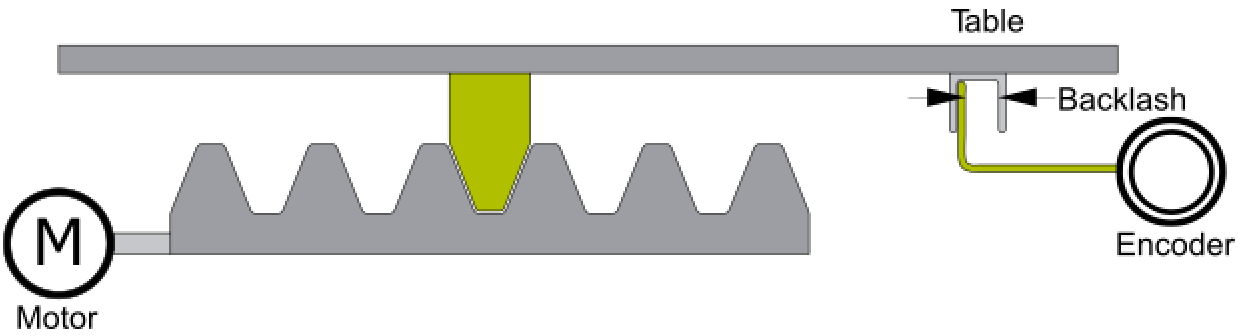


The encoder leads the load (e.g. machine table), so the measured encoder position, leads the actual position of the table. Therefore the table's movement will be too short.

In this case enter a positive correction value for the backlash (= normal case).

**Negative backlash**

Negative backlash occurs in systems where a mechanical tolerance exists between the drive and the measuring system. When the direction of motion is reversed the load immediately moves in the new direction and the measuring system won't detect the changed position. In this case the load will travel further than required. The encoder, which measures the position of the load directly, lags behind the actual position of the machine part.

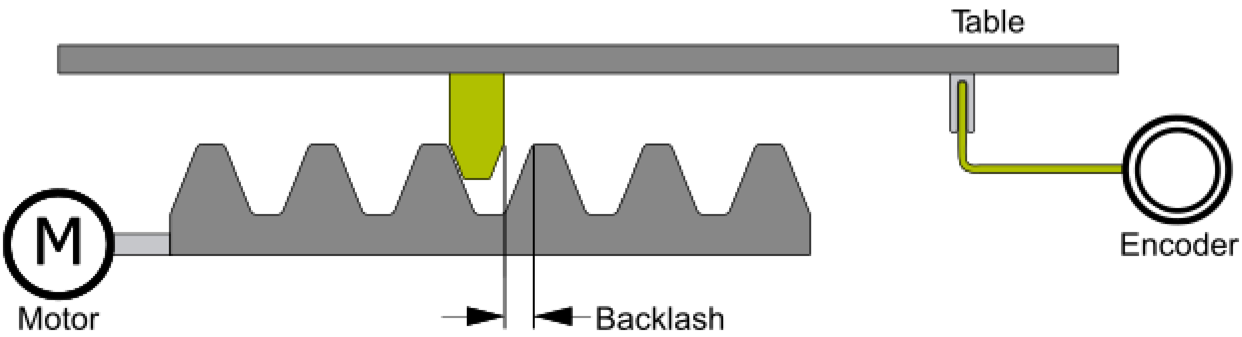



The encoder lags behind the actual position of the load (e.g. machine table). Therefore the table will travel too far.

In this case enter a negative correction value for the backlash.

**Neutral backlash**

In systems, neutral backlash is an exception. In this case the measuring system is directly coupled to the load and electrically connected with the drive. Here the encoder position and the load position are the same. Now when the drive's direction of motion is reversed, the backlash will be automatically compensated. The position control loop is closed around the drivetrain because the encoder is coupled to the load directly. The set position can be reached with no further compensation.




The encoder is directly coupled with the load (e.g. machine table) and ensuring steady state accuracy. No special settings are required.

**General Hints and Notes:**

- Implemention of positive or negative backlash compensation is the same in TwinCAT (only the sign of the backlash value differs). A positive backlash is parameterized as positive value, a negative as negative value.
- A negative backlash is undesirable, because an axis with a backlash in the encoder system is difficult to control (stationery vibrations / oscillations). Typically there are further steps necessary to solve this problem.

- It is not necessary to differentiate between position interface (position control in the drive) and velocity interface (position control in TwinCAT), because they have the same effect. This applies to all variants of backlashes.

- In case of neutral backlash there is no compensation action necessary, even though there is a mechanical backlash. The encoder system is coupled to the machine table therefore enforcing steady state accuracy.

- If a referencing (homing) of an axis is necessary, do the homing with backlash compensation deactivated also position correction deactivated. The last driving direction defines if the left or the right edge is the point of reference by defining a reference position (see NC Implementing the TwinCAT Backlash Compensation [▸ 153]).

## 9.3.2 NC Implementing of the TwinCAT Position Correction

TwinCAT position correction is used for the backlash compensation.

The following table shows a description of TwinCAT position correction for drives in cyclic position and velocity interfaces.

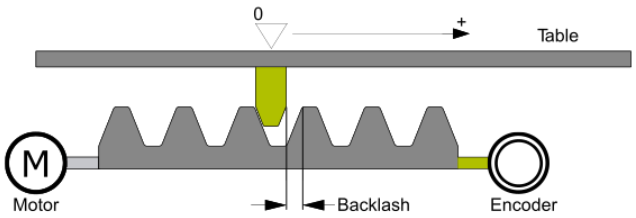
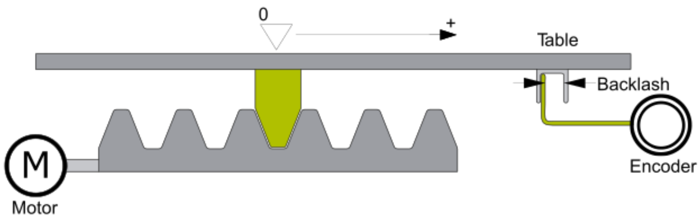
| | **Implementation and effect of Position Correction resp. Backlash Compensation** |
|---|---|
| **Implementation** | 1. The backlash correction is subtracted from the set position which is transmitted to the drive.<br>2. The backlash correction is added to the actual position which is transmitted from the encoder. |
| **Effect in the Position Interface**<br>(position control in the Drive) | Description / Effect:<br>By manipulating the set position which is transmitted to the drive the backlash will be driven (subtraction of position correction, see case 1).<br>To realize that the monitored actual value is correct, this backlash correction is subtracted from the transmitted actual value (addition of the position correction, see case 2). |
| **Effect in the Velocity Interface**<br>(position control in TwinCAT) | Description / Effect:<br>In the velocity interface no set position is transmitted to the drive, so correcting the set position would have no effect.<br>By correcting the actual position which is transmitted from the encoder to the drive a position difference is created. The closed loop position controller in TwinCAT controls, by means of this position difference (“lag error”) that the backlash is driven (addition of the position correction, see case 2). |

## 9.3.3 NC Implementing the TwinCAT Backlash Compensation

Overview of the effect of the backlash compensation differentiating by type of backlash and reference position (left or right edge):

- **Positive backlash**: Drive resp. external encoder has no backlash – backlash value is positive.
- **Negative backlash:** Additional external encoder has a backlash – backlash value is negative.

| | **Implementation and effect of backlash compensation** |
|---|---|
| **Reference position at the left edge** | **Backlash compensation in negative direction**<br>**Positive direction:**<br>• No manipulation<br>**Negative direction:** |

<table>
<tr><th></th><th>Implementation and effect of backlash compensation</th></tr>
<tr><td>
</td><td>• Manipulation of the actual position +backlash<br>• Manipulation of the set position -backlash</td></tr>
<tr><td>Reference position at the right edge<br>
</td><td>Backlash compensation in positive direction<br>Positive direction:<br>• Manipulation of the actual position -backlash<br>• Manipulation of the set position +backlash<br>Negative direction:<br>• No manipulation</td></tr>
</table>

## 9.4 Discrete Axis (two speed)

This TwinCAT axis type (Two-Speed) enables the positioning of a so-called rapid/creep axis. Such an axis can be physically made up of a motor with two speeds (switching of the pole pair numbers), or alternatively by a motor that can be driven at two speeds with the help of a frequency inverter.

The typical positioning of such an axis is first of all made at high speed up to a parameterization distance away from the target position (creep distance in positive or negative direction). From this position you switch to slow traverse, so that the physical velocity (actual velocity) is reduced to a slower constant velocity. The slow traverse is then also switched off at a closer distance from the target (braking distance) and the brake is then activated after a parameterized time (delay time for brake incidence).

This particular positioning sequence is for the sole purpose of ensuring that the axis reaches its target position in the most accurate and repeatable way.

In case a positioning inaccuracy occurs depending on the last physical direction of travel (typical effect of a backlash), then a looping distance is activated. This looping distance has the effect that the target position is always approached from the same direction thus reducing the backlash influence. In case of an axis stop basically the same sequence is run through as for positioning without looping distance. However the priority for an axis stop lies on a shorter braking distance or time and not on the positioning accuracy. There is the separate parameter (looping distance for stop) for stopping in the shortest possible distance.

### General

There are two equal possibilities available for the physical control of the axis, in the form of discrete travel signals.

**Use of the 6 bits in ControlByte**

| | |
|---|---|
| bMinusHigh | Rapid traverse, negative direction |
| bMinusLow | Slow traverse, negative direction |
| bPlusHigh | Rapid traverse, positive direction |
| bPlusLow | Slow traverse, positive direction |
| bBreak | Braking bit |
| bBreakInv | inverted braking bit |

**Use of the 6 bits in ExtControlByte**

| | |
|---|---|
| bDirectionMinus | Negative direction |
| bDirectionPlus | Positive direction |
| bVeloLow | Slow traverse |
| bVeloHigh | Rapid traverse |
| bBreak | Braking bit |
| bBreakInv | inverted braking bit |

A master-slave coupling is not possible with rapid/creep axis.

An axis start will only be initiated if the distance from the target point is in fact larger than the parameterized braking distance.

**Velocity and override**

| Starting velocity value range v | Interpretation of the start velocity at override 100% |
|---|---|
| v > 50 | Rapid traverse |
| 0 < v ≤ 50 | Slow traverse |
| v ≤ 0 | Error |

| Value range override | Interpretation of the override value |
|---|---|
| Override > 50% | Rapid traverse |
| 0% < Override ≤ 50% | Slow traverse |
| Override = 0% | Stationary (tolerance window: <0.01% ) |

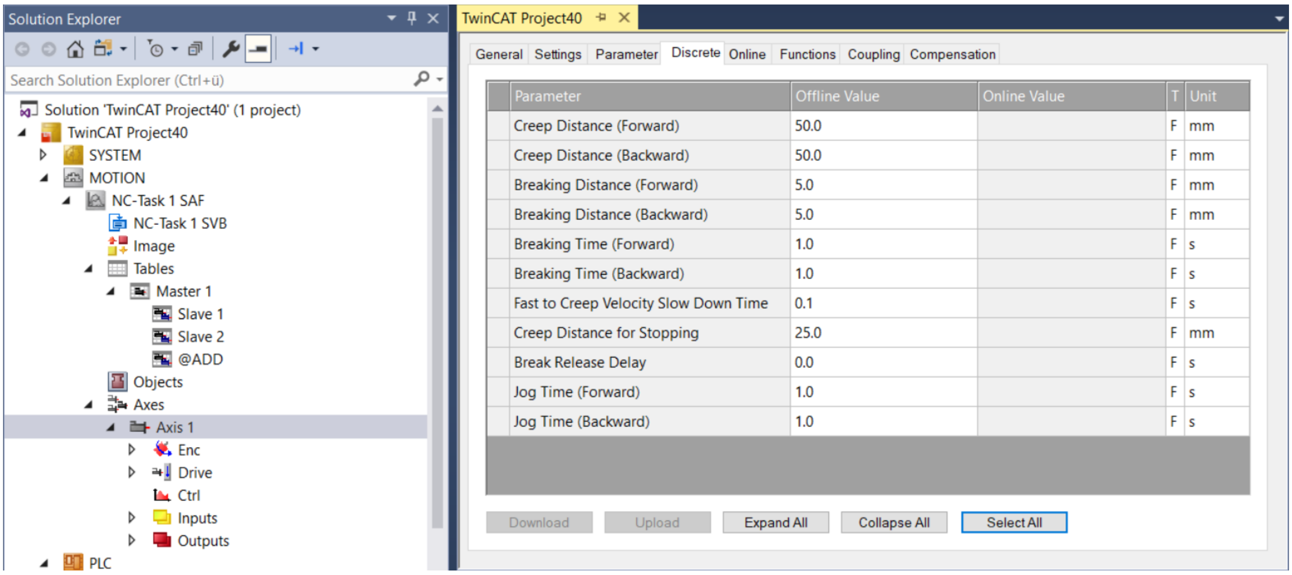
An override modification (also override = 0) only takes effect within the main travel phase.
If the override is set to 0 within one of the braking phases, the initiated braking phase is ended without being influenced.

**I/O configuration: Drive interface for rapid/creep axes NC → I/O (12 bytes)**

| No. | Data type | Byte | Bit | Def.-Range | Variable name | Description |
|---|---|---|---|---|---|---|
| 1 | UINT32 | 0-3 | - | - | nOutData1 | Drive output data 1 (NC->I/O) |
| 2 | UINT32 | 4-7 | - | - | nOutData2 | Drive output data 2 (NC->I/O) |
| 3 | UINT8 | 8 | - | - | nControlByte | Control byte |
| | | | 0 | 0/1 | bMinusHigh | Direction: negative velocity: fast |
| | | | 1 | 0/1 | bMinusLow | Direction: negative velocity: slow |
| | | | 2 | 0/1 | bPlusLow | Direction: positive velocity: slow |
| | | | 3 | 0/1 | bPlusHigh | Direction: positive velocity: fast |
| | | | 4 | 0/1 | - | RESERVE |
| | | | 5 | 0/1 | - | RESERVE |
| | | | 6 | 0/1 | bBreakInv | Inverse braking bit (0 ≡ ACTIVE, 1 ≡ PASSIVE) |
| | | | 7 | 0/1 | bBreak | Braking bit (0 ≡ PASSIVE, 1 ≡ ACTIVE) |
| 4 | UINT8 | 9 | - | - | nExtControlByte | Extended control byte |
| | | | 0 | 0/1 | bDirectionMinus | Direction: negative |
| | | | 1 | 0/1 | bDirectionPlus | Direction: positive |

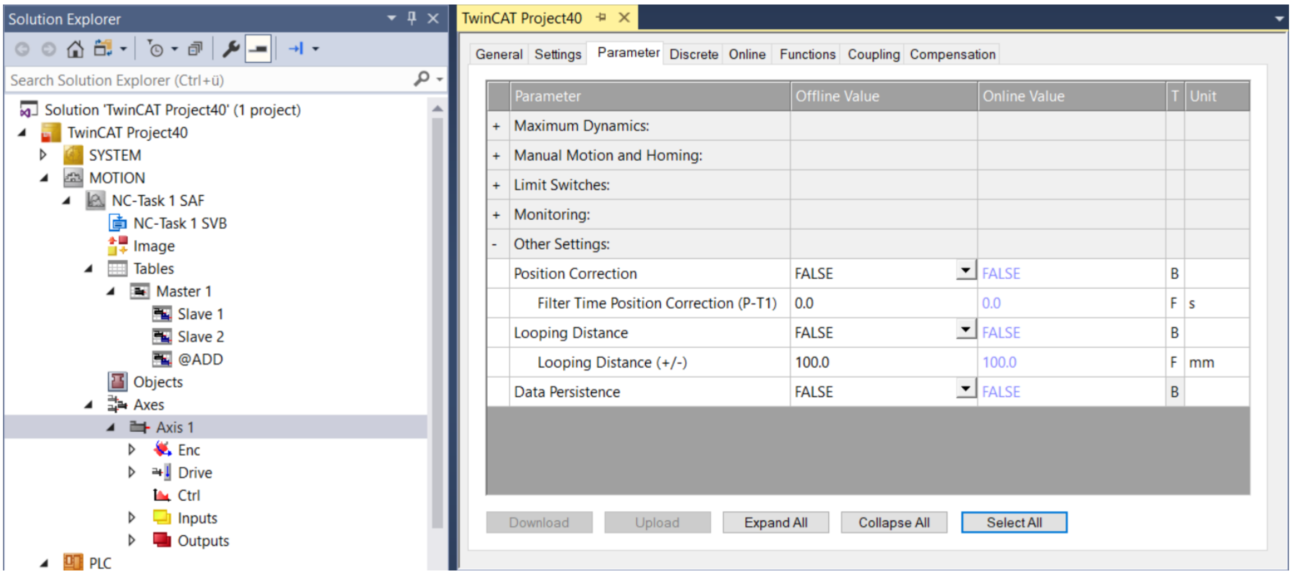
| No. | Data type | Byte | Bit | Def.-Range | Variable name | Description |
|---|---|---|---|---|---|---|
| | | | 2 | 0/1 | bVeloLow | Velocity: slow |
| | | | 3 | 0/1 | bVeloHigh | Velocity: high |
| | | | 4 | 0/1 | - | RESERVE |
| | | | 5 | 0/1 | - | RESERVE |
| | | | 6 | 0/1 | bBreakInv | Inverse braking bit (0 ≡ ACTIVE, 1 ≡ PASSIVE) |
| | | | 7 | 0/1 | bBreak | Braking bit (0 ≡ PASSIVE, 1 ≡ ACTIVE) |
| 5 | UINT16 | 10-11 | - | - | nReserved | Reserved bytes |

**Parameters of the rapid/creep axes**




| Parameter | Description |
|---|---|
| Creep distance positive direction | The creep distance in the positive direction gives the distance to the target position, below which the velocity changes from rapid to creep velocity, if the direction of travel is positive.<br>If a looping distance is selected then this distance is based on the movement reversal point.<br>This distance is denoted by $\mathbf{\Delta p_1}$ in positioning example 1 [▸ 158] |
| Creep distance negative direction | The creep distance in the negative direction gives the distance to the target position, below which the velocity changes from rapid to creep velocity, if the direction of travel is negative.<br>If a looping distance is selected then this distance is based on the movement reversal point. |
| Braking distance positive direction | The braking distance in the positive direction gives the distance to the target position, below which the creep velocity is switched off, if the direction of travel is positive.<br>This distance is denoted by $\mathbf{\Delta p_2}$ in positioning example 1 [▸ 158] |
| Braking distance negative direction | The braking distance in the negative direction gives the distance to the target position, below which the creep velocity is switched off, if the direction of travel is negative. |
| Brake incidence delay time in pos. direction | This delay time gives the start-up delay of the brakes after switching off of the creep velocity, if the direction of travel is positive.<br>In positioning example 1 [▸ 158], this time is between the times $\mathbf{T_4}$ and $\mathbf{T_5}$. |
| Brake incidence delay time in negative direction | This delay time gives the start-up delay of the brakes after switching off of the creep velocity, if the direction of travel is negative. |
| Rapid to creep velocity delay time | This waiting time is between the switching off of the rapid velocity and switching on of the creep velocity. |

| Parameter | Description |
| --- | --- |
| | In positioning example 1 [▶ 158] , this time is between the times $\mathbf{T_2}$ and $\mathbf{T_3}$. |
| Creep distance for stop | The creep distance for stop gives the distance that is covered with the creep velocity after calling up the stop. This creep distance is usually selected shorter than the creep distances in positive and negative direction, since the axis should stop as soon as possible and the exact positioning is not a priority.<br>This distance is denoted by $\mathbf{\Delta p_1}$ in positioning example 3 [▶ 160] |
| Brake release delay | The brake is released immediately the axis is started and after elapse of the brake release delay either the rapid or creep velocity is activated depending on the displacement.<br>In positioning example 1 [▶ 158] , this time is between the times $\mathbf{T_0}$ and $\mathbf{T_1}$. |
| Pulse time in positive direction | This parameter is not evaluated and therefore has no effect. |
| Pulse time in negative direction | This parameter is not evaluated and therefore has no effect. |




| Parameter | Description |
| --- | --- |
| OPERATION MODE: looping distance | The looping distance can be activated with this flag. The looping distance is for approaching the target position always from the same direction. In the case of a positive (negative) looping distance a target position in the positive (negative) direction is increased by the amount of the looping distance and the target then approached from the opposite direction. Consequently the target position in case of a positive looping distance is always approached with negative velocity and a negative looping distance with positive velocity. |
| Looping distance (+ / -) | The looping distance gives the distance, by which the target position is exceeded if necessary so that it is possible to move to the target position from the required direction.<br>This distance is denoted by **creep distance** (looping distance) in positioning example 2 [▶ 159] |

**Axis movement state (nAxisState in cyclic interface):**

Axis movement state (nAxisState in cyclic interface):

| nAxisState | Description |
| --- | --- |
| 0 | Set value generator not active |
| 20 | Axis stopped |
| 21 | Main travel phase: High or low speed travel in relation to the start speed and override |

| nAxisState | Description |
|---|---|
| 22 | Braking phase: High-to-low speed delay time active |
| 23 | Braking phase: Low speed travel |
| 24 | Braking phase: Delay time for brake incidence active |

**Positioning examples:**

Positioning examples

### 1) Positioning A → B, without loop movement

Positioning A → B, without loop movement

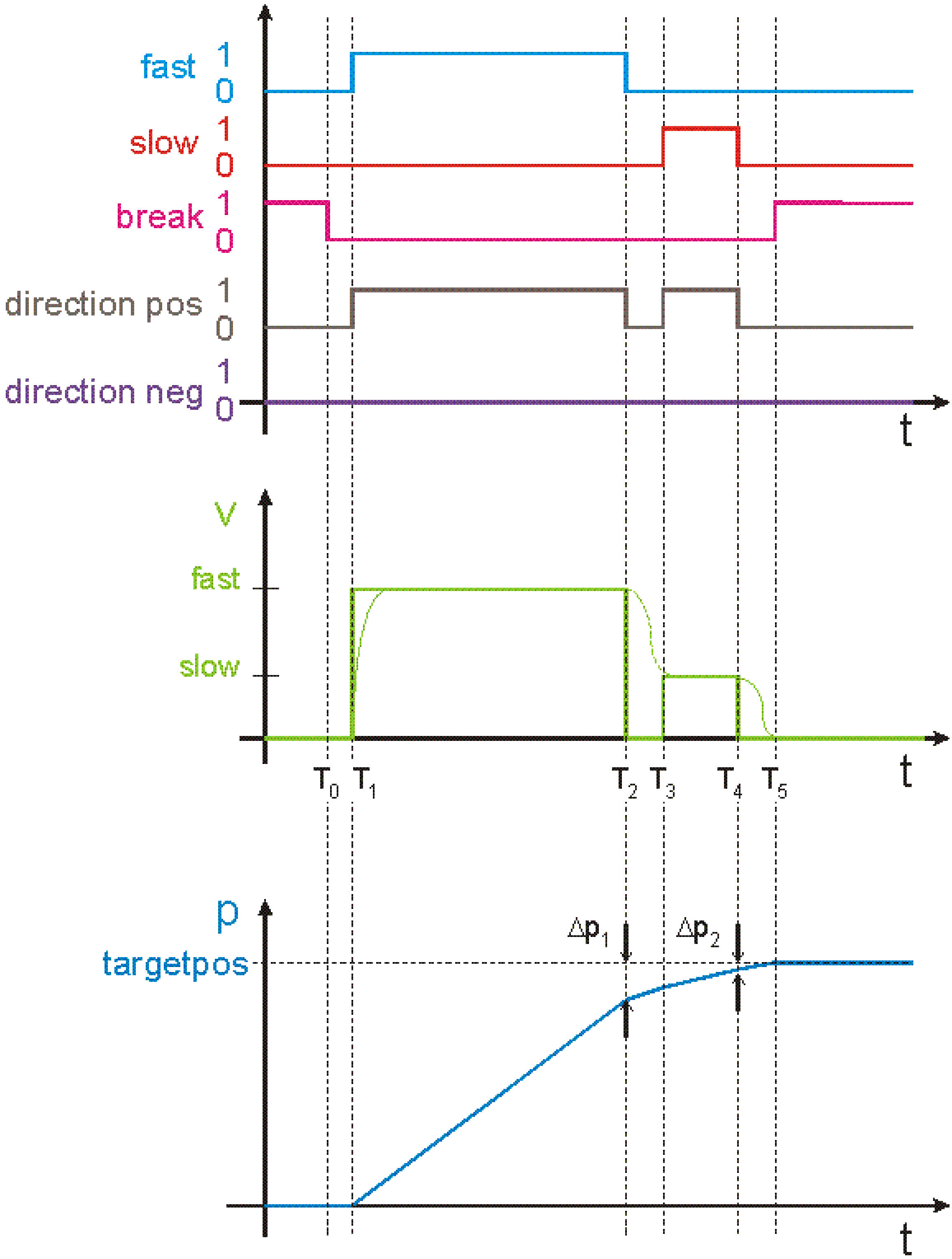



Fig. 1: TcNcTwoSpeed_Positioning1

**2) Positioning A → B, with loop movement > 0.0**

2) Positioning A → B, with loop movement > 0.0

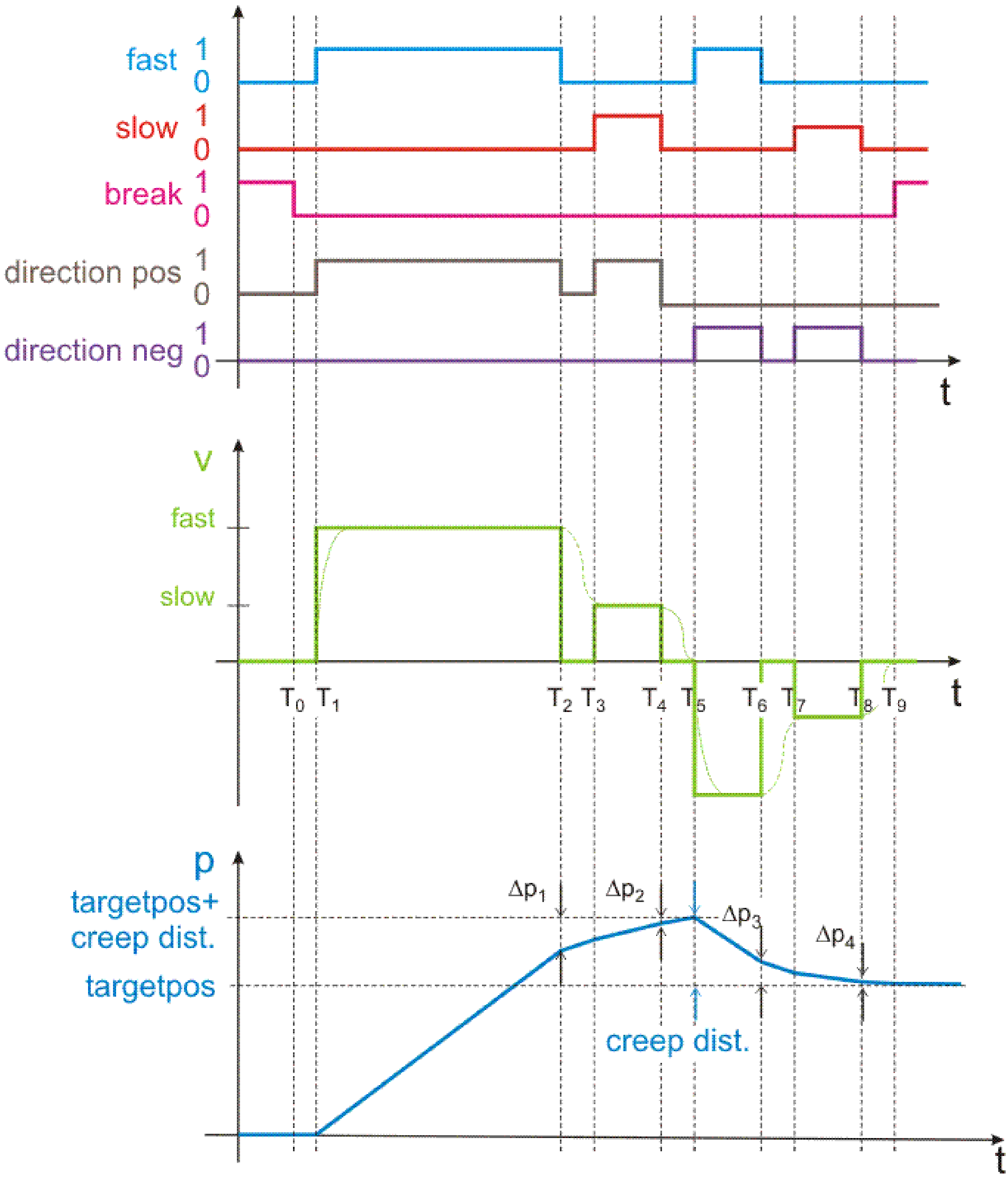
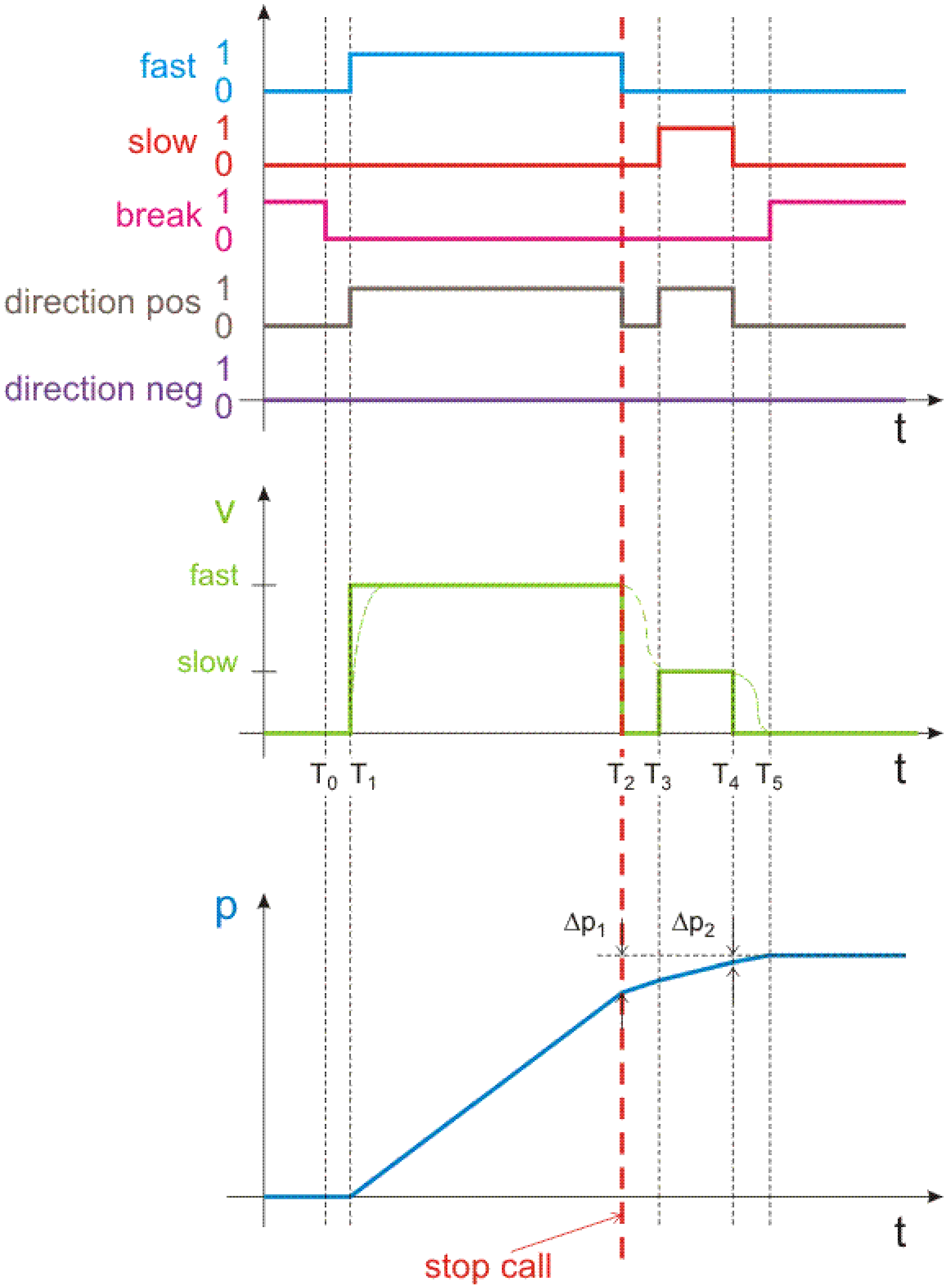



Fig. 2: TcNcTwoSpeed_Positioning2

**3) Stop call up in case of active positioning**


fast
1
0
slow
1
0
break
1
0
direction pos
1
0
direction neg
1
0
t
v
fast
slow
T0 T1 T2 T3 T4 T5
t
p
Δp1
Δp2
t
stop call

# 10 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.

### Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

### 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 our internet page: 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 |
| 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 |
| e-mail: | service@beckhoff.com |

### Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

| | |
|---|---|
| Phone: | +49 5246 963-0 |
| e-mail: | info@beckhoff.com |
| web: | www.beckhoff.com |

More Information:
**www.beckhoff.com/tf5000**

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