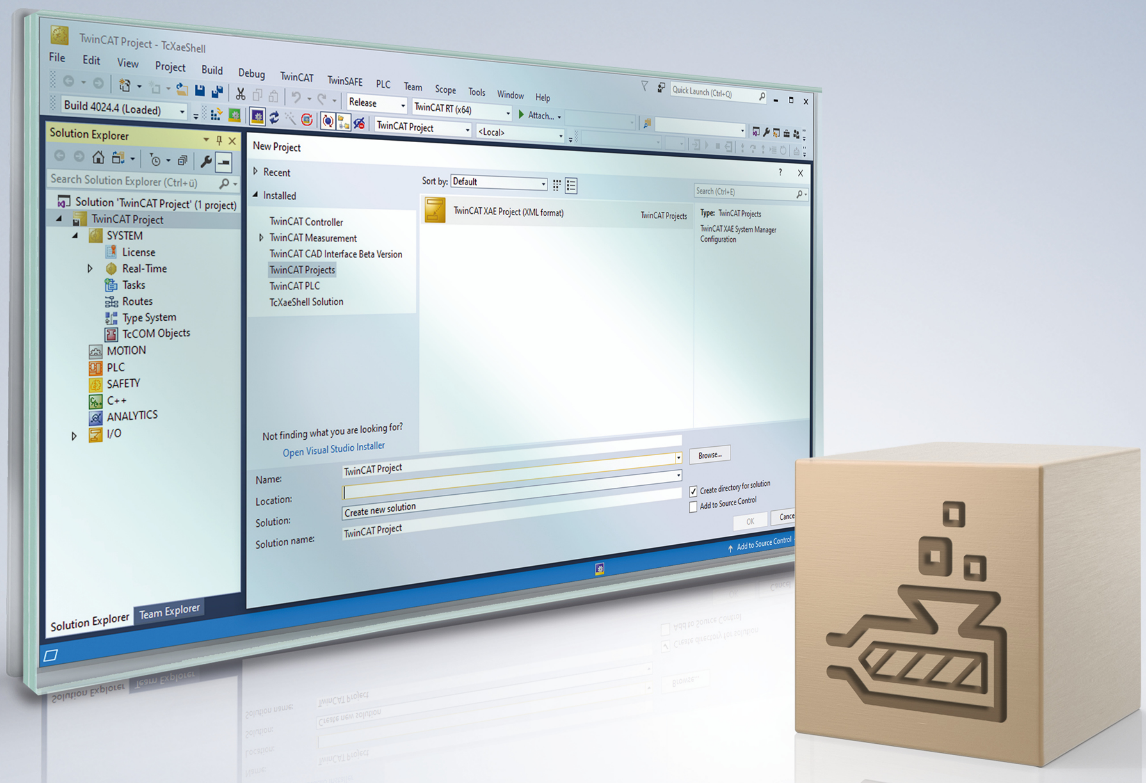
BECKHOFF New Automation Technology

Manual | EN

TF8560

TwinCAT 3 | Plastic Technology Functions



2024-04-23 | Version: 1.1.0

Table of contents

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

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

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.

2 Concept of Libraries

TF8560 TC3 Plastic Technology Functions is a solution that allows the plastics processing industry to rapidly implement TwinCAT 3-based motion tasks.

In plastics processing machines, two different drive types, electric and hydraulic, can be used individually or in combination. The underlying Motion Control libraries for these two drive types are different. Therefore, there is the TwinCAT 3 NC PTP (TF5000) for electric axes and the TwinCAT 3 Hydraulic Positioning (TF5810) for hydraulic axes on the TwinCAT 3 platform. If the control program is developed directly on the basis of these two Motion Control libraries, the customer must re-implement all interfaces that call different libraries when the drive technology is changed.

TC3 Plastic Technology Functions provides a unified interface for the common functions of the hydraulic and electric Motion Control libraries. When developing the control program based on TC3 Plastic Technology Functions, only minimal modification is necessary for a different drive technology.

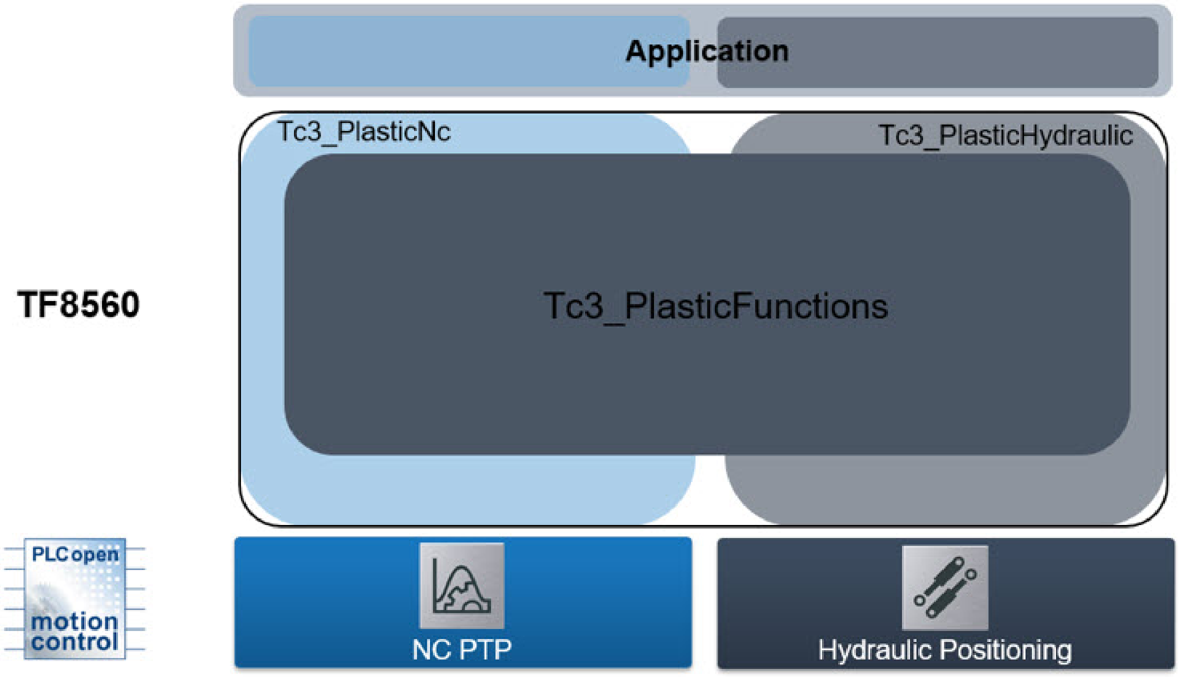
In addition, the elementary motion tasks commonly used in plastics processing, such as cam plates for wall thickness control, multi-segment PTP motion, and pressure control, have been implemented and fully tested in TC3 Plastic Technology Functions, encapsulated as core functions. Customers are freed from building them from scratch and can directly use the provided components to achieve complex functions with little engineering effort.

The basic concept of TC3 Plastic Technology Functions:

- Uniformity: A uniform interface for calling common Motion Control functions for hydraulic and electric axes.
- Ready-to-use: Implementation and encapsulation of essential motion tasks commonly used in plastics processing processes.
- Expandability: Extension or modification of the functions through inheritance.
- Flexibility: Choice of language, object/process-oriented programming approach, multitasking/multi-core capability (to be tested).

TC3 Plastic Technology Functions libraries and license

TC3 Plastic Technology Functions can be considered as an interface between the customer application and the TwinCAT 3 platform. TC3 Plastic Technology Functions consists of three libraries, namely Tc3_PlasticFunctions, Tc3_PlasticNc, and Tc3_PlasticHydraulic. Their dependencies are as shown in the figure below. Tc3_PlasticFunctions implements the common functions for both drive types. The functions specific to the electric or hydraulic axes are implemented respectively in Tc3_PlasticNc and Tc3_PlasticHydraulic and will correspondingly call TF5000 TwinCAT 3 NC PTP or TF5810 TwinCAT 3 Hydraulic Positioning.


Application
Tc3_PlasticNc
Tc3_PlasticHydraulic
TF8560
Tc3_PlasticFunctions
PLCopen motion control
NC PTP
Hydraulic Positioning

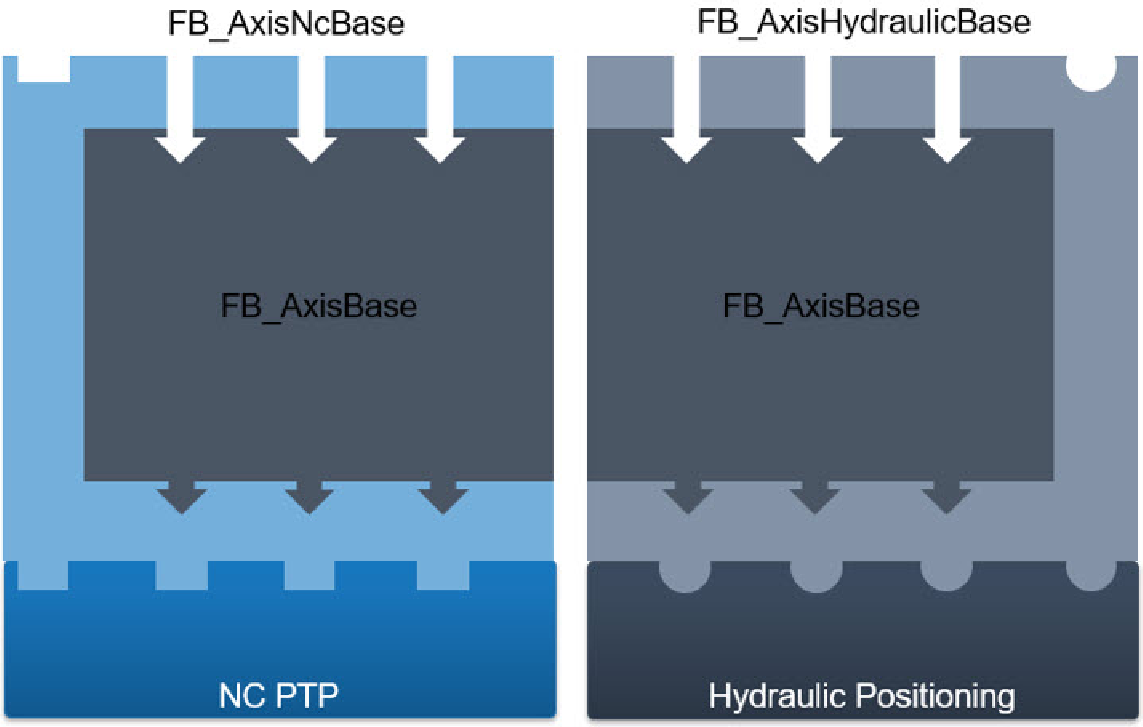
3 Axes

When setting up the functions of an axis, the methods and properties for implementing an elementary motion task are bundled in special function blocks. These are so-called core functions, which can be an active command, such as executing a multi-segment PTP movement, or a passive task, such as displaying the motion state and parameters of an axis, and are described in more detail in the chapter "Core functions concept".

3.1 FB_AxisBase - Virtual axes

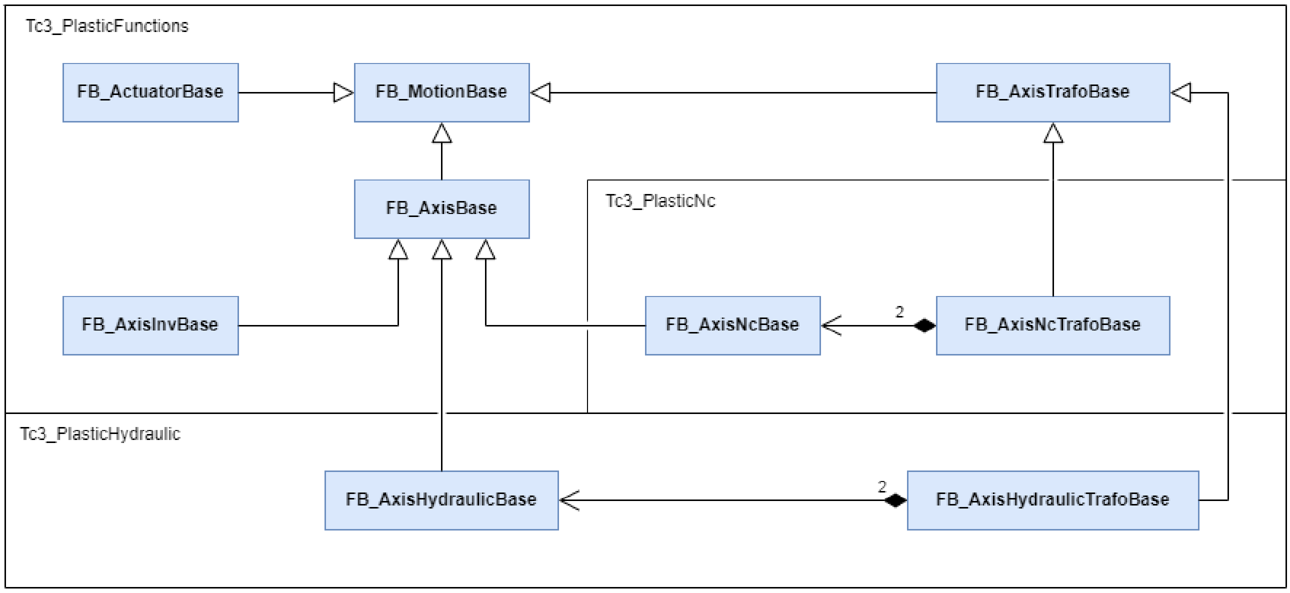
TwinCAT 3 Plastic Technology Functions defines `FB_AxisBase` as a virtual base axis. This provides the core functions required for both electric and hydraulic drives. The ABSTRACT definition means that `FB_AxisBase` cannot be instantiated. This is intentional, as the connection to specific Motion Control libraries is missing and some core functions are only instantiated in the specific versions.

For electric axes `FB_AxisNcBase` and for hydraulic axes `FB_AxisHydraulicBase` is derived from `FB_AxisBase`. On the one hand, the connection to the Motion Control libraries is established. On the other hand, special axis types add their own core functions for their mode of operation. The following figure shows how the axis classes (FBs) in TwinCAT 3 Plastic Technology Functions are to be understood structurally.



Further motion objects

In addition to the basic concept of NC-based and hydraulic-based axes, the Plastic Technology Functions provide further axis and movement types. These include transformation and inverter axes as well as digital linear actuators. The following UML diagram shows the relationship between the individual types and their availability per library.



3.2 Instantiation

FB_AxisNcBase, FB_AxisHydraulicBase and FB_AxisInvBase can be instantiated. Below you will find sample code for creating instances of these axis types. The input variables of these FBs are all defined in the method FB_init(). FB_init() is always called implicitly when initializing an instance of a FB. For a detailed description see FB_init().

Some input variables of FB_init() must be assigned mandatory, while others are optional and can be 0. The following table shows the requirements of FB_AxisNcBase, FB_AxisHydraulicBase, FB_AxisInvBase and FB_ActuatorBase when assigning their input variables. For a detailed explanation of each input variable, see METHOD FB_init().

The open source project TF85xx - Plastic Application, which is based on TF8560 - TwinCAT 3 Plastic Technology Functions, has designed common machine axes and their motion tasks in blow molding and other types of plastics machines. This includes axes such as clamp, blow pin, wall thickness control, injection unit, etc. Customers can use the Plastic Application as the basis for their control system or as a tutorial for TF8560 TwinCAT 3 Plastic Technology Functions as required. The code for the Plastic Application is available here.

Designation	Description
FB_AxisHydraulicBase [▸ 17]	Hydraulic axes, operated with the Tc2_Hydraulics library.
FB_AxisNcBase [▸ 15]	Servo axes, operated with the Tc2_MC2 library.
FB_AxisNcTrafoBase [▸ 16]	Transforming servo axes, operated with the Tc2_MC2 library.
FB_AxisInvBase [▸ 19]	Inverter axes
FB_ActuatorBase	Digital linear actuators

3.2.1 FB_AxisNcBase

This FB creates an axis operated with the Tc2_MC2 library.

Syntax:

```
fbNcAxis:    FB_AxisNcBase
             (
                 AxisName         := 'NcClampAxis1',
                 nPtpPoints       := 10,
                 iProcessHandler := 0,
                 iPosCamLookup   := 0,
                 iVeloCamLookup  := 0
             );
```

Initialization parameters of the FB_init constructor

Designation	Type	Obligatory	Description
AxisName	STRING	Yes	The text-based name of the axis. Used for messages and file names, among other things.
nPtpPoints	INT	Yes	Number of available segments in PTP tables. **The number of points can also be set via Ptp.SetPoints()**
iProcessHandler	I_ProcessHandler	No	A function block for handling pressures and other process variables. **Reserved, currently to be assigned zero**
iPosCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a position cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**
iVeloCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a velocity cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**

Required libraries

NC-based axes require the Tc2_MC2 library and a TwinCAT NC license.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.2.2 FB_AxisNcTrafoBase

This FB creates a transforming axis operated with the Tc2_MC2 and Tc2_MC2_Camming library.

Syntax:

```
fbNcTrafoAxis: FB_AxisNcTrafoBase
               (
                   AxisName        := 'NcAxis',
                   nPtpPoints      := 10,
                   nTrafoPoints    := 181,
                   iProcessHandler := 0,
                   iPosCamLookup   := 0,
                   iVeloCamLookup  := 0
               );
```

Initialization parameters of the FB_init constructor

Designation	Type	Obligatory	Description
AxisName	STRING	Yes	The text-based name of the axis. Used for messages and file names, among other things.
nPtpPoints	INT	Yes	Number of available segments in PTP tables. **The number of points can also be set via Ptp.SetPoints()**
nTrafoPoints	INT	Yes	Number of usable transformer points in the transformation table.
iProcessHandler	I_ProcessHandler	No	A function block for handling pressures and other process variables. **Reserved, currently to be assigned zero**
iPosCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a position cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**
iVeloCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a velocity cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**

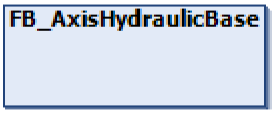
Required libraries

NC-based transformation axes require the Tc2_MC2 and the Tc2_MC2_Camming library and a TwinCAT NC PTP & Camming license.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.2.3 FB_AxisHydraulicBase



This FB creates an axis operated with the Tc2_Hydraulics library.

Syntax:

```
fbHydAxis1: FB_AxisHydraulicBase
            (
                AxisName        := 'HydClampAxis1',
                nPtpPoints      := 10,
                iProcessHandler := 0,
                iPosCamLookup   := 0,
                iVeloCamLookup  := 0,
                iEncoder        := fbHydClampAxisEncIn01,
                iDrive          := fbHydClampAxisDriveOut01,
                iPressureP      := fbHydClampAxis1PrsInP,
                iPressureM      := fbHydClampAxis1PrsInM,
                iPosFilter      := 0,
                iVeloFilter     := 0
            );
```

Initialization parameters of the FB_init constructor

Designation	Type	Obligatory	Description
AxisName	STRING	Yes	The text-based name of the axis. Used for messages and file names, among other things.
nPtpPoints	INT	Yes	Number of available segments in PTP tables. **The number of points can also be set via Ptp.SetPoints()**
iProcessHandler	I_ProcessHandler	No	A function block for handling pressures and other process variables. **Reserved, currently to be assigned zero**
iPosCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a position cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**
iVeloCamLookup	I_CammingLookup [▶ 85]	No	An interface to a FB with a velocity cam plate. **This input is usually assigned zero, as the cam plate is defined at a later point in time.**
iEncoder	I_InputBase	Yes	I/O interface for the encoder. This interface is used to determine the actual position of the axis. Can also be assigned via a property
iDrive	I_OutputBase	Yes	I/O interface for the drive. This interface is used to output the target velocity of the axis. Can also be assigned via a property
iPressureP	I_InputBase	No	I/O interface for a pressure sensor. This interface is used to determine the pressure on the positive cylinder surface. Can also be assigned via a property
iPressureM	I_InputBase	No	I/O interface for a pressure sensor. This interface is used to determine the pressure on the negative cylinder surface. Can also be assigned via a property
iPosFilter	I_Filter	No	Interface for a filter of the actual position. Can also be assigned via a property
iVeloFilter	I_Filter	No	Interface for a filter of the actual velocity. Can also be assigned via a property

Strong filtering

Strong filtering may have a negative effect on the performance of the axis.

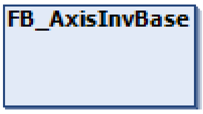
Required libraries

Hydraulic axes require the Tc2_Hydraulics library.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

3.2.4 FB_AxisInvBase



This FB creates an axis operated with a simple inverter. This type of axis is prepared for the use of drive components whose definition is below the possibilities of a servo axis. The biggest problem might be the lack of position feedback. But even very simple drives like frequency inverters may be used to support some tasks in a machine, such as turning an extruder or moving a conveyor belt. The axis does not require position feedback and does not support position-bound functions.

Syntax:

```
fbInvAxis:  FB_AxisInvBase
            (
                AxisName        := 'InvExtruder',
                nPtpPoints      := 10,
                iProcessHandler := 0,
                iPosCamLookup   := 0,
                iVeloCamLookup  := 0
            );
```

Initialization parameters of the FB_init constructor

Designation	Type	Obligatory	
AxisName	STRING	Yes	Used for messages and file names, among other things.
nPtpPoints	INT	Yes	Number of available segments in PTP tables.
iProcessHandler	I_ProcessHandler	No	Optional: A function block for handling pressures and other process variables.
iPosCamLookup	I_CammingLookup	No	Optional: A lookup function block with a PvsP camming table.
iVeloCamLookup	I_CammingLookup	No	Optional: A lookup function block with a VvsP camming table.

Required libraries

Inverter axes are fully implemented in Tc3_PlasticFunctions and do not require the libraries mentioned above.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.3 Properties and methods

Some of the properties of the axes are standard variables (e.g. `iNcSampleAxis.Name` of type `STRING`) and the required information can be processed directly. Other properties are so-called CoreFunctions and return a type `INTERFACE` to the CoreFunction itself.

For example, the axis has a property of type `I_Power` with the name Power, which is the `INTERFACE` of `FB_Power`. This interface can be used to access properties and methods provided there.

The constantly active algorithms for cyclic calls are not executed in the main part (body) of the FBs, but in its cycle method. An axis is therefore only executed if its `Cyclic()` methods are called. In addition, the cycle method of an axis should be called exactly once for each PLC cycle, as these can also execute setpoint generation and control algorithms.

```
//  CoreFunction method calls
iNcSampleAxis.Power.DoPower(TRUE)
iNcSampleAxis.Power.FeedEnable(TRUE, TRUE);

// CoreFunction property access
bEnabled := iNcSampleAxis.Power.Status;

// Cyclic call of the axis
iNcSampleAxis.Cyclic();
```

3.4 Access and embedding

The TC3 Plastic Technology Functions creates the corresponding INTERFACEs for each function block (FB). For a detailed description see INTERFACE concept. In this document the following naming rules are applied:

`FB_Xyz`: Declaration of a function block

`fbXyz`: an instance of `FB_Xyz`

`I_Xyz`: the INTERFACE declaration corresponding to `FB_Xyz`

`iXyz`: a variable of `I_Xyz` that is instantiated with `fbXyz`

NOTICE

Do not access the `fbXyz` instance of an axis directly

We suggest that you don't access the `fbXyz` of an axis directly, but instead access the properties and methods of the `FB_Xyz` via `iXyz`, as is shown below. This is because the development of `I_Xyz` in TC3_Plastic Technology Functions only considers the properties and methods that are needed externally to operate `FB_Xyz`. In contrast to that, the properties and methods that are only to be called by other methods inside `FB_Xyz` do not appear in `I_Xyz`. As a result, `iXyz` is clearer than `fbXyz` and customers can find the information they need faster than with `fbXyz`, while the risk of misuse is prevented.

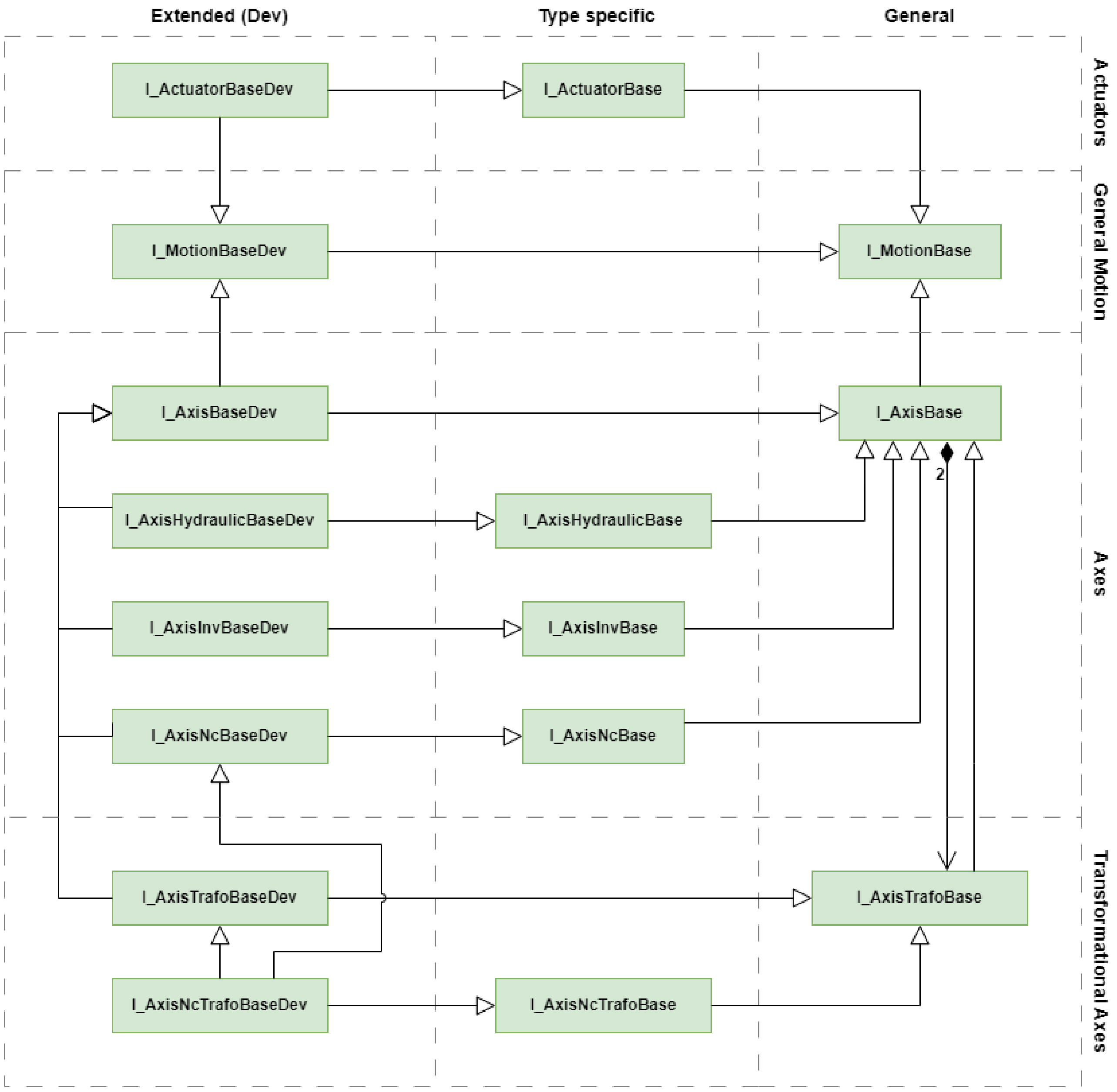
```
sName := fbNcAxis1.Name         // not preferred
sName := iNcClamp1.Name         // preferred
```

It should be noted that TC3 Plastic Technology Functions creates two INTERFACEs with different information content for each function block of the axes. For example, `FB_AxisBase` has the interfaces `I_AxisBase` and `I_AxisBaseDev`. `I_AxisBase` contains the necessary functions of the axes for most common cases. In contrast, `I_AxisBaseDev` provides more information access and allows users to implement more complex functions. For the sake of simplicity and security, it is recommended to use the general interface (e.g. `I_AxisBase)`.

```
iNcAxis1:             I_AxisNcBase := fbNcAxis1;
iNcAxis1Dev:          I_AxisNcBaseDev := fbNcAxis1;

iHydAxis1:            I_AxisHydraulicBase := fbHydAxis1;
iHydAxis1Dev:         I_AxisHydraulicBaseDev := fbHydAxis1;
```

The following UML diagram shows the available interfaces of the individual axes and their inheritance structure.



Name	Description
I_MotionBase [▸ 56]	General motion unit
I_MotionBaseDev [▸ 57]	
I_AxisBase [▸ 22]	General axis
I_AxisBaseDev [▸ 23]	
I_AxisNcBase [▸ 26]	NC-based axis
I_AxisNcBaseDev [▸ 27]	
I_AxisHydraulicBase [▸ 29]	Hydraulic-based axis.
I_AxisHydraulicBaseDev [▸ 31]	.
I_AxisInvBase [▸ 34]	Inverter-based axis

Name	Description
I_AxisInvBaseDev [▸ 35]	
I_AxisTrafoBase [▸ 41]	General transformation axis
I_AxisTrafoBaseDev [▸ 42]	
I_AxisNcTrafoBase [▸ 45]	NC-based transformation axis
I_AxisNcTrafoBaseDev [▸ 47]	
I_ActuatorBase [▸ 58]	Digital linear actuator
I_ActuatorBaseDev [▸ 59]	

3.4.1 I_AxisBase

Represents the general interface for an axis.

Syntax:

```
INTERFACE I_AxisBase EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▸ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.

Name	Type	Access	Description
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

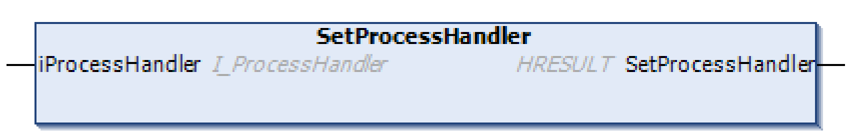
Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.4.1.1 SetProcessHandler



Reserved for extension: An interface to a FB for automatic pressure controller activation is entered.

3.4.2 I_AxisBaseDev

Represents the extended interface for an axis.

Syntax:

```
INTERFACE I_AxisBaseDev EXTENDS I_AxisBase, I_MotionBaseDev
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▸ 19].
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▸ 146].
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.

Name	Type	Access	Description
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
SetTorqueLimiting(iTorqueLimiting)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(iCoreFunc)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(eNewState)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

3.4.3 I_AxisNcBase

Represents the general interface for an NC axis.

Syntax:

```
INTERFACE I_AxisNcBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
ActualsNc [▶ 76]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▶ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 97] core function.
Homing [▶ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▶ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.

Name	Type	Access	Description
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.4.4 I_AxisNcBaseDev

Represents the extended interface for an NC axis.

Syntax:

```
INTERFACE I_AxisNcBaseDev EXTENDS I_AxisBaseDev, I_AxisNcBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 19].
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 146].
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.

Name	Type	Access	Description
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
ActualsNc [▸ 76]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▸ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▸ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▸ 163]	I_TeachUpdate	Get	
ToolAdaption [▸ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▸ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▸ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).

Name	Type	Access	Description
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▸ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetNcAxisRef()	The address to the mapping interface (`AXIS_REF`) between NC and PLC.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(`nErrorCode`)	This method can be used to set the axis to an error state.
SetTorqueLimiting(`iTorqueLimiting`)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(`iCoreFunc`)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(`eNewState`)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▸ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.4.5 I_AxisHydraulicBase

Represents the general interface for a hydraulic axis.

Syntax:

```
INTERFACE I_AxisHydraulicBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)

Name	Type	Access	Description
ActualsHydraulics [▶ 74]	I_ActualsHydraulic	Get	Provision of current actual values (positions, velocities, pressures, etc.). It is an extended version of Actuals.
AutoIdent [▶ 77]	I_AutoIdent	Get	Automatic characteristic measurement of the hydraulic axis.
DirectOutput [▶ 93]	I_DirectOutput	Get	Direct output via the drive interface of the axis.
UseDatFile	BOOL	Get, Set	A TRUE here signals that the axis loads its parameters with function blocks of the hydraulics library from a file during startup. ***Notice*** **A TRUE must be set before the first Cyclic call.**
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▶ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 97] core function.
Homing [▶ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▶ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.

Name	Type	Access	Description
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

3.4.6 I_AxisHydraulicBaseDev

Represents the extended interface for a hydraulic axis.

Syntax:

```
INTERFACE I_AxisHydraulicBaseDev EXTENDS I_AxisBaseDev, I_AxisHydraulicBase, I_ExternalHydAxisLibRef
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
Drive	I_OutputBase	Get, Set	I/O interface for communication with the axis output hardware.
Encoder	I_InputBase	Get, Set	I/O interface for communication with the actual value acquisition of the axis.
PosFilter	I_Filter	Get, Set	Filter interface for filtering the actual position of the axis.
PressureInputM	I_InputBase	Get, Set	I/O interface for determining the actual pressure of the axis acting in the negative direction of movement.
PressureInputP	I_InputBase	Get, Set	I/O interface for determining the actual pressure of the axis acting in the positive direction of movement.
VeloFilter	I_Filter	Get, Set	Filter interface for filtering the actual velocity of the axis.
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 19].

Name	Type	Access	Description
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▸ 146].
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
ActualsHydraulics [▸ 74]	I_ActualsHydraulic	Get	Provision of current actual values (positions, velocities, pressures, etc.). It is an extended version of Actuals.
AutoIdent [▸ 77]	I_AutoIdent	Get	Automatic characteristic measurement of the hydraulic axis.
DirectOutput [▸ 93]	I_DirectOutput	Get	Direct output via the drive interface of the axis.
UseDatFile	BOOL	Get, Set	A TRUE here signals that the axis loads its parameters with function blocks of the hydraulics library from a file during startup. ***Notice*** **A TRUE must be set before the first Cyclic call.**
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.

Name	Type	Access	Description
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▸ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▸ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▸ 163]	I_TeachUpdate	Get	
ToolAdaption [▸ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▸ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▸ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▸ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetHydAxisRef()	The address to the mapping interface (`AXIS_REF_BkPlcMc`) between hydraulics library and PLC.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(`nErrorCode`)	This method can be used to set the axis to an error state.
SetTorqueLimiting(`iTorqueLimiting`)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(`iCoreFunc`)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(`eNewState`)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▸ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

3.4.7 I_AxisInvBase

Represents the general interface for an inverter axis.

Syntax:

```
INTERFACE I_AxisInvBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▶ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 97] core function.
Homing [▶ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▶ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.

Name	Type	Access	Description
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.4.8 I_AxisInvBaseDev

Represents the extended interface for an inverter axis.

Syntax:

```
INTERFACE I_AxisInvBaseDev EXTENDS I_AxisBaseDev, I_AxisInvBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
Device	I_InvDevice	Get, Set	I/O interface for communication with the axis inverter
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 19].
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 146].

Name	Type	Access	Description
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▶ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 97] core function.
Homing [▶ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▶ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

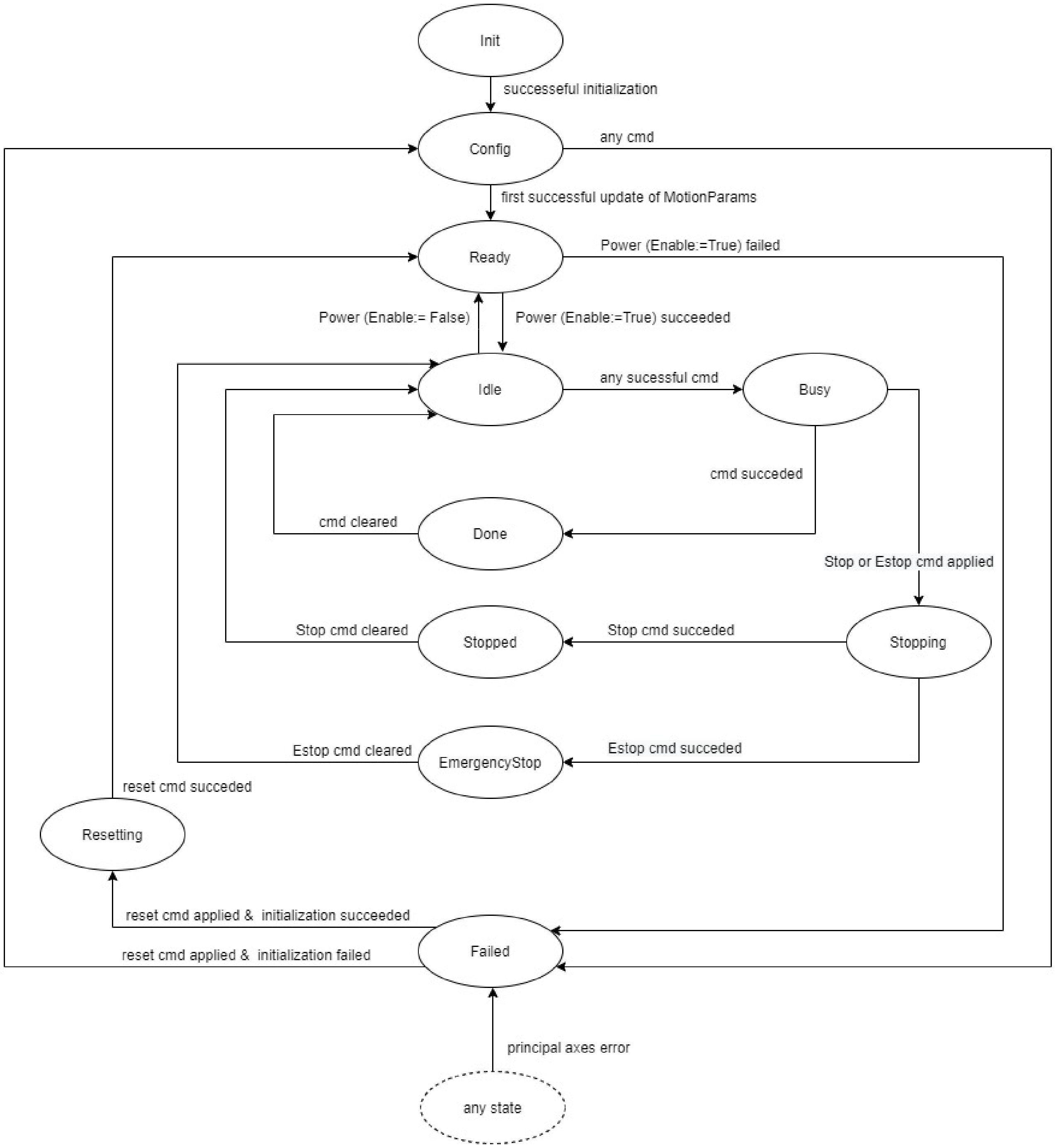
Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
SetTorqueLimiting(iTorqueLimiting)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(iCoreFunc)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(eNewState)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.5 States (state machine)

The state of an axis is declared via an enumeration of the type E_AxisState. This information is provided by each axis as a property with the name State (e.g. iNcSampleAxis.State). An overview of all states can be represented as follows:



State	Description
eInit	The axis is in the initialization phase and must be initialized according to the application requirements. In this state the axis is not ready for operation.
eConfig	The axis applies a series of parameters from the subordinate drive technology (NC, hydraulics library). Settings that are important for correct operation are checked for correspondence to the motion technology. ADS and mapping connections are also tested at the same time.
eReady	The axis has been successfully initialized and configured. It is ready to accept an enable, given via iAxis.Power.
eIdle	The axis is enabled and ready to accept motion commands (e.g. JogP()).
eBusy	The axis is processing a command (e.g. JogP()).
eDone	The axis has successfully completed a command (e.g. TableMove()).
eStopping	The axis is in the state of processing a stop or Estop command.
eStopped	The Stop command was successful.
eEmergencyStop	The Estop command was successful.

State	Description
eResetting	The axis was instructed by Reset() to initiate the change from the state eFailed to the state eReady.
eFailed	The axis is in the error state.

3.6 Transformation axes

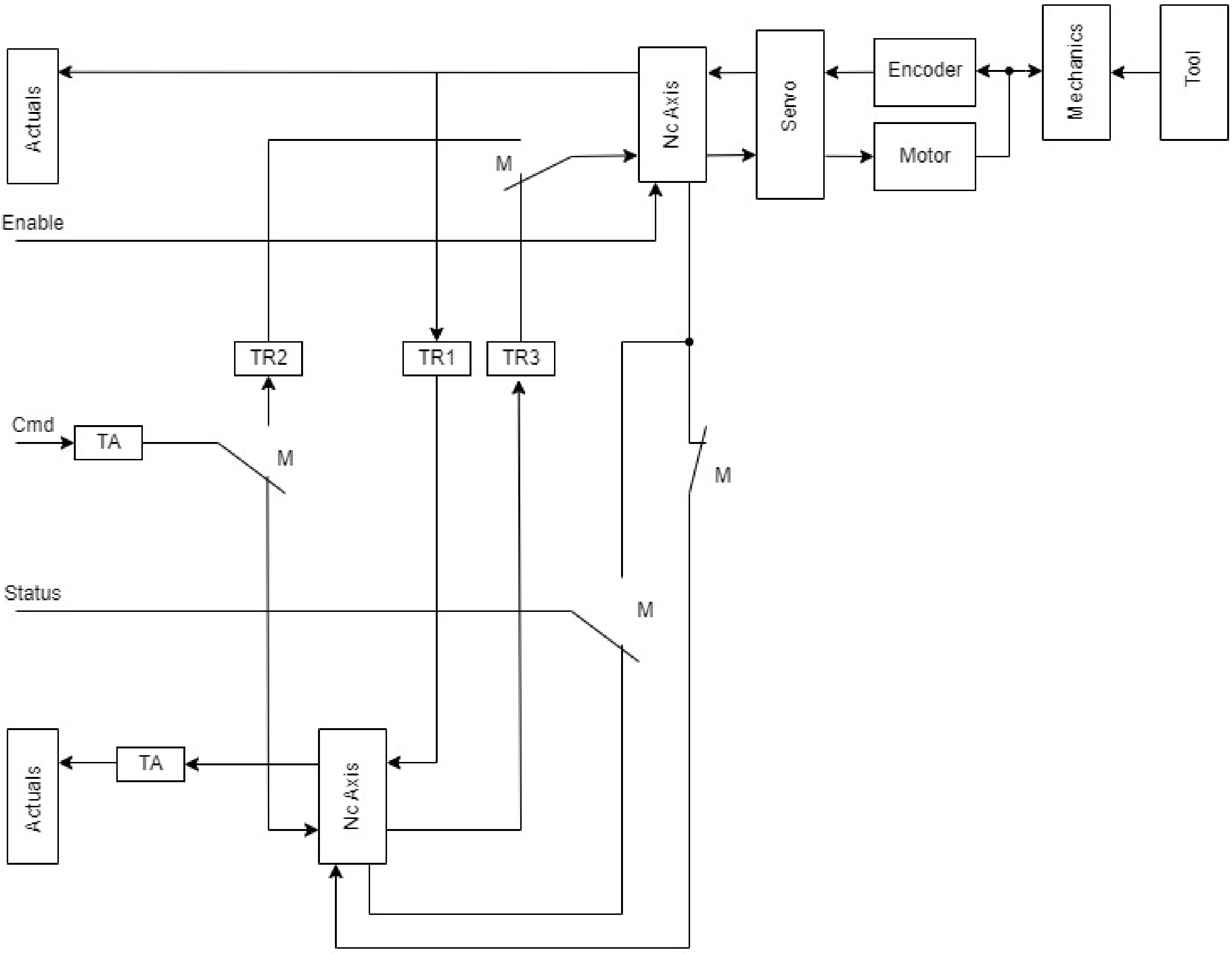
The mechanical solution for some axis types requires the design of a non-linear lever mechanism. The tool path does not follow the drive movement in a constant ratio. This enables the optimization of axis properties for varying requirements in individual areas of the travel. For the best possible embedding of this non-linearity, this design must be represented accordingly by a suitable software concept.

Typical mechanics for a transforming axis would be:

- Crank drives
- Scotch yoke
- customized concepts.
- etc…

3.6.1 Construction of a transforming axis

A transformation axis is a container object that implements the same interface as a standard Nc axis. Inside this object there are two local standard axis objects, named Load Side and Drive Side. These internal objects are used to handle the specific requirements for the effective tool and the actuator part. There is almost no exchange between the application project and these internal objects because the usual interactions use the interfaces of the container object.



Used symbols

Symbol	Description
Actuals	A common core function that contains information about the current situation of the axis. There are alternative sub-versions for the different axis types.
TA: Tool Adaptation	A common core function that is used to handle the differences between the axis and the effective tool.
TRx: Transformation	A core function specific to transforming axes. It is used to convert actual values of the drive side axis into load side actual values (TR1) and load side setpoints (TR2) or setpoints (TR3) into drive side values.
MM: Operation mode select	In this figure the "Full transformation mode" is selected.

3.6.2 Semi-transformation mode

In this operation mode all commands are forwarded to the Drive Side axis. To avoid unexpected position lag errors, the Load Side axis is not enabled.

The actual position and velocity values of the Load Side are updated using translated values from the Drive Side.

Any commanded motion will be executed by the Drive Side using translated target position values. A commanded move to 100.0 will make the tool travel to 100.0 mm, no matter what Drive Side motor angle is required.

Velocity cannot be translated

The commanded velocity cannot be translated because the result would depend on the position.

No constant velocity

Because the profile generation is executed by the Drive Side axis, the tool will not travel with constant velocity.

No position or velocity camming

A position or velocity camming is not supported.

3.6.3 Full transformation mode

In this operation mode, almost all commands are forwarded to the Load Side axis. Enabling the container object will enable both internal axes.

The actual position and velocity values of the Load Side are updated using translated values from the Drive Side.

The transformation will be performed by converting the output of the Load Side profile calculation. Again, a commanded move to 100.0 will make the tool travel to 100.0 mm, no matter what Drive Side motor angle is required.

Constant velocity

Because the profile generation is executed by the Load Side axis, the tool will travel with constant velocity.

Excessive Drive Side velocity values required

In some areas of the travel, even low tool velocities may require excessive Drive Side velocity values.

Full transformation temporarily paused

For Jog or Homing commands, the full transformation is temporarily paused. All position and velocity values are used following Drive Side definitions without any translation.

3.6.4 I_AxisTrafoBase

Represents the general interface for a transformation axis.

Syntax:

```
INTERFACE I_AxisTrafoBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
ActualsDrive	I_MotionActuals	Get	Provision of current actual values of the drive side (positions, velocities, etc.).
MotionSetpointsDrive	I_MotionSetpoints	Get	Provision of current setpoints on the drive side (position, velocity, etc.).
Suspend	BOOL	Get, Set	Temporarily suspends the full transformation.
Suspended	BOOL	Get	The full transformation is temporarily suspended.
Synchronise	BOOL	Get, Set	Activates the full transformation of the axis.
Synchronised	BOOL	Get	The full transformation is active.
Transformation [▸ 166]	I_Transformation	Get	Transformation function of the axis.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.

Name	Type	Access	Description
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.6.5 I_AxisTrafoBaseDev

Represents the extended interface for a transformation axis.

Syntax:

```
INTERFACE I_AxisTrafoBaseDev EXTENDS I_AxisBaseDev, I_AxisTrafoBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
DoTransforming	BOOL	Get, Set	[INTERNAL] For internal use only.
DriveSideBase	I_AxisBaseDev	Get	Interface to the drive side of the transformation axis.
IsTransforming	BOOL	Get, Set	[INTERNAL] For internal use only.
LoadSideBase	I_AxisBaseDev	Get	Interface to the load side of the transformation axis.
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 19].
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 146].
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
ActualsDrive	I_MotionActuals	Get	Provision of current actual values of the drive side (positions, velocities, etc.).
MotionSetpointsDrive	I_MotionSetpoints	Get	Provision of current setpoints on the drive side (position, velocity, etc.).
Suspend	BOOL	Get, Set	Temporarily suspends the full transformation.
Suspended	BOOL	Get	The full transformation is temporarily suspended.
Synchronise	BOOL	Get, Set	Activates the full transformation of the axis.
Synchronised	BOOL	Get	The full transformation is active.
Transformation [▶ 166]	I_Transformation	Get	Transformation function of the axis.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▶ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.

Name	Type	Access	Description
Estop [▶ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 97] core function.
Homing [▶ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▶ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▶ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▶ 163]	I_TeachUpdate	Get	
ToolAdaption [▶ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▶ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▶ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
SetTorqueLimiting(iTorqueLimiting)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(iCoreFunc)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(eNewState)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.

Name	Description
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▶ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.6.6 I_AxisNcTrafoBase

Represents the general interface for a transforming NC axis.

Syntax:

```
INTERFACE I_AxisNcTrafoBase EXTENDS I_AxisTrafoBase, I_AxisNcBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
ActualsDriveNc	I_NcActuals	Get	Provision of current actual values of the drive side (positions, velocities, torque, etc.). It is an extended version of Actuals.
ActualsNc [▶ 76]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
ActualsDrive	I_MotionActuals	Get	Provision of current actual values of the drive side (positions, velocities, etc.).
MotionSetpointsDrive	I_MotionSetpoints	Get	Provision of current setpoints on the drive side (position, velocity, etc.).
Suspend	BOOL	Get, Set	Temporarily suspends the full transformation.
Suspended	BOOL	Get	The full transformation is temporarily suspended.
Synchronise	BOOL	Get, Set	Activates the full transformation of the axis.
Synchronised	BOOL	Get	The full transformation is active.
Transformation [▶ 166]	I_Transformation	Get	Transformation function of the axis.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▶ 81]	I_CammingMain	Get	Functions for processing cam plates.

Name	Type	Access	Description
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▸ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▸ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▸ 163]	I_TeachUpdate	Get	
ToolAdaption [▸ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▸ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▸ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▸ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▸ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.6.7 I_AxisNcTrafoBaseDev

Represents the extended interface for a transforming NC axis.

Syntax:

```
INTERFACE I_AxisNcTrafoBaseDev EXTENDS I_AxisTrafoBaseDev, I_AxisNcBaseDev, I_AxisNcTrafoBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
DoTransforming	BOOL	Get, Set	[INTERNAL] For internal use only.
DriveSideBase	I_AxisBaseDev	Get	Interface to the drive side of the transformation axis.
IsTransforming	BOOL	Get, Set	[INTERNAL] For internal use only.
LoadSideBase	I_AxisBaseDev	Get	Interface to the load side of the transformation axis.
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▸ 19].
AxisIsNc	BOOL	Get	TRUE, if the axis is based on TwinCAT NC.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▸ 146].
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
ActualsDriveNc	I_NcActuals	Get	Provision of current actual values of the drive side (positions, velocities, torque, etc.). It is an extended version of Actuals.
ActualsDrive	I_MotionActuals	Get	Provision of current actual values of the drive side (positions, velocities, etc.).
MotionSetpointsDrive	I_MotionSetpoints	Get	Provision of current setpoints on the drive side (position, velocity, etc.).
Suspend	BOOL	Get, Set	Temporarily suspends the full transformation.
Suspended	BOOL	Get	The full transformation is temporarily suspended.
Synchronise	BOOL	Get, Set	Activates the full transformation of the axis.
Synchronised	BOOL	Get	The full transformation is active.
Transformation [▸ 166]	I_Transformation	Get	Transformation function of the axis.

Name	Type	Access	Description
ActualsNc [▸ 76]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
AutoTorqueLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for torque limitation if the type of a servo drive (CoE, SoE) has been detected.
AxisName	STRING	Get, Set	See property *.Name* The axis name must be unique. This is of particular importance for hydraulic axes. The library used here uses the axis name as file name for loading and saving the axis parameters.
BadSwitchOver	BOOL	Get	Reserved for future extension.
Camming [▸ 81]	I_CammingMain	Get	Functions for processing cam plates.
CycleTime	LREAL	Get	The call cycle time of the most important axis functionalities.
CycleTimeValid	BOOL	Get	TRUE, if the call cycle time was determined to be valid.
DisableSoftEnd [▸ 95]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▸ 98]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▸ 97] core function.
Homing [▸ 100]	I_HomingMain	Get	Provision of a selection of homing procedures.
MotionSetpoints [▸ 134]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▸ 146]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▸ 158]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▸ 161]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
SwitchOver	BOOL	Get	Reserved for future extension.
TeachPosition [▸ 163]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems.
TeachUpdate [▸ 163]	I_TeachUpdate	Get	
ToolAdaption [▸ 165]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
TorqueLimiting [▸ 164]	I_TorqueLimiting	Get	Torque limitation of the axis.
VelocityFeed [▸ 172]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.

Name	Type	Access	Description
State [▸ 37]	E_AxiState	Get	The current state of the axis.

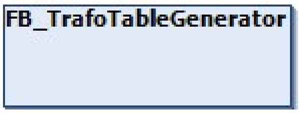
Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
GetNcDriveAxisRef()	The address on the mapping interface (`AXIS_REF`) of the drive side between NC and PLC.
GetNcAxisRef()	The address to the mapping interface (`AXIS_REF`) between NC and PLC.
ConvertCountToPos	[INTERNAL] For internal use only.
ConvertPosToCount	[INTERNAL] For internal use only.
GoErrorBase(`nErrorCode`)	This method can be used to set the axis to an error state.
SetTorqueLimiting(`iTorqueLimiting`)	This method connects a function block for torque limitation with the axis.
AppendCorefunction(`iCoreFunc`)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(`eNewState`)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
GetProcessHandler()	Reserved for future extension.
SetProcessHandler() [▸ 23]	Reserved for future extension.
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

3.6.8 FB_TrafoTableGenerator



Represents the base class for generation algorithms of transformation cam plates.

The following pre-implemented geometries are included:

1. FB_ClampTableGenerator [▸ 50] - Geometry of a typical clamp
2. FB_CrankTableGenerator [▸ 54] - Geometry of a crank drive
3. FB_ScotchYokeTableGenerator [▸ 55] - Geometry of a Scotch yoke

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_TrafoTableGenerator
```

Properties

Name	Type	Access	Description
LookUp	I_CammingLookUp [▶ 85]	Get, Set	Interface to the table to be written
DriveHighEnd	LREAL	Get, Set	Upper limit of the drive position.
DriveLowEnd	LREAL	Get, Set	Lower limit of the drive position.
DriveStep	LREAL	Get	Step size of the drive positions.
ParamValid	BOOL	Get	The set parameters have valid values.
Points	INT	Get, Set	Number of points to be defined.

Methods

Name	Description
DefineTable()	Starts the calculation of the transformation table.
Reset()	Resets generator calculations.

Methods are abstract

The methods are defined as `ABSTRACT` and must be implemented in inheriting classes.

Interfaces

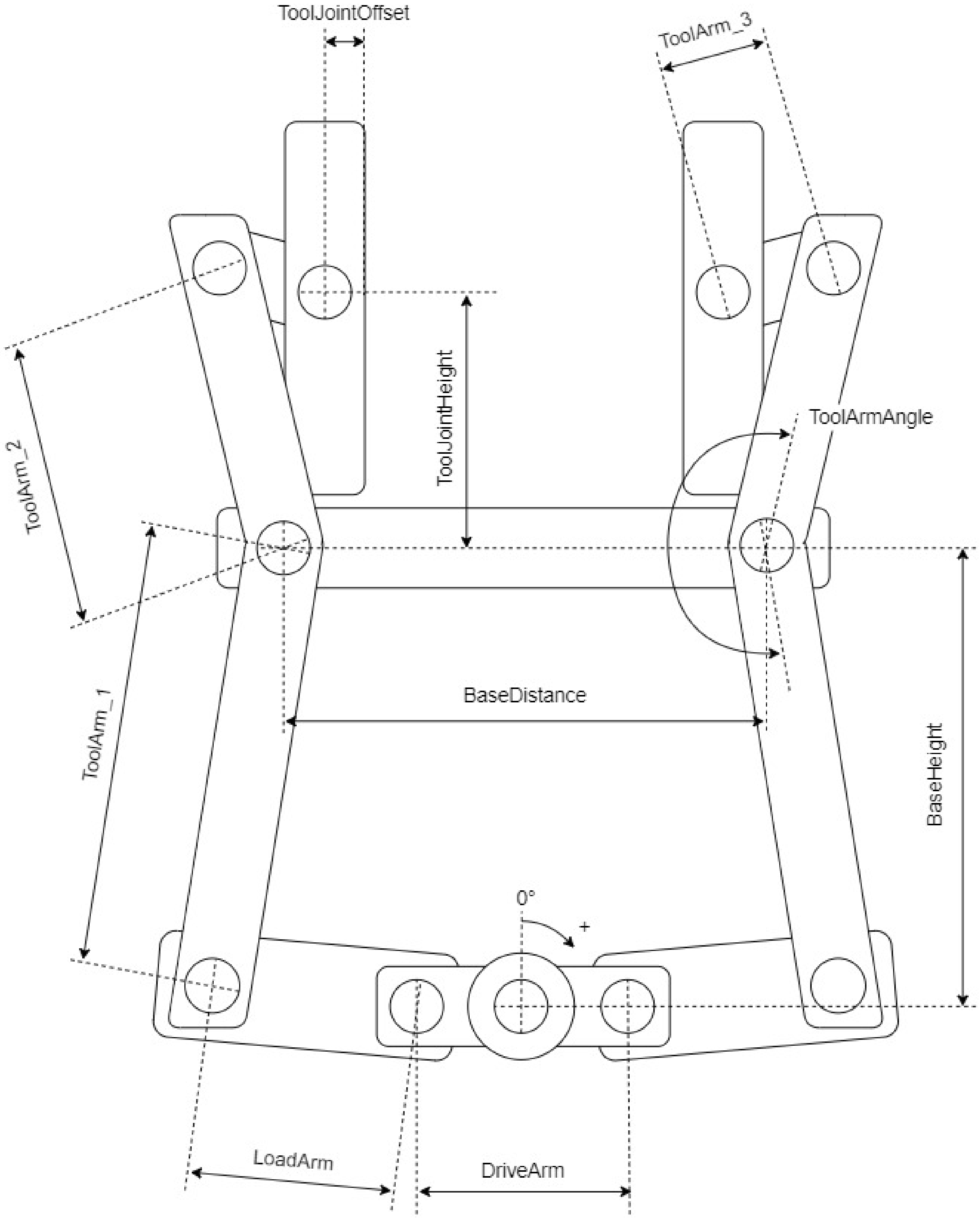
Type	Description
I_TrafoTableGenerator	Standard interface on FB_TrafoTableGenerator.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.6.9 FB_ClampTableGenerator

Implements a calculation algorithm for a standard clamp mechanism.



Names of the dimension designations in the sketch

The dimension designations are the same as the names of the properties of the class to be configured.

Syntax:

```
FUNCTION_BLOCK FB_ClampTableGenerator EXTENDS FB_TrafoTableGenerator
```

Properties

Name	Type	Access	Description
LookUp	I_CammingLookUp [▸ 85]	Get, Set	Interface to the table to be written
AsymHalf	I_ClampTableGenerator_Half	Get	Interface to an asymmetrical half of the clamp. *Notice* **If this property is called at least once, the class switches to asymmetrical operation mode.**
BaseDistance	LREAL	Get, Set	Distance between the two stationary bearing points in the center of the mechanism.
BaseHeight	LREAL	Get, Set	Height of the fixed center of the mechanism to the flange point of the motor. **Optional if** `DriveArm` **and** `LoadArm` **are aligned in the closed position.**
DriveArm	LREAL	Get, Set	Lever arm attached to the drive.
LoadArm	LREAL	Get, Set	Transfer arm to the tool arms.
ToolArmAngle	LREAL	Get, Set	Internal angle between the two parts of the side tool arm.
ToolArm_1	LREAL	Get, Set	Lower part of the lateral tool arm.
ToolArm_2	LREAL	Get, Set	Upper part of the lateral tool arm.
ToolArm_3	LREAL	Get, Set	Horizontal tool arm.
ToolJointHeight	LREAL	Get, Set	Height of the anchor point (joint) of the tool mounted in the direction of movement. **Optional if** `ToolArm_3` **remains almost horizontal.**
ToolJointOffset	LREAL	Get, Set	Offset of the tool clamping surface to the anchor point mounted in the direction of movement. **In the recommended configuration of this parameter, the transformation results in the distance between the tool clamping surfaces. To use the distance of the tool opening, it is recommended to use the core function** `ToolAdaption` **.**
DriveHighEnd	LREAL	Get, Set	Upper limit of the drive position.
DriveLowEnd	LREAL	Get, Set	Lower limit of the drive position.
DriveStep	LREAL	Get	Step size of the drive positions.
ParamValid	BOOL	Get	The set parameters have valid values.
Points	INT	Get, Set	Number of points to be defined.

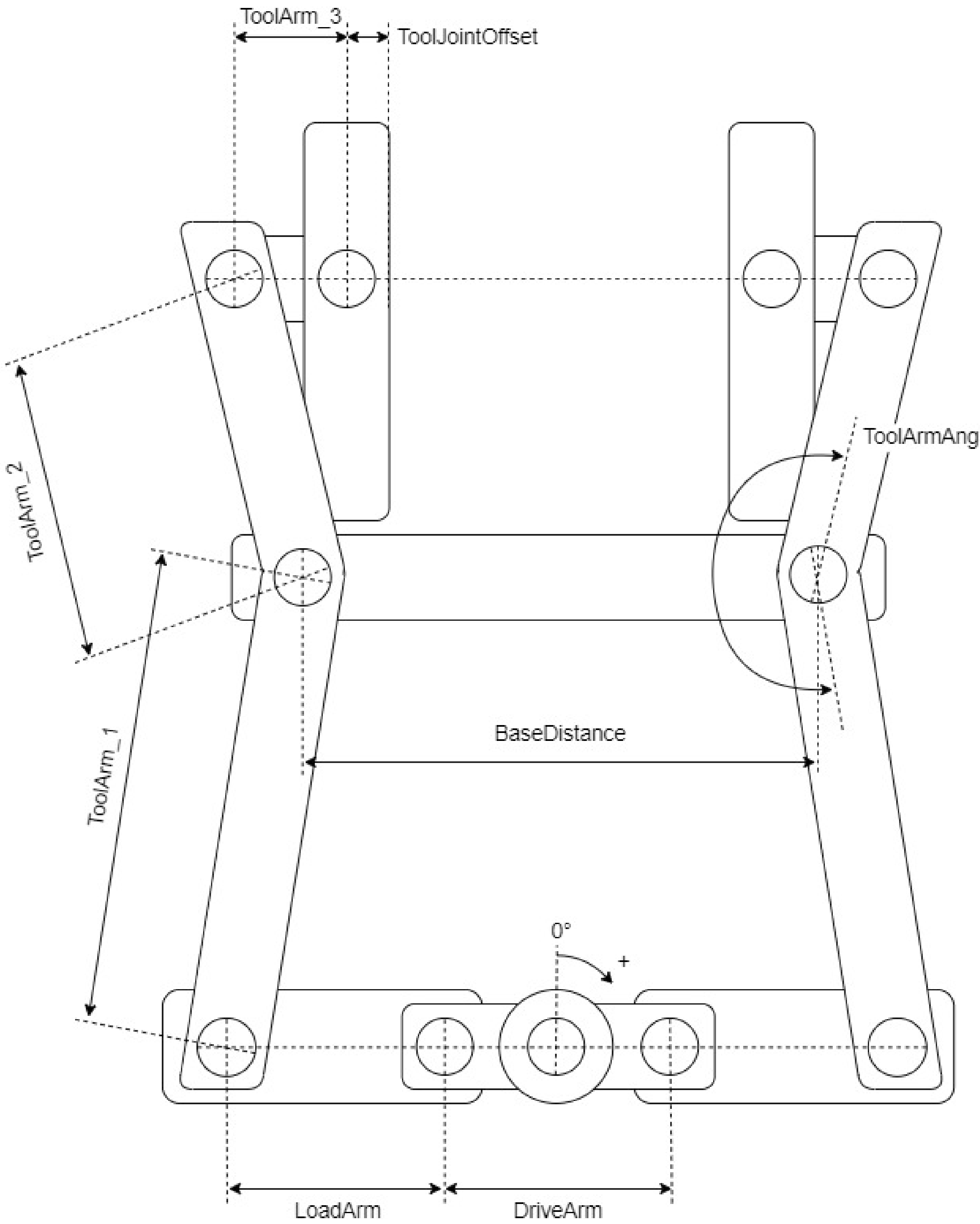
Methods

Name	Description
DefineTable()	Starts the calculation of the transformation table.
Reset()	Resets generator calculations.

Interfaces

Type	Description
I_TrafoTableGenerator	Standard interface on FB_TrafoTableGenerator.
I_ClampTableGenerator	Standard interface on FB_ClampTableGenerator.

Drawing without optional parameters



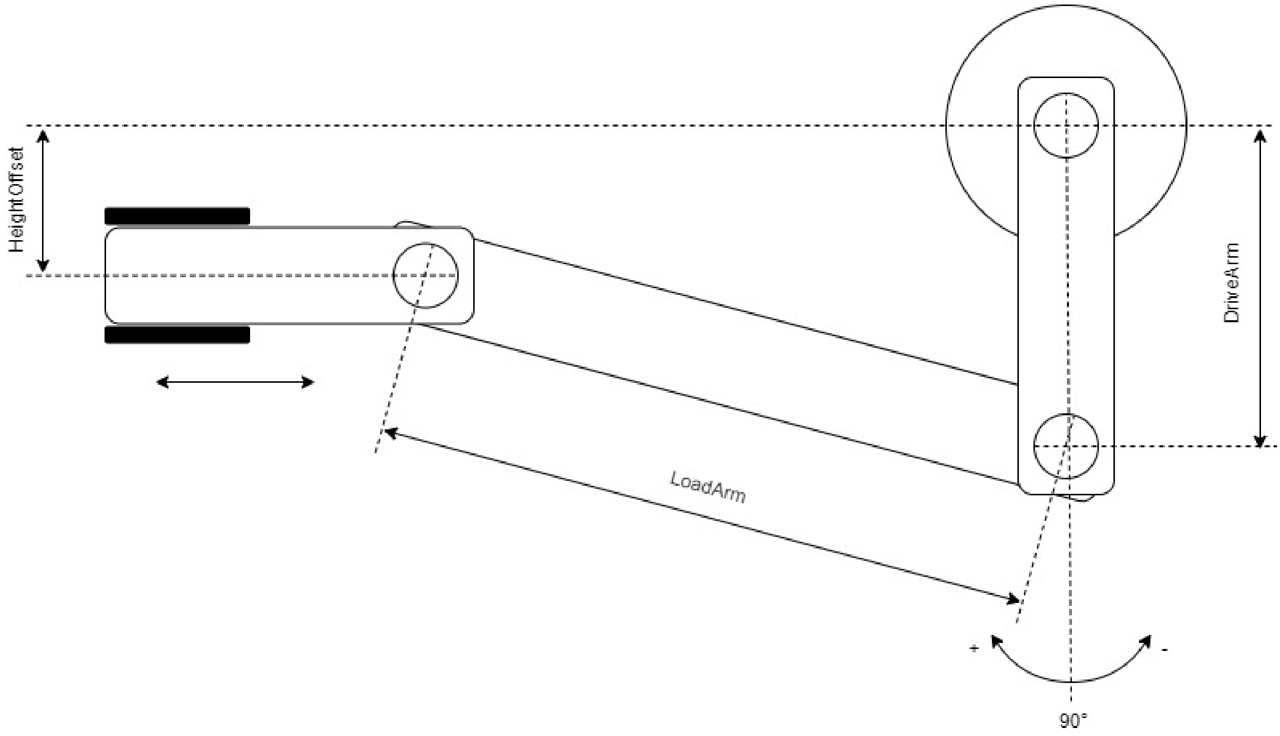
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.6.10 FB_CrankTableGenerator

FB_CrankTableGenerator

Implements a calculation algorithm for a crank mechanism.



Names of the dimension designations in the sketch

The dimension designations are the same as the names of the properties of the class to be configured.

Syntax:

```
FUNCTION_BLOCK FB_CrankTableGenerator EXTENDS FB_TrafoTableGenerator
```

Properties

Name	Type	Access	Description
LookUp	I_CammingLookUp [▸ 85]	Get, Set	Interface to the table to be written
DriveArm	LREAL	Get, Set	Lever arm attached to the drive.
HeightOffset	LREAL	Get, Set	Height offset between the guiding direction of the load and the pivot point of the drive.
LoadArm	LREAL	Get, Set	Transfer arm to the guided load side.
DriveHighEnd	LREAL	Get, Set	Upper limit of the drive position.
DriveLowEnd	LREAL	Get, Set	Lower limit of the drive position.
DriveStep	LREAL	Get	Step size of the drive positions.
ParamValid	BOOL	Get	The set parameters have valid values.
Points	INT	Get, Set	Number of points to be defined.

Methods

Name	Description
DefineTable()	Starts the calculation of the transformation table.
Reset()	Resets generator calculations.

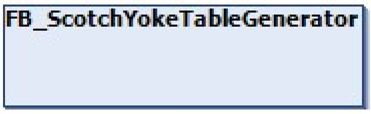
Interfaces

Type	Description
I_TrafoTableGenerator	Standard interface on FB_TrafoTableGenerator.
I_CrankTableGenerator	Standard interface on FB_CrankTableGenerator.

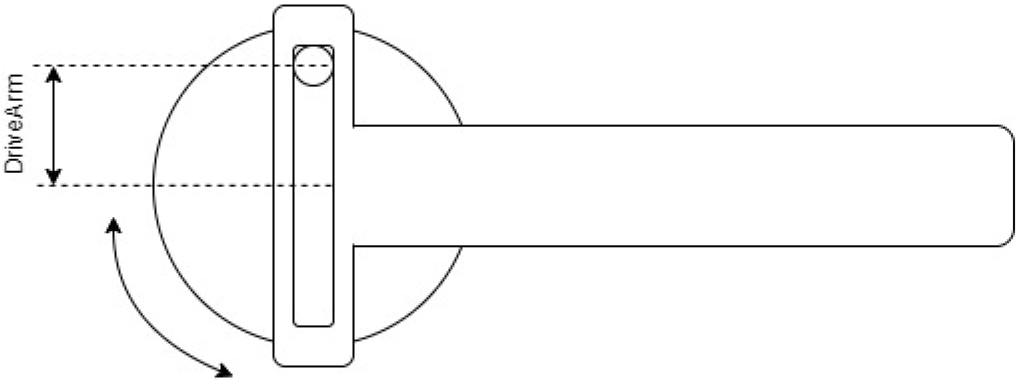
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.6.11 FB_ScotchYokeTableGenerator


FB_ScotchYokeTableGenerator

Implements a calculation algorithm for a scotch yoke mechanism.



Names of the dimension designations in the sketch

The dimension designations are the same as the names of the properties of the class to be configured.

Syntax:

```
FUNCTION_BLOCK FB_ScotchYokeTableGenerator EXTENDS FB_TrafoTableGenerator
```

Properties

Name	Type	Access	Description
LookUp	I_CammingLookUp [▶ 85]	Get, Set	Interface to the table to be written
DriveArm	LREAL	Get, Set	Radius of the eccentric rotation.
DriveHighEnd	LREAL	Get, Set	Upper limit of the drive position.
DriveLowEnd	LREAL	Get, Set	Lower limit of the drive position.
DriveStep	LREAL	Get	Step size of the drive positions.

Name	Type	Access	Description
ParamValid	BOOL	Get	The set parameters have valid values.
Points	INT	Get, Set	Number of points to be defined.

Methods

Name	Description
DefineTable()	Starts the calculation of the transformation table.
Reset()	Resets generator calculations.

Interfaces

Type	Description
I_TrafoTableGenerator	Standard interface on FB_TrafoTableGenerator.
I_ScotchYokeTableGenerator	Standard interface on FB_ScotchYokeTableGenerator.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.7 Linear actuators

Some handling and/or processing devices require a very simple type of drive technology. From the process side, it is of secondary importance whether the drive is operated with a pneumatic/hydraulic cylinder or a miniature servomotor with spindle, for example. The requirement of the application is a constant linear movement with a base position and a working position.

The TwinCAT 3 Plastic Technology Functions offer a simplified variant of the type `FB_AxisBase` for this purpose: `FB_MotionBase`. Based on this motion class, simple linear movements can be programmed for both axes and digital actuators. The digital actuators can be instantiated in a project via the `FB_ActuatorBase` class.

3.7.1 I_MotionBase

Represents the general interface for a motion unit.

Syntax:

```
INTERFACE I_MotionBase EXTENDS I_MessageBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.

Name	Type	Access	Description
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.7.2 I_MotionBaseDev

Represents the extended interface for a motion unit.

Syntax:

```
INTERFACE I_MotionBaseDev EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
Jog [▶ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.

Name	Description
AppendCorefunction(iCoreFunc)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(eNewState)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.7.3 I_ActuatorBase

Represents the general interface for an actuator.

Syntax:

```
INTERFACE I_ActuatorBase EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
ActualsActuator [▸ 73]	I_ActualsActuator	Get	Provision of current actual values (positions, velocities, motion times, etc.). It is an extended version of Actuals.
Tuning [▸ 171]	I_ActuatorTuning	Get	Tuning function for calibrating the movement time.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▸ 37]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
DoBase(bEnable)	Command for the movement to the base position (Jog.DoJogM()).
DoWork(bEnable)	Command to move to the working position (Jog.DoJogP()).
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

3.7.4 I_ActuatorBaseDev

Represents the extended interface for an actuator.

Syntax:

```
INTERFACE I_ActuatorBaseDev EXTENDS I_MotionBaseDev, I_ActuatorBase
```

Properties

Name	Type	Access	Description
Actuals [▸ 72]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
FeedbackBase	I_InputBase	Get, Set	I/O interface to an end position sensor of the base position.
FeedbackPower	I_InputBase	Get, Set	I/O interface to the status feedback of a power release.
FeedbackWork	I_InputBase	Get, Set	I/O interface to an end position sensor of the working position.
OutputBase	I_OutputBase	Get, Set	I/O interface for controlling the base position.
OutputPower	I_OutputBase	Get, Set	I/O interface for controlling a power release.
OutputWork	I_OutputBase	Get, Set	I/O interface for controlling the working position.
CmdCurrent	UDINT	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdCurrentId	E_CurrentCmd	Get, Set	[INTERNAL] A counter that assigns an identification to each activated command.
CmdNext	UDINT	Get	[INTERNAL] The next value to be used as CmdCurrent.
CoreDebug	I_CoreDebug	Get	[INTERNAL] Debugging support
CycleTime	LREAL	Get	[INTERNAL] In the startup phase, the axis will determine the cycle time of the PLC task that runs the axis function. The axis will not execute any function if it could not update this value.
CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
ActualsActuator [▸ 73]	I_ActualsActuator	Get	Provision of current actual values (positions, velocities, motion times, etc.). It is an extended version of Actuals.
Tuning [▸ 171]	I_ActuatorTuning	Get	Tuning function for calibrating the movement time.
Jog [▸ 126]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▸ 129]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▸ 134]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▸ 37]	E_AxiState	Get	The current state of the axis.

I/O interfaces optional

All I/O interfaces of an actuator are optional. The feedback signals can be simulated over time via `MotionParams.ActuatorXyz.FeedbackDelay`.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
AppendCorefunction(`iCoreFunc`)	[INTERNAL] For internal use only.
EnterCriticalSection()	[INTERNAL] For internal use only.
ForceState(`eNewState`)	This method changes the state of the axis with high priority.
LeaveCriticalSection()	[INTERNAL] For internal use only.
ReadCycleTime()	[INTERNAL] Used to determine the cycle time.
DoBase(`bEnable`)	Command for the movement to the base position (`Jog.DoJogM()`).
DoWork(`bEnable`)	Command to move to the working position (`Jog.DoJogP()`).
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4 Core functions

In plastics machines, an axis must perform a variety of elementary motion tasks, such as multi-stage PTP movements, pressure control, homing, and so on. Usually, these motion commands are independent of each other and the axis performs only one motion command at a time. TwinCAT 3 Plastic Technology Functions implements and encapsulates each of these motion tasks in a separate FB called a Corefunction.

The Corefunctions were defined in a standardized format. The base axis defined in TwinCAT 3 Plastic Technology Functions provides the common Corefunctions. By deriving a specific axis, the user can also replace or supplement the function of a Corefunction without influencing the behavior of other Corefunctions. This makes the TwinCAT 3 Plastic Technology Functions flexible and easy to customize.

4.1 Embedding core functions in an axis

A core function cannot operate independently because it contains only the code implementation of the motion task, but is not connected to the Motion Control library. When an axis is initialized, the core function receives the interface of the axis, while the axis contains an interface of the core function. At the same time, the core function receives interfaces to Motion Control library functions.

The axis creates a concatenated list to call its core functions. This list is used during the operation of the axis for the following tasks:

- Signaling an Online Change
- Passing on information about the cycle time
- Cyclic calling of methods
- Command for resetting errors

The core functions have access to the information of the axis. In addition, it is possible to call the interfaces of other core functions of the axis if this is necessary for the coordinated execution of tasks.

4.2 The basics of core functions (FB_CoreFunction)

All Corefunctions have a number of common features that are implemented in a `ABSTRACT FB_Corefunction`. A number of Corefunctions are derived from FB_CorefunctionFeedback [▶ 63] and provide further properties and methods. Through the inheritance of `FB_Corefunction`, each Corefunction receives the properties and methods described here.

Core functions are instantiated within motion objects as local elements. At startup, the core function receives an interface to its host axis and, if necessary, to a drive-related adaptation.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_Corefunction EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
AxisState	E_AxisState	Get, Set	[INTERNAL] The current state of the axis state machine.
Connected	BOOL	Get	[INTERNAL] The Corefunction is connected to an axis.
CycleTime	LREAL	Get, Set	[INTERNAL] The cycle time of the PLC task from which the method `Cyclic()` of the Corefunction is executed.

Name	Type	Access	Description
CycleTimeValid	BOOL	Get, Set	[INTERNAL] A TRUE indicates that the `CycleTime` of the Corefunction is defined.
FailedState	BOOL	Get, Set	If IsActivated is TRUE at the same time: Signals the failure of an accepted command. If IsActivated is FALSE at the same time: Signals the rejection of a command.
HasFeedback	BOOL	Get	The core function has responded to a pending command.
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
IsActivated	BOOL	Get	The core function has an accepted command pending.
IsLocalCmd	BOOL	Get	Signals that the axis is assigned with a command of this core function.
NextCore	I_Corefunction	Get, Set	[INTERNAL] This property is part of the execution chain and must not be affected by the application task.
OnlineChangeMark	BOOL	Get, Set	[INTERNAL] The axis uses this property to signal an Online Change to the Corefunction.
ReadyState	BOOL	Get, Set	The Corefunction is ready for operation, but is not commandable at this time. Possible reasons are: - The axis is not enabled. - Another Corefunction is active. - The Corefunction does not support any commands.

Methods

Name	Description
ApplyCommand()	[INTERNAL] Applies the command to the axis.
ClearStates()	[INTERNAL] Clears up some of the possible states.
Cyclic()	[INTERNAL] This method is called cyclically by the axis.
DoReset(b)	[INTERNAL] This method is called when the axis is reset.
RemoveCommand()	[INTERNAL] Removes the command from the axis.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

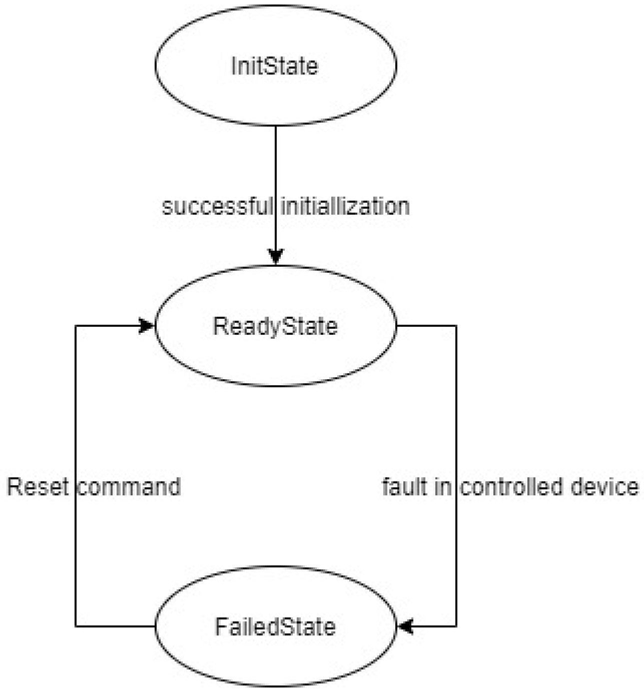
4.3 Permanently active core functions

A Corefunction from this group does not accept any commands and remains active as soon as a connection with the axis is established, i.e. the Corefunction remains in the ReadyState. The state transitions and conditions of the constantly active Corefunctions are shown in the figure below.

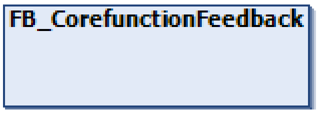
The following Corefunctions are of this type:

Permanently active Core-functions	Description
Actuals [▶ 72]	Contains the state of the axis.
ActualsActuator [▶ 73]	Is derived from Actuals and interprets position and velocity as Boolean.
ActualsHydraulics [▶ 74]	Derived from Actuals and extends it to include hydraulic-specific elements.
ActualsNc [▶ 76]	Is derived from Actuals and extends it with NC-specific elements.
MotionParams [▶ 129]	Provides access to a range of axis parameters.
MotionSetpoints [▶ 134]	Contains the current setpoints of the axis.
PtpLookUp [▶ 154]	Contains the multi-segment PTP move command.
ToolAdaption [▶ 165]	Contains the parameters of a tool adaptation.

State machine



4.4 Commanded core functions (FB_CorefunctionFeedback)



These Corefunctions are activated by a command. The functions of this group are derived from a `ABSTRACT FB_CorefunctionFeedback`. As a result, all Corefunctions of this group have additional properties compared to the variants derived from `FB_Corefunction`.

Non-functional situation

A situation that does not allow the use of the core function. This may be caused by a failed or missing initialization or any other problem that causes damage to the axis or its sub-components. In this case the axis is permanently in `InitState`.

Idling situation

In this situation, the axis has no active, failed or completed command pending. As a rule, there are some requirements that must be considered when activating the core function. If it can be activated, it reports `IdleState`, otherwise `ReadyState`. Refer to the core function documentation for details.

No command

The core functions of the always active group do not support a command (and do not need one). Consequently, they will not report `IdleState` at any time.

Active situation

While the core function is actively performing its task, it reports `BusyState`. This situation ends when a fault is detected in the controlled component or device, or when another function has taken over control. For edge-triggered core functions and some statically controlled core functions, this situation ends when the task is successfully completed. In these cases, the core function is changed to the final situation.

Performed task is aborted

Statically controlled core functions leave this situation by terminating the executed task and falling back into idle mode as soon as the command of the core function is cleared.

Final situation

In this situation, the signals are determined by the result of the previous active situation. A `DoneState` reports a successfully completed task. A `FailedState` or `AbortedState` indicates a failed result or an abort by another function.

Check the command input

In the next cycle, after the result of the above active situation has been signaled, the core function starts checking the command input. If the input is `FALSE`, the core function falls back to idle.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_CorefunctionFeedback EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
AxisState	E_AxisState	Get, Set	[INTERNAL] The current state of the axis state machine.
AbortedState	BOOL	Get, Set	Signals the abort of a command by another Corefunction.
BusyState	BOOL	Get, Set	Signals the active execution of a command.
DoneState	BOOL	Get, Set	Signals the successful execution of a command.
IdleState	BOOL	Get, Set	The Corefunction is ready for operation and commandable.
IsCommanded	BOOL	Get	Signals the pending of a command.
Connected	BOOL	Get	[INTERNAL] The Corefunction is connected to an axis.
CycleTime	LREAL	Get, Set	[INTERNAL] The cycle time of the PLC task from which the method `Cyclic()` of the Corefunction is executed.
CycleTimeValid	BOOL	Get, Set	[INTERNAL] A TRUE indicates that the `CycleTime` of the Corefunction is defined.
FailedState	BOOL	Get, Set	If IsActivated is TRUE at the same time: Signals the failure of an accepted command. If IsActivated is FALSE at the same time: Signals the rejection of a command.

Name	Type	Access	Description
HasFeedback	BOOL	Get	The core function has responded to a pending command.
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
IsActivated	BOOL	Get	The core function has an accepted command pending.
IsLocalCmd	BOOL	Get	Signals that the axis is assigned with a command of this core function.
NextCore	I_Corefunction	Get, Set	[INTERNAL] This property is part of the execution chain and must not be affected by the application task.
OnlineChangeMark	BOOL	Get, Set	[INTERNAL] The axis uses this property to signal an Online Change to the Corefunction.
ReadyState	BOOL	Get, Set	The Corefunction is ready for operation, but is not commandable at this time. Possible reasons are: - The axis is not enabled. - Another Corefunction is active. - The Corefunction does not support any commands.

Methods

Name	Description
ApplyCommand()	[INTERNAL] Applies the command to the axis.
ClearStates()	[INTERNAL] Clears up some of the possible states.
Cyclic()	[INTERNAL] This method is called cyclically by the axis.
DoReset(b)	[INTERNAL] This method is called when the axis is reset.
RemoveCommand()	[INTERNAL] Removes the command from the axis.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.4.1 Edge-triggered core functions

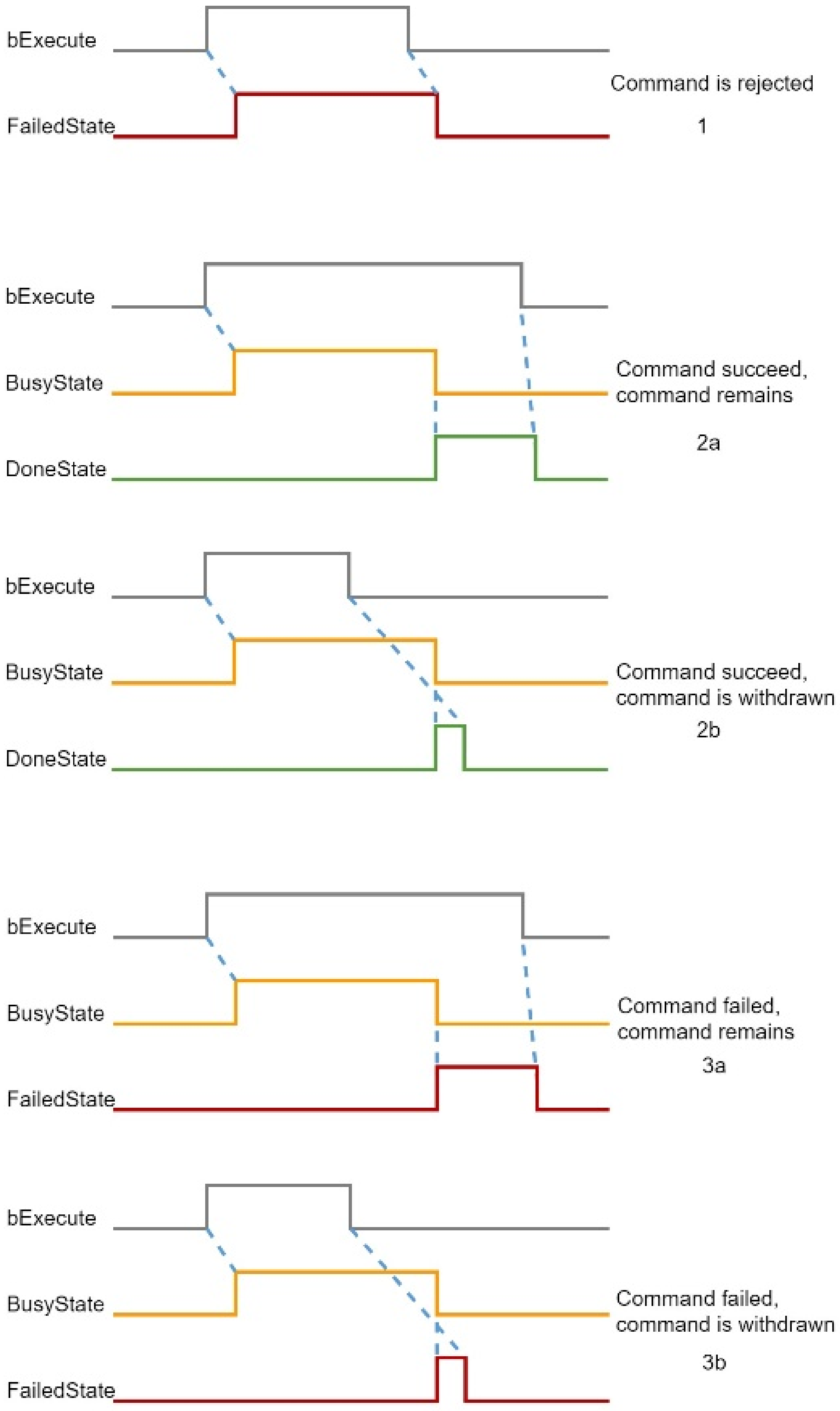
The command accepted by these core functions is usually called `bExecute` (with the exception of `Power`, which is described below).

The rising edge of `bExecute` triggers a series of checks to determine whether the execution conditions are met. If the execution conditions are not met, the command is rejected and the core function is set to `FailedState` (case 1 in the figure below). If the execution conditions are met, the command is accepted and the core function enters `BusyState`. The falling edge of `bExecute` does not directly trigger a response in `BusyState` (case b below). If the command was executed successfully, the core function enters the `DoneState` (case 2 below), otherwise it enters the `FailedState` (case 3 below). At this time it is checked if `bExcute` is still `TRUE` and the state of the core function is changed in the next PLC cycle. The state transition is shown in the figure below.

The following functions are edge-triggered core functions:

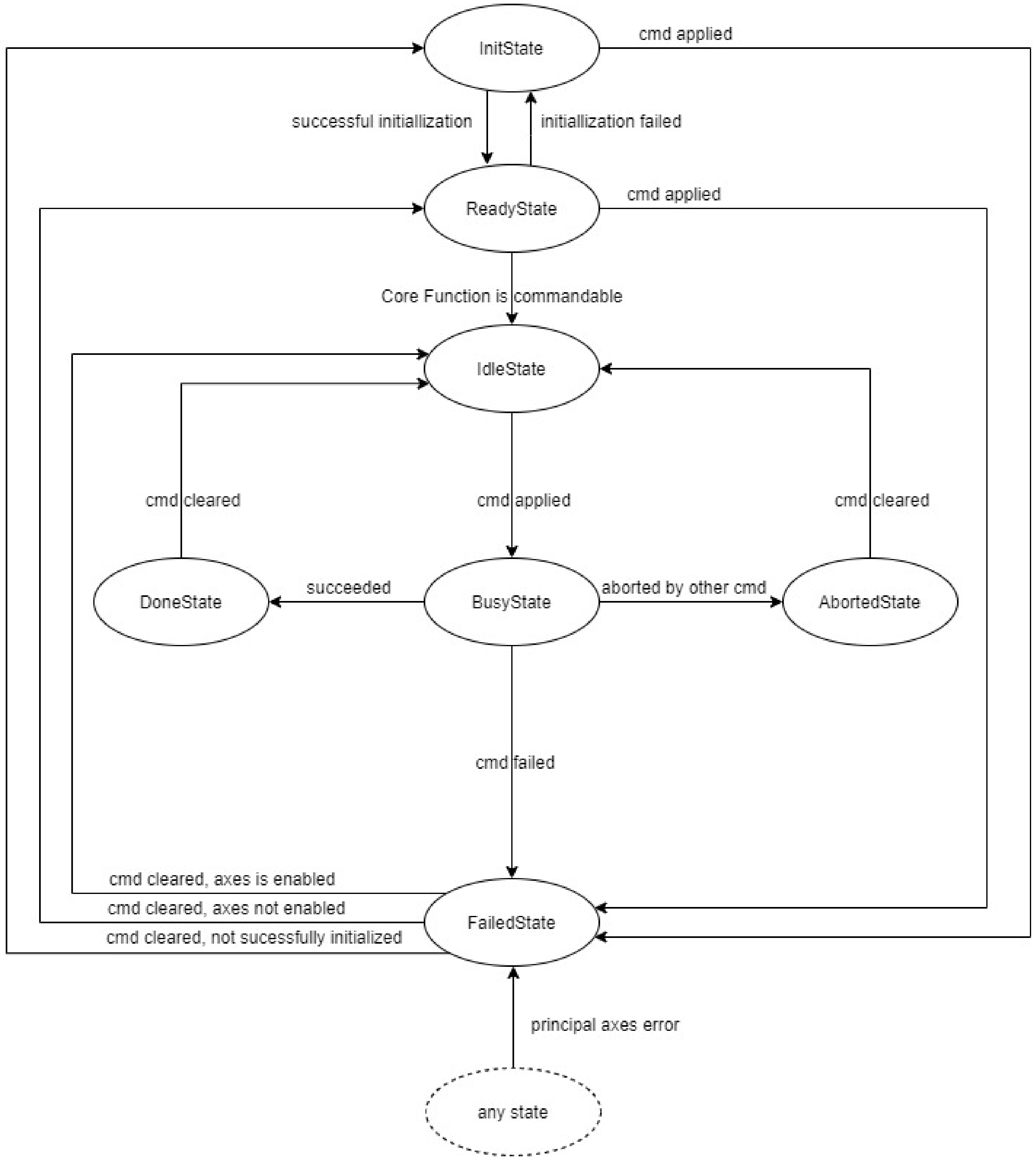
Corefunction	Description
DisableSoftEnd [▶ 95]	Is used to temporarily disable and re-enable the soft limit switches of the axis.
Homing [▶ 100]	Provides a range of homing methods.
Power [▶ 134]	Used to activate and deactivate the axis.
Ptp [▶ 146]	Is used to perform multi-segment PTP movements.
Reset [▶ 158]	Is used to reset the error state of axis and devices.
SetPosition [▶ 158]	Is used to change the actual position of the axis.
Estop [▶ 98]	Command for emergency shutdown using the maximum dynamic parameters to stop the axis.
Stop [▶ 161]	Command for stop with the standard dynamic parameters for stopping the axis.
TeachPosition [▶ 163]	Used for referencing analog encoder systems.
TeachUpdate [▶ 163]	Used for referencing analog encoder systems. Is used after TeachPosition to calculate the measured values.
AutoIdent [▶ 77]	A special core function for hydraulic axes. It is used to identify the characteristic velocity behavior of the axis.
DirectOutput [▶ 93]	A special core function for hydraulic axes. It is used to send output signals directly to the control device.
Tuning [▶ 171]	A special core function for actuators. It is used for automatic optimization such as motion time measurement.

Timing Digram



Edge triggered

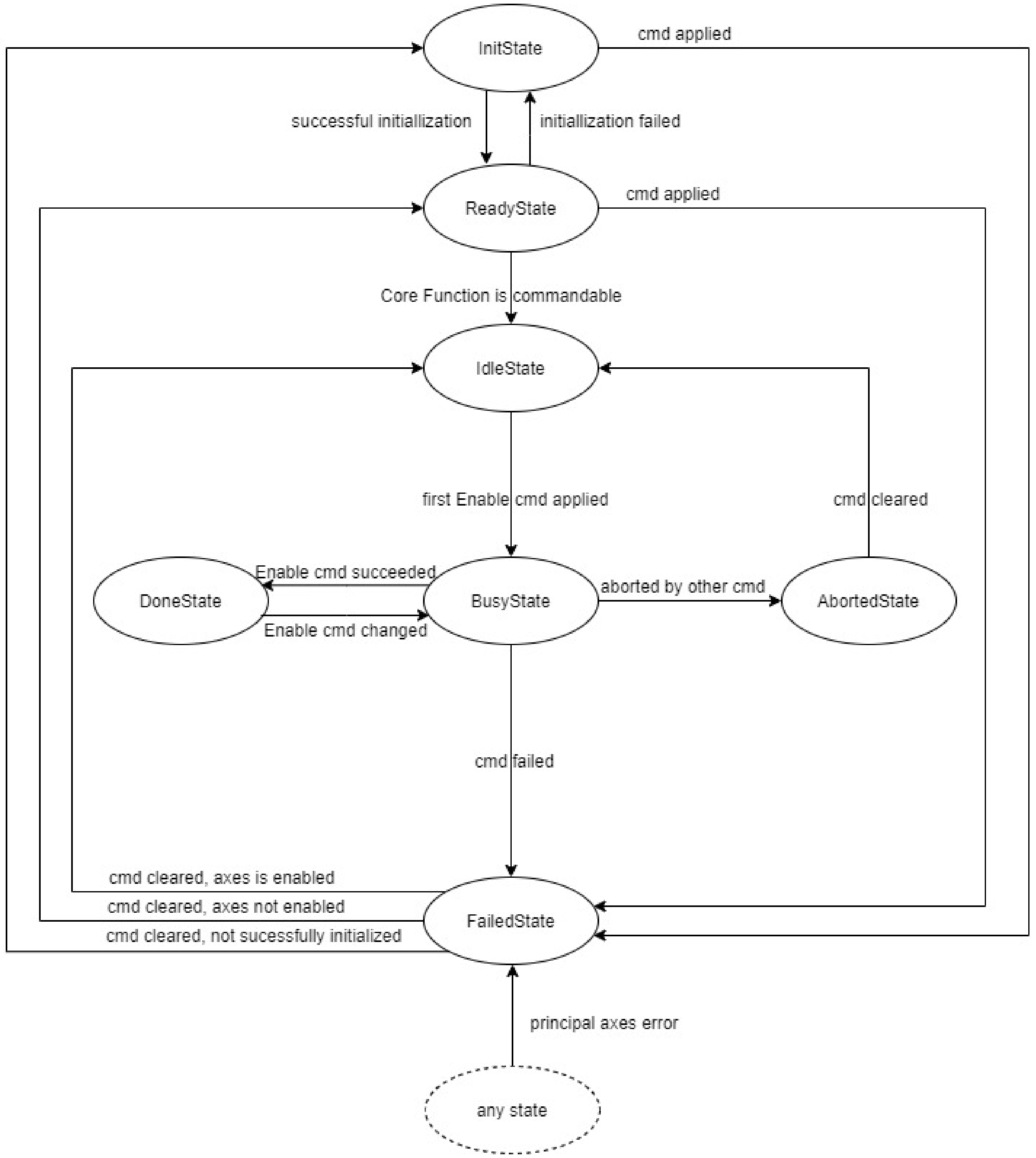
State machine



Exceptional case: Power

Power has a deviant behavior

Although power is an edge-triggered core function, the command for Power is called `bEnable`. Power has a different state machine than other edge-triggered core functions: with `DoneState` it returns to `BusyState` when the `bEnable` signal changes.



4.4.2 Statically controlled core functions

The command that is accepted by these core functions is usually called `bEnable`.

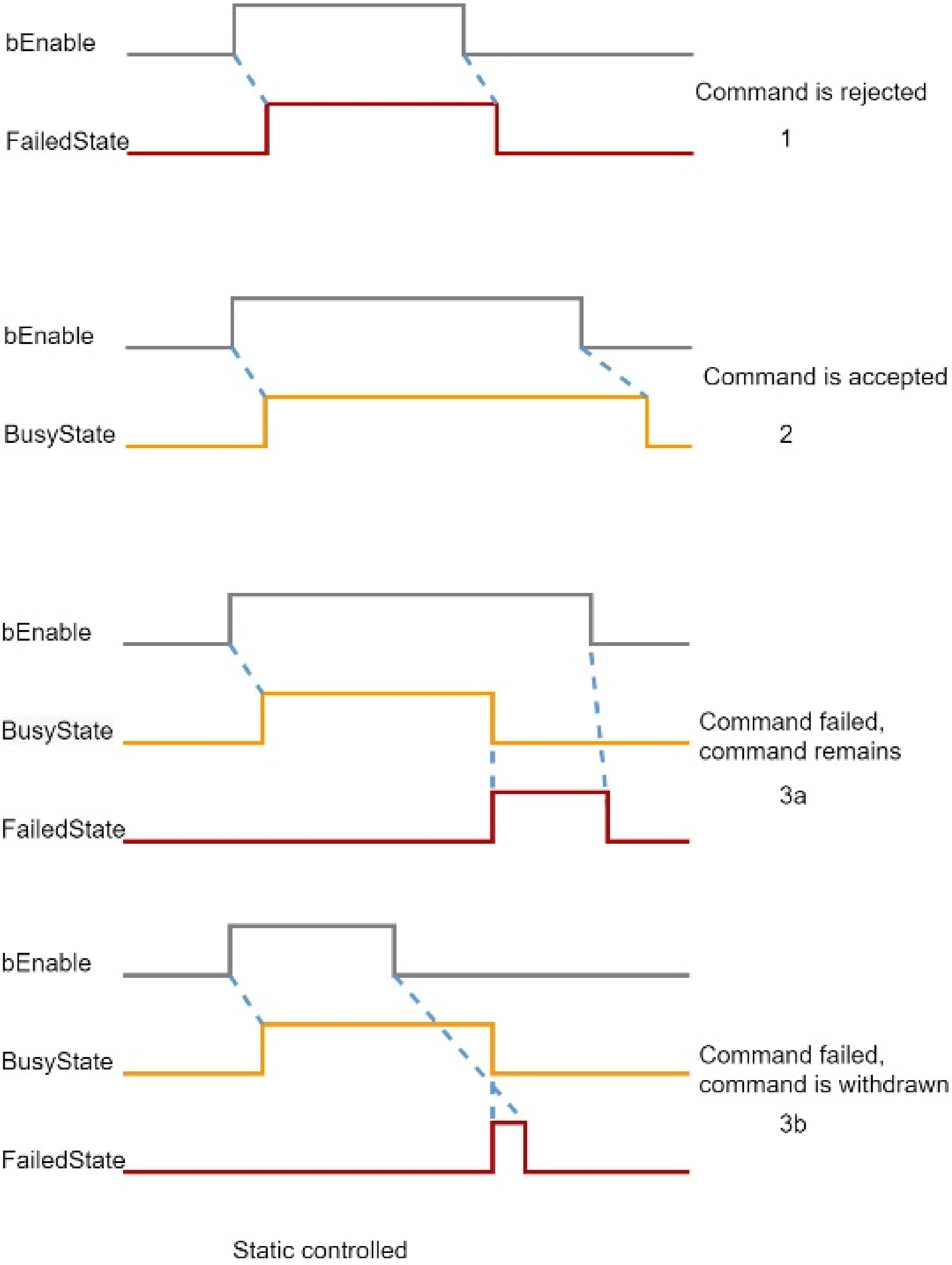
The rising edge of `bEnable` triggers a series of checks to determine whether the execution conditions are met. If the execution conditions are not met, the command is rejected and the core function enters the `FailedState` (case 1 in the figure below). If the execution conditions are met, the command is accepted and the core function enters `BusyState`. The falling edge of `bEnable` triggers a response to terminate execution (case 2 below).

If an error occurs during the execution of the command, the core function enters the `FailedState` (case 3 below). At this time it is checked if `bEnable` is still `TRUE` and the state of the core function is changed in the next PLC cycle.

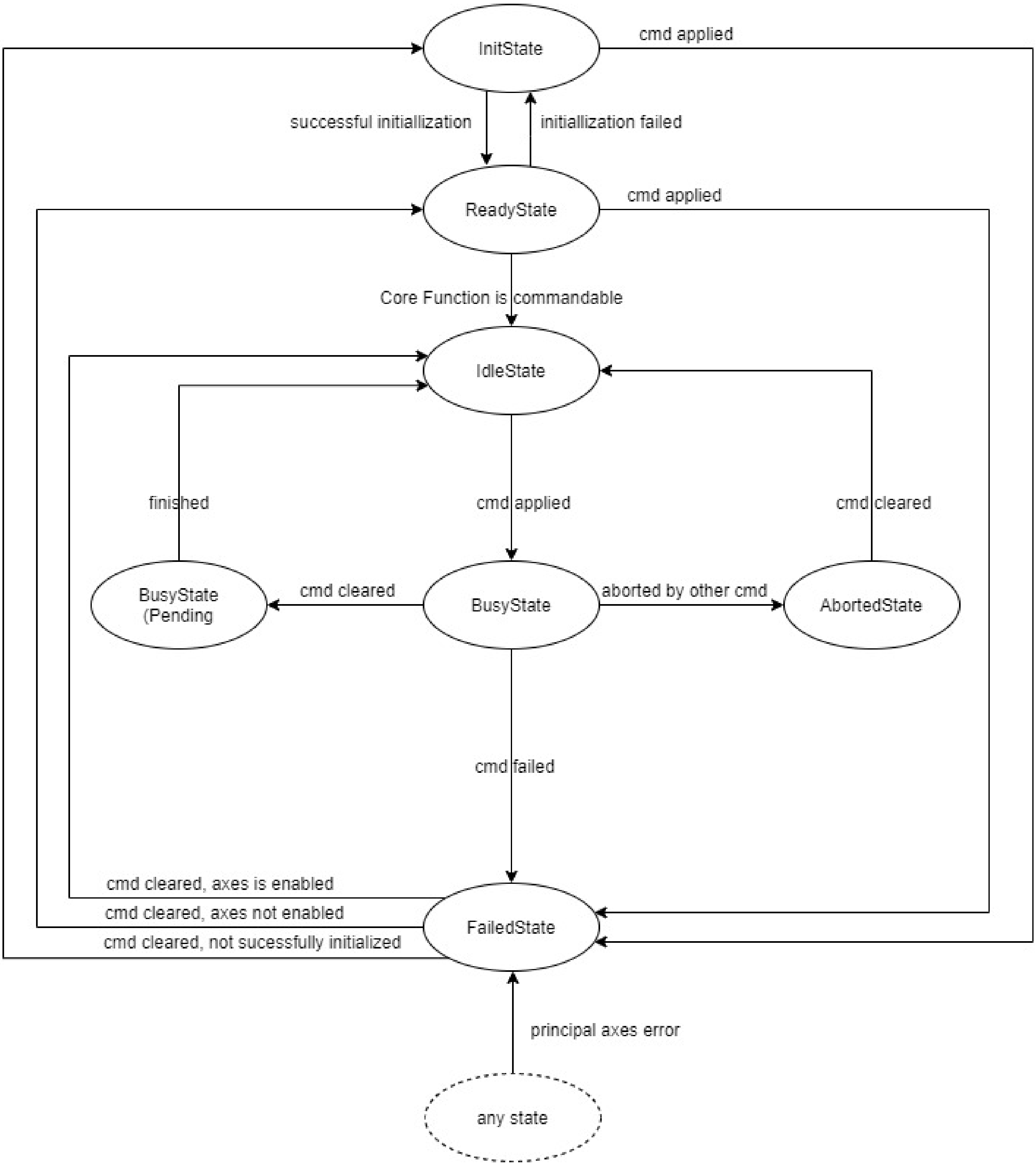
The following core functions are statically controlled:

Corefunction	Description
Camming [▶ 81]	Is used to activate setpoint generation that is controlled by a guide value.
ExternalGenerating [▶ 97]	Is used internally for setpoint generation by TwinCAT 3 Plastic Technology Functions.
Jog [▶ 126]	Is used to command a movement without a specified target.
PressureControl [▶ 137]	Reserved for future extension.
VelocityFeed [▶ 172]	Is used to activate a movement without a specified target and a velocity that is controlled by a guide value.
Transformation [▶ 166]	A special core function for transforming axes. It is used for coupling the two internal axes of a transforming axis.

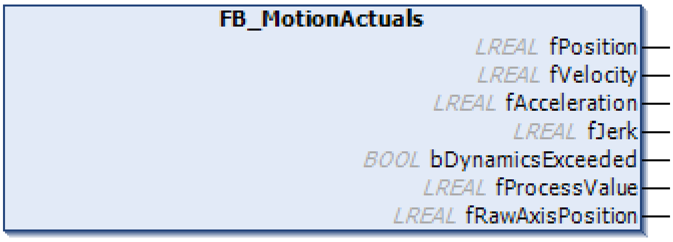
Timing Diagram



State machine



4.5 Actuals



```
FB_MotionActuals
                  LREAL fPosition
                  LREAL fVelocity
              LREAL fAcceleration
                      LREAL fJerk
          BOOL bDynamicsExceeded
              LREAL fProcessValue
          LREAL fRawAxisPosition
```

This core function is not instantiated directly. It is used as a common part of type-specific core functions like ActualsHydraulics, ActualsNc or ActualsActuator.

All derivatives of Actuals are members of the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_MotionActuals EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
Acceleration	LREAL	Get	The current actual acceleration.
DynamicsExceeded	BOOL	Get, Set	TRUE if the dynamic limits of the axis were exceeded during a currently active camming. This signal is only deleted when camming is stopped.
Jerk	LREAL	Get	The current actual jerk.
Position	LREAL	Get	The current actual position.
ProcessValue	LREAL	Get, Set	The current actual process value.
RawAxisPosition	LREAL	Get	The unconverted current actual position.
SetEvent	BOOL	Get, Set	Reserved.
Velocity	LREAL	Get	The current actual velocity.

Interfaces

Type	Description
I_MotionActuals	Standard interface on FB_MotionActuals.

State interface

In a non-functional situation, the core function reports InitState. Otherwise ReadyState is reported.

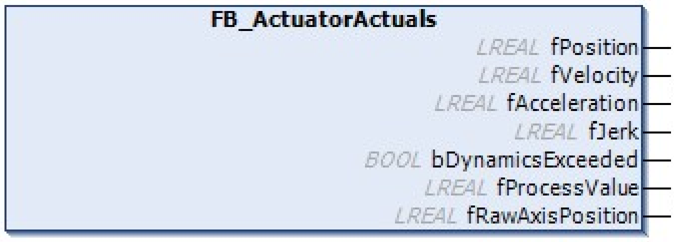
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.6 ActualsActuator

Exclusive function for actuators

This core function is only available for actuators.



This core function is used to display a compilation of actual values of an actuator. It is an extension of Actuals and extends the range of its parent element by adding specific interpretations for actuators.

Like all derivatives of Actuals, this core function belongs to the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_ActuatorActuals EXTENDS FB_MotionActuals
```

Properties

Name	Type	Access	Description
Acceleration	LREAL	Get	The current actual acceleration.
IsBase	BOOL	Get	Base position reached.
IsWork	BOOL	Get	Working position reached.
MoveTimeBase	LREAL	Get	Current/last movement time to the base position.
MoveTimeWork	LREAL	Get	Current/last movement time to the working position.
ToBase	BOOL	Get	Movement to the base position.
ToWork	BOOL	Get	Movement to the working position.
DynamicsExceeded	BOOL	Get, Set	TRUE if the dynamic limits of the axis were exceeded during a currently active camming. This signal is only deleted when camming is stopped.
Jerk	LREAL	Get	The current actual jerk.
Position	LREAL	Get	The current actual position.
ProcessValue	LREAL	Get, Set	The current actual process value.
RawAxisPosition	LREAL	Get	The unconverted current actual position.
SetEvent	BOOL	Get, Set	Reserved.
Velocity	LREAL	Get	The current actual velocity.

Alias of position and velocity

The derivative for actuators defines the following positions and velocities

- Position: 0 = `IsBase`, 1 = Intermediate position, 2 = `IsWork`
- Velocity: -1 = `ToBase`, 0 = Standstill, 1 = `ToWork`

Interfaces

Type	Description
I_MotionActuals	Standard interface on FB_MotionActuals.
I_ActuatorActuals	Standard interface on FB_ActuatorActuals.

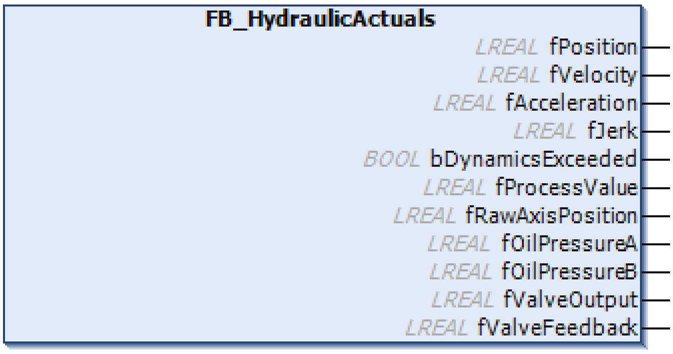
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4.7 ActualsHydraulics

Exclusive function for hydraulic axes

This core function is only available for hydraulic axis types.



This core function is used to display a compilation of actual values of the axis. It is an extension of Actuals and extends the range of its parent element by adding specific values for hydraulic axes.

Like all derivatives of Actuals, this core function belongs to the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_HydraulicActuals EXTENDS FB_MotionActuals
```

Properties

Name	Type	Access	Description
Acceleration	LREAL	Get	The current actual acceleration.
OilPressureA	LREAL	Get	The current actual pressure on the A-side of the cylinder.
OilPressureB	LREAL	Get	The current actual pressure on the B-side of the cylinder.
ValveFeedback	LREAL	Get	The current feedback value (slider actual position) of the valve.
ValveOutput	LREAL	Get	The current output value (slider set position) for the valve.
DynamicsExceeded	BOOL	Get, Set	TRUE if the dynamic limits of the axis were exceeded during a currently active camming. This signal is only deleted when camming is stopped.
Jerk	LREAL	Get	The current actual jerk.
Position	LREAL	Get	The current actual position.
ProcessValue	LREAL	Get, Set	The current actual process value.
RawAxisPosition	LREAL	Get	The unconverted current actual position.
SetEvent	BOOL	Get, Set	Reserved.
Velocity	LREAL	Get	The current actual velocity.

The terms A and B

The terms **A** and **B** are used in the definition of the hydraulics library, i.e. the side of the cylinder that makes the axis move in positive direction is regarded as the **A** side.

Sometimes no feedback

Not all types of proportional valves support this kind of feedback.

Interfaces

Type	Description
I_MotionActuals	Standard interface on FB_MotionActuals.

Type	Description
I_HydraulicActuals	Standard interface on FB_HydraulicActuals.

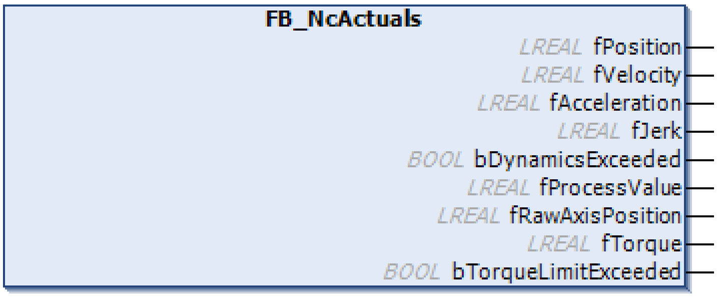
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

4.8 ActualsNc

Exclusive function for NC axes

This core function is only available for NC axis types.



This core function is used to display a compilation of actual values of the axis. It is an extension of Actuals and extends the range of its parent element by adding specific values for NC axes.

Like all derivatives of Actuals, this core function belongs to the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_NcActuals EXTENDS FB_MotionActuals
```

Properties

Name	Type	Access	Description
Acceleration	LREAL	Get	The current actual acceleration.
Torque	LREAL	Get	The current torque actual value.
TorqueLimitExceeded	BOOL	Get	TRUE if the torque limit has been reached.
DynamicsExceeded	BOOL	Get, Set	TRUE if the dynamic limits of the axis were exceeded during a currently active camming. This signal is only deleted when camming is stopped.
Jerk	LREAL	Get	The current actual jerk.
Position	LREAL	Get	The current actual position.
ProcessValue	LREAL	Get, Set	The current actual process value.
RawAxisPosition	LREAL	Get	The unconverted current actual position.
SetEvent	BOOL	Get, Set	Reserved.
Velocity	LREAL	Get	The current actual velocity.

Clear this property

TorqueLimitExceeded is set by the library, but not cleared. The application must be sure to clear this property at the start of a function that has to be monitored.

Interfaces

Type	Description
I_MotionActuals	Standard interface on FB_MotionActuals.
I_NcActuals	Standard interface on FB_NcActuals.

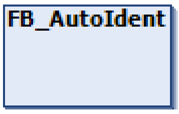
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

4.9 AutoIdent

Exclusive function for hydraulic axes

This core function is only available for hydraulic axis types.



This core function is used to analyze the non-linear transfer characteristic of hydraulic axes. It belongs to the group of edge-triggered core functions.

Methods

Name	Description
DoAutoIdent() [▶ 78]	Activates and terminates the measurement procedure.
SetParameter() [▶ 79]	Sets the parameters for the measurement procedure.

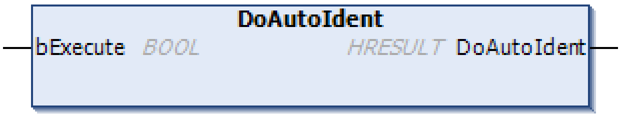
Interfaces

Type	Description
I_AutoIdent	Standard interface on FB_AutoIdent

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

4.9.1 DoAutoIdent()



This method is used to activate the core function.

Syntax:

```
METHOD DoAutoIdent : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoAutoIdent	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

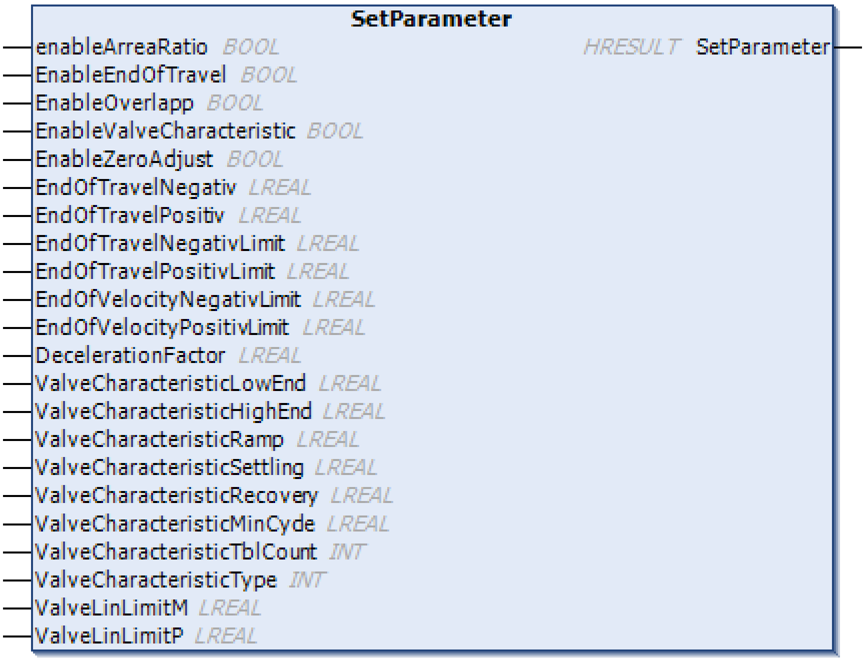
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge starts the identification. A falling edge cancels a still active identification with an invalid table.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

4.9.2 SetParameter()



SetParameter

enableArreaRatio BOOL
EnableEndOfTravel BOOL
EnableOverlapp BOOL
EnableValveCharacteristic BOOL
EnableZeroAdjust BOOL
EndOfTravelNegativ LREAL
EndOfTravelPositiv LREAL
EndOfTravelNegativLimit LREAL
EndOfTravelPositivLimit LREAL
EndOfVelocityNegativLimit LREAL
EndOfVelocityPositivLimit LREAL
DecelerationFactor LREAL
ValveCharacteristicLowEnd LREAL
ValveCharacteristicHighEnd LREAL
ValveCharacteristicRamp LREAL
ValveCharacteristicSettling LREAL
ValveCharacteristicRecovery LREAL
ValveCharacteristicMinCycle LREAL
ValveCharacteristicTblCount INT
ValveCharacteristicType INT
ValveLinLimitM LREAL
ValveLinLimitP LREAL

HRESULT SetParameter

This method can be used to set the parameters for identification.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    EnableArreaRatio            : BOOL;
    EnableEndOfTravel           : BOOL;
    EnableOverlapp              : BOOL;
    EnableValveCharacteristic   : BOOL;
    EnableZeroAdjust            : BOOL;
    EndOfTravelNegativ          : LREAL;
    EndOfTravelPositiv          : LREAL;
    EndOfTravelNegativLimit     : LREAL;
    EndOfTravelPositivLimit     : LREAL;
    EndOfVelocityNegativLimit   : LREAL;
    EndOfVelocityPositivLimit   : LREAL;
    DecelerationFactor          : LREAL;
    ValveCharacteristicLowEnd   : LREAL;
    ValveCharacteristicHighEnd  : LREAL;
    ValveCharacteristicRamp     : LREAL;
    ValveCharacteristicSettling: LREAL;
    ValveCharacteristicRecovery: LREAL;
    ValveCharacteristicMinCycle: LREAL;
    ValveCharacteristicTblCount: INT;
    ValveCharacteristicType     : INT;
    ValveLinLimitM              : LREAL;
    ValveLinLimitP              : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALID DATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.

E_AdsErr	Return value	Cause
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Designation	Type	Description
EnableArreaRatio	BOOL	A TRUE will request a separate identification of effects caused by cylinder asymmetry.
EnableEndOfTravel	BOOL	A TRUE will request a separate identification of mechanical limits of traveling.
EnableOverlapp	BOOL	A TRUE will request a separate identification of effects caused by valve overlap.
EnableValveCharacteristic	BOOL	A TRUE will request the identification of the valves transfer characteristic.
EnableZeroAdjust	BOOL	A TRUE will request a separate identification of effects caused by valve offset.
EndOfTravelNegativ	LREAL	A mechanical travel limit. This value may be found by identification or entered by using an HMI.
EndOfTravelPositiv	LREAL	A mechanical travel limit. This value may be found by identification or entered by using an HMI.
EndOfTravelNegativLimit	LREAL	This value defines a control value limit. The identification will be concluded for that direction if the output to the control device has reached the limit.
EndOfTravelPositivLimit	LREAL	This value defines a control value limit. The identification will be concluded for that direction if the output to the control device has reached the limit.
EndOfVelocityNegativLimit	LREAL	This value defines an actual velocity limit. The identification will be concluded for that direction if the actual velocity exceeds the limit.
EndOfVelocityPositivLimit	LREAL	This value defines an actual velocity limit. The identification will be concluded for that direction if the actual velocity exceeds the limit.
DecelerationFactor	LREAL	This value defines the limits of travel that will be used for the identification.
ValveCharacteristicLowEnd	LREAL	This value defines the limits of travel that will be used for the identification.
ValveCharacteristicHighEnd	LREAL	This value defines the limits of travel that will be used for the identification.
ValveCharacteristicRamp	LREAL	This parameter defines the ramping to the output value currently under investigation.
ValveCharacteristicSettling	LREAL	This parameter defines the delay for starting the investigation after ramping to the output value.
ValveCharacteristicRecovery	LREAL	This parameter defines a recovery time before continuing the identification in the opposite direction.
ValveCharacteristicMinCycle	LREAL	This value specifies a minimum limit for identification.
ValveCharacteristicTblCount	INT	This parameter defines the number of points in the linearization table. *Notice* **This value must be an odd number. It must be in the range of 5 to 1001. Recommended values are 101, 201 or 401.**
ValveCharacteristicType	INT	This parameter is reserved to indicate the use of valves with special behavior details.

Designation	Type	Description
		For more details, see the hydraulics library documentation.
ValveLinLimitM	LREAL	This value limits the use of the linearization table.
ValveLinLimitP	LREAL	This value limits the use of the linearization table.

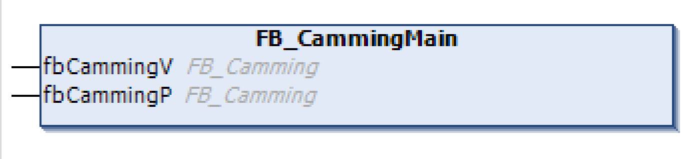
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

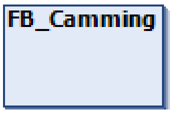
4.10 Camming

Core function is not supported by inverter axes

This core function is not supported by inverter axes. Any use will report DEVICE_NOTINIT and trigger an error message.



The property provides two options: Camming.Pos for position camming and Camming.Velo for velocity camming. To do this, two classes of the same type are instantiated, but with different parameterizations.



All these functions belong to the group of statically controlled core functions.

Both types of camming use a default value to identify a control value within a camming lookup table. For more information, see CammingLookUp [▶ 85].

Any useful information can serve as a default value. Common options are a time (`LREAL` variable that starts with zero and is cyclically updated by adding the cycle time of the PLC tasks) or the position of another axis.

⚠ WARNING

Unexpected responses of the controlled axis

Unsuitable default values or table points can lead to unexpected reactions of the controlled axis. This may result in risk of accident or damage.

Using a camming requires several steps:

- A camming lookup table must be instantiated. There is no rule for the number of these tables. A table used once can be used again at a later time by repeating the following steps.
- The table must be defined by loading the table points. Make sure that the point data corresponds to the camming type (position, velocity) for which it is to be used.
- The properties of the table must be updated.
- The table must be assigned to the camming with the method `SetLookupinterface()`. At this time, camming must not be commanded.
- An initial default value must be specified with the SetGuidingValue() method.
- At the right time use `DoCamming(bEnable:=TRUE)` to enable the function. Make sure that the axis is in a situation (position, velocity) that matches the situation of the camming table.

- Make sure that the default value is updated cyclically.
- At the right time use `DoCamming(bEnable:=FALSE)` to disable the function. Pay attention to the situation (position, velocity) of the axis.

Syntax:

```
FUNCTION_BLOCK FB_Camming EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
Overrun	BOOL	Get	TRUE if the current default value is not within the range of the camming lookup table.
Synchronize	BOOL	Get, Set	With Synchronize=FALSE the axis should follow every setpoint change immediately. A TRUE requires the axis to follow setpoint changes with respect to the dynamic limit parameters.
Synchronized	BOOL	Get	TRUE if Synchronize=TRUE and the setpoint changes are within the limits of the dynamic parameters of the axis.
UseAsPosition	BOOL	Get, Set	This property defines the camming rule. A TRUE causes the core function to act as a position camming. A FALSE will cause it to act as a velocity camming. This property is set during startup. The application must not change its setting.

Methods

Name	Description
DoCamming() [▸ 82]	This method is used to enable and disable the core function.
SetGuidingValue() [▸ 83]	This method is used to update the default value.
SetLookupInterface() [▸ 84]	This method must be used to connect a camming table.

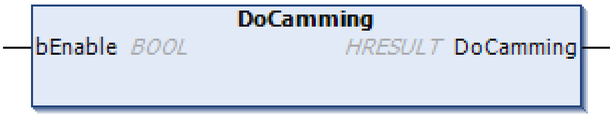
Interfaces

Type	Description
I_Camming	Standard interface on FB_Camming

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.1 DoCamming



This method is used to enable and disable the core function.

Syntax:

```
METHOD DoCamming : HRESULT
VAR_INPUT
    bEnable: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoCamming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

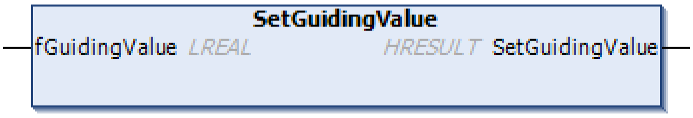
Inputs

Name	Type	Description
bEnable	BOOL	A TRUE activates the camming. A FALSE ends the camming and slows down a still existing movement.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.2 SetGuidingValue



This method must be used cyclically to update the default value.

Syntax:

```
METHOD SetGuidingValue : HRESULT
VAR_INPUT
    fGuidingValue: LREAL;
END_VAR
```

Return value

Name	Type	Description
SetGuidingValue	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

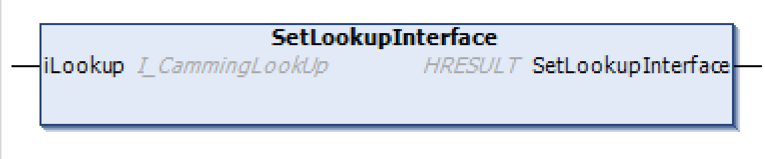
Inputs

Name	Type	Description
fGuidingValue	LREAL	The default value valid in the next cycle.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.3 SetLookupInterface



This method must be used to connect a camming table.

Syntax:

```
METHOD SetLookupInterface: HRESULT
VAR_INPUT
    iLookup: I_CammingLookUp;
END_VAR
```

Return value

Name	Type	Description
SetLookupInterface	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
iLookUp	I_CammingLookUp [▶ 85]	The camming table to use.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4 FB_CammingLookUp

This class is able to dynamically allocate memory and make it available as a table. Accordingly, this function block acts as a LookUp table for the cam plate core function.

Syntax:

```
FUNCTION_BLOCK FB_CammingLookUp EXTENDS FB_MessageBase
```

Initialization parameters of the FB_init constructor

Name	Type	Description
Points	INT	Defines the number of points = 0 enables subsequent assignment via `SetPoints(…)`
OpMode	E_LookupMode [▶ 92]	Defines the operation mode of the LookUp table

Properties

Name	Type	Access	Description
InvalidPoint	BOOL	Get	Is set by a call to `.LookUp()` or `.LookDown()` if the requested point is outside the valid range.
OpMode	E_LookupMode [▶ 92]	Get	Operation mode of the table
Overrun	BOOL	Get	Additional information on InvalidPoint: TRUE -> InvalidPoint is above the definition range FALSE -> InvalidPoint is below the definition range

Methods

Name	Description
GetPoint() [▶ 86]	Returns an index-addressed point in the table
GetPoints() [▶ 87]	Returns the number of points available in the table
GetProperties() [▶ 87]	Returns the master limit values of the table
GetValid()	Checks the table for validity

Name	Description
LookDown() [▸ 88]	Translates a slave value to a matching master value
LookUp() [▸ 89]	Translates a master value to a matching slave value
SetPoint() [▸ 90]	Sets an index-addressed point in the table
SetPointNonEquidistant() [▸ 90]	Sets any point in the table
SetPoints() [▸ 91]	Defines the number of assignable points once
SetProperties() [▸ 92]	Sets the master limit values of the table

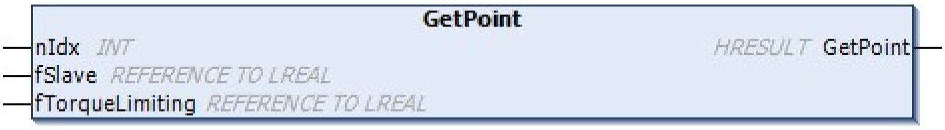
Interfaces

Type	Description
I_CammingLookUp	Standard interface on FB_CammingLookUp

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.1 GetPoint()



This method returns a point in the table addressed by `nIdx`

Syntax:

```
METHOD GetPoint : HRESULT
VAR_INPUT
    nIdx:                INT;
    fSlave:              REFERENCE TO LREAL;
    fTorqueLimiting:     REFERENCE TO LREAL;
END_VAR
```

Return value

Name	Type	Description
GetPoint	HRESULT	Return value with feedback on the success of the execution

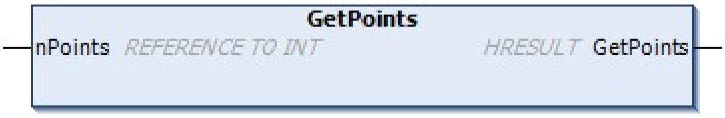
Inputs

Name	Type	Description
nIdx	INT	Index of the requested table point
fSlave	REFERENCE TO LREAL	Slave value of the table point
fTorqueLimiting	REFERENCE TO LREAL	TorqueLimiting value (secondary slave value) of the table point

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.2 GetPoints()



This method returns the number of points available in the table

Syntax:

```
METHOD GetPoint : HRESULT
VAR_INPUT
    nPoints:          REFERENCE TO INT;
END_VAR
```

Return value

Name	Type	Description
GetPoint	HRESULT	Return value with feedback on the success of the execution

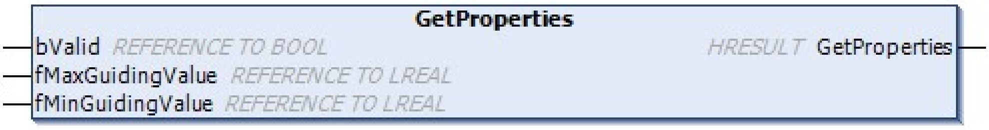
Inputs

Name	Type	Description
nPoints	REFERENCE TO INT	Number of available points

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.3 GetProperties()


GetProperties
bValid REFERENCE TO BOOL
fMaxGuidingValue REFERENCE TO LREAL
fMinGuidingValue REFERENCE TO LREAL
HRESULT GetProperties

This method returns the master limit values of the table. These are used to break down the master position in equidistant operation modes of the table.

Syntax:

```
METHOD SetProperties : HRESULT
VAR_INPUT
    bValid              : REFERENCE BOOL;
    fMaxGuidingValue    : REFERENCE LREAL;
    fMinGuidingValue    : REFERENCE LREAL;
END_VAR
```

Return value

Name	Type	Description
GetProperties	HRESULT	Return value with feedback on the success of the execution

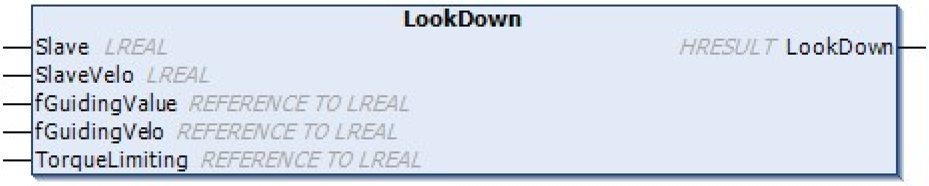
Inputs

Name	Type	Description
bValid	REFERENCE TO BOOL	The table is validated
fMaxGuidingValue	REFERENCE TO LREAL	Maximum value of the master
fMinGuidingValue	REFERENCE TO LREAL	Minimum value of the master

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.4 LookDown()


LookDown
Slave LREAL
SlaveVelo LREAL
fGuidingValue REFERENCE TO LREAL
fGuidingVelo REFERENCE TO LREAL
TorqueLimiting REFERENCE TO LREAL
HRESULT LookDown

This method translates a slave value to a matching master value. If the value lies between two defined points in the table, the selected operation mode is used for interpolation.

Syntax:

```
METHOD LookDown : HRESULT
VAR_INPUT
    Slave:              LREAL;
    SlaveVelo:          LREAL;
    fGuidingValue:      REFERENCE TO LREAL;
    fGuidingVelo:       REFERENCE TO LREAL;
    TorqueLimiting:     REFERENCE TO LREAL;
END_VAR
```

Return value

Name	Type	Description
LookDown	HRESULT	Return value with feedback on the success of the execution

Inputs

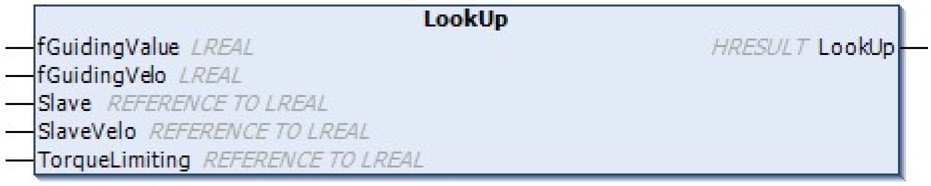
Name	Type	Description
Slave	LREAL	Primary slave value
SlaveVelo	LREAL	Tertiary slave value *Notice* **Only available in second-order equidistant operation mode**
fGuidingValue	REFERENCE TO LREAL	Resulting master value for the requested slave value

Name	Type	Description
fGuidingVelo	REFERENCE TO LREAL	Master value for the requested tertiary slave value. This is determined taking into account the first derivative of the primary slave value *Notice* **Only available in second-order equidistant operation mode**
TorqueLimiting	REFERENCE TO LREAL	Secondary slave value for the requested primary slave value

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.5 LookUp()



This method translates a master value to a matching slave value. If the value lies between two defined points in the table, the selected operation mode is used for interpolation.

Syntax:

```
METHOD LookUp : HRESULT
VAR_INPUT
    fGuidingValue:      LREAL;
    fGuidingVelo:       LREAL;
    Slave:              REFERENCE TO LREAL;
    SlaveVelo:          REFERENCE TO LREAL;
    TorqueLimiting:     REFERENCE TO LREAL;
END_VAR
```

Return value

Name	Type	Description
LookDown	HRESULT	Return value with feedback on the success of the execution.

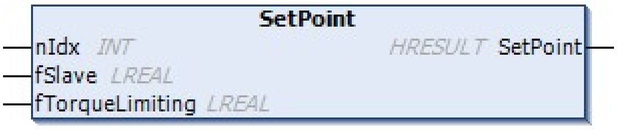
Inputs

Name	Type	Description
fGuidingValue	LREAL	Master value
fGuidingVelo	LREAL	Reserved for later use.
Slave	REFERENCE TO LREAL	Resulting slave value.
SlaveVelo	REFERENCE TO LREAL	Reserved for later use.
TorqueLimiting	REFERENCE TO LREAL	Secondary slave value for the requested primary master value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.6 SetPoint()



This method sets a point in the table addressed via `nIdx`.

Syntax:

```
METHOD SetPoint : HRESULT
VAR_INPUT
    nIdx                : INT;
    fSlave              : LREAL;
    fTorqueLimiting     : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetPoint	HRESULT	Return value with feedback on the success of the execution.

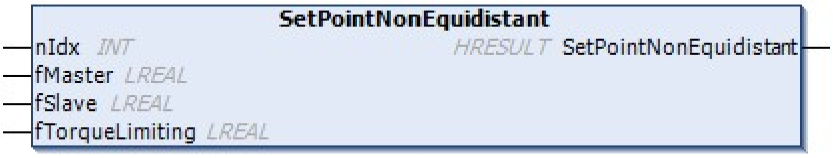
Inputs

Name	Type	Description
nIdx	INT	Index of the table point to be set.
fSlave	LREAL	Slave value of the table point.
fTorqueLimiting	LREAL	TorqueLimiting value (secondary slave value) of the table point.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.7 SetPointNonEquidistant()



This method sets a point in the table addressed via `nIdx` that has a specific master position. To do this, the table must be initialized in a non-equidistant operation mode.

Syntax:

```
METHOD SetPointNonEquidistant : HRESULT
VAR_INPUT
    nIdx                : INT;
    fMaster             : LREAL;
    fSlave              : LREAL;
    fTorqueLimiting     : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetPointNonEquidistant	HRESULT	Return value with feedback on the success of the execution.

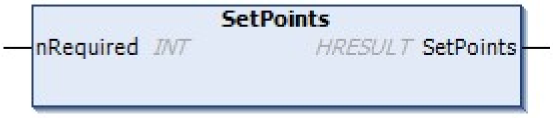
Inputs

Name	Type	Description
nIdx	INT	Index of the table point to be set.
fMaster	LREAL	Master value of the table point.
fSlave	LREAL	Slave value of the table point.
fTorqueLimiting	LREAL	TorqueLimiting value (secondary slave value) of the table point.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.8 SetPoints()



This method sets the number of available points in the table.

Method can only be executed once

If this method is executed for the second time or a number of points > 0 is passed in the initialization, this method fails. The number of points can only be defined once.

Syntax:

```
METHOD SetPoints : HRESULT
VAR_INPUT
    nRequired:          INT;
END_VAR
```

Return value

Name	Type	Description
GetPoint	HRESULT	Return value with feedback on the success of the execution.

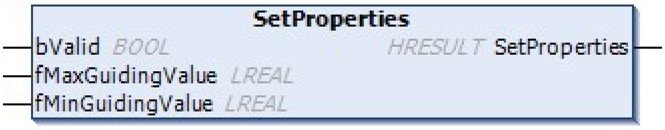
Inputs

Name	Type	Description
nRequired	INT	Number of points requested.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.4.9 SetProperties()



This method sets the master limit values of the table. These are used to break down the master position in equidistant operation modes of the table.

Syntax:

```
METHOD SetProperties : HRESULT
VAR_INPUT
    bValid              : BOOL;
    fMaxGuidingValue    : LREAL;
    fMinGuidingValue    : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetProperties	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
bValid	BOOL	Defines the valid initialization of the table.
fMaxGuidingValue	LREAL	Maximum value of the master.
fMinGuidingValue	LREAL	Minimum value of the master.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.10.5 E_LookupMode

Operation mode of the FB_CammingLookUp class.

Syntax:

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_LookupMode :
(
    eNoneEqui := 1,
    eNoneEquiLinIpol,

    eEquiLinIpol,
    eEquiSecondOrder
);
END_TYPE
```

Values

Name	Number	Equidistant	Interpolation
eNoneEqui	1	No	/

Name	Number	Equidistant	Interpolation
eNoneEquiLinIpol	2		Linear
eEquiLinIpol	3	Yes	Linear
eEquiSecondOrder	4		Second order (parabola)

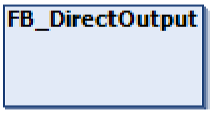
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.11 DirectOutput

Exclusive function for hydraulic axes

This core function is only available for hydraulic axis types.


FB_DirectOutput

This core function is used to directly control the output of a hydraulic axis. There will be no monitoring of position limits.

This core function is a member of the group of statically-controlled core functions.

Syntax:

```
FUNCTION_BLOCK FB_DirectOutput EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
OutputReference	LREAL	Get, Set	This property defines the value that must be specified as OutPutValue to cause a full-scale output to the controlled device.
OutputValue	LREAL	Get, Set	This property is used to define the output to the device. OutputReference as a scaling here.
Ramptime	LREAL	Get, Set	RampTime is used to define the time for ramping from zero to the full-scale value.

The output value is influenced by RampTime

Any change in the output whose amount is less than the full-scale value requires a proportional part of RampTime.

Methods

Name	Description
DoActivate() [▶ 94]	This method enables and disables the direct output.

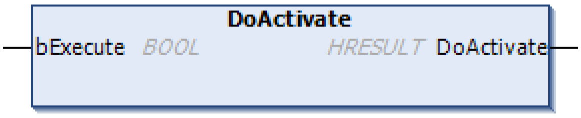
Interfaces

Type	Description
I_DirectOutput	Standard interface on FB_DirectOutput.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

4.11.1 DoActivate



This method enables and disables the direct output.

Syntax:

```
METHOD DoActivate : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoActivate	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bExecute	BOOL	A TRUE enables the output. A FALSE ramps the output to zero and disables the core function.

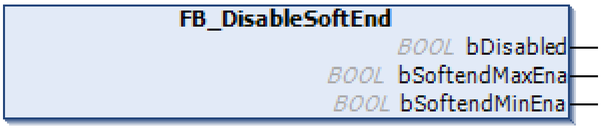
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

4.12 DisableSoftEnd

Core function is not supported by inverter axes

This core function is not supported by inverter axes. Any use will report DEVICE_NOTINIT and trigger an error message.



This core function is used to temporarily disable and re-enable the software position limits of the axis.

DisableSoftEnd is a member of the group of edge-triggered core functions.

Properties

Name	Type	Access	Description
Disabled	BOOL	Get	A TRUE signal reports active disabling.

Methods

Name	Description
DoDisable() [▶ 95]	This method disables the set software position limits.
ReEnable() [▶ 96]	This method re-enables the set software position limits.

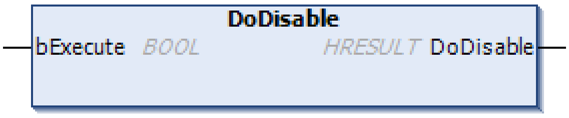
Interfaces

Type	Description
I_DisableSoftEnd	Standard interface on FB_DisableSoftEnd

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.12.1 DoDisable



This method disables the set software position limits.

Syntax:

```
METHOD DoDisable: HRESULT
VAR_INPUT
    bExecute: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoDisable	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

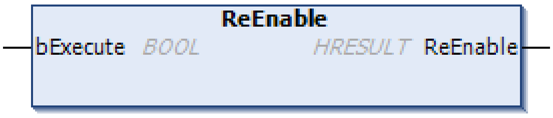
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers the disabling.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.12.2 ReEnable



This method re-enables the set software position limits.

Syntax:

```
METHOD ReEnable: HRESULT
VAR_INPUT
    bExecute    : BOOL;
END_VAR
```

Return value

Name	Type	Description
ReEnable	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.

E_AdsErr	Return value	Cause
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers the re-enabling.

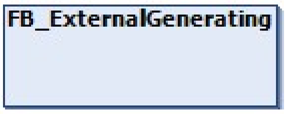
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.13 ExternalGenerating

Not available via the axis interfaces

This core function is not available via the axis interfaces. Direct access by the application is not required.


FB_ExternalGenerating

This core function is used for setpoint generation generated in TwinCAT 3 Plastic Technlogy Functions. It belongs to the group of statically controlled core functions.

Syntax:

```
FUNCTION_BLOCK FB_ExternalGenerating EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
Position	LREAL	Get, Set	Transfer of the set position.
TorqueLimiting	LREAL	Get, Set	Transfer of the applied torque limit.
Velocity	LREAL	Get, Set	Transfer of the target velocity.

Methods

Name	Description
DoEnable() [▸ 98]	Enables/disables external setpoint generation.

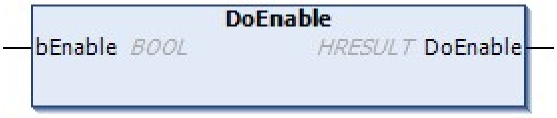
Interfaces

Type	Description
I_ExternalGenerating	Standard interface on FB_ExternalGenerating.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.13.1 DoEnable()



This method enables the external setpoint generation.

Syntax:

```
METHOD DoEnable : HRESULT
VAR_INPUT
    bEnable         : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoEnable	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bEnable	BOOL	A rising edge triggers the external setpoint generation. A falling edge stops the external setpoint generation with a final stop ramp.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.14 Estop

This core function is used to trigger an emergency shutdown operation. It will use the maximum dynamic parameters that are allowed for this axis by the underlying motion technology.

Properties

Name	Type	Access	Description
NoCreeping	BOOL	Get, Set	A TRUE in this property avoids the creep phase at the end of the stopping process of a hydraulic axis.

Methods

Name	Description
DoEstop() [▸ 99]	A rising edge triggers the stop.

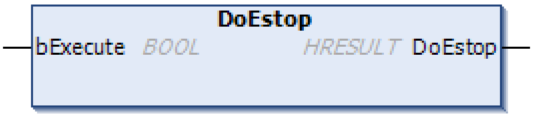
Interfaces

Type	Description
I_Estop	Standard interface on FB_Estop.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.14.1 DoEstop



This method is used to trigger the stop.

Syntax:

```
METHOD DoEstop : HRESULT
VAR_INPUT
    bExecute: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoEstop	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bExecute	BOOL	A rising edge at this input triggers the stop.

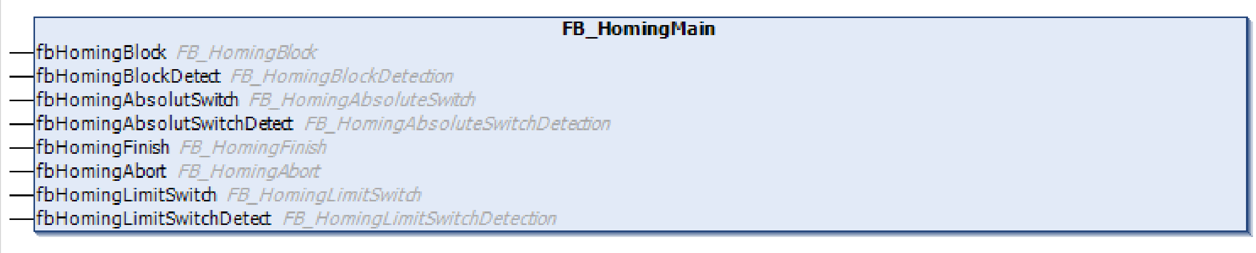
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15 Homing

Core function is not supported by inverter axes

This core function is not supported by inverter axes. Any use will report DEVICE_NOTINIT and trigger an error message.



```
FB_HomingMain
fbHomingBlock FB_HomingBlock
fbHomingBlockDetect FB_HomingBlockDetection
fbHomingAbsolutSwitch FB_HomingAbsoluteSwitch
fbHomingAbsolutSwitchDetect FB_HomingAbsoluteSwitchDetection
fbHomingFinish FB_HomingFinish
fbHomingAbort FB_HomingAbort
fbHomingLimitSwitch FB_HomingLimitSwitch
fbHomingLimitSwitchDetect FB_HomingLimitSwitchDetection
```

The property offers access to a range of homing functions.

All these functions belong to the group of edge-triggered core functions.

There are two groups of homing functions: Initiating or continuing functions (AbsoluteSwitch, AbsoluteSwitchDetect, Block, BlockDetect, LimitSwitch, LimitSwitchDetect) and terminating functions (Abort, Finish).

Triggering a function of the first group changes the behavior of the axis by activating the homing mode. If this change has already been made by another function in this group, this has no effect. In this mode, the underlying motion technology disables a number of mechanisms such as lag monitoring, velocity pre-control, software position limit switch, etc.

NOTICE

Unexpected behavior

Axes in homing mode may respond to motion commands in an unexpected manner.

As the last step of a homing procedure, the functions of the second group are used to terminate the homing mode and set the axis to a normal behavior.

Properties

Name	Type	Access	Description
Abort [▸ 101]	I_HomingAbort	Get	An active homing is aborted without success.
AbsoluteSwitch [▸ 103]	I_HomingAbsoluteSwitch		Homing is performed at a position reported by a binary sensor.
AbsoluteSwitchDetect [▸ 106]	I_HomingAbsoluteSwitchDetection		The actual position is recorded at a position reported by a binary sensor.
Block [▸ 110]	I_HomingBlock		Homing is performed on a mechanical stop.
BlockDetect [▸ 113]	I_HomingBlockDetection		The actual position is recorded at a mechanical stop.

Name	Type	Access	Description
Finish [▶ 117]	I_HomingFinish		An active homing is completed successfully.
LimitSwitch [▶ 119]	I_HomingLimitSwitch		Homing is performed at a position reported by a hardware limit switch.
LimitSwitchDetect [▶ 123]	I_HomingLimitSwitchDetection		The actual position is recorded at a position reported by a hardware limit switch.

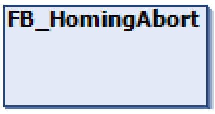
Interfaces

Type	Description
I_HomingMain	Standard interface on FB_HomingMain.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.1 Abort



This core function can be used to abort a homing in case of a problem.

Abort required

Abort is also required if a homing function fails.

Methods

Name	Description
DoAbort() [▶ 102]	A rising edge triggers the abort.
SetParameter() [▶ 102]	The procedure for returning to normal operation can be specified.

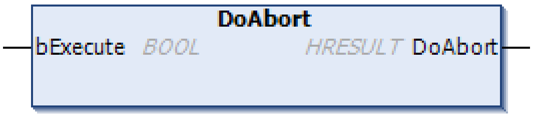
Interfaces

Type	Description
I_HomingAbort	Standard interface on FB_HomingAbort.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.1.1 DoAbort()



This method triggers the abort in case of a rising edge.

Syntax:

```
METHOD DoAbort : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoAbort	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALID DATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

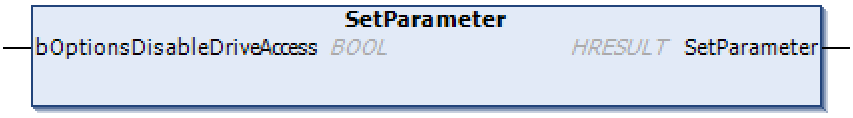
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers the abort.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.1.2 SetParameter()



The procedure for returning to normal operation can be specified.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    bOptionsDisableDriveAccess : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
bOptionsDisableDriveAccess	BOOL	A TRUE prevents ADS communication via the fieldbus from being used to change parameters of the controlled device in order to return to normal operation.

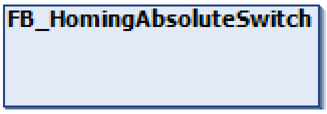
Non-Beckhoff servo drives

This option must be used for most non-Beckhoff servo drives.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.2 AbsoluteSwitch



FB_HomingAbsoluteSwitch

This core function performs homing by searching for a fixed edge of a signal. A specified position is set at this location.

Properties

Name	Type	Access	Description
AbsoluteSwitch	BOOL	Get, Set	This signal indicates the homing position.
NegativeLimitSwitch	BOOL	Get, Set	The hardware limit switch at the lower end of the available travel path.
PositiveLimitSwitch	BOOL	Get, Set	The hardware limit switch at the upper end of the available travel path.

The action of the limit switches is defined by PLCopen standards.

Methods

Name	Description
DoHoming() [▸ 104]	This method triggers the homing.
SetParameter() [▸ 105]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▸ 105]	A set of travel parameters for the homing procedure is set.

Interfaces

Type	Description
I_HomingAbsoluteSwitch	Standard interface on FB_HomingAbsoluteSwitch.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.2.1 DoHoming()

DoHoming
bExecute BOOL
HRESULT DoHoming

This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

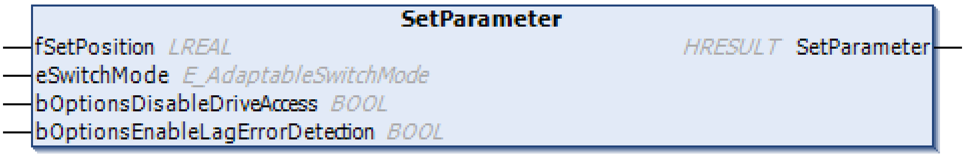
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.2.2 SetParameter()


SetParameter
fSetPosition LREAL
eSwitchMode E_AdaptableSwitchMode
bOptionsDisableDriveAccess BOOL
bOptionsEnableLagErrorDetection BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fSetPosition                       : LREAL;
    eSwitchMode                        : E_AdaptableSwitchMode;
    bOptionsDisableDriveAccess         : BOOL;
    bOptionsEnableLagErrorDetection : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

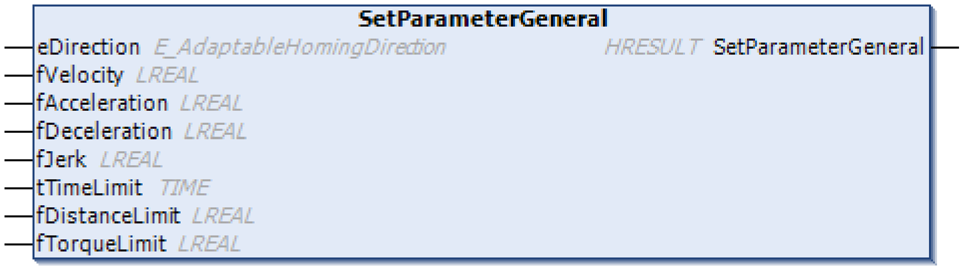
Inputs

Name	Description
fSetPosition	Specifies the value that is applied to the actual position at the homing event.
eSwitchMode	Defines how the AbsoluteSwitch is interpreted as a homing event.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. ***Notice*** **This option must be used for most non-Beckhoff servo drives.**
bOptionsEnableLagErrorDetection	A TRUE will enable the lag error detection while the function is being executed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.2.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

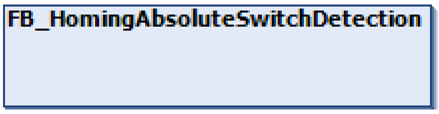
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.3 AbsoluteSwitchDetect



FB_HomingAbsoluteSwitchDetection

This core function performs homing by searching for a fixed edge of a signal. At this location the actual position is latched and reported.

Properties

Name	Type	Access	Description
AbsoluteSwitch	BOOL	Get, Set	This signal indicates the homing position.
NegativeLimitSwitch	BOOL	Get, Set	The hardware limit switch at the lower end of the available travel path.
PositiveLimitSwitch	BOOL	Get, Set	The hardware limit switch at the upper end of the available travel path.
RecordedPosition	LREAL	Get	The position latched at the location of the signal.

The action of the limit switches is defined by PLCopen standards.

Methods

Name	Description
DoHoming() [▶ 107]	This method triggers the homing.
SetParameter() [▶ 108]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 109]	A set of travel parameters for the homing procedure is set.

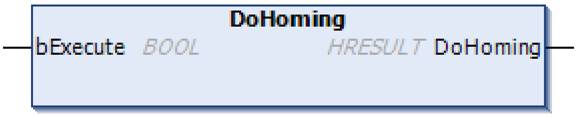
Interfaces

Type	Description
I_HomingAbsoluteSwitchDetect	Standard interface on FB_HomingAbsoluteSwitchDetect.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.3.1 DoHoming()



This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

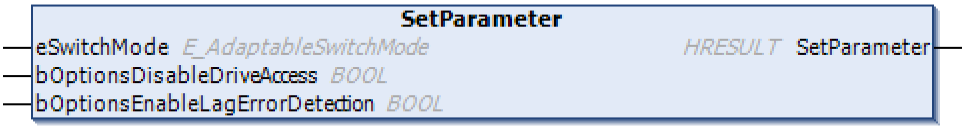
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.3.2 SetParameter()


SetParameter
eSwitchMode E_AdaptableSwitchMode
bOptionsDisableDriveAccess BOOL
bOptionsEnableLagErrorDetection BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    eSwitchMode                     : E_AdaptableSwitchMode;
    bOptionsDisableDriveAccess      : BOOL;
    bOptionsEnableLagErrorDetection : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

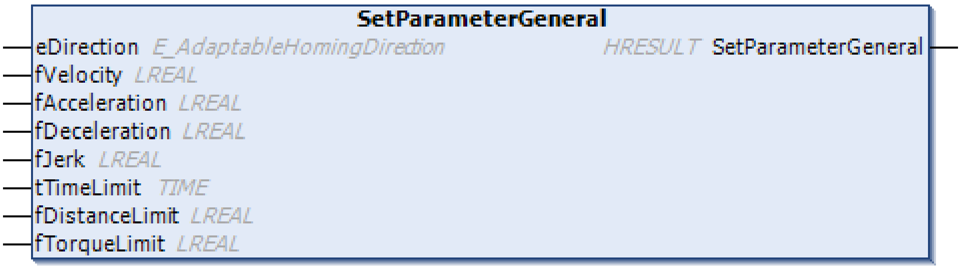
Inputs

Name	Description
eSwitchMode	Defines how the AbsoluteSwitch is interpreted as a homing event.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. *Notice* **This option must be used for most non-Beckhoff servo drives.**
bOptionsEnableLagErrorDetection	A TRUE will enable the lag error detection while the function is being executed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.3.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

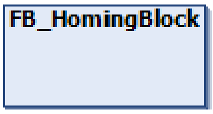
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.4 Block



This core function performs homing by searching for a mechanical stop. At this location the actual position is latched and reported.

Methods

Name	Description
DoHoming() [▶ 110]	This method triggers the homing.
SetParameter() [▶ 111]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 112]	A set of travel parameters for the homing procedure is set.

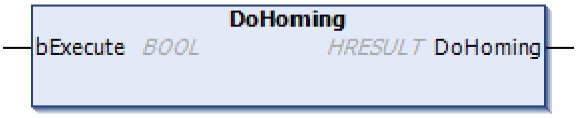
Interfaces

Type	Description
I_HomingBlock	Standard interface on FB_HomingBlock.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.4.1 DoHoming()



This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

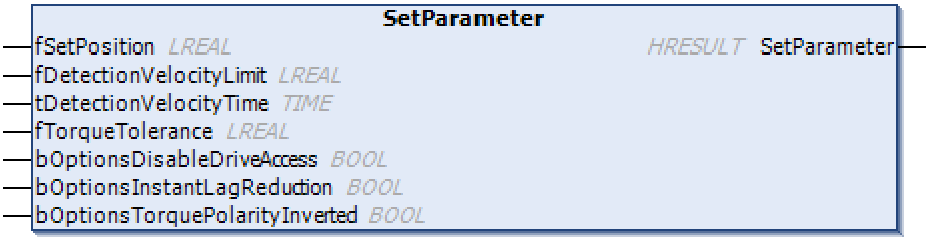
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.4.2 SetParameter()


SetParameter
fSetPosition LREAL
fDetectionVelocityLimit LREAL
tDetectionVelocityTime TIME
fTorqueTolerance LREAL
bOptionsDisableDriveAccess BOOL
bOptionsInstantLagReduction BOOL
bOptionsTorquePolarityInverted BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fSetPosition                   : LREAL;
    fDetectionVelocityLimit        : LREAL;
    tDetectionVelocityTime         : TIME;
    fTorqueTolerance               : LREAL;
    bOptionsDisableDriveAccess     : BOOL;
    bOptionsInstantLagReduction    : BOOL;
    bOptionsTorquePolarityInverted : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Description
fSetPosition	The position to be assigned to the homing event.
fDetectionVelocityLimit	A velocity threshold for the detection of the homing event. For details see below.
tDetectionVelocityTime	A filter time for the detection of the homing event. For details see below.
fTorqueTolerance	A torque threshold for the detection of the homing event. For details see below.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. *Notice* **This option must be used for most non-Beckhoff servo drives.**
bOptionsInstantLagReduction	A TRUE causes the lag error to be cleared (set position := actual position) when the homing event is detected.
bOptionsTorquePolarityInverted	A TRUE causes an inverted evaluation of the torque. *Notice* **This inversion must be used if the signs of the torque and the direction of movement do not match. This may be caused by direction-reversing mechanics (gears, etc.).**

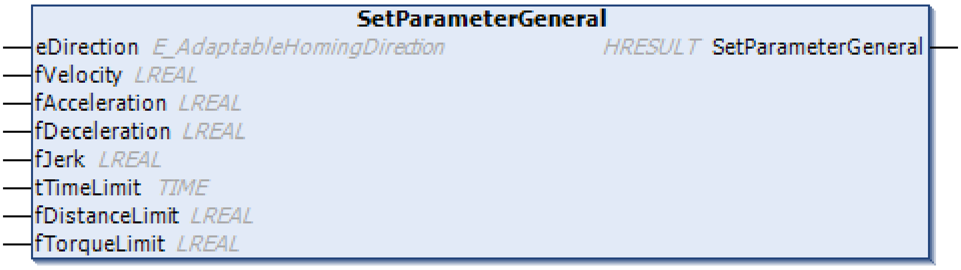
The homing event

A mechanical block as a homing event is detected if at the same time the torque is below the torque limit by less than `fTorqueTolerance` and the actual velocity has been continuously below `fDetectionVelocityLimit` since `tDetectionVelocityTime`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.4.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

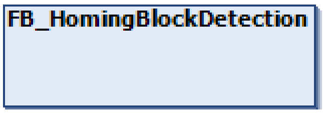
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.5 BlockDetect



FB_HomingBlockDetection

This core function performs homing by searching for a mechanical stop. At this location the actual position is latched and reported.

Properties

Name	Type	Access	Description
RecordedPosition	LREAL	Get	The position latched at the location of the signal.

Methods

Name	Description
DoHoming() [▶ 114]	This method triggers the homing.
SetParameter() [▶ 115]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 116]	A set of travel parameters for the homing procedure is set.

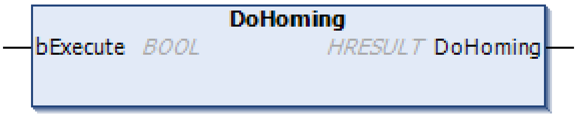
Interfaces

Type	Description
I_HomingBlockDetect	Standard interface on FB_HomingBlockDetect.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.5.1 DoHoming()



This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

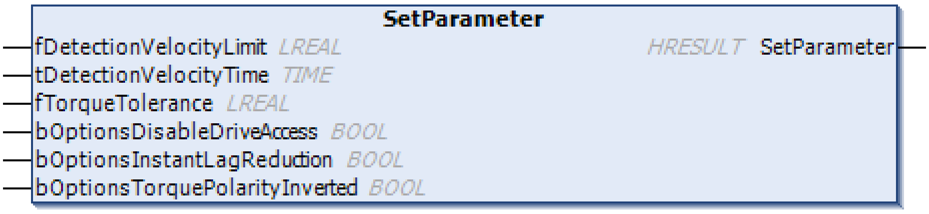
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.5.2 SetParameter()


SetParameter
fDetectionVelocityLimit LREAL
tDetectionVelocityTime TIME
fTorqueTolerance LREAL
bOptionsDisableDriveAccess BOOL
bOptionsInstantLagReduction BOOL
bOptionsTorquePolarityInverted BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fDetectionVelocityLimit        : LREAL;
    tDetectionVelocityTime         : TIME;
    fTorqueTolerance               : LREAL;
    bOptionsDisableDriveAccess     : BOOL;
    bOptionsInstantLagReduction    : BOOL;
    bOptionsTorquePolarityInverted : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Description
fDetectionVelocityLimit	A velocity threshold for the detection of the homing event. For details see below.
tDetectionVelocityTime	A filter time for the detection of the homing event. For details see below.
fTorqueTolerance	A torque threshold for the detection of the homing event. For details see below.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. *Notice* **This option must be used for most non-Beckhoff servo drives.**
bOptionsInstantLagReduction	A TRUE causes the lag error to be cleared (set position := actual position) when the homing event is detected.
bOptionsTorquePolarityInverted	A TRUE causes an inverted evaluation of the torque. *Notice* **This inversion must be used if the signs of the torque and the direction of movement do not match. This may be caused by direction-reversing mechanics (gears, etc.).**

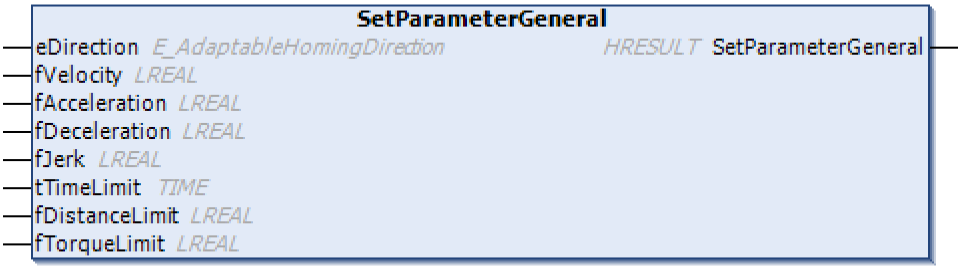
The homing event

A mechanical block as a homing event is detected if at the same time the torque is below the torque limit by less than `fTorqueTolerance` and the actual velocity has been continuously below `fDetectionVelocityLimit` since `tDetectionVelocityTime`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.5.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.

Name	Type	Description
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

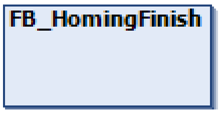
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.6 Finish



This core function must be used to successfully complete a homing.

Abort required

Abort is required if a homing function fails.

Methods

Name	Description
DoFinish() [▶ 117]	A rising edge triggers the termination.
SetParameter() [▶ 118]	The procedure for returning to normal operation can be specified.

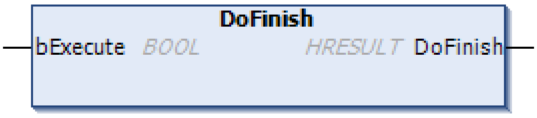
Interfaces

Type	Description
I_HomingFinish	Standard interface on FB_HomingFinish.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.6.1 DoFinish()



This method triggers the termination on a rising edge.

Syntax:

```
METHOD DoFinish : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoFinish	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

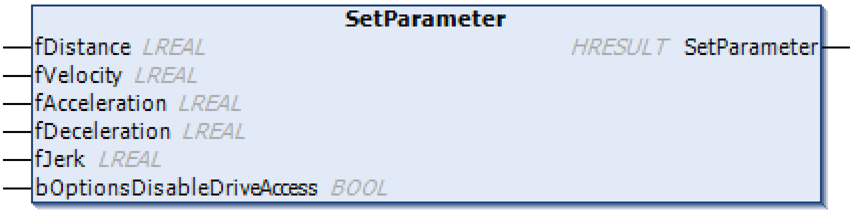
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers the termination.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.6.2 SetParameter()



The procedure for returning to normal operation can be specified.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fDistance : LREAL;
    fVelocity : LREAL;
    fAcceleration : LREAL;
    fDeceleration : LREAL;
    fJerk : LREAL;
    bOptionsDisableDriveAccess : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

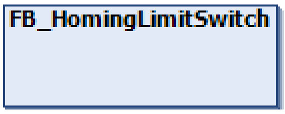
Inputs

Name	Type	Description
fDistance	LREAL	Here you can define a distance by which the axis should move away from the homing position. **In many cases, there is a torque at the end of a homing procedure. The mechanics of the axis can be relieved by a movement suitably selected in amount and direction.**
fVelocity	LREAL	The velocity to be commanded for this.
fAcceleration	LREAL	The acceleration to be commanded for this.
fDeceleration	LREAL	The deceleration to be commanded for this.
fJerk	LREAL	The jerk to be commanded for this.
bOptionsDisableDriveAccess	BOOL	A TRUE prevents ADS communication via the fieldbus from being used to change parameters of the controlled device in order to return to normal operation. ***Notice*** **The option** `bOptionsDisableDriveAccess` **must be used for most non-Beckhoff servo drives.**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.7 LimitSwitch



This core function performs homing by searching for a fixed edge of a signal. A specified position is set at this location.

Properties

Name	Type	Access	Description
NegativeLimitSwitch	BOOL	Get, Set	The hardware limit switch at the lower end of the available travel path.
PositiveLimitSwitch	BOOL	Get, Set	The hardware limit switch at the upper end of the available travel path.

The action of the limit switches is defined by PLCopen standards.

Methods

Name	Description
DoHoming() [▸ 120]	This method triggers the homing.
SetParameter() [▸ 121]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▸ 122]	A set of travel parameters for the homing procedure is set.

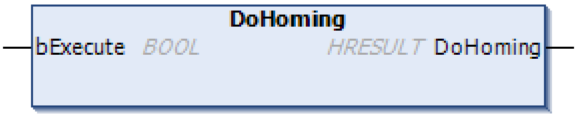
Interfaces

Type	Description
I_HomingLimitSwitch	Standard interface on FB_HomingLimitSwitch.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.7.1 DoHoming()



This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

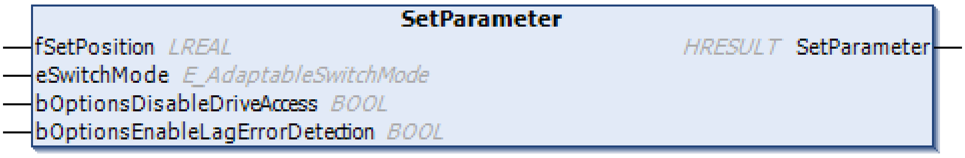
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.7.2 SetParameter()


SetParameter
fSetPosition LREAL
eSwitchMode E_AdaptableSwitchMode
bOptionsDisableDriveAccess BOOL
bOptionsEnableLagErrorDetection BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fSetPosition                    : LREAL;
    eSwitchMode                     : E_AdaptableSwitchMode;
    bOptionsDisableDriveAccess      : BOOL;
    bOptionsEnableLagErrorDetection : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

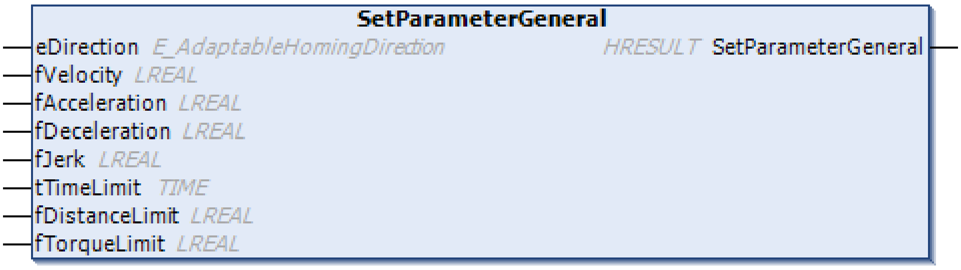
Inputs

Name	Description
fSetPosition	Specifies the value that is applied to the actual position at the homing event.
eSwitchMode	Defines how the AbsoluteSwitch is interpreted as a homing event.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. **This option must be used for most non-Beckhoff servo drives.**
bOptionsEnableLagErrorDetection	A TRUE will enable the lag error detection while the function is being executed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.7.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

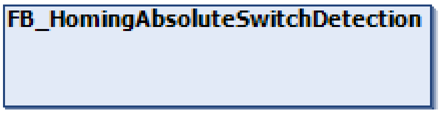
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.8 LimitSwitchDetect



This core function performs homing by searching for a fixed edge of a signal. At this location the actual position is latched and reported.

Properties

Name	Type	Access	Description
NegativeLimitSwitch	BOOL	Get, Set	The hardware limit switch at the lower end of the available travel path.
PositiveLimitSwitch	BOOL	Get, Set	The hardware limit switch at the upper end of the available travel path.
RecordedPosition	LREAL	Get	The position latched at the location of the signal.

The action of the limit switches is defined by PLCopen standards.

Methods

Name	Description
DoHoming() [▸ 123]	This method triggers the homing.
SetParameter() [▸ 124]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▸ 125]	A set of travel parameters for the homing procedure is set.

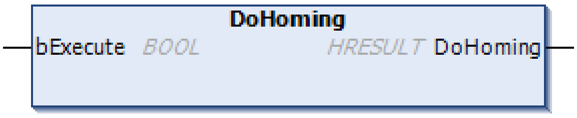
Interfaces

Type	Description
I_HomingLimitSwitchDetect	Standard interface on FB_HomingLimitSwitchDetect.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.8.1 DoHoming()



This method triggers the homing.

Syntax:

```
METHOD DoHoming : HRESULT
VAR_INPUT
    bExecute : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoHoming	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALID DATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDS TATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

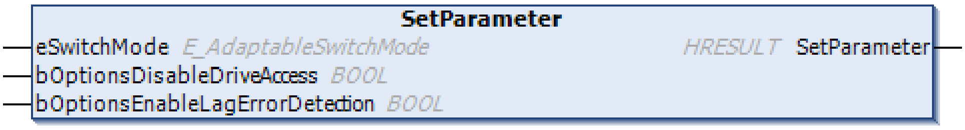
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers homing.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.8.2 SetParameter()


SetParameter
eSwitchMode E_AdaptableSwitchMode
bOptionsDisableDriveAccess BOOL
bOptionsEnableLagErrorDetection BOOL
HRESULT SetParameter

A set of parameters specific to this homing procedure is set.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    eSwitchMode                      : E_AdaptableSwitchMode;
    bOptionsDisableDriveAccess       : BOOL;
    bOptionsEnableLagErrorDetection  : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

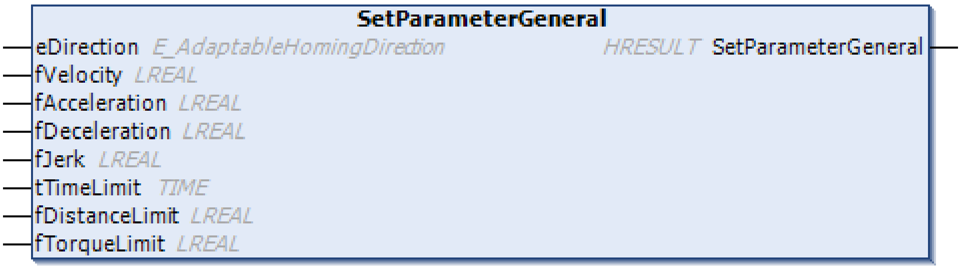
Inputs

Name	Description
eSwitchMode	Defines how the AbsoluteSwitch is interpreted as a homing event.
bOptionsDisableDriveAccess	A TRUE prevents the underlying motion technology from using ADS communication via the fieldbus to change parameters of the controlled device and return to normal operation. **This option must be used for most non-Beckhoff servo drives.**
bOptionsEnableLagErrorDetection	A TRUE will enable the lag error detection while the function is being executed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.15.8.3 SetParameterGeneral()



A general set of travel parameters for the homing procedure is set.

Syntax:

```
METHOD SetParameterGeneral : HRESULT
VAR_INPUT
    eDirection     : E_AdaptableHomingDirection;
    fVelocity      : LREAL;
    fAcceleration  : LREAL;
    fDeceleration  : LREAL;
    fJerk          : LREAL;
    tTimeLimit     : TIME;
    fDistanceLimit : LREAL;
    fTorqueLimit   : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameterGeneral	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
eDirection	E_AdaptableHomingDirection	The movement with which the homing event is found.
fVelocity	LREAL	The velocity of the movement used in the search of the homing event.

Name	Type	Description
fAcceleration	LREAL	The acceleration of the movement used in the search of the homing event.
fDeceleration	LREAL	The deceleration of the movement used in the search of the homing event.
fJerk	LREAL	The jerk of the movement used in the search of the homing event.
tTimeLimit	TIME	The timeout limit of the core function.
fDistanceLimit	LREAL	The maximum distance that may be traveled in the search of the homing event.
fTorqueLimit	LREAL	The limit of the torque applied in the search of the homing event.

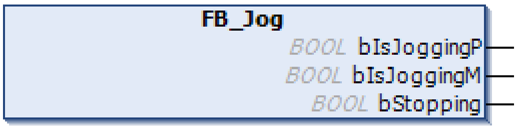
Torque limitation in the event of a mechanical blockage

The torque limitation is used to prevent damage if the axis encounters a mechanical blockage without having detected the homing event.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.16 Jog



This core function is used to start and stop the axis with a certain velocity but without a defined target position.

Properties

Name	Type	Access	Description
IsJoggingM	BOOL	Get	A TRUE signals an active jogging in negative direction.
IsJoggingP	BOOL	Get	A TRUE signals an active jogging in positive direction.
IsStopping	BOOL	Get	A TRUE signals the stopping of the axis after an active jogging.
TorqueLimiting	LREAL	Get, Set	This sets the torque limit during jogging.

Methods

Name	Description
DoJogM() [▸ 127]	This method performs the start and stop of the axis in the negative direction.
DoJogP() [▸ 128]	This method performs the start and stop of the axis in the positive direction.
SetParameter() [▸ 128]	This method is used to set the parameters of a movement in jogging mode.

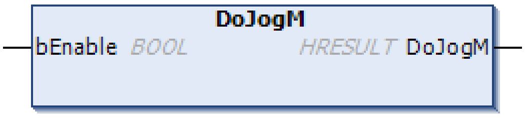
Interfaces

Type	Description
I_Jog	Standard interface on FB_Jog.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.16.1 DoJogM()



This method performs the start and stop of the axis in the negative direction.

Syntax:

```
METHOD DoJogM : HRESULT
VAR_INPUT
    bEnable : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoJogM	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

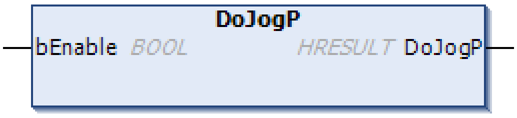
Inputs

Name	Type	Description
bEnable	BOOL	A rising edge starts a jogging in negative direction. A falling edge triggers a stop of the axis.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.16.2 DoJogP()



This method performs the start and stop of the axis in the positive direction.

Syntax:

```
METHOD DoJogP : HRESULT
VAR_INPUT
    bEnable : BOOL;
END_VAR
```

Return value

Name	Type	Description
DoJogP	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALID DATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

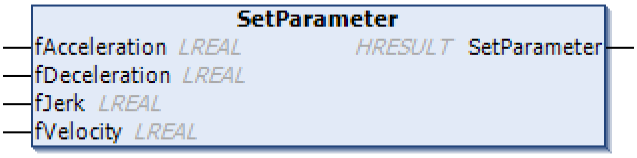
Inputs

Name	Type	Description
bEnable	BOOL	A rising edge starts a jogging in positive direction. A falling edge triggers a stop of the axis.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.16.3 SetParameter()



This method is used to set the parameters of a movement in jogging mode.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
    fAcceleration : LREAL;
    fDeceleration : LREAL;
    fJerk          : LREAL;
    fVelocity      : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

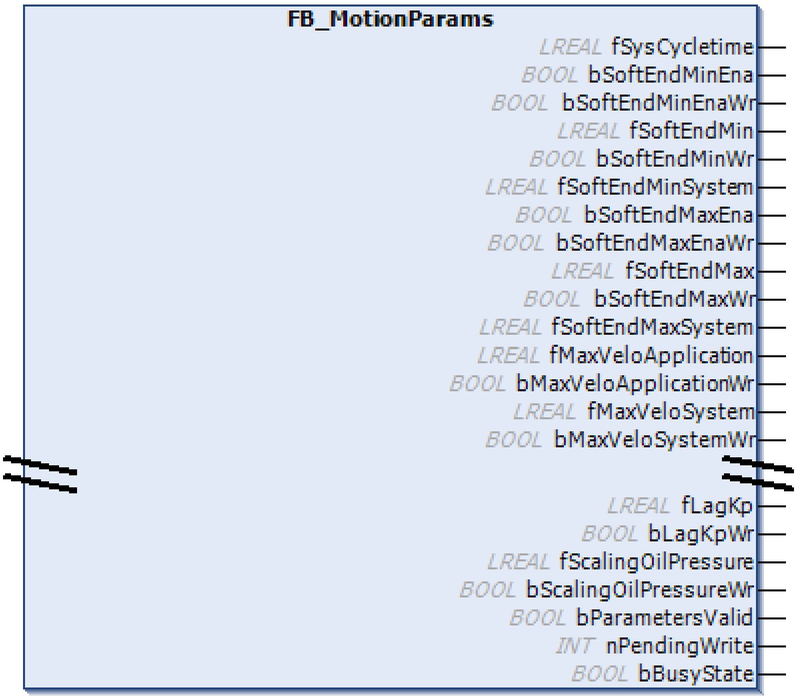
Inputs

Name	Type	Description
fAcceleration	LREAL	The commanded acceleration.
fDeceleration	LREAL	The commanded deceleration.
fJerk	LREAL	The commanded jerk.
fVelocity	LREAL	The commanded velocity.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.17 MotionParams



This core function is used to display a range of parameter values for a motion unit.

Support of the parameters depends on the technology

Some parameters are only available for specific drive technologies. Please note the comments in the "Choice" column!

Function is asynchronous

An implementation of runtime switching via this function block must always take the state of the function block into account. A written parameter does not become active immediately.

- The BusyState can be checked by setting it to FALSE

Axes must be disabled

Some parameters cannot be written while an axis is enabled.

Syntax:

```
FUNCTION_BLOCK FB_MotionParams EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Selection	Description
ActuatorBase [▶ 133]	I_Actuator ParamsH	Get	Actuators only	Access to parameters for the base position of an actuator.
ActuatorWork [▶ 133]	I_Actuator ParamsH	Get	Actuators only	Access to parameters for the working position of an actuator.
AsymTargeting	BOOL	Get, Set	Hydraulics only	TRUE if the parameters for the target approach are direction-dependent. Otherwise, the parameters for the negative direction are ignored and the parameters for the positive direction are used regardless of the direction.
AutoBrakeCalculation	BOOL	Get, Set	Hydraulics only	TRUE if the braking distance is determined automatically.
AxisIsNc	BOOL	Get	/	TRUE, if the axis is operated with TwinCAT NC.
BrakeDistanceM	LREAL	Get, Set	Hydraulics only	The braking distance time in negative direction. See also AsymTargeting.
BrakeDistanceP	LREAL	Get, Set	Hydraulics only	The braking distance time in positive direction.
CreepDistanceM	LREAL	Get, Set	Hydraulics only	The creep distance in negative direction. See also AsymTargeting.
CreepDistanceP	LREAL	Get, Set	Hydraulics only	The creep distance in positive direction.
CreepSpeedM	LREAL	Get, Set	Hydraulics only	The creep speed in negative direction.
CreepSpeedP	LREAL	Get, Set	Hydraulics only	The creep speed in positive direction.
CycleTime	LREAL	Get	/	The cycle time of the task in which the real-time functions of the axis are executed.
DriveIsCoE	BOOL	Get	/	TRUE, if the axis has a CoE interface.
DriveIsServo	BOOL	Get	/	TRUE if the axis is a servo drive.
DriveIsSimulated	BOOL	Get	/	TRUE if the drive of the axis is simulated.
DriveIsSoE	BOOL	Get	/	TRUE, if the axis has a SoE interface.
DriveReversed	BOOL	Get, Set	Not inverter	TRUE if the drive of the axis is inverted. *Notice* **The outputs are swapped for actuators.**
DriveType	UDINT	Get	/	A numeric identifier for the type of the connected drive.

Name	Type	Access	Selection	Description
				Notice **The meaning of the constants is defined in the Tc2 NC or the Tc2_Hydraulics library.**
EncoderInterpolation	LREAL	Get, Set	NC and hydraulics only	The divider for the encoder evaluation. It specifies the number of increments corresponding to the distance specified by EncoderWeighting.
EncoderIsAnalog	BOOL	Get	/	TRUE if the encoder of the axis uses an analog input.
EncoderIsSimulated	BOOL	Get	/	TRUE if the encoder of the axis is simulated.
EncoderReversed	BOOL	Get, Set	Not inverter	TRUE if the encoder of the axis is inverted. *Notice* **The feedback inputs are swapped for actuators.**
EncoderType	UDINT	Get	/	A numeric identifier for the type of the connected encoder. *Notice* **The meaning of the constants is defined in the Tc2 NC or the Tc2_Hydraulics library.**
EncoderWeighting	LREAL	Get, Set	NC and hydraulics only	The factor for encoder evaluation. It specifies the distance corresponding to a number of increments specified by EncoderInterpolation.
EncoderZeroShift	LREAL	Get, Set	NC and hydraulics only	The zero offset shift of the encoder.
HasTorqueLimiting	BOOL	Get	/	TRUE if the axis has a torque limit.
LagControlled	BOOL	Get	/	TRUE if the axis has a position controller.
LagCtrlKp	LREAL	Get, Set	NC and hydraulics only	The kP factor of the position controller.
LagFilter	LREAL	Get, Set	NC and hydraulics only	The filter time of the lag error monitoring.
LagLimit	LREAL	Get, Set	NC and hydraulics only	The threshold value of the lag error monitoring.
LagMonitored	BOOL	Get, Set	NC and hydraulics only	TRUE, if the lag error monitoring of the axis is active.
MaxAccApplication	LREAL	Get, Set	Not actuators	An additional restriction on acceleration.
MaxAcceleration	LREAL	Get, Set	Not actuators	The maximum permissible acceleration. *Notice* **A write operation on this parameter is always also applied to MaxAccApplication.**
MaxDecApplication	LREAL	Get, Set	Not actuators	An additional limitation of the deceleration.
MaxDeceleration	LREAL	Get, Set	Not actuators	The maximum permissible deceleration. *Notice* **A write operation on this parameter is always also applied to MaxDecApplication.**
MaxJerk	LREAL	Get, Set	NC and hydraulics only	The maximum permissible jerk.
MaxVeloApplication	LREAL	Get, Set	Not actuators	An additional velocity restriction.

Name	Type	Access	Selection	Description
MaxVeloSystem	LREAL	Get, Set	Not actuators	The maximum velocity of the axis. *Notice* **This parameter acts as a reference value for the maximum velocity output of the drive interface.**
MinVeloApplication	LREAL	Get, Set	Not NC and actuators	The minimum velocity of the axis.
Persist	BOOL	Get, Set	NC and hydraulics only	Enables the persistent saving of parameter write operations of this function block on the target system.
ScalingOilPressure	LREAL	Get, Set	Hydraulics only	The scaling factor for the actual pressure acquisition.
SoftEndMax	LREAL	Get, Set	NC and hydraulics only	The maximum software limit switch.
SoftEndMaxEna	BOOL	Get, Set	NC and hydraulics only	Enable for the maximum software limit switch.
SoftEndMaxSystem	LREAL	Get, Set	NC and hydraulics only	For the load side of transforming axes: The converted maximum software limit switch of the drive side. For the drive side of transforming axes: The converted maximum software limit switch of the load side. For non-transforming axes: A copy of the maximum software limit switch.
SoftEndMin	LREAL	Get, Set	NC and hydraulics only	The minimum software limit switch.
SoftEndMinEna	BOOL	Get, Set	NC and hydraulics only	Enable for the minimum software limit switch.
SoftEndMinSystem	LREAL	Get, Set	NC and hydraulics only	For the load side of transforming axes: The converted minimum software limit switch of the drive side. For the drive side of transforming axes: The converted minimum software limit switch of the load side. For non-transforming axes: A copy of the minimum software limit switch.
Valid	BOOL	Get	/	TRUE if all parameters are valid.
MinVeloJog	*LREAL*	*Get, Set*	*Obsolete*	Returns the value of MinVeloApplication.

Interfaces

Type	Description
I_MotionParams	Standard interface on FB_MotionParams.

State interface

This core function implements a signal pattern that is different from the standards.

Name	Type	Access	Description
BusyState	BOOL	Get	TRUE, if the function block writes changed parameters.

Name	Type	Access	Description
ReadyState	BOOL	Get	TRUE if the function block has loaded all parameters at least once.

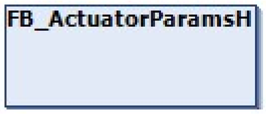
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4.17.1 ActuatorParamsH

Exclusive function for actuators

This core function is only available for actuators.



FB_ActuatorParamsH

This core function combines several parameters per end position of an actuator.

The core function is a member of the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_ParamActuatorH EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
FeedbackDelay	LREAL	Get, Set	Delay [s] of the feedback signal.
FeedbackDelayOff	LREAL	Get, Set	Delay [s] of the falling edge of the feedback signal.
HoldOutput	BOOL	Get, Set	When the motion commands are canceled, the output should remain active.
InvertFeedback	BOOL	Get, Set	The interpretation of the feedback signal is inverted (InPos = `FALSE`).
LatchFeedback	BOOL	Get, Set	The feedback signal is saved after the end position has been reached.
ReturnTime	LREAL	Get, Set	Time [s] after which the actuator automatically commands itself back in the opposite direction.
ReturnTimeEna	BOOL	Get, Set	Enables the time-based reset command.
Timeout	LREAL	Get, Set	Time [s] after the end position must be reached. Otherwise, the actuator goes into the error state.
UseImpulse	BOOL	Get, Set	The output is canceled when the end position is reached.

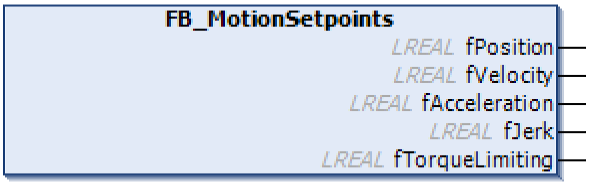
Interfaces

Type	Description
I_ActuatorParamsH	Standard interface on FB_ActuatorParamsH.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4.18 MotionSetpoints



This core function offers a range of current setpoints.

This core function is a member of the group of permanently active core functions.

Syntax:

```
FUNCTION_BLOCK FB_MotionSetpoints EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
Acceleration	LREAL	Get	The current acceleration setpoint.
Jerk	LREAL	Get	The current jerk setpoint.
Position	LREAL	Get	The current position setpoint.
TorqueLimiting	LREAL	Get	The current setpoint for torque limitation.
Velocity	LREAL	Get	The current velocity setpoint.

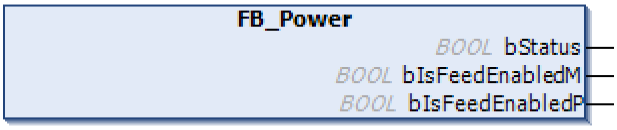
Interfaces

Type	Description
I_MotionSetpoints	Standard interface on FB_ MotionSetpoints.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.19 Power



This core function is used to enable the operation of the controlled device.

Syntax:

```
FUNCTION_BLOCK FB_Power EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
IsFeedEnabledM	BOOL	Get	TRUE if the axis is enabled for active movement in negative direction.
IsFeedEnabledP	BOOL	Get	TRUE if the axis is enabled for active movement in positive direction.
Override	LREAL	Get, Set	A factor for scaling commanded velocities. *Notice* **The effect is significantly influenced by the type of axis and its parameterization.**
Status	BOOL	Get	TRUE, if the axis is enabled for active operation. *Notice* **For an active movement, the direction-related enable is also required.**

Methods

Name	Description
DoPower() [▶ 135]	Enable for active operation of the axis.
FeedEnable() [▶ 136]	Directional enables for commanding active axis movements.

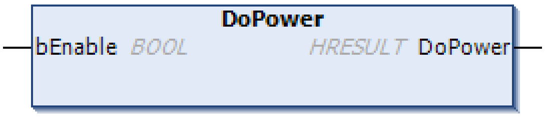
Interfaces

Type	Description
I_Power	Standard interface on FB_Power.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.19.1 DoPower()



This method is used to enable or disable the axis for active operation. If this requires an exchange of signals with a device, this exchange is performed and monitored.

Syntax:

```
METHOD DoPower: HRESULT
VAR_INPUT
    bEnable: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoPower	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

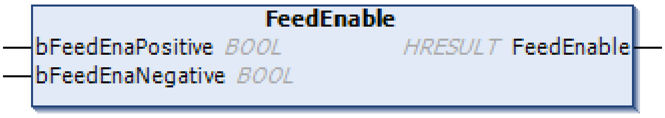
Inputs

Name	Type	Description
bEnable	BOOL	A rising edge starts the enable process. A falling edge starts the disable process.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.19.2 FeedEnable()



This method is used to define directional enables for active movements of the axis.

Syntax:

```
METHOD FeedEnable:   HRESULT
VAR_INPUT
    bFeedEnaPositive: BOOL;
    bFeedEnaNegative: BOOL;
END_VAR
```

Return value

Name	Type	Description
FeedEnable	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

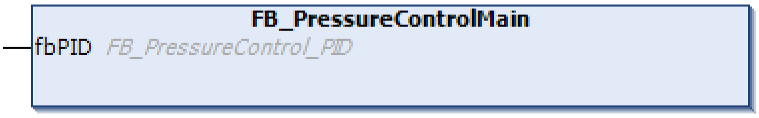
Inputs

Name	Type	Description
bFeedEnaPositive	BOOL	A TRUE enables active movements in positive direction.
bFeedEnaNegative	BOOL	A TRUE enables active movements in negative direction.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.20 PressureControl



FB_PressureControlMain

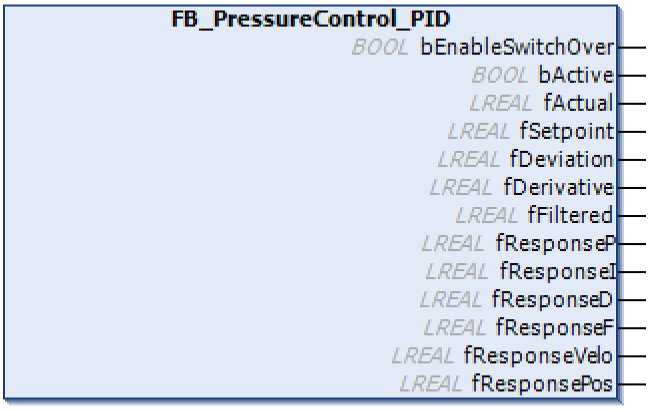
fbPID FB_PressureControl_PID

This core function is provided to allow access to a number of controller types. Currently, there is an extended PID controller [▶ 137].

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.20.1 PressureControl.PID



FB_PressureControl_PID

BOOL bEnableSwitchOver
BOOL bActive
LREAL fActual
LREAL fSetpoint
LREAL fDeviation
LREAL fDerivative
LREAL fFiltered
LREAL fResponseP
LREAL fResponseI
LREAL fResponseD
LREAL fResponseF
LREAL fResponseVelo
LREAL fResponsePos

This core function implements an extended PID controller functionality that can be used for a number of tasks:

- Holding pressure control in injection molding machines
- Back pressure control in injection molding machines
- Others

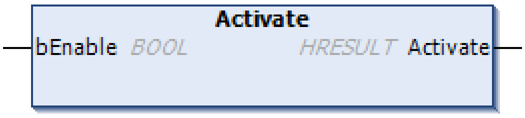
Properties

Name	Type	Description
IsEnabled	BOOL	A TRUE signals the active state of the controller.

Methods

Name	Description
Activate [▶ 138]	Enable/disable the controller.
EnableSwitchOver [▶ 138]	Enable for automatic activation by a PressureHandler.
GetActual [▶ 139]	The actual value of the controller is determined.
GetParams [▶ 139]	An interface to the connected parameter set is determined.
SetParams [▶ 140]	A parameter set is connected to the controller.
Setpoint [▶ 140]	The setpoint of the controller is set.
SwitchOver [▶ 140]	Automatic activation by a PressureHandler.

4.20.1.1 Activate



This method is used to activate and deactivate the controller.

Syntax:

```
METHOD Activate: HRESULT
VAR_INPUT
    bEnable: BOOL;
END_VAR
```

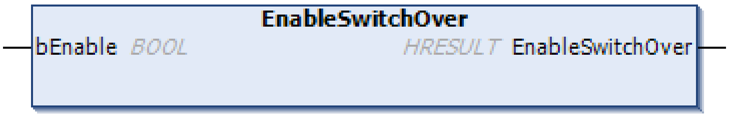
Return value

Name	Type	Description
Activate	HRESULT	See below

Inputs

Name	Type	Description
bEnable	BOOL	Enable for the controller.

4.20.1.2 EnableSwitchOver



This method can be used to enable or disable automatic activation by a pressure handler.

Syntax:

```
METHOD EnableSwitchOver: HRESULT
VAR_INPUT
    bEnable: BOOL;
END_VAR
```

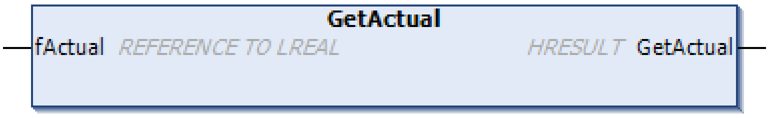
Return value

Name	Type	Description
EnableSwitchOver	HRESULT	See below

Inputs

Name	Type	Description
bEnable	BOOL	The enable for automatic activation.

4.20.1.3 GetActual



The actual value of the controller is determined.

Syntax:

```
METHOD GetActual : HRESULT
VAR_INPUT
fActual: REFERENCE TO LREAL;
END_VAR
```

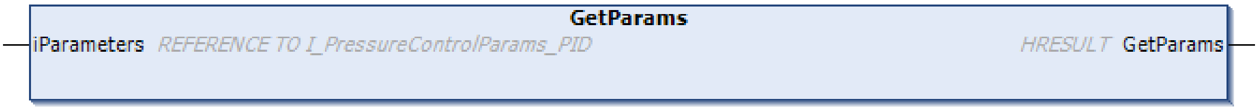
Return value

Name	Type	Description
GetActual	HRESULT	See below

Inputs

Name	Type	Description
fActual	REFERENCE TO LREAL	A reference to the variable to be updated with the actual value.

4.20.1.4 GetParams



An interface to the parameters of the controller is determined.

Syntax:

```
METHOD GetParams: HRESULT
VAR_INPUT
    iParameters: REFERENCE TO I_PressureControlParams_PID;
END_VAR
```

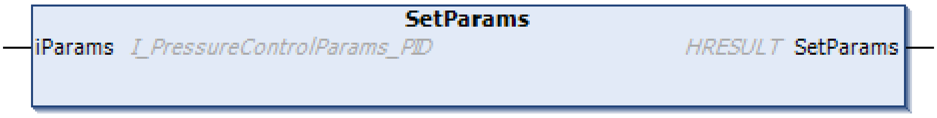
Return value

Name	Type	Description
GetParams	HRESULT	See below

Inputs

Name	Type	Description
iParameters	REFERENCE TO I_PressureControlParams_PID	A reference to the variable to be updated with the interface.

4.20.1.5 SetParams



A parameter set is connected to the controller.

Syntax:

```
METHOD SetParams: HRESULT
VAR_INPUT
    iParams: I_PressureControlParams_PID [▶ 141];
END_VAR
```

Return value

Name	Type	Description
SetParams	HRESULT	See below

Inputs

Name	Type	Description
iParameters	I_PressureControlParams_PID	An interface to the parameter set.

4.20.1.6 Setpoint

Setpoint
fValue
LREAL
HRESULT
Setpoint

The setpoint of the controller is defined.

Syntax:

```
METHOD Setpoint: HRESULT
VAR_INPUT
    fValue: LREAL;
END_VAR
```

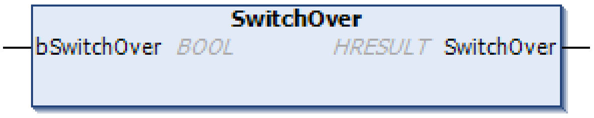
Return value

Name	Type	Description
Setpoint	HRESULT	See below

Inputs

Name	Type	Description
fValue	LREAL	The setpoint.

4.20.1.7 SwitchOver



This method can be used by a pressure handler function block.

A TRUE at bSwitchover will activate the controller if EnableSwitchOver(TRUE) has been called before.

Syntax:

```
METHOD SwitchOver: HRESULT
VAR_INPUT
    bSwitchOver: BOOL;
END_VAR
```

Return value

Name	Type	Description
SwitchOver	HRESULT	See below

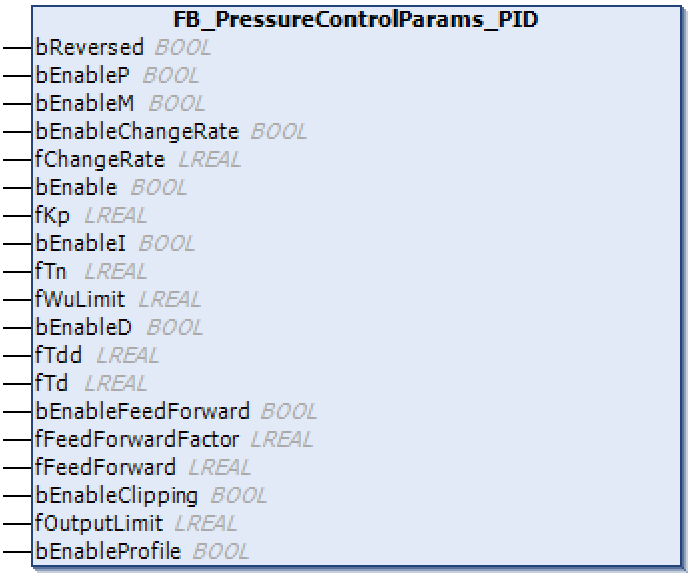
The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bSwitchOver	BOOL	The activation signal of the pressure handler.

4.20.2 FB_PressureControlParams_PID



This function block contains a parameter set to be used by a FB_PressureControl_PID [▶ 137] function block.

Syntax:

```
FUNCTION_BLOCK FB_PressureControlParams_PID IMPLEMENTS I_PressureControlParams_PID
END_VAR
VAR_OUTPUT
END_VAR
```

Properties

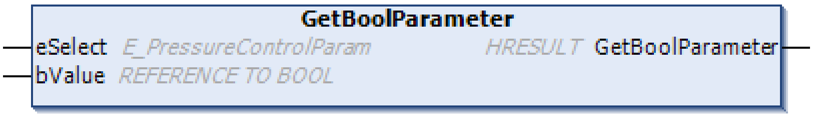
Name	Type	Access	Properties
ChangeRate	LREAL	Get, Set	The ramp rate to use. **See EnableChangeRate for details.**
Enable	BOOL	Get, Set	This property is used to activate and deactivate the proportional component of the PID controller. **For details see Kp.**
EnableChangeRate	BOOL	Get, Set	A setpoint ramp function can be activated and deactivated. **The internally used setpoint for the controller may be updated to the given setpoint by a limited rate, given as ChangeRate. This parameter is given in pressure units per second.** **A setting of ChangeRate:=0.0 or EnableChangeRate:=FALSE disables the ramp function and causes the internal setpoint to immediately follow the specified setpoint.**
EnableClipping	BOOL	Get, Set	A limiting function for the output can be activated and deactivated. **For details see OutputLimit.**
EnableD	BOOL	Get, Set	The differential component of the PID controller can be activated and deactivated. **For details see Td.**
EnableFeedForward	BOOL	Get, Set	A velocity pre-control can be activated and deactivated. **For details see FeedForward.**
EnableI	BOOL	Get, Set	This property is used to enable and disable the integrating component of the PID controller. For details see Tn below.
EnableM	BOOL	Get, Set	This property is used to enable and disable a negative output of the controller.
EnableP	BOOL	Get, Set	This property is used to enable or disable a positive output of the controller.
FeedForward	LREAL	Get, Set	A velocity feed forward component. **When EnableFeedForward is set to TRUE, the value of FeedForward is multiplied by FeedForwardFactor and added to the PID response output.** **An active back pressure controller is used to match the injector backward velocity to the effect of the dosing axis. This function can be used to achieve a more dynamic adjustment to changes in turn rate.**
FeedForwardFactor	LREAL	Get, Set	This property is a parameter of the velocity feed forward function. **For details see FeedForward.**
Kp	LREAL	Get, Set	The proportional gain of the PID controller. Enable must be TRUE to allow the calculation. **The unit is velocity unit per pressure unit.**
OutputLimit	LREAL	Get, Set	A limit for the response of the controller.

Name	Type	Access	Properties
Reversed	BOOL	Get, Set	This property is used to invert the output of the PID controller. **In a number of applications, the axis must move in the positive direction to relieve excess pressure. Typical examples are holding pressure and back pressure controllers in injection molding machines.**
Td	LREAL	Get, Set	The differential component of the PID controller. **The response is calculated if EnableD is TRUE and Td and Tdd are >=cycle time, otherwise it is zero.** **The unit is velocity units * second per pressure unit.**
Tdd	LREAL	Get, Set	A parameter of the differential component of the PID controller. For details see Td above.
Tn	LREAL	Get, Set	The integrating component of the PID controller. The response is calculated if EnableI is TRUE and Tn >= cycle time, otherwise it is zero. The output is limited to WuLimit. **The unit is velocity units per (pressure unit * second).**
WuLimit	LREAL	Get, Set	A parameter of the integrating component of the PID controller. **For details see Tn above.**

Methods

Name	Description
GetBoolParameter [▶ 143]	This method is used to read BOOL parameters of the controller. For details see E_PressureControlParam [▶ 145].
GetFloatParameter [▶ 144]	This method is used to read LREAL parameters of the controller. For details see E_PressureControlParam [▶ 145].
SetBoolParameter [▶ 144]	This method is used to define BOOL parameters of the controller. For details see E_PressureControlParam [▶ 145].
SetFloatParameter [▶ 145]	This method is used to define LREAL parameters of the controller. For details see E_PressureControlParam [▶ 145].

4.20.2.1 GetBoolParameter



This method is used to read BOOL parameters of the controller. For details see E_PressureControlParam [▶ 145].

Syntax:

```
METHOD GetBoolParameter : HRESULT
VAR_INPUT
    eSelect: E_PressureControlParam;
    bValue : REFERENCE TO BOOL;
END_VAR
```

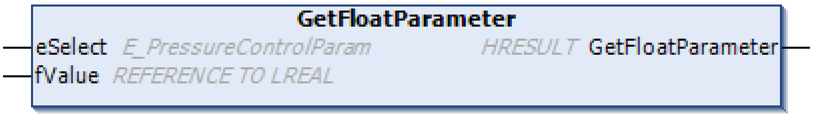
Return value

Name	Type	Description
GetBoolParameter	HRESULT	See below

Inputs

Name	Type	Description
eSelect	E_PressureControlParam	The selection of the parameter.
bValue	REFERENCE TO BOOL	A reference to the variable to be updated with the parameter.

4.20.2.2 GetFloatParameter


GetFloatParameter
eSelect E_PressureControlParam
HRESULT GetFloatParameter
fValue REFERENCE TO LREAL

This method is used to read LREAL parameters of the controller. For details see E_PressureControlParam [▶ 145].

Syntax:

```
METHOD GetFloatParameter : HRESULT
VAR_INPUT
    eSelect: E_PressureControlParam;
    fValue : REFERENCE TO LREAL;
END_VAR
```

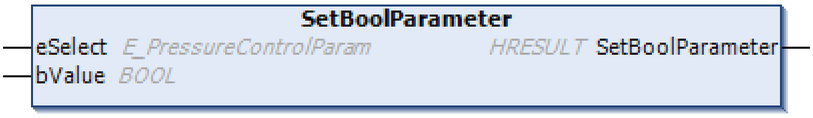
Return value

Name	Type	Description
GetFloatParameter	HRESULT	See below

Inputs

Name	Type	Description
eSelect	E_PressureControlParam	The selection of the parameter.
fValue	REFERENCE TO LREAL	A reference to the variable to be updated with the parameter.

4.20.2.3 SetBoolParameter



This method is used to define BOOL parameters of the controller. For details see E_PressureControlParam [▶ 145].

Syntax:

```
METHOD SetBoolParameter : HRESULT
VAR_INPUT
    eSelect: E_PressureControlParam;
    bValue : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetBoolParameter	HRESULT	See below

Inputs

Name	Type	Description
eSelect	E_PressureControlParam	The selection of the parameter.
bValue	BOOL	The value with which the parameter is to be defined.

4.20.2.4 SetFloatParameter

SetFloatParameter
eSelect E_PressureControlParam
HRESULT SetFloatParameter
fValue LREAL

This method is used to define LREAL parameters of the controller. For details see E_PressureControlParam [▶ 145].

Syntax:

```
METHOD SetFloatParameter : HRESULT
VAR_INPUT
    eSelect: E_PressureControlParam;
    fValue : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetFloatParameter	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
eSelect	E_PressureControlParam	The selection of the parameter.
fValue	LREAL	The value with which the parameter is to be defined.

4.20.3 E_PressureControlParam

The values of this enumeration are used by GetBoolParameter(), GetFloatParameter(), SetBoolParameter() and SetFloatParameter() of FB_PressureControlParams_PID [▶ 137].

```
TYPE E_PressureControlParam :
(
eKp := 1,
eTn,
eTd,
eTdd,
//
eWuLimit,
eOutLimit,
//
eChangeRate,
eFeedForward,
eFeedForwardFactor,

eEnable,
eEnableP,
eEnableM,
//
eReversed,
//
eEnableFF,
eEnableI,
eEnableD,
eEnableClipping,
eEnableChangeRate
);
END_TYPE
```

4.21 Ptp

Core function is not supported by inverter axes

This core function is not supported by inverter axes. Any use will report DEVICE_NOTINIT and trigger an error message.

This core function is used to perform multi-segment movements with the possibility to switch to torque or constant output clamping at the end.

Syntax:

```
FUNCTION_BLOCK FB_Ptp EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
ActiveSegment	INT	Get	The number of the point that is currently being approached.
IsClamping	BOOL	Get	TRUE if the core function has switched the axis to clamping.
MovingNegative	BOOL	Get	TRUE if the axis is actively moving in the negative direction.
MovingPositive	BOOL	Get	TRUE if the axis is actively moving in the positive direction.
NumberOfPoints	INT	Get	The number of points that the core function can store.

Methods

Name	Description
CheckPoint() [▶ 147]	The specified segment is compared with the axis parameters.

Name	Description
DoMove() [▶ 148]	The execution is triggered.
GetClampPoint() [▶ 148]	A segment of the clamping table is read back.
GetPoint() [▶ 149]	A segment of the lookup table is read back.
GetUpdatedPoint() [▶ 150]	A segment of the lookup table is read back.
InvalidateClampPoint()	All segments in the clamping table are marked as invalid.
InvalidateTable()	All segments in the lookup table are marked as invalid.
SetClampPoint() [▶ 150]	A segment of the clamping table is defined.
SetPoint() [▶ 151]	A segment of the lookup table is defined.
UpdatePosition() [▶ 152]	The target position of the segment is changed after it is defined.

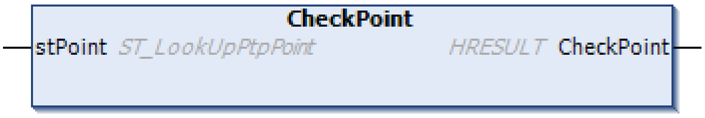
Interfaces

Type	Description
I_Ptp	Standard interface on FB_Ptp.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.1 CheckPoint()



The transferred point is compared with the axis parameters. The result is only tested as `SUCCEEDED()` if the target does not exceed any of the enabled software position limits and does not fall below the minimum velocity.

Syntax:

```
METHOD CheckPoint: HRESULT
VAR_INPUT
    stPoint: ST_LookUpPtpPoint;
END_VAR
```

Return value

Name	Type	Description
CheckPoint	HRESULT	Return value with feedback on the validity of the point.

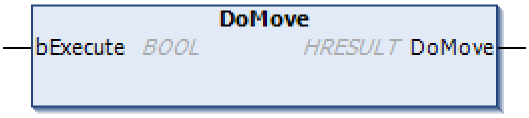
Inputs

Name	Type	Description
stPoint	ST_LookUpPtpPoint [▶ 152]	The point to be checked.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.2 DoMove()



The PTP movement is triggered.

Syntax:

```
METHOD DoMove: HRESULT
VAR_INPUT
    bExecute: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoMove	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

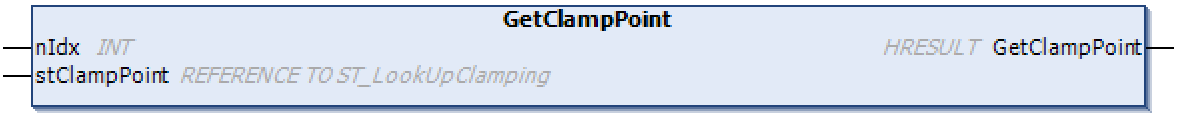
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge triggers the command.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.3 GetClampPoint()



A point in the clamping table is read back.

Syntax:

```
METHOD GetClampPoint: HRESULT
VAR_INPUT
    nIdx:           INT;
    stClampPoint:   REFERENCE TO ST_LookUpClamping;
END_VAR
```

Return value

Name	Type	Description
GetClampPoint	HRESULT	Return value with feedback on the success of the execution.

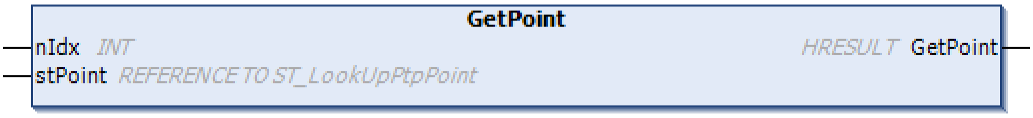
Inputs

Name	Type	Description
nIdx	INT	The index of the point.
stClampPoint	REFERENCE TO ST_LookUpClamping [▸ 153]	A reference to the variable that is to be updated with the requested point.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.4 GetPoint()



A point of the lookup table is read back.

Syntax:

```
METHOD GetPoint: HRESULT
VAR_INPUT
    nIdx   : INT;
    stPoint: REFERENCE TO ST_LookUpPtpPoint;
END_VAR
```

Return value

Name	Type	Description
GetPoint	HRESULT	Return value with feedback on the success of the execution.

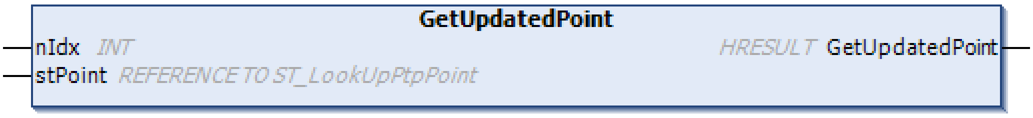
Inputs

Name	Type	Meeting
nIdx	INT	The index of the interpolation point.
stPoint	REFERENCE TO ST_LookUpPtpPoint [▸ 152]	A reference to the variable that is to be updated with the requested point.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.5 GetUpdatedPoint()



A point of the lookup table is read back.

Syntax:

```
METHOD GetUpdatedPoint: HRESULT
VAR_INPUT
    nIdx    : INT;
    stPoint : REFERENCE TO ST_LookUpPtpPoint;
END_VAR
```

Return value

Name	Type	Description
GetUpdatedPoint	HRESULT	Return value with feedback on the success of the execution.

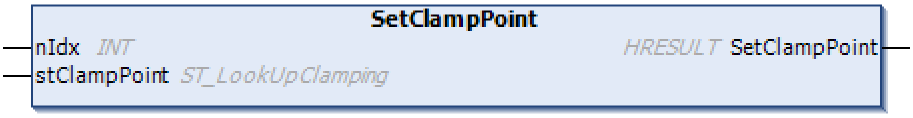
Inputs

Name	Type	Meeting
nIdx	INT	The index of the segment.
stPoint	REFERENCE TO ST_LookUpPtpPoint [▸ 152]	A reference to the variable that is to be updated with the requested point.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.6 SetClampPoint()



A segment of the clamping table is defined.

Syntax:

```
METHOD SetClampPoint: HRESULT
VAR_INPUT
    Idx: INT;
END_VAR
```

Return value

Name	Type	Description
SetClampPoint	HRESULT	Return value with feedback on the success of the execution.

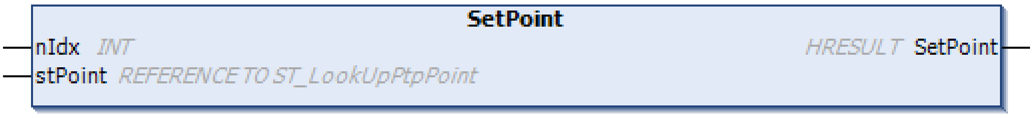
Inputs

Name	Type	Description
nIdx	INT	The index of the clamping point.
stClampPoint	ST_LookUpClamping [▶ 153]	The clamping point to be used.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.7 SetPoint()



A point of the lookup table is defined.

Syntax:

```
METHOD SetPoint: HRESULT
VAR_INPUT
    nIdx    : INT;
    stPoint: REFERENCE TO ST_LookUpPtpPoint;
END_VAR
```

Return value

Name	Type	Description
SetPoint	BOOL	Return value with feedback on the success of the execution.

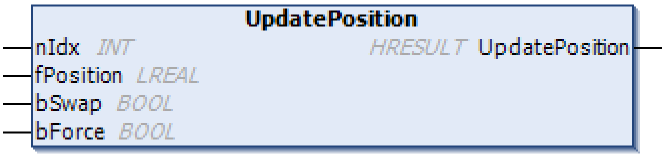
Inputs

Name	Type	Description
nIdx	INT	The index of the segment.
stPoint	REFERENCE TO ST_LookUpPtpPoint [▶ 152]	A reference to the variable that is to be updated.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.8 UpdatePosition()



The target position of the point is changed after it is defined.

Syntax:

```
METHOD UpdatePosition:  HRESULT
VAR_INPUT
    nIdx      : INT;
    fPosition: LREAL;
    bSwap     : BOOL;
END_VAR
```

Return value

Name	Type	Description
UpdatePosition	BOOL	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
nIdx	INT	The index of the point to be updated.
fPosition	LREAL	The new target position.
bSwap	BOOL	With `bSwap=TRUE` the direction related enables (PositiveOnly, NegativeOnly) are swapped.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.9 ST_LookUpPtpPoint

Such a structure contains the information of a PTP point.

Syntax:

```
// ATTENTION: pointer arithmetics in arrays of this type
// be sure sizeof is the same as distance in arrays
//
TYPE ST_LookUpPtpPoint :
STRUCT
    Position:        LREAL;
    Velocity:        LREAL;
    Acceleration:    LREAL;
    Deceleration:    LREAL;
    Jerk:            LREAL;

    Limiting:        LREAL;                 // Pressure or torque

    Valid:           BOOL;                  // 1 byte
    PositiveOnly:    BOOL;                  // 1 byte
    NegativeOnly:    BOOL;                  // 1 byte
    balign:          ARRAY[4..8] OF BOOL;   // 5 bytes alignment to 8 bytes
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Position	LREAL	Target position of the PTP point. If there is another point in the motion profile, it will be blended with the BlendingLow rule.
Velocity	LREAL	Velocity with which the point is to be approached.
Acceleration	LREAL	Acceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Deceleration	LREAL	Deceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Jerk	LREAL	Jerk with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Limiting	LREAL	Limitation (torque or pressure) with which the point is to be approached. **If 0 is transferred, the idle value of the axis is used.**
Valid	BOOL	Marks the point as valid for use with the next command.
PositiveOnly	BOOL	The PTP movement should only be executed if the resulting movement will be in a positive direction.
NegativeOnly	BOOL	The PTP movement should only be executed if the resulting movement will be in the negative direction.
balign	ARRAY [4..8] OF BOOL	[INTERNAL] Adjustment to a byte length of the structure divisible by 8.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.21.10 ST_LookUpClamping

Such a structure contains the information of a clamping point.

Syntax:

```
// ATTENTION: pointer arithmetics in arrays of this type
// be sure sizeof is the same as distance in arrays
//
TYPE ST_LookUpClamping :
STRUCT
    Position:        LREAL;
    Velocity:        LREAL;
    Acceleration:    LREAL;                 // velocity units per second
    Limiting:        LREAL;                 // pressure or torque
    LimitingRamp:    LREAL;                 // limiting units per second

    Duration:        LREAL;                 // seconds

    Valid:           BOOL;                  // 1 byte
    PositiveOnly:    BOOL;                  // 1 byte
    NegativeOnly:    BOOL;                  // 1 byte
    balign:          ARRAY[4..8] OF BOOL;   // 5 bytes alignment to 8 bytes
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Position	LREAL	Target position of the clamping point.
Velocity	LREAL	Velocity with which the clamp feed should be moved.
Acceleration	LREAL	Acceleration with which the clamping velocity should be moved. **If 0 is transferred, the default values of the axis are used.**
Limiting	LREAL	Limitation (torque or pressure) with which the clamping is to be executed. **If 0 is transferred, the idle value of the axis is used.**
LimitingRamp	LREAL	Ramp to the limitation value [limitation unit / s].
Duration	LREAL	Duration over which the clamping point is to be executed.
Valid	BOOL	Marks the clamping point as valid for use with the next command.
PositiveOnly	BOOL	The clamping movement should only be carried out if the resulting movement will be in a positive direction.
NegativeOnly	BOOL	The clamping movement should only be carried out if the resulting movement will be in the negative direction.
balign	ARRAY [4..8] OF BOOL	[INTERNAL] Adjustment to a byte length of the structure divisible by 8.

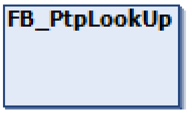
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.22 PtpLookUp

Not available via the axis interfaces

This core function is not available via the axis interfaces. Direct access by the application is not required.



This core function is used to store the definition of a complex multi-segment movement.

Connection during startup

The internal PtpLookUp core function of the axis is connected to the Ptp core function at startup.

Properties

Name	Type	Access	Description
NumberOfPoints	INT	Get	The number of motion points that the core function can store.

Definition at the instantiation

The capacity of the core function must be defined at instantiation. See Instantiation [▶ 15].

Methods

Name	Description
GetPoint() [▶ 155]	A point from the lookup table is read back.
Invalidate()	All points in the lookup table are marked as invalid.
ReadMaster() [▶ 156]	A master value is determined that corresponds to the specified slave value.
SetPoint() [▶ 156]	A point of the lookup table is defined.
UpdatePosition() [▶ 157]	In some applications, the target position of the segments must be changed after definition. **This method is used, for example, with transforming axes.**

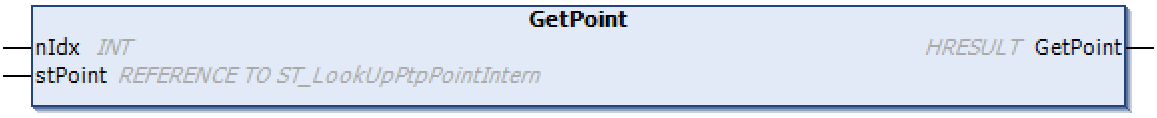
Interfaces

Type	Description
I_PtpLookUp	Standard interface on FB_PtpLookUp.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.22.1 GetPoint()



A point from the lookup table is read back.

Syntax:

```
METHOD GetPoint: HRESULT
VAR_INPUT
    nIdx    : INT;
    stPoint: REFERENCE TO ST_LookUpPtpPointIntern;
END_VAR
```

Return value

Name	Type	Description
GetPoint	HRESULT	Return value with feedback on the success of the execution.

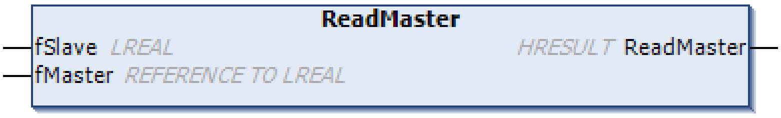
Inputs

Name	Type	Description
nIdx	INT	The index of the point. Permissible range 1 ... NumberOfPoints.
stPoint	REFERENCE TO ST_LookUpPtpPointIntern	A reference to the variable to be updated with the point data.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.22.2 ReadMaster()



This method finds a master value that corresponds to the specified slave value.

Syntax:

```
METHOD ReadMaster : HRESULT
VAR_INPUT
    fSlave:      LREAL;
    fMaster:     REFERENCE TO LREAL;
END_VAR
```

Return value

Name	Type	Description
ReadMaster	HRESULT	Return value with feedback on the success of the execution.

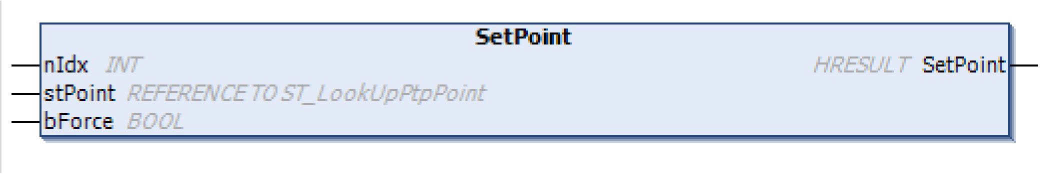
Inputs

Name	Type	Description
fSlave	LREAL	The specified slave value.
fMaster	REFERENCE TO LREAL	A reference to the variable to be updated with the master position.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.22.3 SetPoint()



A point of the lookup table is defined.

Syntax:

```
METHOD SetPoint: HRESULT
VAR_INPUT
    nIdx   : INT;
    stPoint: REFERENCE TO ST_LookUpPtpPoint;
    bForce : BOOL;
END_VAR
```

Return value

Name	Type	Description
SetPoint	HRESULT	Return value with feedback on the success of the execution.

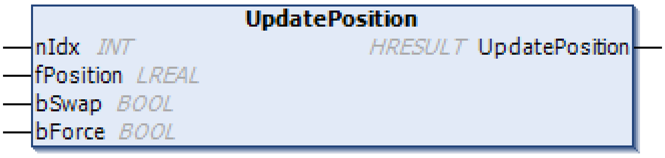
Inputs

Name	Type	Description
nIdx	INT	The index of the point. Permissible range 1 ... NumberOfPoints.
stPoint	REFERENCE TO ST_LookUpPtpPoint [▶ 152]	A reference to the variable to be updated with the point data.
bForce	BOOL	If TRUE, the segment will be updated even if the axis executes a command. *Notice* **Make sure that the axis does not execute a Ptp command.**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.22.4 UpdatePosition()



In some use cases, the target position of the segments must be modified after definition. This method is used for transforming axes, for example.

Syntax:

```
METHOD UpdatePosition: HRESULT
VAR_INPUT
    nIdx     : INT;
    fPosition: LREAL;
    bSwap    : BOOL;
    bForce   : BOOL;
END_VAR
```

Return value

Name	Type	Description
UpdatePosition	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
nIdx	INT	The index of the point. Permissible range 1 ... NumberOfPoints.
fPosition	LREAL	The value to be used for the update.
bSwap	BOOL	With `bSwap=TRUE` the direction related enables (PositiveOnly, NegativeOnly) are swapped.
bForce	BOOL	If TRUE, the point will be updated even if the axis executes a command. *Notice* **Make sure that the axis does not execute a Ptp command.**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.23 Reset

Not available via the axis interfaces

This core function is not available via the axis interfaces. Direct access by the application is not required.

This core function resets the error state of an axis in a controlled manner. The `DoReset()` method can be accessed directly via the motion interface `I_MotionBase` and the interfaces inherited from it.

Syntax:

```
FUNCTION_BLOCK FB_Reset EXTENDS FB_CorefunctionFeedback
```

Methods

Name	Description
DoReset()	Resets the error state of an axis.

Interfaces

Type	Description
I_Reset	Standard interface on FB_Reset.

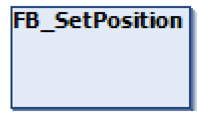
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.24 SetPosition

Core function is not supported by inverter axes

This core function is not supported by inverter axes. Any use will report DEVICE_NOTINIT and trigger an error message.



This core function is used to change the actual position without physically moving the axis. It updates the offset of the position encoder function.

Properties

Name	Type	Access	Description
Mode	BOOL	Get, Set	`Mode = TRUE` causes the actual position to be changed by an amount specified as Target. `Mode = FALSE` causes the actual position to be set to the value specified as the target. **This property can also be set using the SetParameter() method.**
Target	LREAL	Get, Set	The new position value is set here. **This property can also be set using the SetParameter() method.**

Methods

Name	Description
DoSetPosition() [▶ 159]	A rising edge at the input `bExecute` triggers the setting of the position.
SetParameter() [▶ 160]	Here the new position and the operation mode of the core function are defined.

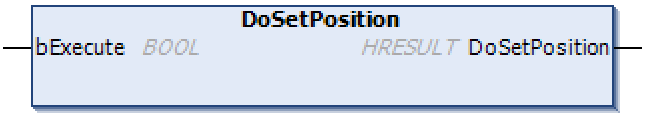
Interfaces

Type	Description
I_SetPosition	Standard interface on FB_SetPosition.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.24.1 DoSetPosition()



This method triggers the core function.

Syntax:

```
METHOD DoSetPosition : HRESULT
VAR_INPUT
    bExecute:        BOOL;
END_VAR
```

Return value

Name	Type	Description
DoSetPosition	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.

E_AdsErr	Return value	Cause
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

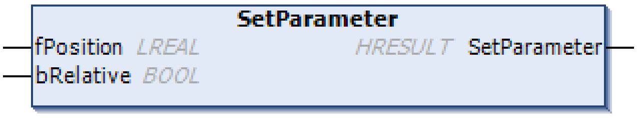
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge starts the core function.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.24.2 SetParameter()



Here the new position and the operation mode of the core function are defined.

Syntax:

```
METHOD SetParameter : HRESULT
VAR_INPUT
fPosition: LREAL;
bRelative: BOOL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

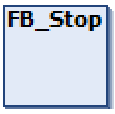
Inputs

Name	Type	Description
fPosition	LREAL	The position value to be used is defined here.
bRelative	BOOL	`bRelative = TRUE` causes the actual position to be changed by an amount specified as fPosition. `bRelative = FALSE` causes the actual position to be set to the value specified as fPosition.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.25 Stop



This core function is used to perform a stop operation using certain parameters.

Properties

Name	Type	Access	Description
NoCreeping	BOOL	Get, Set	A TRUE in this property avoids the creep phase at the end of the stopping process of the hydraulics library.

Methods

Name	Description
DoStop() [▶ 161]	A rising edge triggers the stop.
SetParameter() [▶ 162]	This method is used to define the dynamic parameters of the operation.

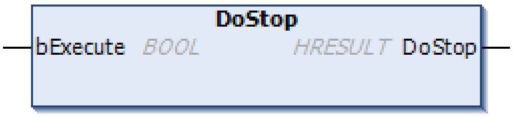
Interfaces

Type	Description
I_Stop	Standard interface on FB_Stop.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.25.1 DoStop()



This method is used to trigger the stop.

Syntax:

```
METHOD DoStop : HRESULT
VAR_INPUT
    bExecute: BOOL;
END_VAR
```

Return value

Name	Type	Description
DoStop	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

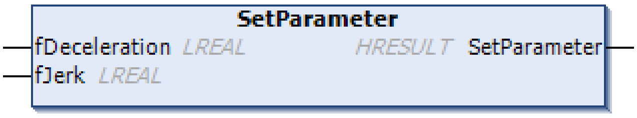
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge at this input triggers the stop.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.25.2 SetParameter()



This method is used to define the dynamic parameters of the operation.

Syntax:

```
METHOD SetParameter: HRESULT
VAR_INPUT
    fDeceleration: LREAL;
    fJerk        : LREAL;
END_VAR
```

Return value

Name	Type	Description
SetParameter	HRESULT	Return value with feedback on the success of the execution.

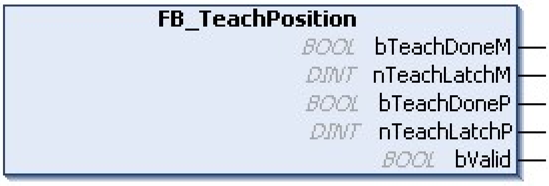
Inputs

Name	Type	Description
fDeceleration	LREAL	The deceleration to be used.
fJerk	LREAL	The jerk to be used.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.26 TeachPosition



FB_TeachPosition

BOOL bTeachDoneM

DINT nTeachLatchM

BOOL bTeachDoneP

DINT nTeachLatchP

BOOL bValid

This core function is used for referencing analog encoder systems.

Syntax:

```
FUNCTION_BLOCK FB_TeachPosition EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
TeachLatchM	LREAL	Get, (Set)	The recorded value when executing `DoTeachM()` in analog increments.
TeachLatchPTeachLatchP	LREAL	Get, (Set)	The recorded value when executing `DoTeachP()` in analog increments.
Valid	BOOL	Get	Two valid values were recorded.

Methods

Name	Description
DoTeach(`bExecute`)	Sets the axis to teaching mode.
DoTeachM(`bExecute`)	Triggers the recording of the lower position.
DoTeachP(`bExecute`)	Triggers the recording of the upper position.

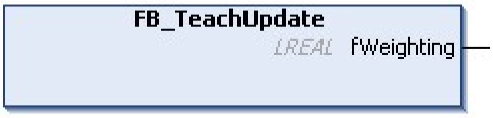
Interfaces

Type	Description
I_TeachPosition	Standard interface on FB_TeachPosition.
I_TeachPositionDev	Extended interface with access to the setters of the latch properties.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4.27 TeachUpdate



FB_TeachUpdate

LREAL fWeighting

This core function is used for referencing analog encoder systems. It is used after TeachPosition [▸ 163] to calculate the measured values.

Syntax:

```
FUNCTION_BLOCK FB_TeachUpdate EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
SetUpperPosition	BOOL	Get, Set	FALSE = The set position is used for the lower measured value. TRUE = The set position is used for the upper measured value.

Methods

Name	Description
DoUpdate(bExecute)	Calculates the measurement and parameters for encoder offset and scaling.
SetPosition(fPosition)	Configures the setpoint at one of the measuring points.
SetStroke(fStroke)	Configures the stroke between the two measuring points.

Interfaces

Type	Description
I_TeachUpdate	Standard interface on FB_TeachUpdaten.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.28 TorqueLimiting

This function is responsible for the torque limitation of all axis functions. The following function blocks are automatically available internally for NC axes:

- FB_TorqueLimitingCoE - DS402 (AX8000) based TorqueLimiting
- FB_TorqueLimitngSoE - AX5000 based TorqueLimiting

No core function

This function does not belong to any classification of core functions. It is only mentioned in the context of core functions, as it can be achieved via the properties of an axis.

Syntax:

```
FUNCTION_BLOCK FB_TorqueLimitingBase EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
Activate	BOOL	Get, Set	Activates the use of the ActualValue as an active limit.
ActiveValue	LREAL	Get	Current torque value output.
ActualTorque	LREAL	Get, Set	Current actual torque value of the axis.
ConnectedToDrive	BOOL	Get	An internal connection to a drive unit has been established.
DefaultFb	I_TorqueLimitingFb	(Get, Set)	Access to the standard function block for torque limitation if specific handling is used.
Direction	E_AdaptableDirection	Get, Set	Direction in which the torque limit should act.
IdleValue	LREAL	Get, Set	The idle value for the torque limitation to which the axis returns after a command.
MaxValue	LREAL	Get, Set	The effective limitation of the torque limit for each command.
NominalValue	LREAL	Get, Set	The normalized value in relation to the ReferenceValue.
ReferenceValue	LREAL	Get, Set	Reference value for any commanded torque limit.

Methods

Name	Description
ReturnToIdle()	[INTERNAL] Implements the behavior to end a command.

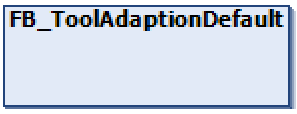
Interfaces

Type	Description
I_TorqueLimiting	Standard interface on FB_TorqueLimitingBase.
I_TorqueLimitingFb	Extended interface for use as a core function.

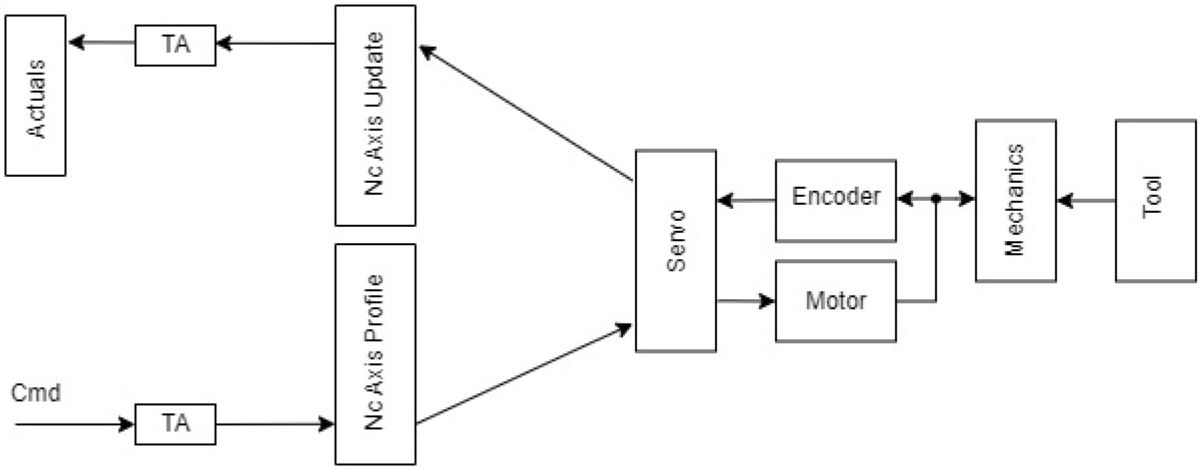
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.29 ToolAdaption



This core function is used to convert between axis coordinates and tool working position. An offset and an optional inversion are taken into account.



Properties

Name	Type	Access	Description
Inverting	BOOL	Get, Set	A TRUE here signals a reversal of direction between axis and tool movement.
Offset	LREAL	Get, Set	The difference between axis and tool position.

Methods

Name	Description
AxisPosition()	The axis position is determined for a given tool position.
AxisVelocity()	The axis velocity is determined for a given tool velocity.
ToolPosition()	The tool position is determined for a given axis position.
ToolVelocity()	The tool velocity is determined for a given axis velocity.

Interfaces

Type	Description
I_ToolAdaption	Standard interface on FB_ToolAdaptionDefault.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30 Transformation

Exclusive function for transforming axes

This core function is only available for transforming axes.

FB_NcTransformation

This core function manages the coupling of the two internal axes of a transforming axis.

To do this, the transformation works internally with three tables for different translation paths:

1. Load table - Stored table in the internal load axis.
    - Is generated from the transferred table.
    - Is used to translate a commanded load position into a drive position.
2. Drive table - Stored table in the internal drive axis.
    - Is the inverse function of the transferred load table.
    - Is used in the full transformation for synchronization of the drive side.
3. Encoder table - Transferred table of the property `Table`.
    - Is used to generate the encoder position of the load side.

Syntax:

```
FUNCTION_BLOCK FB_NcTransformation EXTENDS FB_TrafoCorefunction
```

Properties

Name	Type	Access	Description
IsUpdating	BOOL	Get	TRUE as long as a previously passed table is being processed internally.
LowerLimit	LREAL	Get, Set	Lower limit from which the drive table is linearized.
Reversed	LREAL	Get, Set	Reserved for future use.
Table	I_CammingLookUp [▶ 85]	Get, Set	Interface for the transformation table to be transferred.
UpperLimit	LREAL	Get, Set	Reserved for future use (counterpart to LowerLimit).
Valid	BOOL	Get	The transferred table was loaded successfully.

Methods

Name	Description
GetSlot()	Reserves the index of a slot for asynchronous translation function and returns it.
ReleaseSlot(`nSlot`)	Releases a reserved slot.
TranslateFromDrive() [▶ 168]	Translates a drive position into a load position using the load table.
TranslateFromLoad() [▶ 168]	Translates a load position into a drive position using the drive table.
TranslateToDrive() [▶ 169]	Translates a load position into a drive position using the load table.
TranslateToDrivePrio()	[INTERNAL] Prioritized execution of the `TranslateToDrive()` method.
TranslateToLoad() [▶ 170]	Translates a drive position into a load position using the drive table.
TranslateToLoadPrio()	[INTERNAL] Prioritized execution of the `TranslateToLoad()` method.
UpdateToLinLoad() [▶ 171]	Translates a position in a linearized way, depending on the set LowerLimit properties.

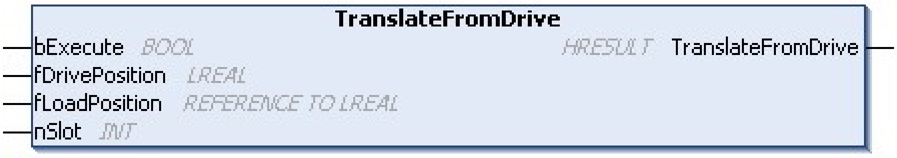
Interfaces

Type	Description
I_Transformation	Standard interface on FB_NcTransformation.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30.1 TranslateFromDrive()



This method is used to calculate a load position using the transformation table of the load table.

Syntax:

```
METHOD TranslateFromDrive : HRESULT
VAR_INPUT
    bExecute:        BOOL;
    fDrivePosition:  LREAL;
    fLoadPosition:   REFERENCE TO LREAL;
    nSlot:           INT;
END_VAR
```

Return value

Name	Type	Description
TranslateFromDrive	HRESULT	Return value with feedback on the success of the execution.

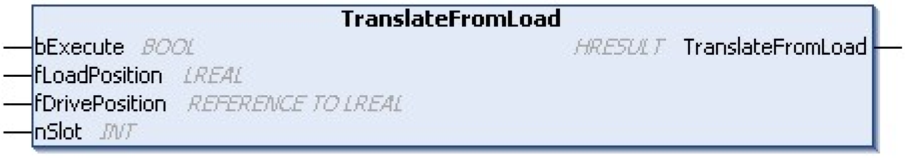
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge starts the asynchronous execution of this method.
fDrivePosition	LREAL	Drive position to be translated.
fLoadPosition	REFERENCE TO LREAL	Translated load position.
nSlot	INT	The slot to be used for the asynchronous query.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30.2 TranslateFromLoad()



This method is used to calculate a drive position using the transformation table of the drive side.

Syntax:

```
METHOD TranslateFromLoad : HRESULT
VAR_INPUT
    bExecute:        BOOL;
    fLoadPosition:   LREAL;
    fDrivePosition:  REFERENCE TO LREAL;
    nSlot:           INT;
END_VAR
```

Return value

Name	Type	Description
TranslateFromLoad	HRESULT	Return value with feedback on the success of the execution.

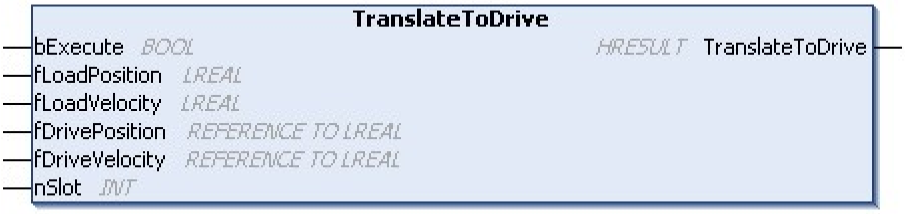
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge starts the asynchronous execution of this method.
fLoadPosition	LREAL	Load position to be translated.
fDrivePosition	REFERENCE TO LREAL	Translated drive position.
nSlot	INT	The slot to be used for the asynchronous query.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30.3 TranslateToDrive()



This method is used to calculate a drive position using the transformation table of the load table.

Syntax:

```
METHOD TranslateToDrive : HRESULT
VAR_INPUT
    bExecute:         BOOL;
    fLoadPosition:    LREAL;
    fLoadVelocity:    LREAL;
    fDrivePosition:   REFERENCE TO LREAL;
    fDriveVelocity:   REFERENCE TO LREAL;
    nSlot:            INT;
END_VAR
```

Return value

Name	Type	Description
TranslateToDrive	HRESULT	Return value with feedback on the success of the execution.

Inputs

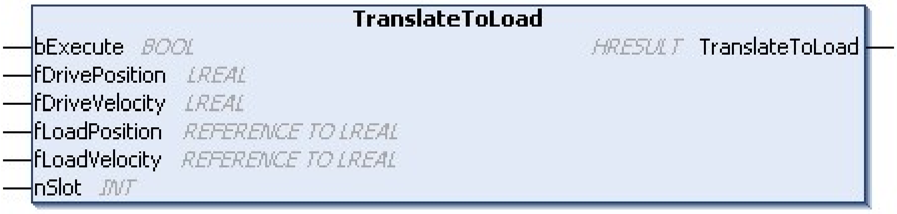
Name	Type	Description
bExecute	BOOL	A rising edge starts the asynchronous execution of this method.
fLoadPosition	LRELA	Load position to be translated.
fLoadVelocity	LREAL	Load velocity to be translated.
fDrivePosition	REFERENCE TO LREAL	Translated load position.

Name	Type	Description
fDriveVelocity	REFERENCE TO LREAL	Translated load velocity.
nSlot	INT	The slot to be used for the asynchronous query.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30.4 TranslateToLoad()



This method is used to calculate a load position via the transformation table of the drive side.

Syntax:

```
METHOD TranslateToLoad : HRESULT
VAR_INPUT
    bExecute:        BOOL;
    fDrivePosition:  LREAL;
    fDriveVelocity:  LREAL;
    fLoadPosition:   REFERENCE TO LREAL;
    fLoadVelocity:   REFERENCE TO LREAL;
    nSlot:           INT;
END_VAR
```

Return value

Name	Type	Description
TranslateToLoad	HRESULT	Return value with feedback on the success of the execution.

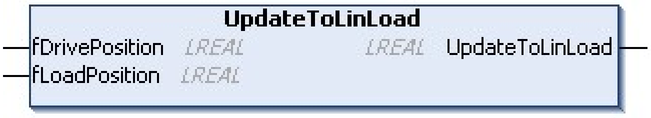
Inputs

Name	Type	Description
bExecute	BOOL	A rising edge starts the asynchronous execution of this method.
fDrivePosition	LREAL	Drive position to be translated.
fDriveVelocity	LREAL	Drive velocity to be translated.
fLoadPosition	REFERENCE TO LREAL	Translated load position.
fLoadVelocity	REFERENCE TO LREAL	Translated load velocity.
nSlot	INT	The slot to be used for the asynchronous query.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.30.5 UpdateToLinLoad()



This method is used to translate a position in a linearized way, depending on the set `LowerLimit` property.

Syntax:

```
METHOD UpdateToLinLoad : LREAL
VAR_INPUT
    fDrivePosition:   LREAL;
    fLoadPosition:    LREAL;
END_VAR
```

Return value

Name	Type	Description
UpdateToLinLoad	LREAL	Linearized load position.

Inputs

Name	Type	Description
fDrivePosition	LREAL	Current drive position.
fLoadPosition	LREAL	Current load position.

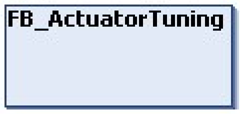
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.31 Tuning

Exclusive function for actuators

This core function is only available for actuators.



This core function is used to automatically optimize an actuator.

Syntax:

```
FUNCTION_BLOCK FB_ActuatorTuning EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
TimeoutFactor	LREAL	Get, Set	Factor (multiplier) for determining the timeout of the movement time.

Methods

Name	Description
DoAbort(`bExecute`)	Cancels the current procedure.
DoTuneTimeout(`bExecute`)	Starts a sequence of commands to the end positions of the actuator and measures the time of the movement. *Notice* **Make sure that the connected end positions can be reached mechanically.**

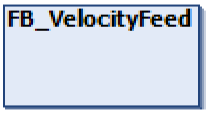
Interfaces

Type	Description
I_ActuatorTuning	Standard interface on FB_ActuatorTuning.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

4.32 VelocityFeed


FB_VelocityFeed

This core function is used to start the axis at a commanded velocity without a defined target position.

Properties

Name	Type	Access	Description
GearFactor	LREAL	Get, Set	The factor with which to respond to GuidingValue.
GuidingValue	LREAL	Get, Set	The commanded velocity.

Decouple the axis

A GearFactor of 0.0 will effectively decouple the axis from the GuidingValue.

Move in the opposite direction

A negative GearFactor causes the axis to move in the opposite direction.

Methods

Name	Description
DoFeed() [▸ 173]	A rising edge triggers the core function.

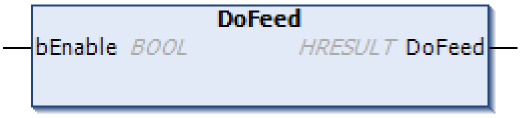
Interfaces

Type	Description
I_VelocityFeed	Standard interface on FB_VelocityFeed.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

4.32.1 DoFeed()



A rising edge at `bEnable` triggers the core function, while a falling edge causes it to stop and return to idle.

Syntax:

```
METHOD DoFeed : HRESULT
VAR_INPUT
    bEnable:    BOOL;
END_VAR
```

Return value

Name	Type	Description
DoFeed	HRESULT	See below

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_BUSY	0x9B00 0708	The axis is busy performing another core function.
DEVICE_INVALIDDATA	0x9B00 0706	Not all data and parameters for the core function are available and valid.
DEVICE_INVALIDSTATE	0x9B00 0712	The state of the axis does not allow the execution.
DEVICE_NOTINIT	0x9B00 0718	The core function is not or not completely initialized.
NOERR	0x1B00 0000	The core function has accepted the command.

Inputs

Name	Type	Description
bEnable	BOOL	A rising edge triggers the core function, while a falling edge causes it to stop and return to idle.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5 I/O

The integration of input and output signals in TwinCAT is realized via mapping. For this purpose, it is relatively simple to define a variable as the carrier of I/O information. When transferring this information, however, the specific data type of the information must be known. To ensure that the interface defined in the TwinCAT 3 Plastic Technology Functions can work independently of the data type, the library creates general I/O classes (FBs) and specific classes derived from them to ensure maximum flexibility and expandability. With the generalization of the I/O type, other tasks such as the processing of diagnostic information can also be implemented.

5.1 Base (Abstract)

5.1.1 IoBase

This class is the basis for all I/O elements. It defines a rudimentary basic scope of methods.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_IoBase EXTENDS FB_MessageBase
```

Methods

Name	Description
Cyclic()	Cycle method
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

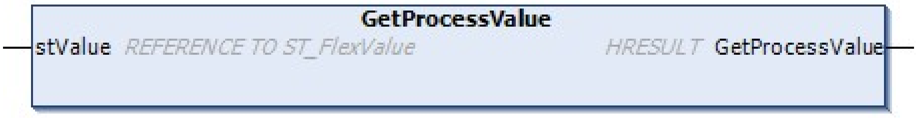
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.1.1 GetProcessValue()



This method is used to read the terminal-specific input/output value.

Syntax:

```
METHOD GetProcessValue : HRESULT
VAR_INPUT
    stValue:      REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
GetProcessValue	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

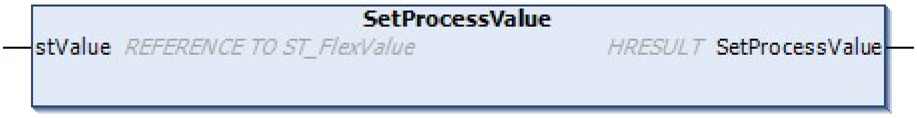
Inputs

Name	Type	Description
stValue	REFERENCE TO ST_FlexValue [▸ 214]	Reference to a variable to which the input/output value is to be written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.1.2 SetProcessValue()



This method is used to write the terminal-specific input/output value.

Syntax:

```
METHOD SetProcessValue : HRESULT
VAR_INPUT
    stValue:      REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
SetProcessValue	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOL NOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMP ATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOT SUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

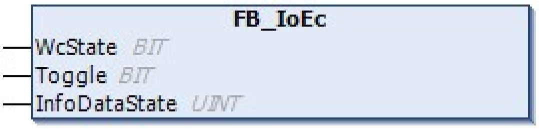
Inputs

Name	Type	Description
stValue	REFERENCE TO ST_FlexValue [▶ 214]	Reference to a variable whose value is to be set to the input/output value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.2 IoEc



This class is the basis for EtherCAT-based I/O elements.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_IoEc EXTENDS FB_IoBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetProcessIoStatus() [▶ 177]	Reads the EtherCAT status of the input/output element.

Name	Description
SetProcessIoStatus() [▶ 178]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 177]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 178]	Writes the terminal-specific input/output value.

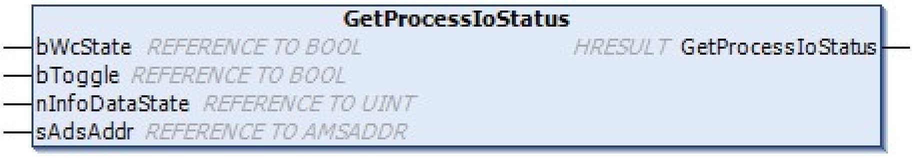
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.2.1 GetProcessIoStatus()


GetProcessIoStatus
bWcState REFERENCE TO BOOL
bToggle REFERENCE TO BOOL
nInfoDataState REFERENCE TO UINT
sAdsAddr REFERENCE TO AMSADDR
HRESULT GetProcessIoStatus

This method is used to read the EtherCAT status of an I/O element.

Syntax:

```
METHOD GetProcessIoStatus : HRESULT
VAR_INPUT
    bWcState:                REFERENCE TO BOOL;
    bToggle:                 REFERENCE TO BOOL;
    nInfoDataState:          REFERENCE TO UINT;
    sAdsAddr:                REFERENCE TO AMSADDR;
END_VAR
```

Return value

Name	Type	Description
GetProcessIoStatus	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

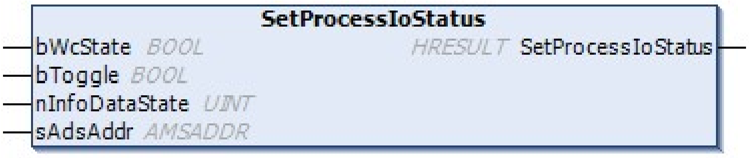
Inputs

Name	Type	Description
bWcState	REFERENCE TO BOOL	Reference to a variable to which the WcState is to be written.
bToggle	REFERENCE TO BOOL	Reference to a variable to which the toggle is to be written.
nInfoDataState	REFERENCE TO UINT	Reference to a variable to which the InfoData state is to be written.
sAdsAddr	REFERENCE TO AMSADDR	Reference to a variable to which the AdsAddr is to be written. *Notice* **Only available with IoEcAds [⏵ 179].**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.2.2 SetProcessIoStatus()



This method is used to write the EtherCAT status of an I/O element.

Syntax:

```
METHOD SetProcessIoStatus : HRESULT
VAR_INPUT
    bWcState:                BOOL;
    bToggle:                 BOOL;
    nInfoDataState:          UINT;
    sAdsAddr:                AMSADDR;
END_VAR
```

Return value

Name	Type	Description
SetProcessIoStatus	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

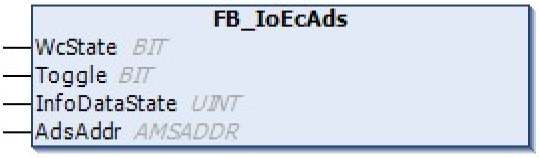
Inputs

Name	Type	Description
bWcState	BOOL	WcState to be written to the input variable.
bToggle	BOOL	Toggle to be written to the input variable.
nInfoDataState	UINT	InfoData State to be written to the input variable.
sAdsAddr	AMSADDR	AdsAddr to be written to the input variable. *Notice* **Only available with IoEcAds [▸ 179].**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.3 IoEcAds



```
FB_IoEcAds
WcState BIT
Toggle BIT
InfoDataState UINT
AdsAddr AMSADDR
```

This class is the basis for EtherCAT-based I/O elements with ADS interface.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_IoEcAds EXTENDS FB_IoEc
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetProcessIoStatus() [▸ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▸ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▸ 174]	Reads the terminal-specific input/output value.

Name	Description
SetProcessValue() [▸ 175]	Writes the terminal-specific input/output value.

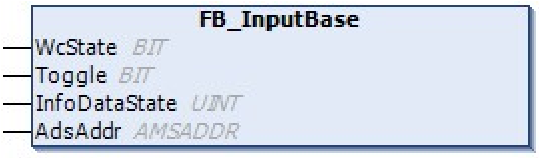
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.4 InputBase



This class is the basis for I/O input elements.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_InputBase EXTENDS FB_IoEcAds
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Scale for scaling the input value.

Methods

Name	Description
Cyclic()	Cycle method

Name	Description
GetChannelState() [▶ 181]	Reads the status word of the input element.
SetChannelState() [▶ 182]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

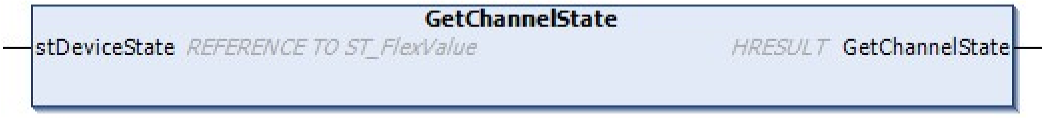
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputBase	Standard interface on FB_InputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.4.1 GetChannelState()



This method is used to read the channel state of an I/O element.

Syntax:

```
METHOD GetChannelState : HRESULT
VAR_INPUT
    stDeviceState:            REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
GetChannelState	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOL NOTFOUND	0x9B00 0710	The transferred reference is invalid.

E_AdsErr	Return value	Cause
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

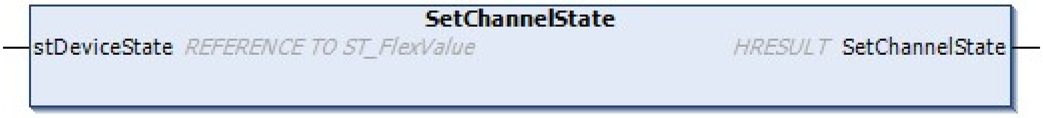
Inputs

Name	Type	Description
stDeviceState	REFERENCE TO ST_FlexValue [▶ 214]	Reference to a variable to which the device state is to be written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.4.2 SetChannelState()



This method is used to write the channel state of an I/O element.

Syntax:

```
METHOD SetChannelState : HRESULT
VAR_INPUT
    stDeviceState:              REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
SetChannelState	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

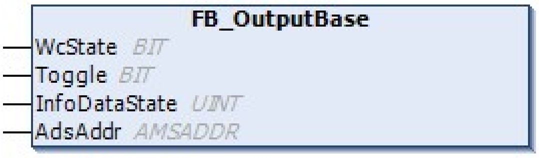
Inputs

Name	Type	Description
stDeviceState	REFERENCE TO ST_FlexValue [▶ 214]	Variable with the state to be written to the input variable.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5 OutputBase



This class is the basis for I/O output elements.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_OutputBase EXTENDS FB_IoEcAds
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 184]	Reads the status word of the output element.
GetControlword() [▶ 185]	Reads the control word of the output element.
GetProcessFeedback() [▶ 186]	Reads the check signal of the output element.
SetChannelState() [▶ 186]	Writes the channel state of the output element.
SetControlword() [▶ 187]	Writes the control word of the output element.

Name	Description
SetProcessFeedback() [▶ 188]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

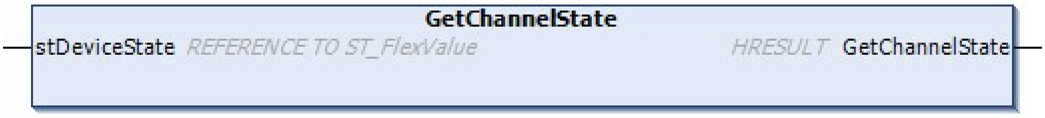
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputBase	Standard interface on FB_OutputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.1.5.1 GetChannelState()



This method is used to read the channel state of an I/O element.

Syntax:

```
METHOD GetChannelState : HRESULT
VAR_INPUT
    stDeviceState:            REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
GetChannelState	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.

E_AdsErr	Return value	Cause
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. Notice **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

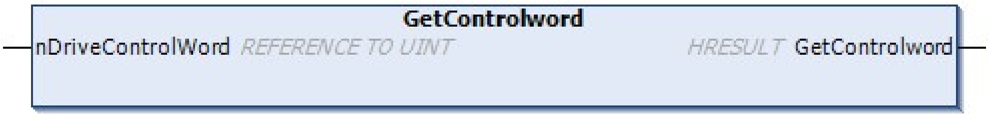
Inputs

Name	Type	Description
stDeviceState	REFERENCE TO ST_FlexValue [▶ 214]	Reference to a variable to which the device state is to be written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5.2 GetControlword()



This method is used to read the control word of an I/O output element.

Syntax:

```
METHOD GetControlword : HRESULT
VAR_INPUT
    nDriveControlWord:        REFERENCE TO UINT;
END_VAR
```

Return value

Name	Type	Description
GetControlword	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. Notice **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

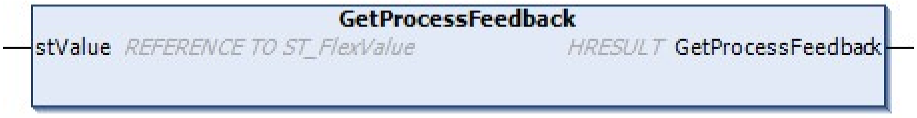
Inputs

Name	Type	Description
nDriveControlWord	REFERENCE TO UINT	Reference to a variable to which the channel state is to be written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5.3 GetProcessFeedback()



This method is used to read the check signal of an I/O output element.

Syntax:

```
METHOD GetProcessFeedback : HRESULT
VAR_INPUT
    stValue:                REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
GetProcessFeedback	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

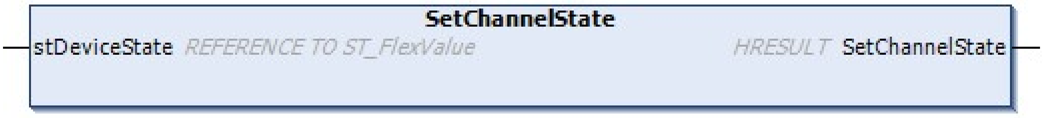
Inputs

Name	Type	Description
stValue	REFERENCE TO ST_FlexValue [▶ 214]	Reference to a variable to which the check value is to be written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5.4 SetChannelState()



This method is used to write the channel state of an I/O element.

Syntax:

```
METHOD SetChannelState : HRESULT
VAR_INPUT
    stDeviceState:             REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
SetChannelState	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. *Notice* **This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

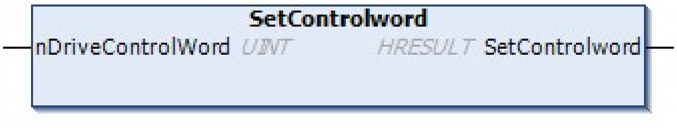
Inputs

Name	Type	Description
stDeviceState	REFERENCE TO ST_FlexValue [▶ 214]	Variable with the state to be written to the input variable.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5.5 SetControlword()



This method is used to write the control word of an I/O output element.

Syntax:

```
METHOD SetControlword : HRESULT
VAR_INPUT
    nDriveControlWord:         UINT;
END_VAR
```

Return value

Name	Type	Description
SetControlword	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. **Notice This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

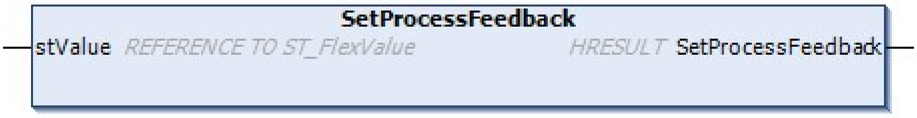
Inputs

Name	Type	Description
nDriveControlWord	UINT	Control word to be written to the input variable.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.1.5.6 SetProcessFeedback()



This method is used to write the check signal of an I/O output element.

Syntax:

```
METHOD SetProcessFeedback : HRESULT
VAR_INPUT
    stValue:                REFERENCE TO ST_FlexValue;
END_VAR
```

Return value

Name	Type	Description
SetProcessFeedback	HRESULT	Return value with feedback on the success of the execution.

The return value of the method is of type HRESULT. The following return values are to be expected.

E_AdsErr	Return value	Cause
DEVICE_ERROR	0x9B00 0700	The I/O element is in error state.
DEVICE_SYMBOLNOTFOUND	0x9B00 0710	The transferred reference is invalid.
DEVICE_INCOMPATIBLE	0x9B00 070E	The variable type of the transferred structure does not match the I/O element.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. **Notice This return value is not marked as an error!**
DEVICE_NOERR	0x1B00 000	Execution was successful.

Inputs

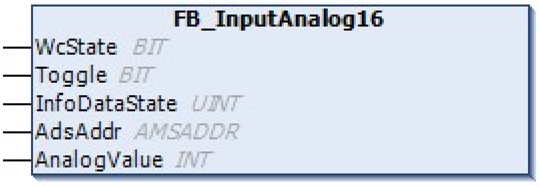
Name	Type	Description
stValue	REFERENCE TO ST_FlexValue [▸ 214]	Variable with the check signal that is to be written to the input variable.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.2 Analog

5.2.1 InputAnalog16



FB_InputAnalog16
WcState BIT
Toggle BIT
InfoDataState UINT
AdsAddr AMSADDR
AnalogValue INT

This class can be used for a 16-bit analog input element.

Syntax:

```
FUNCTION_BLOCK FB_InputAnalog16 EXTENDS FB_InputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AnalogValue	INT	Input	Numerical signed analog input value.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Scale for scaling the input value.

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▸ 181]	Reads the status word of the input element.
SetChannelState() [▸ 182]	Writes the status word of the input element.

Name	Description
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

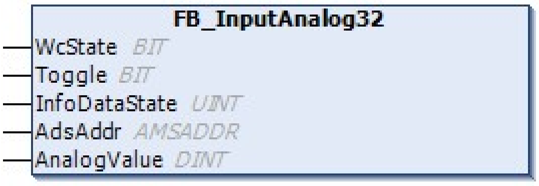
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputAnalog16	Standard interface on FB_InputAnalog16.
I_InputBase	Standard interface on FB_InputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.2.2 InputAnalog32



This class can be used for a 32-bit analog input element.

Syntax:

```
FUNCTION_BLOCK FB_InputAnalog32 EXTENDS FB_InputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AnalogValue	DINT	Input	Numerical signed analog input value.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Scale for scaling the input value.

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 181]	Reads the status word of the input element.
SetChannelState() [▶ 182]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

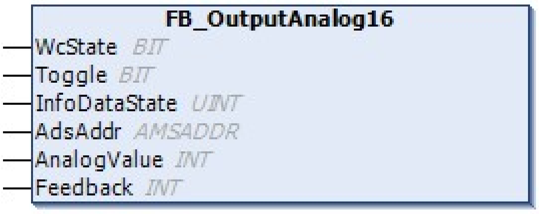
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputAnalog32	Standard interface on FB_InputAnalog32.
I_InputBase	Standard interface on FB_InputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.2.3 OutputAnalog16



This class can be used for a 16-bit analog output element.

Syntax:

```
FUNCTION_BLOCK FB_OutputAnalog16 EXTENDS FB_OutputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
AnalogValue	INT	Output	Numerical signed analog output value.
Feedback	INT	Input	Numerical signed check value for the analog output value.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 181]	Reads the status word of the output element.
GetControlword() [▶ 185]	Reads the control word of the output element.
GetProcessFeedback() [▶ 186]	Reads the check signal of the output element.
SetChannelState() [▶ 182]	Writes the channel state of the output element.
SetControlword() [▶ 187]	Writes the control word of the output element.
SetProcessFeedback() [▶ 188]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

Interfaces

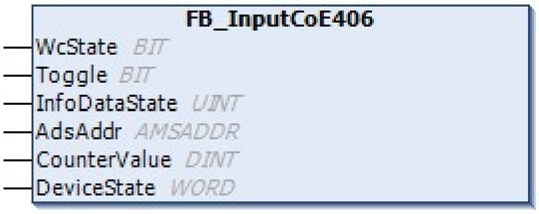
Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputAnalog16	Standard interface on FB_OutputAnalog16.
I_OutputBase	Standard interface on FB_OutputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.3 CoE

5.3.1 InputCoE406



This class can be used for a DS406 based encoder. In this context, the term CoE (CAN over EtherCAT) refers to the transmission of the CiA DS406 protocol via EtherCAT.

Syntax:

```
FUNCTION_BLOCK FB_InputCoE406 EXTENDS FB_InputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
CounterValue	DINT	Input	Counter value of the encoder.
DeviceState	WORD	Input	Status word of the encoder.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Scale for scaling the input value.

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 181]	Reads the status word of the input element.
SetChannelState() [▶ 182]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.

Name	Description
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

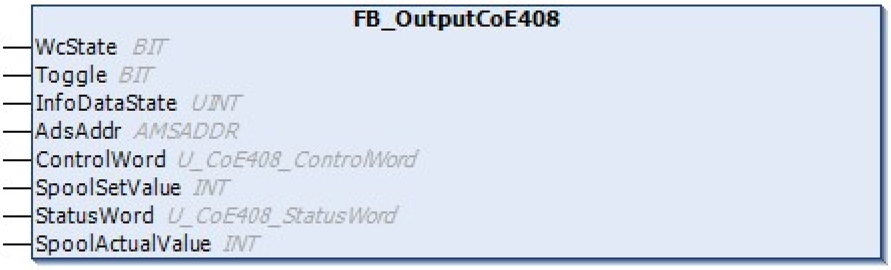
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputCoE406	Standard interface on FB_InputCoE406.
I_InputBase	Standard interface on FB_InputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.3.2 OutputCoE408



FB_OutputCoE408
WcState BIT
Toggle BIT
InfoDataState UINT
AdsAddr AMSADDR
ControlWord U_CoE408_ControlWord
SpoolSetValue INT
StatusWord U_CoE408_StatusWord
SpoolActualValue INT

This class can be used for a DS408 based hydraulic valve. In this context, the term CoE (CAN over EtherCAT) refers to the transmission of the CiA DS408 protocol via EtherCAT.

Syntax:

```
FUNCTION_BLOCK FB_OutputCoE408 EXTENDS FB_OutputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
ControlWord	U_CoE408_ControlWord [▶ 195]	Output	Union for the control word of the valve.
SpoolSetValue	INT	Output	Setpoint for the valve coil.
StatusWord	U_CoE408_StatusWord [▶ 196]	Input	Union for the status word of the valve.
SpoolActualValue	INT	Input	Check signal for the position of the valve coil.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 181]	Reads the status word of the output element.
GetControlword() [▶ 185]	Reads the control word of the output element.
GetProcessFeedback() [▶ 186]	Reads the check signal of the output element.
SetChannelState() [▶ 182]	Writes the channel state of the output element.
SetControlword() [▶ 187]	Writes the control word of the output element.
SetProcessFeedback() [▶ 188]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputCoE408	Standard interface on FB_OutputCoE408.
I_OutputBase	Standard interface on FB_OutputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.3.2.1 U_CoE408_ControlWord

Union for decoding a CoE408 control word

Syntax:

```
TYPE U_CoE408_ControlWord :
UNION
    {attribute 'tc_no_symbol'}
    bits:            ST_CoE408_ControlWord;
    {attribute 'tc_no_symbol'}
    w:               WORD;
END_UNION
END_TYPE
```

Values

Name	Type	Description
bits	ST_CoE408_ControlWord [▶ 196]	Breakdown of the word into bits.
w	WORD	Word as a complete data type.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.3.2.2 ST_CoE408_ControlWord

Structure for breaking down a CoE408 control word.

Syntax:

```
TYPE ST_CoE408_ControlWord :
STRUCT
    Disable:                BIT;
    HoldEnable:             BIT;
    DeviceModeActiveEnable: BIT;
    Reset:                  BIT;
END_STRUCT
END_TYPE
```

Bits

Name	Offset	Description
Disable	0	Disable valve.
HoldEnable	1	Enable valve hold mode.
DeviceModeActiveEnable	2	Generally enable the valve.
Reset	3	Reset error state.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.3.2.3 U_CoE408_StatusWord

Union for decoding a CoE408 status word.

Syntax:

```
TYPE U_CoE408_StatusWord :
UNION
    {attribute 'tc_no_symbol'}
    bits:           ST_CoE408_StatusWord;
    {attribute 'tc_no_symbol'}
    w:              WORD;
END_UNION
END_TYPE
```

Values

Name	Type	Description
bits	ST_CoE408_StatusWord [▶ 197]	Breakdown of the word into bits.
w	WORD	Word as a complete data type.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.3.2.4 ST_CoE408_StatusWord

Structure for the breakdown of a CoE408 status word.

Syntax:

```
TYPE ST_CoE408_StatusWord :
STRUCT
    Disabled:                BIT;
    HoldEnabled:             BIT;
    DeviceModeActiveEnabled: BIT;
    Ready:                   BIT;
END_STRUCT
END_TYPE
```

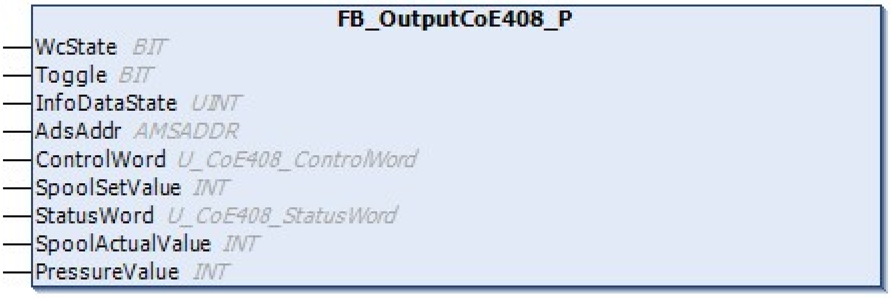
Bits

Name	Offset	Description
Disabled	0	Valve is disabled.
HoldEnabled	1	Valve hold mode is enabled.
DeviceModeActiveEnabled	2	Valve is generally enabled.
Ready	3	Device is ready to operate.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.3.3 OutputCoE408_P



This class is an extension of the CoE408-based valve with integrated pressure measurement.

Syntax:

```
FUNCTION_BLOCK FB_OutputCoE408_P EXTENDS FB_OutputCoE408
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
PressureValue	INT	Input	Actual pressure value of the internal valve pressure measurement.

Name	Type	Direction	Description
ControlWord	U_CoE408_ControlWord [▸ 195]	Output	Union for the control word of the valve.
SpoolSetValue	INT	Output	Setpoint for the valve coil.
StatusWord	U_CoE408_StatusWord [▸ 196]	Input	Union for the status word of the valve.
SpoolActualValue	INT	Input	Check signal for the position of the valve coil.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
Pressure	I_InputBase	Get	Interface for integrated pressure measurement

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▸ 181]	Reads the status word of the output element.
GetControlword() [▸ 185]	Reads the control word of the output element.
GetProcessFeedback() [▸ 186]	Reads the check signal of the output element.
SetChannelState() [▸ 182]	Writes the channel state of the output element.
SetControlword() [▸ 187]	Writes the control word of the output element.
SetProcessFeedback() [▸ 188]	Writes the check signal of the output element.
GetProcessIoStatus() [▸ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▸ 175]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▸ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▸ 175]	Writes the terminal-specific input/output value.

Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputCoE408	Standard interface on FB_OutputCoE408.
I_OutputBase	Standard interface on FB_OutputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.

Type	Description
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.4 Digital

5.4.1 IoDigital

FB_IoDigital

This class is the basis for digital I/O elements.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_IoDigital EXTENDS FB_IoBase
```

Properties

Name	Type	Access	Description
DelayOff	LREAL	Get, Set	Delay [s] of a falling edge of the digital signal.
DelayOn	LREAL	Get, Set	Delay [s] of a rising edge of the digital signal.
In	BOOL	Get, Set	Digital input value.
Invert	BOOL	Get, Set	Inverts the interpretation of the input value.
Q	BOOL	Get	Interpreted input value.
QF	BOOL	Get	Falling edge from Q.
QR	BOOL	Get	Rising edge from Q.

Methods

Name	Description
Cyclic()	Cycle method
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

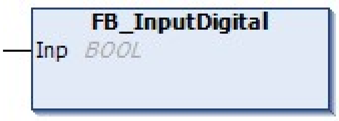
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.4.2 InputDigital



This class is used for digital input signals.

Syntax:

```
FUNCTION_BLOCK FB_InputDigital EXTENDS FB_IoDigital
```

I/O variables

Name	Type	Direction	Description
Inp	BOOL	Input	Digital input signal

Properties

Name	Type	Access	Description
DelayOff	LREAL	Get, Set	Delay [s] of a falling edge of the digital signal.
DelayOn	LREAL	Get, Set	Delay [s] of a rising edge of the digital signal.
In	BOOL	Get, Set	Digital input value.
Invert	BOOL	Get, Set	Inverts the interpretation of the input value.
Q	BOOL	Get	Interpreted input value.
QF	BOOL	Get	Falling edge from Q.
QR	BOOL	Get	Rising edge from Q.

Methods

Name	Description
Cyclic()	Cycle method
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

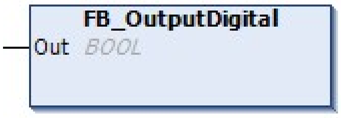
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputBase	Compatible standard interface for inputs. **This class implements hidden placeholders for compatibility.**
I_InputDigital	Standard interface on FB_InputDigital.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.4.3 OutputDigital



This class is used for digital output signals.

Syntax:

```
FUNCTION_BLOCK FB_OutputDigital EXTENDS FB_IoDigital
```

I/O variables

Name	Type	Direction	Description
Out	BOOL	Output	Digital output signal.

Properties

Name	Type	Access	Description
DelayOff	LREAL	Get, Set	Delay [s] of a falling edge of the digital signal.
DelayOn	LREAL	Get, Set	Delay [s] of a rising edge of the digital signal.
In	BOOL	Get, Set	Digital input value.
Invert	BOOL	Get, Set	Inverts the interpretation of the input value.
Q	BOOL	Get	Interpreted input value.
QF	BOOL	Get	Falling edge from Q.
QR	BOOL	Get	Rising edge from Q.

Methods

Name	Description
Cyclic()	Cycle method
GetProcessValue() [▶ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 175]	Writes the terminal-specific input/output value.

Interfaces

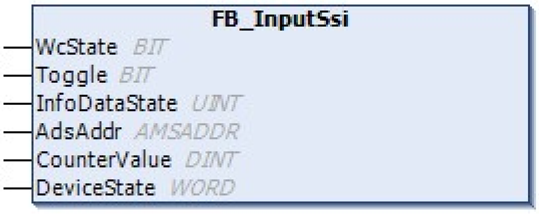
Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputBase	Compatible standard interface for outputs. **This class implements hidden placeholders for compatibility.**
I_OutputDigital	Standard interface on FB_OutputDigital.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.5 Serial

5.5.1 InputSsi



FB_InputSsi
WcState BIT
Toggle BIT
InfoDataState UINT
AdsAddr AMSADDR
CounterValue DINT
DeviceState WORD

This class can be used for an SSI based encoder.

Syntax:

```
FUNCTION_BLOCK FB_InputSsi EXTENDS FB_InputBase
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
CounterValue	DINT	Input	Counter value of the encoder.
DeviceState	WORD	Input	Status word of the encoder.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Bit switching with the update rate of the mapping.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Scale for scaling the input value.

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 181]	Reads the status word of the input element.
SetChannelState() [▶ 182]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 174]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 175]	Writes the EtherCAT status of the input/output element.

Name	Description
GetProcessValue() [▸ 174]	Reads the terminal-specific input/output value.
SetProcessValue() [▸ 175]	Writes the terminal-specific input/output value.

Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputSsi	Standard interface on FB_InputSsi.
I_InputBase	Standard interface on FB_InputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6 Device (inverter)

5.6.1 InvDeviceBase

This class is the basis for inverter I/O elements.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_InvDeviceBase
```

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.

Name	Type	Access	Description
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Direction of the signals

The properties essentially form the interface between the inverter and the inverter axis.

- Methods without a setter are usually feedback signals from the inverter to the axis.
- Methods with setters are command signals from the axis to the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

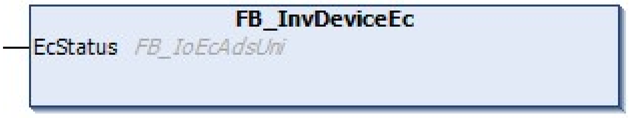
Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.2 InvDeviceEc



This class is the basis for inverters with EtherCAT status.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_InvDeviceEc EXTENDS FB_InvDeviceBase
```

I/O variables

Name	Type	Direction	Description
EcStatus	FB_IoEcAdsUni [▶ 179]	Input	EtherCAT status of the inverter.

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

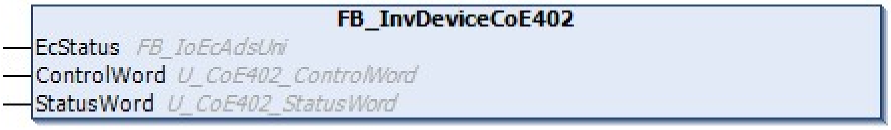
Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.3 InvDeviceCoE402


FB_InvDeviceCoE402
EcStatus FB_IoEcAdsUni
ControlWord U_CoE402_ControlWord
StatusWord U_CoE402_StatusWord

This class is the basis for DS402-based inverters. In this context, the term CoE (CAN over EtherCAT) refers to the transmission of the CiA DS402 protocol via EtherCAT.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_InvDeviceCoE402 EXTENDS FB_InvDeviceEc
```

I/O variables

Name	Type	Direction	Description
EcStatus	FB_IoEcAdsUni [▶ 179]	Input	EtherCAT status of the inverter.
ControlWord	U_CoE402_ControlWord [▶ 207]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 208]	Input	Status word of the inverter.

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
TolStandstill	LREAL	Get, Set	Tolerance with which the current velocity is interpreted as standstill.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InverterCoE402	Standard interface on FB_InvDeviceCoE402.
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.3.1 U_CoE402_ControlWord

Union for decoding a CoE402 control word.

Syntax:

```
TYPE U_CoE402_ControlWord :
UNION
    {attribute 'tc_no_symbol'}
    bits:            ST_CoE402_ControlWord;
    {attribute 'tc_no_symbol'}
    w:               WORD;
END_UNION
END_TYPE
```

Values

Name	Type	Description
bits	ST_CoE402_ControlWord [▶ 207]	Breakdown of the word into bits.
w	WORD	Word as a complete data type.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.3.2 ST_CoE402_ControlWord

Structure for breaking down a CoE402 control word.

Syntax:

```
TYPE ST_CoE402_ControlWord :
STRUCT
    SwitchOn:            BIT;
    EnableVoltage:       BIT;
    QuickStopDisable:    BIT;
    EnableOperation:     BIT;
    Reserve_4:           BIT;
    Reserve_5:           BIT;
    Reserve_6:           BIT;
    Reset:               BIT;
    Halt:                BIT;
END_STRUCT
END_TYPE
```

Bits

Name	Offset	Description
SwitchOn	0	Switch on the device.
EnableVoltage	1	Enable supply.
QuickStopDisable	2	Disable quick stop.
EnableOperation	3	Enable power.
Reset	7	Reset error state.
Halt	8	Stop the inverter.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.3.3 U_CoE402_StatusWord

Union for decoding a CoE402 status word.

Syntax:

```
TYPE U_CoE402_StatusWord :
UNION
    {attribute 'tc_no_symbol'}
    bits:             ST_CoE402_StatusWord;
    {attribute 'tc_no_symbol'}
    w:                WORD;
END_UNION
END_TYPE
```

Values

Name	Type	Description
bits	ST_CoE402_StatusWord [▶ 208]	Breakdown of the word into bits.
w	WORD	Word as a complete data type.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.3.4 ST_CoE402_StatusWord

Structure for the breakdown of a CoE402 status word.

Syntax:

```
TYPE ST_CoE402_StatusWord :
STRUCT
    ReadyToSwitchOn:    BIT;
    SwitchedOn:         BIT;
    OperationEnabled:   BIT;
    Fault:              BIT;
    VoltageEnabled:     BIT;
    QuickStopInactive:  BIT;
    SwitchOnDisabled:   BIT;
    Warning:            BIT;
    Reserve_8:          BIT;
    Remote:             BIT;
END_STRUCT
END_TYPE
```

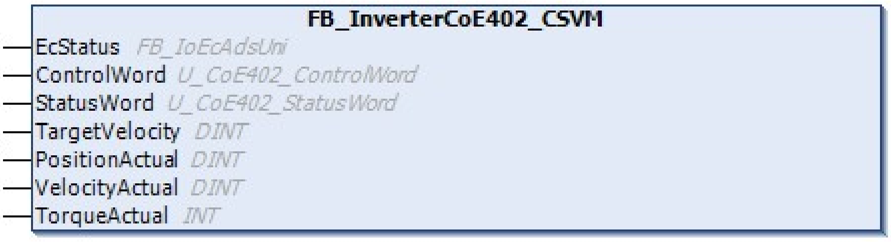
Bits

Name	Offset	Description
ReadyToSwitchOn	0	Inverter is ready to be switched on.
SwitchedOn	1	Inverter is switched on.
OperationEnabled	2	Power is enabled.
Fault	3	Inverter is in error state.
VoltageEnabled	4	Supply is enabled.
QuickStopInactive	5	Quick stop is disabled.
SwitchOnDisabled	6	Inverter is blocked from being switched on.
Warning	7	Inverter reports a warning.
Remote	9	Inverter runs in remote mode and can be controlled via the interface.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.4 InverterCoE402_CSVM



This class can be used for DS402 inverters in **C**yclic**S**ynchronous**V**elocity**M**ode.

Syntax:

```
FUNCTION_BLOCK FB_InverterCoE402_CSVM EXTENDS FB_InvDeviceCoE402
```

I/O variables

Name	Type	Direction	Description
EcStatus	FB_IoEcAdsUni [▶ 179]	Input	EtherCAT status of the inverter.
TargetVelocity	DINT	Output	Target velocity of the inverter.
PositionActual	DINT	Input	Actual position of the inverter.
VelocityActual	DINT	Input	Actual velocity of the inverter.
TorqueActual	INT	Input	Current torque.

Name	Type	Direction	Description
ControlWord	U_CoE402_ControlWord [▶ 207]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 208]	Input	Status word of the inverter.

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
OutputFactor	LREAl	Get, Set	Multiplier for the velocity output.
TorqueFactor	LREAL	Get, Set	Multiplier for interpretation of the actual torque.
TolStandstill	LREAL	Get, Set	Tolerance with which the current velocity is interpreted as standstill.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

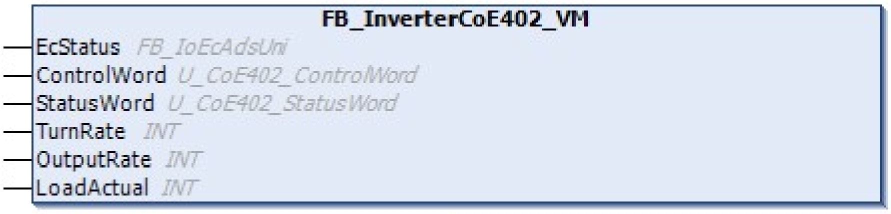
Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InverterCoE402	Standard interface on FB_InvDeviceCoE402.
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.5 InverterCoE402_VM



```
FB_InverterCoE402_VM
EcStatus FB_IoEcAdsUni
ControlWord U_CoE402_ControlWord
StatusWord U_CoE402_StatusWord
TurnRate INT
OutputRate INT
LoadActual INT
```

This class can be used for DS402 inverters in **V**elocity**M**ode.

Syntax:

```
FUNCTION_BLOCK FB_InverterCoE402_VM EXTENDS FB_InvDeviceCoE402
```

I/O variables

Name	Type	Direction	Description
EcStatus	FB_IoEcAdsUni [▶ 179]	Input	EtherCAT status of the inverter.
TurnRate	INT	Output	Set speed for the inverter.
OutputRate	INT	Input	Actual speed of the inverter.
LoadActual	INT	Input	Current load of the inverter motor.
ControlWord	U_CoE402_ControlWord [▶ 207]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 208]	Input	Status word of the inverter.

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
TolStandstill	LREAL	Get, Set	Tolerance with which the current velocity is interpreted as standstill.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

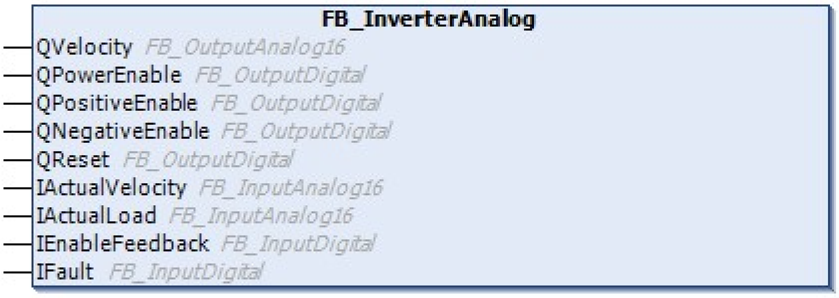
Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InverterCoE402	Standard interface on FB_InvDeviceCoE402.
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.6.6 InverterAnalog


FB_InverterAnalog
QVelocity FB_OutputAnalog16
QPowerEnable FB_OutputDigital
QPositiveEnable FB_OutputDigital
QNegativeEnable FB_OutputDigital
QReset FB_OutputDigital
IActualVelocity FB_InputAnalog16
IActualLoad FB_InputAnalog16
IEnableFeedback FB_InputDigital
IFault FB_InputDigital

This class is used for inverters with an analog interface.

Syntax:

```
FUNCTION_BLOCK FB_InverterAnalog EXTENDS FB_InvDeviceBase
```

I/O variables

Name	Type	Direction	Description
QVelocity	FB_OutputAnalog16 [▶ 191]	Output	Velocity output
QPowerEnable	FB_OutputDigital [▶ 201]	Output	Power enable
QPositiveEnable	FB_OutputDigital [▶ 201]	Output	Positive directional enable
QNegativeEnable	FB_OutputDigital [▶ 201]	Output	Negative directional enable
QReset	FB_OutputDigital [▶ 201]	Output	Reset the inverter error state.
IActualVelocity	FB_InputAnalog16 [▶ 189]	Input	Message of the current velocity.

Name	Type	Direction	Description
IActualLoad	FB_InputAnalog16 [▸ 189]	Input	Message of the current load.
IEnableFeedback	FB_InputDigital [▸ 200]	Input	Message of power enable.
IFault	FB_InputDigital [▸ 200]	Input	Inverter error

Properties

Name	Type	Access	Description
ActVelocity	LREAL	Get	Current velocity of the inverter motor.
TolStandstill	LREAL	Get, Set	Tolerance with which the current velocity is interpreted as standstill.
Connected	BOOL	Get	[INTERNAL] Class has been connected to an axis.
EnableN	BOOL	Get, Set	Negative directional enable of the axis.
EnableP	BOOL	Get, Set	Positive directional enable of the axis.
EnablePower	BOOL	Get, Set	Power enable of the axis.
EnableSupply	BOOL	Get, Set	Supply enable of the axis.
Error	BOOL	Get	Inverter is in error state.
ErrorID	UDINT	Get	Error code of the inverter error.
Load	LREAL	Get	Moad of the inverter.
PowerEnabled	BOOL	Get	Feedback of the power enable of the inverter.
Reset	BOOL	Get, Set	Reset signal of the axis.
Reseted	BOOL	Get, Set	Feedback of the successful reset.
SetVelocity	LREAL	Get, Set	Set velocity of the axis.
Stop	BOOL	Get, Set	Stop signal of the axis for possible stop functions of the inverter.
Stopped	BOOL	Get	Feedback of the successful stop of the inverter.
SupplyEnabled	BOOL	Get	Feedback of the supply enable of the inverter.

Direction of the signals

The properties essentially form the interface between the inverter and the inverter axis.

- Methods without a setter are usually feedback signals from the inverter to the axis.
- Methods with setters are command signals from the axis to the inverter.

Methods

Name	Description
Cyclic()	Cycle method
Connect()	Connects the inverter device to the axis.
GoError()	Simulates an error of the inverter device.

Interfaces

Type	Description
I_InvDeviceBase	Standard interface on FB_InvDeviceBase **Defines the interface to the axis.**
I_InverterAnalog	Standard interface on FB_InverterAnalog.
I_InvDeviceBaseDev	Extended interface on FB_InvDeviceBase.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.7 ST_FlexValue

Structure for transferring information without specifying the data type.

Syntax:

```
TYPE ST_FlexValue :
STRUCT
    VarType:    E_FlexValue;    // 2 byte
    Value:      U_FlexValue;    // 8 byte
END_STRUCT
END_TYPE
```

Values

Name	Type	Description
VarType	E_FlexValue [▶ 214]	Enumeration for identifying the value to be read.
Value	U_FlexValue [▶ 215]	Union to which the value is written.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

5.7.1 E_FlexValue

Enumeration to identify a value of the type `U_FlexValue`.

Syntax:

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_FlexValue :
(
    eILLEGAL := -1,
    //
    // 1 bit
    //
    eBOOL := 0,
    //
    // 8 bit
    //
    eSINT := 8,
    eUSINT,
    eBYTE,
    //
    // 16 bit
    //
    eINT := 16,
    eUINT,
    eWORD,
    //
    // 32 bit
    //
    eDINT := 32,
    eUDINT,
    eDWORD,
    eREAL,
    //
    // 64 bit
    //
    eLINT := 64,
    eULINT,
    eLWORD,
    eLREAL
);
END_TYPE
```

Values

Name	Number	Bit width
eILLEGAL	-1	/
eBOOL	0	1 bit
eSINT	8	8 bits
eUSINT	9	
eBYTE	10	
eINT	16	16 bits
eUINT	17	
eWORD	18	
eDINT	32	32 bits
eUDINT	33	
eDWORD	34	
eREAL	35	
eLINT	64	64 bits
eULINT	65	
eLWORD	66	
eLREAL	67	

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

5.7.2 U_FlexValue

Union to hold the information in the data type `ST_FlexValue`.

Syntax:

```
TYPE U_FlexValue :
UNION
    b:          BOOL;
    //
    si:         SINT;
    usi:        USINT;
    bt:         BYTE;
    //
    i:          INT;
    ui:         UINT;
    w:          WORD;
    //
    di:         DINT;
    udi:        UDINT;
    dw:         DWORD;
    re:         REAL;
    //
    li:         LINT;
    uli:        ULINT;
    lw:         LWORD;
    lr:         LREAL;
END_UNION
END_TYPE
```

Values

Name	Type
b	BOOL
si	SINT
usi	USINT
bt	BYTE
i	INT
ui	UINT
w	WORD
di	DINT
udi	UDINT
dw	DWORD
re	REAL
li	LINT
uli	ULINT
lw	LWORD
lr	LREAL

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

6 MachineData

6.1 FB_MdContainer

This class is able to write data from a list of components to a file. The file format is binary and secured via a CRC checksum. This automatically makes it more difficult for unauthorized users to manipulate this file.

Syntax:

```
FUNCTION_BLOCK FB_MdContainer
```

Properties

Name	Type	Access	Description
Busy	BOOL	Get	The container is busy reading or writing.
Done	BOOL	Get	Reading or writing has been successfully completed.
Error	BOOL	Get	An error has occurred while reading or writing.
FilePath	STRING	Get, Set	Path under which the machine data is to be saved.
Load	BOOL	Get, Set	Command to read the file.
Name	STRING	Get, Set	Name of the machine data container.
Save	BOOL	Get, Set	Command for writing the file.

Methods

Name	Description
AddComponent()	Adds another component to the container.
ClearChain()	Deletes the list of all components.
Cyclic()	Cycle method

Interfaces

Type	Description
I_MdContainer	Standard interface on FB_MdContainer.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

6.2 FB_MdComponent

The class represents the basis of a machine data component.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_MdComponent
```

Properties

Name	Type	Access	Description
ComponentIndex	USINT	Get, Set	Index of the instance of the component. Identifies a component if there are several components of the same type in a container.
ComponentType	USINT	Get, Set	Index of the component type.
Connected	BOOL	Get	Component is connected to the container.
CrcChecked	BOOL	Get, Set	CRC checksum has been checked.
EndMark	BOOL	Get, Set	Is set when the component has accepted the last parameter.
MdNextComponent	I_MdComponent	Get	Interface to the next component.

Methods

Name	Description
AddNextComponent()	Adds the subsequent component to the component (list).
ClearChain()	Resets the interface to the next component and calls its `ClearChain()` method.

Event-driven methods (callback methods)

Name	Description
MdNextParameter() [▶ 218]	Gets parameters from the application.
MdSetParameter() [▶ 219]	Passes parameters to the application.

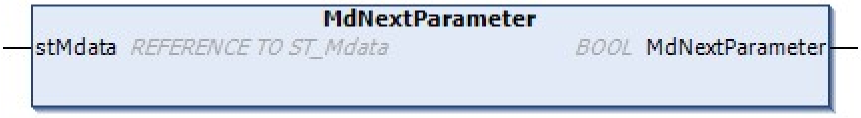
Interfaces

Type	Description
I_MdContainer	Standard interface on FB_MdContainer.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

6.2.1 MdNextParameter()



This method is used to get the parameters from the application in order to write them to the file.

Syntax:

```
METHOD ABSTRACT MdNextParameter : BOOL
VAR_INPUT
    stMdata:    REFERENCE TO ST_Mdata;
END_VAR
```

Return value

Name	Type	Description
MdNextParameter	BOOL	TRUE = `stMdata` contains a valid parameter. FALSE = No further data available.

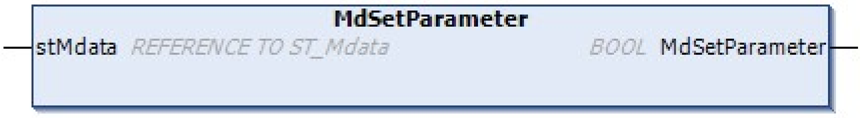
Inputs

Name	Type	Description
stMdata	REFERENCE TO ST_Mdata [▶ 220]	Reference to a variable to which the component should write the information of the parameter.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

6.2.2 MdSetParameter()



This method is used to set the parameters of the application that are read from the file.

Syntax:

```
METHOD ABSTRACT MdSetParameter : BOOL
VAR_INPUT
    stMdata:    REFERENCE TO ST_Mdata;
END_VAR
```

Return value

Name	Type	Description
MdSetParameter	BOOL	TRUE = Parameter is applied. FALSE = Parameter cannot be used.

Inputs

Name	Type	Description
stMdata	REFERENCE TO ST_Mdata [▶ 220]	Reference to a variable that contains the parameter loaded from the file.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

6.3 ST_Mdata

This structure defines a parameter data set for the machine data.

Syntax:

```
TYPE ST_Mdata :
STRUCT
    nParamIndex:     UDINT;            //  4 byte
    nComponentType:  USINT;            //  1 byte
    nComponentIndex: USINT;            //  1 byte
    stFlex:          ST_FlexValue;     // 10 byte
END_STRUCT                             //-------------
END_TYPE                               // 16 byte
```

Values

Name	Type	Description
nParamIndex	UDINT	Individual index of the written parameter.
nComponentType	USINT	Components Type [▸ 217] of the written parameter.
nComponentIndex	USINT	Components Index [▸ 217] of the written parameter.
stFlex	ST_FlexValue [▸ 214]	Data type-independent parameter.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7 Utilities

7.1 Filter

Filters are FBs that may be applied to reduce noise in actual values like axis position or velocity, pressures, or forces. To be compatible with the intended use they must implement at least the pre-defined INTERFACE I_Filter. Tc3 Plastic libraries supply a range of basic filter FBs. Find details below.

7.1.1 FB_FilterBase

Base class for filter function blocks for filtering floating point numbers.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_FilterBase EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
Output	LREAL	Get	The filtered value.

Methods

Name	Description
Cyclic	Cycle method
CyclicUpdate [▸ 221]	Cyclic transfer of a new input value.

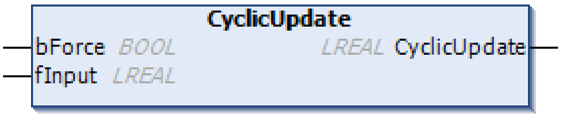
Interfaces

Type	Description
I_Filter	Basic interface for filter function blocks.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.1.1.1 CyclicUpdate



Cyclic transfer of a new value.

Syntax:

```
METHOD CyclicUpdate: LREAL
VAR_INPUT
    bForce:     BOOL;
    fInput:     LREAL;
END_VAR
```

Inputs

Name	Type	Description
bForce	BOOL	The filtered value is updated with the input value regardless of filter parameters if this input is TRUE. **Notice A TRUE overwrites the parameterized filter behavior!**
fInput	LREAL	New input value

Outputs

Name	Type	Description
CyclicUpdate	LREAL	The filtered value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.1.2 FB_FilterPt1

This class, derived from `FB_FilterBase`, implements a filter of type PT1.

Syntax:

```
FUNCTION_BLOCK FB_FilterPt1 EXTENDS FB_FilterBase
```

Initialization parameters of the FB_init constructor

Name	Type	Description
fTau	LREAL	Filter time constant [s]

Properties

Name	Type	Access	Description
Output	LREAL	Get	The filtered value.
Tau	LREAL	Get, Set	The filter time constant [s]

Methods

Name	Description
Cyclic	Cycle method
CyclicUpdate [▸ 221]	Cyclic transfer of a new input value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.1.3 FB_FilterSlewRateLimit

This class derived from `FB_FilterBase` implements a filter that limits the rate of rise (i.e. the ramp rate).

Syntax:

```
FUNCTION_BLOCK FB_FilterSlewRateLimit EXTENDS FB_FilterBase
```

Initialization parameters of the FB_init constructor

Name	Type	Description
fChangeRate	LREAL	The maximum rate of change of the filtered value.

Properties

Name	Type	Access	Description
Output	LREAL	Get	The filtered value.
ChangeRateLimit	LREAL	Get, Set	The maximum rate of change of the filtered value.

Methods

Name	Description
Cyclic	Cycle method
CyclicUpdate [▸ 221]	Cyclic transfer of a new input value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.1.4 FB_FilterSrlPt1

This class derived from `FB_FilterBase` implements a PT1 filter and a ramp limiting filter.

Syntax:

```
FUNCTION_BLOCK FB_FilterSrlPt1 EXTENDS FB_FilterBase
```

Properties

Name	Type	Access	Description
Output	LREAL	Get	The filtered value.
Tau	LREAL	Get, Set	The filter time constant [s]
ChangeRateLimit	LREAL	Get, Set	The maximum rate of change of the filtered value.

Methods

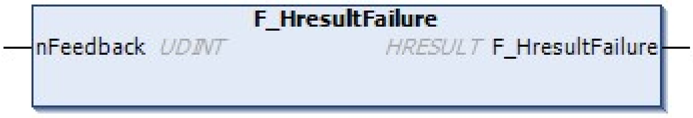
Name	Description
Cyclic	Cycle method
CyclicUpdate [▶ 221]	Cyclic transfer of a new input value.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.2 Hresult

7.2.1 F_HresultFailure



This method is used to generate a `HRESULT` value that is interpreted as an error.

Syntax:

```
FUNCTION F_HresultFailure : HRESULT
VAR_INPUT
    nFeedback:          UDINT;
END_VAR
```

Return value

Name	Type	Description
F_HresultFailure	HRESULT	Generated `HRESULT` value.

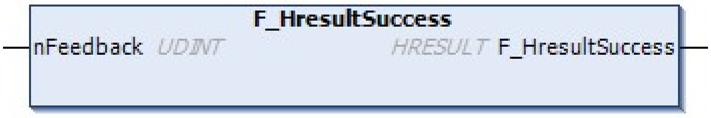
Inputs

Name	Type	Description
nFeedback	UDINT	Error code to be integrated into the HRESULT.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.2.2 F_HresultSuccess



This method is used to generate a HRESULT value that is interpreted as a success.

Syntax:

```
FUNCTION F_HresultSuccess : HRESULT
VAR_INPUT
    nFeedback:            UDINT;
END_VAR
```

Return value

Name	Type	Description
F_HresultSuccess	HRESULT	Generated HRESULT value.

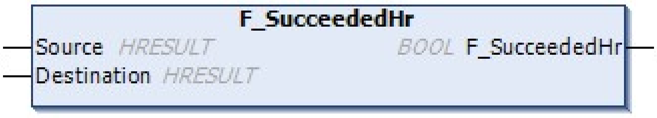
Inputs

Name	Type	Description
nFeedback	UDINT	Error code to be integrated into the HRESULT.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.2.3 F_SucceededHr



This method checks a HRESULT value for success and copies the result to a transferred variable if the check fails.

Syntax:

```
FUNCTION F_SucceededHr : BOOL
VAR_INPUT
    Source:                 HRESULT;
END_VAR
VAR_IN_OUT
    Destination:            HRESULT;
END_VAR
```

Inputs/outputs

Name	Type	Description
Destination	HRESULT	Variable to which the `HRESULT` value is to be copied in the event of an error.

Inputs

Name	Type	Description
Source	HRESULT	Error code to be checked.

Return value

Name	Type	Description
F_SucceededHr	BOOL	TRUE if the check was successful.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.3 Messages

7.3.1 FB_Message

This class is able to send messages via the EventLogger.

Syntax:

```
FUNCTION_BLOCK FB_Message EXTENDS FB_TcMessage
```

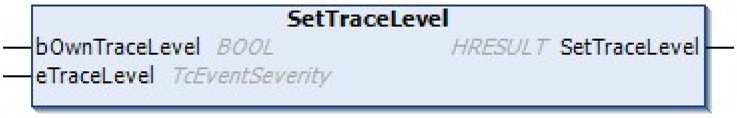
Methods

Name	Description
SetTraceLevel() [▶ 226]	Fixes the severity of the events thrown by this instance.
Reset()	Creates a `SUCCESS_EVENT`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.3.1.1 SetTraceLevel()



This method fixes the severity of the events thrown by this instance.

Syntax:

```
METHOD SetTraceLevel : HRESULT
VAR_INPUT
    bOwnTraceLevel:  BOOL;               // activate internal trace level
    eTraceLevel:     TcEventSeverity;    // set internal trace level
END_VAR
```

Return value

Name	Type	Description
SetTraceLevel	HRESULT	Return value with feedback on the success of the execution.

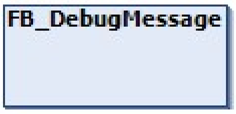
Inputs

Name	Type	Description
bOwnTraceLevel	BOOL	Activates the fixed severity level.
eTraceLevel	TcEventSeverity	Severity level

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.3.2 FB_XyzMessage



These classes take into account various static parameters of the library in order to report messages in the EventLogger.

Syntax:

```
FUNCTION_BLOCK FB_XyzMessage EXTENDS FB_Message
```

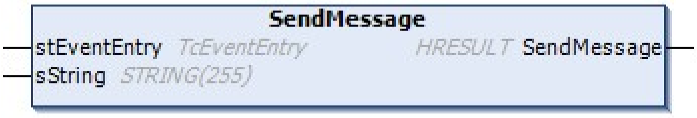
Methods

Name	Description
SetTraceLevel() [▶ 226]	Fixes the severity of the events thrown by this instance.
SendMessage() [▶ 228]	Generates a message in the EventLogger.
Reset()	Creates a `SUCCESS_EVENT`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.3.2.1 SendMessage()



This method sends a message to the EventLogger.

Syntax:

```
METHOD SendMessage : HRESULT
VAR_INPUT
    stEventEntry:       TcEventEntry;       // TcEventEntry to be created and send
    sString:            STRING(255);        // additional STRING attribute
END_VAR
```

Return value

Name	Type	Description
SendMessage	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
stEventEntry	TcEventEntry	Message to be generated and sent.
sString	STRING(255)	Optional additional information as text.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.3.3 FB_MessageBase

This class is the basic scope of the EventLogger interface of most axes and core functions.

Class is Abstract

Since the class is defined as `ABSTRACT`, the class cannot be instantiated and must be implemented using inheritance.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_MessageBase
```

Properties

Name	Type	Access	Description
Error	BOOL	Get	Error state
ResultMessage	I_TcMessage	Get	Access to the internal instance of the error messages.

Methods

Name	Description
DoReset(bExecute)	Resets the error.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4 Simulation

A simulation requires the emulation of a process that is influenced by a controller. This is realized by sending control values to output devices that control actuators. The state of the process is detected by sensors, which respond by sending signals to input devices.

Typical examples for these devices are:

- servo drives and valves with direct fieldbus interface
- rotary encoders and sensors with direct fieldbus interface
- rotary encoders and sensors via analog input terminals
- servo drives and valves via analog output terminals

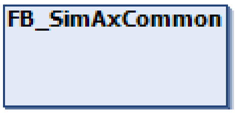
The most efficient way to connect a process simulation with a control implementation such as an application based on TC3 Plastic Functions is to replace the real I/O devices with compatible container objects.

In such a concept, the simulation must perform the following tasks:

- emulate the functionality of the output device
- simulate the behavior of the actuator
- simulate the effect on the process
- simulate the behavior of the sensor
- emulate the functionality of the input device

7.4.1 General simulation components

7.4.1.1 FB_SimAxCommon



This class is used as a common platform for carrying out various types of axis simulations. It is not intended to instantiate objects.

Syntax:

```
FUNCTION_BLOCK FB_SimAxCommon
```

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	This runtime value provides information about the simulated axis position. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**
AbsolutSwitch	BOOL	Get	This runtime value provides information about the simulated feedback signal as defined by AbsSwitchHighSelect.
AbsSwitchHighEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=TRUE: AbsolutSwitch will be TRUE if the simulated actuator position is >= AbsSwitchHighEnd. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**
AbsSwitchHighSelect	BOOL	Get, Set	This parameter defines the active AbsolutSwitch. A TRUE will select AbsSwitchHighEnd while a FALSE makes the AbsSwitchLowEnd the active AbsolutSwitch.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=FALSE: AbsolutSwitch will be TRUE if the simulated actuator position is <= AbsSwitchLowEnd. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**
CycleTime	LREAL	Get, Set	This parameter must be initialized with the update calling cycle of the Cyclic() method and will be forwarded to any sub-component if required. It is used to define any time behavior.
EncoderInterpolation	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor denominator • Hydraulics library: encoder, inc. Interpolation
EncoderNoiseLevel	LREAL	Get, Set	The simulated axis offers the possibility to disturb the reported position with a pseudo-random white noise. This is realized by a FB_Noise() function block. The parameter EncoderNoiseLevel is forwarded as SetWhiteNoiseLevel.
EncoderWeighting	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor numerator • Hydraulics library: encoder, weighting factor
EncoderZeroShift	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology:

Name	Type	Access	Description
			• NC: rotary encoder, parameter, position bias • Hydraulics library: encoder, zero offset
HighSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value less than or equal to this parameter.
HighSideEndswitch	LREAL	Get, Set	This parameter defines the UpperEndSwitch threshold that becomes TRUE when the simulated position of the actuator is >= this parameter.
HighSideSpringLengt	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the upper side of the actuator stroke.
LowerEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes TRUE if the position of the simulated actuator is >= HighSideEndswitch.
LowSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value above or equal to this parameter.
LowSideEndswitch	LREAL	Get, Set	This parameter defines the LowerEndSwitch threshold that becomes TRUE when the simulated position of the actuator is <= this parameter.
LowSideSpringLength	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the lower side of the actuator stroke.
MovingMass	LREAL	Get, Set	This parameter can be used to define a moving mass. It is used to calculate dynamic acceleration and deceleration forces, torques or pressures.
NoBlock	BOOL	Get, Set	If this parameter is set to TRUE, the use of the springs and blocks on the upper and lower sides will be disabled.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. It becomes TRUE if the position of the simulated actuator is <= LowSideEndswitch.

Methods

Name	Description
GoFaultState() [▸ 232]	This method can be used to simulate an error situation on the simulated axis.

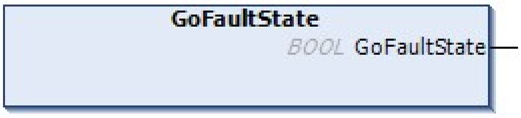
Interfaces

Type	Description
I_SimAxCommon	Standard interface for FB_SimAxCommon.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.1.1.1 GoFaultState()



This method can be used to simulate an error situation on the simulated axis.

The simulated axis does not respond to setpoints, updates the simulated drive state machine and reports the problem via the status word.

Syntax:

```
METHOD GoFaultState : BOOL
```

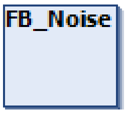
Return value

Name	Type	Description
GoFaultState	BOOL	Reserved for future implementation

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.1.2 FB_Noise



This FB is used to generate a pseudo-random signal that resembles a white noise disturbance. Noise signals may be used to simulate a common problem of analog sensors and interfaces.

Objects of this type are typically used as local elements in an implementation of an axis simulation.

Syntax:

```
FUNCTION_BLOCK FB_Noise
```

Methods

Name	Description
Cyclic [▸ 233]	Cyclic call to generate a new value.
SetLineNoiseLevel [▸ 233]	Determination of the level of an influence by a supply network.
SetSparkNoiseLevel [▸ 233]	Determination of the level of an influence by static discharges.
SetWhiteNoiseLevel [▸ 234]	Setting the level of the noise signal.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.1.2.1 Cyclic

Method to be called cyclically to generate a new value.

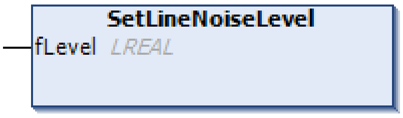
Syntax:

```
METHOD Cyclic : LREAL
```

Outputs

Name	Type	Description
Cyclic	LREAL	The new value of the noise signal

7.4.1.2.2 SetLineNoiseLevel



Determination of the level of an influence by a supply network.

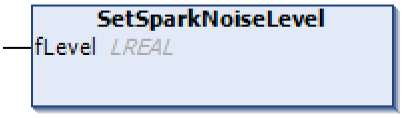
Syntax:

```
METHOD SetLineNoiseLevel
VAR_INPUT
fLevel: LREAL;
END_VAR
```

Inputs

Name	Type	Description
fLevel	LREAL	Level of an influence by a supply network. **Recommended standard: depending on the use of the signal.**

7.4.1.2.3 SetSparkNoiseLevel



Determination of the level of an influence by static discharges.

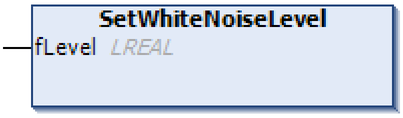
Syntax:

```
METHOD SetSparkNoiseLevel
VAR_INPUT
fLevel: LREAL;
END_VAR
```

Inputs

Name	Type	Description
fLevel	LREAL	Level of an influence by static discharges. **Recommended standard: depending on the use of the signal.**

7.4.1.2.4 SetWhiteNoiseLevel



Setting the level of the noise signal.

Syntax:

```
METHOD SetWhiteNoiseLevel
VAR_INPUT
fLevel: LREAL;
END_VAR
```

Inputs

Name	Type	Description
fLevel	LREAL	Level of the noise signal. **Recommended standard: depending on the use of the signal.**

7.4.2 Simulation of an EtherCAT based servo drive axis

TwinCAT 3 Technology Functions offers two types of EtherCAT-based servo drives: SoE and CoE devices. Both protocols can be used in real machines, but simulating the real behavior is an additional challenge. In CoE drives, a device reset command is triggered via a ControlWord/StatusWord mechanism, while SoE drives use acyclic ADS via EtherCAT communication. Since the simulation of the SoE mechanism is much more complicated, only a CoE-based simulation is provided.

Implementing a simulated servo axis

A simulation requires an implementation that can run in one of two places:

- in the same context as the control implementation
- in a separate PLC or even on a separate CPU

Instantiation in the same context:

Designation	Type	Description
`{attribute 'TcContextName':='FastTask'}`		Instructs TwinCAT to update the I/O of the FB_SimCoE402_Servo class in a specified task. The task should be identical to the call of the `Cyclic()` method of the simulation block.
`fbNcAxisSim`	FB_SimCoE402_Servo [▶ 236]	Creates an instance of the simulation block.
`bNcAxisSimGoError`	BOOL	Creates a signal with which the simulation triggers an error state and requests a reset procedure.
`fbNcAxisSimLimiter`	FB_SimTorqueLimitingCoE402_Servo [▶ 241]	Creates an instance of a FB that is needed to provide the simulation with the torque limitation capability. This FB will exchange data with the control functionality via an interface access.

Instantiation in a separate context:

Designation	Type	Description
`{attribute 'TcContextName':='FastTask'}`		Instructs TwinCAT to update the I/O of the FB_SimCoE402_Servo class in a specified task. The task should be identical to the call of the `Cyclic()` method of the simulation block.
`fbNcAxisSim`	FB_SimCoE402_Servo [▸ 236]	Creates an instance of the simulation block.
`bNcAxisSimGoError`	BOOL	Creates a signal with which the simulation triggers an error state and requests a reset procedure.
`{attribute 'TcContextName':='FastTask'}`		Instructs the following instance to be updated in a specified task. The task should be identical to the call of the `Cyclic()` method of the simulation block.
`fbNcAxisSimLimiter`	FB_SimTorqueLimitingCoE402_Mapped [▸ 241]	Creates an instance of a FB that is needed to provide the simulation with the torque limitation capability. This FB exchanges data with the control functionality via mapping.

Initialization of a simulated servo axis

Initialization is required during the start-up phase. The following list can be used as an example.

```
IF NOT bInitialised AND bSimulation THEN
    fbNcAxisSim.EncoderWeighting:=5.0;
    fbNcAxisSim.EncoderZeroShift:=100.0;
    fbNcAxisSim.EncoderNoiseLevel:=0.001;
    fbNcAxisSim.HighSideBlock:=700.0;
    fbNcAxisSim.HighSideSpringLength:=1.0;
    fbNcAxisSim.LowSideBlock:=100.0;
    fbNcAxisSim.LowSideSpringLength:=1.0;
    fbNcAxisSim.PositionLagKp:=1.0;
    fbNcAxisSim.HighSideEndswitch:=699.0;
    fbNcAxisSim.LowSideEndswitch:=101.0;
    fbNcAxisSim.AbsSwitchHighEnd:=679.0;
    fbNcAxisSim.AbsSwitchLowEnd:=21.0;
    // the next line must be used if the simulation
    // is implemented in a separate context
    fbNcAxisSimLimiterMapped.ConnectToSim(ipSim:=fbNcAxis2Sim);
    // the next two lines must be used if the simulation
    // is implemented in the same context
    fbNcAxisSimLimiter.ConnectToSim(ipSim:=fbNcAxis2Sim);
    iNcAxis.SetTorqueLimiting(fbNcAxisSimLimiter);
END_IF
```

The last line defines the class used by the axis for torque limitation when the simulation is performed in the same context. In this case the penultimate line is required to connect this class to the drive simulation.

Above these lines is an example of implementation in a separate context.

Mapping of a simulated servo axis

The simulation FB provides a local instance of a mapping interface defined as follows:

```
NcAdapt : FB_SimCoE402_ServoNcAdapt;
```

The supplied mapping structures for encoder and drive inputs and outputs are compatible with the TwinCAT NC interfaces and can thus be used.

Functionality of the CoE402 simulation

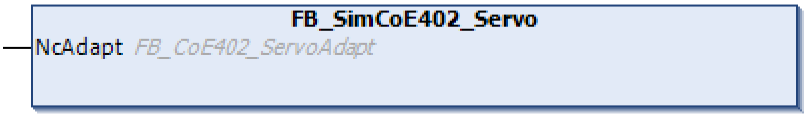
All mechanisms that use mapped interfaces are supported in a way that is expected by TwinCAT NC. This also includes torque limitation.

Since there is no way to emulate ADS communication, some mechanisms cannot be supported.

- Simulated axes cannot be supported by DriveManager. There is no servo drive and no motor. The commissioning requires just parameters and must be done hard coded in the application project.

- The Advanced Homing library of TwinCAT NC uses the ADS communication to disable some functions of the AX servo drives while the homing procedures are executed. The core functions of the TC3 Plastic Functions provide methods to avoid the use of this communication. The homing mechanisms are also ready for use with simulated servo axes.

7.4.2.1 FB_SimCoE402_Servo



This class implements a simulation of a CoE402 servo drive. The FB offers mapping elements for CoE interfaces of servo drives. Mapping is done in the same way as for the real drive unit.

Syntax:

```
FUNCTION_BLOCK FB_SimCoE402_Servo EXTENDS FB_SimAxCommon
VAR_INPUT
    NcAdapt: FB_CoE402_ServoAdapt;
END_VAR
```

I/O variables

Name	Type	Direction	Description
NcAdapt	FB_CoE402_ServoAdapt	Input/Output	The mapping interface for the NC axis.

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	This runtime value provides information about the simulated axis position. ***Notice*** **The simulated position of the actuator can deviate from the reported axis position.**
ActualTorque	LREAL	Get	The simulated actual torque.
ExternalTorque	LREAL	Get, Set	Here, an external torque can be specified from the simulation of a simulated process.
Inertia	LREAL	Get	Moment of inertia [kg * m³], formed from `MovingMass` and a chaining of gears.
PositionLagKp	LREAL	Get, Set	The position control gain of the simulated servo.
Reversed	BOOL	Get, Set	TRUE if the direction of rotation of the simulated servo is inverted.
StatusWord	WORD	Get	The status word of the servo.
TorqueLimitNegative	LREAL	Get, Set	The negative torque limit.
TorqueLimitPositive	LREAL	Get, Set	The positive torque limit.
AbsolutSwitch	BOOL	Get	This runtime value provides information about the simulated feedback signal as defined by AbsSwitchHighSelect.
AbsSwitchHighEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=TRUE: AbsolutSwitch will be TRUE if the simulated actuator position is >= AbsSwitchHighEnd.

Name	Type	Access	Description
			***Notice* The simulated position of the actuator can deviate from the reported axis position.**
AbsSwitchHighSelect	BOOL	Get, Set	This parameter defines the active AbsolutSwitch. A TRUE will select AbsSwitchHighEnd while a FALSE makes the AbsSwitchLowEnd the active AbsolutSwitch.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=FALSE: AbsolutSwitch will be TRUE if the simulated actuator position is <= AbsSwitchLowEnd. ***Notice* The simulated position of the actuator can deviate from the reported axis position.**
CycleTime	LREAL	Get, Set	This parameter must be initialized with the update calling cycle of the Cyclic() method and will be forwarded to any sub-component if required. It is used to define any time behavior.
EncoderInterpolation	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor denominator • Hydraulics library: encoder, inc. Interpolation
EncoderNoiseLevel	LREAL	Get, Set	The simulated axis offers the possibility to disturb the reported position with a pseudo-random white noise. This is realized by a FB_Noise() function block. The parameter EncoderNoiseLevel is forwarded as SetWhiteNoiseLevel.
EncoderWeighting	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor numerator • Hydraulics library: encoder, weighting factor
EncoderZeroShift	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: rotary encoder, parameter, position bias • Hydraulics library: encoder, zero offset
HighSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value less than or equal to this parameter.
HighSideEndswitch	LREAL	Get, Set	This parameter defines the UpperEndSwitch threshold that becomes TRUE when the simulated position of the actuator is >= this parameter.
HighSideSpringLengt	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the upper side of the actuator stroke.

Name	Type	Access	Description
LowerEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes TRUE if the position of the simulated actuator is >= HighSideEndswitch.
LowSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value above or equal to this parameter.
LowSideEndswitch	LREAL	Get, Set	This parameter defines the LowerEndSwitch threshold that becomes TRUE when the simulated position of the actuator is <= this parameter.
LowSideSpringLength	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the lower side of the actuator stroke.
MovingMass	LREAL	Get, Set	This parameter can be used to define a moving mass. It is used to calculate dynamic acceleration and deceleration forces, torques or pressures.
NoBlock	BOOL	Get, Set	If this parameter is set to TRUE, the use of the springs and blocks on the upper and lower sides will be disabled.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. It becomes TRUE if the position of the simulated actuator is <= LowSideEndswitch.

NOTICE

When using a simulated servo drive, a FB of type FB_SimTorqueLimitingCoE402_Servo [▸ 241] or FB_SimTorqueLimitingCoE402_Mapped [▸ 241] must be used to perform the data exchange between TwinCAT 3 Plastic Functions and the simulation.

Methods

Name	Description
GoFaultState() [▸ 232]	This method can be used to simulate an error situation on the simulated axis.
Cyclic()	This method must be called cyclically.
Gear() [▸ 239]	Starts a chaining of gear factors.
GearBox() [▸ 239]	Rotary transmission (gearbox).
GearRackNPinion() [▸ 240]	Linear transmission by rack and pinion.
GearScrew() [▸ 240]	Linear transmission by spindle.

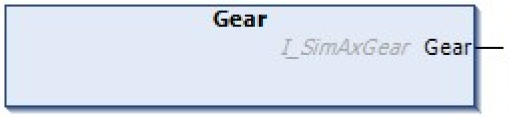
Interfaces

Type	Description
I_SimAxCommon	Standard interface for FB_SimAxCommon.
I_SimCoE402_Servo	Standard interface for FB_SimCoE402_Servo.
I_SimAxGear	Gear interface for coupling gear ratios of the axis.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

7.4.2.1.1 Gear()



This method is used to start the chaining of gear factors.

Effects on the actual position

The gear methods are used to determine the moment of inertia and the encoder weighting. Check the resulting `EncoderWeighting` to ensure that it matches the NC.

Syntax:

```
METHOD GearBox : I_SimAxGear
```

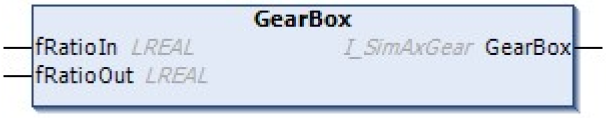
Return value

Name	Type	Description
GearBox	I_SimAxGear	Gear interface on the axis to configure a chaining of gearboxes.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0) Tc3_PlasticNc (>= v3.12.5.0)

7.4.2.1.2 GearBox()



This method can be used to configure a rotary transmission factor of the gearbox.

Effects on the actual position

The gear methods are used to determine the moment of inertia and the encoder weighting. Check the resulting `EncoderWeighting` to ensure that it matches the NC.

Syntax:

```
METHOD GearBox : I_SimAxGear
VAR_INPUT
    fRatioIn:     LREAL;
    fRatioOut:    LREAL;
END_VAR
```

Return value

Name	Type	Description
GearBox	I_SimAxGear	Gear interface on the axis to configure a chaining of gearboxes.

Inputs

Name	Type	Description
fRatioIn	LREAL	Number of teeth of the gearbox input.

Name	Type	Description
fRatioOut	LREAL	Number of teeth of the gearbox output.

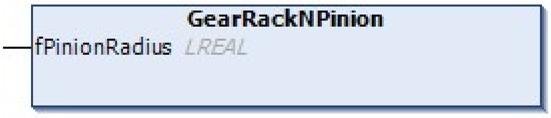
Mathematical relationship of the inputs

`fRationIn` and `fRatioOut` can be regarded as the numerator and denominator of a mathematical fraction. This allows the transmission ratio to be specified as a fraction.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0) Tc3_PlasticNc (>= v3.12.5.0)

7.4.2.1.3 GearRackNPinion()



This method is used to finalize a chaining of gears by means of a rack.

Effects on the actual position

The gear methods are used to determine the moment of inertia and the encoder weighting. Check the resulting `EncoderWeighting` to ensure that it matches the NC.

Syntax:

```
METHOD GearRackNPinion
VAR_INPUT
    fPinionRadius:    LREAL;    // [m]
END_VAR
```

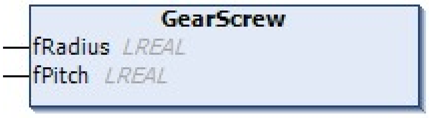
Inputs

Name	Type	Description
fPinionRadius	LREAL	Effective radius [m] of the pinion on the rack

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0) Tc3_PlasticNc (>= v3.12.5.0)

7.4.2.1.4 GearScrew()



This method is used to finalize a chaining of gears by means of a spindle.

Effects on the actual position

The gear methods are used to determine the moment of inertia and the encoder weighting. Check the resulting `EncoderWeighting` to ensure that it matches the NC.

Syntax:

```
METHOD GearScrew
VAR_INPUT
    fRadius:        LREAL;    // [m]
    fPitch:         LREAL;    // [m]
END_VAR
```

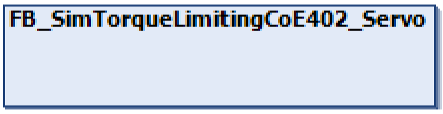
Inputs

Name	Type	Description
fRadius	LREAL	Radius of the spindle [m]
fPitch	LREAL	Pitch of the spindle [m]

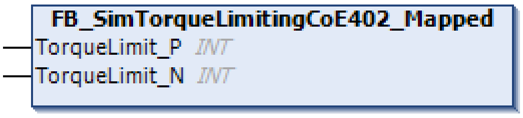
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0) Tc3_PlasticNc (>= v3.12.5.0)

7.4.2.2 FB_SimTorqueLimitingCoE402_Servo



If the simulation is performed in the same task in which the process software of the axis is running: A FB of this type must be used to provide a simulated servo drive with a torque limitation.



If the simulation is not performed in the same task in which the process software of the axis is running: A FB of this type must be used to provide a simulated servo drive with a torque limitation and to connect it by a mapping.

Properties

Name	Type	Access	Description
Activate	BOOL	Get, Set	A TRUE signals that a torque limitation is active.

Methods

Name	Description
ConnectToSim() [▸ 242]	This method establishes a connection between the torque-limiting FB and the simulated servo drive.
Cyclic	This method, to be called cyclically, performs all calculations and decisions.

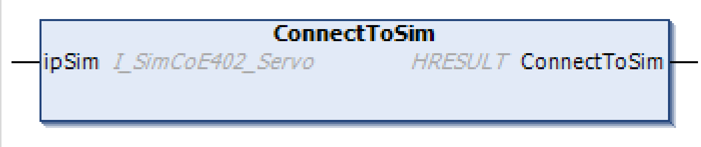
Interfaces

Type	Description
I_TorqueLimitingFb	Interface for linking with the TorqueLimiting function of an axis.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

7.4.2.2.1 ConnectToSim()



This method establishes a connection between the torque-limiting FB and the simulated servo drive. This connection replaces the EtherCAT communication with a real CoE402-based servo drive.

Syntax:

```
METHOD ConnectToSim : HRESULT
VAR_INPUT
    ipSim:    I_SimCoE402_Servo;
END_VAR
```

Return value

Name	Type	Description
ConnectToSim	HRESULT	Return value with feedback on the success of the execution.

Inputs

Name	Type	Description
ipSim	I_SimCoE402_Servo	The simulation block for the simulation of a CoE402 servo.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticNc v3.12.4.26 or higher

7.4.3 Simulation of an inverter axis

This simulation corresponds to an inverter axis and can be used to replace a real inverter drive. For this purpose, compatible mapping elements are provided and a behavior very similar to that of a real axis is emulated.

Implementation of a simulated inverter axis

A simulation requires an implementation that can run in one of two places:

- in the same context as the control implementation
- in a separate runtime or even on a separate CPU

Implementation in both contexts is recommended as described below:

Designation	Type	Description
`{attribute 'TcContextName':='FastTask'}`		Instructs FB_SimCoE402_Servo to be updated in a specific task. In this case it must be the calling task that executes the Cyclic() method of the simulation block.
`fbInvAxisSim`	FB_SimCoE402_Inverter_Xyz [▶ 243]	Creates an instance of the simulation FB.
`bInvAxisSimGoError`	BOOL	Creates a signal for the simulation to report an error state and request a reset procedure.

Range of supported inverters

The term FB_SimCoE402_Inverter_Xyz above must be replaced by one of the following options:

`FB_SimCoE402_Inverter_VM`

`FB_SimCoE402_Inverter_CSVM`

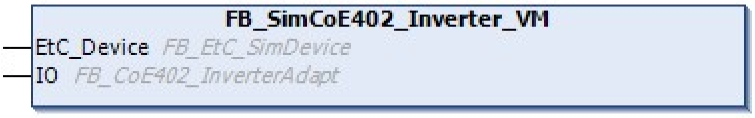
Initialization of a simulated inverter axis

In the startup phase, it is necessary to implement an initialization sequence as described below.

```
IF NOT bInitialised AND bSimulation THEN
    fbInverterSimDS402.CycleTime := 0.002;
    fbInverterSimDS402.MaxTurnRate := 1380.0;
    fbInverterSimDS402.MinTurnrate := 45.0;
    fbInverterSimDS402.RampTime := 2.0;
    fbInverterSimDS402.OutputFactor := 1.0;
    fbInverterSimDS402.ReferenceLoad := 100.0;
END_IF
```

The parameters must correspond to the behavior of the simulated device, not to the intended use.

7.4.3.1 FB_SimCoE402_Inverter_VM



This simulation corresponds to an inverter axis and can be used to replace a real inverter drive by providing compatible mapping elements and emulating behavior very similar to that of a real axis.

The FB provides mapping elements for CoE interfaces of basic frequency inverters (FI) that match the interfaces implemented by the adapter FB for this drive type. Mapping is done in the same way as for the real drive unit.

Syntax:

```
FUNCTION_BLOCK FB_SimCoE402_Inverter_VM EXTENDS FB_SimCoE402_Inverter
VAR_INPUT
    IO : FB_CoE402_InverterAdapt;
END_VAR
```

I/O variables

Name	Type	Direction	Description
EtC_Device	FB_Etc_SimDevice	Output	Simulated EtherCAT diagnostic signals.
I/O	FB_CoE402_InverterAdapt	Input/Output	The mapping interface for the inverter axis.

Properties

Name	Type	Access	Description
CycleTime	LREAL	Get, Set	Cycle time with which the simulation is executed.
Load	LREAL	Get, Set	Here you can specify a load that is claimed by a simulated process.
MaxTurnRate	LREAL	Get, Set	The maximum turn rate of the drive.
MinTurnrate	LREAL	Get, Set	A minimum turn rate can be specified here. If a value > 0.0 is specified, the drive ignores set turn rates below this threshold and a dead band is created. If the default is less than 10 RPM, it is assumed that the inverter supports vector control.
OutputFactor	LREAL	Get, Set	Multiplier for the output.
RampTime	LREAL	Get, Set	Here the time for the ramp from zero to `MaxTurnRate` or vice versa is defined.
ReferenceLoad	LREAL	Get, Set	A reference value for Load.
VectorControlled	BOOL	Get	A TRUE specifies that the actual turn rate is load independent. If FALSE, the inverter responds to a load torque with a proportional slip. The simulation assumes a slip of 0 to 50 % for a load of 0 to 100 % of the reference load.

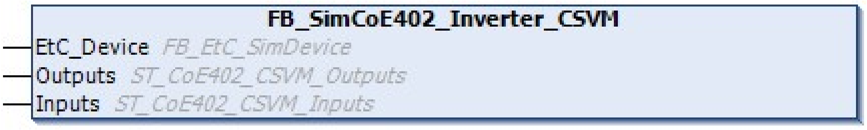
Methods

Name	Description
Cyclic()	Cycle method
TriggerError()	A call to this method places the simulated drive into the error state.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

7.4.3.2 FB_SimCoE402_Inverter_CSVM


FB_SimCoE402_Inverter_CSVM
EtC_Device FB_EtC_SimDevice
Outputs ST_CoE402_CSVM_Outputs
Inputs ST_CoE402_CSVM_Inputs

This simulation corresponds to an inverter axis and can be used to replace a real inverter drive by providing compatible mapping elements and emulating behavior very similar to that of a real axis.

The FB provides mapping elements for CoE interfaces of frequency inverters that implement a servo-like architecture and that match the interfaces implemented by the adapter FB for this type of drives. Mapping is done in the same way as for the real drive unit.

Syntax:

```
FUNCTION_BLOCK FB_SimCoE402_Inverter_CSVM EXTENDS FB_SimCoE402_Inverter
VAR_INPUT
    Outputs AT %I* : ST_CoE402_CSVM_Outputs;
    Inputs AT %Q*  : ST_CoE402_CSVM_Inputs;
END_VAR
```

I/O variables

Name	Type	Direction	Description
EtC_Device	FB_Etc_SimDevice	Output	Simulated EtherCAT diagnostic signals
Outputs	ST_CoE402_CSVM_Outputs	Input	The mapping interface for the inverter outputs.
Inputs	ST_CoE402_CSVM_Inputs	Output	The mapping interface for the inverter inputs.

Properties

Name	Type	Access	Description
CycleTime	LREAL	Get, Set	Cycle time with which the simulation is executed.
Load	LREAL	Get, Set	Here you can specify a load that is claimed by a simulated process.
MaxTurnRate	LREAL	Get, Set	The maximum turn rate of the drive.
MinTurnrate	LREAL	Get, Set	A minimum turn rate can be specified here. If a value > 0.0 is specified, the drive ignores set turn rates below this threshold and a dead band is created. If the default is less than 10 RPM, it is assumed that the inverter supports vector control.
OutputFactor	LREAL	Get, Set	The factor used to exchange turn rates with the device.
RampTime	LREAL	Get, Set	Here the time for the ramp from zero to MaxTurnRate or vice versa is defined.
ReferenceLoad	LREAL	Get, Set	A reference value for Load.
VectorControlled	BOOL	Get	A TRUE specifies that the actual turn rate is load independent. If FALSE, the inverter responds to a load torque with a proportional slip. The simulation assumes a slip of 0 to 50 % for a load of 0 to 100 % of the reference load.

Methods

Name	Description
Cyclic()	This method must be called cyclically by the application.
TriggerError()	A call to this method places the simulated drive into the error state.

Requirements

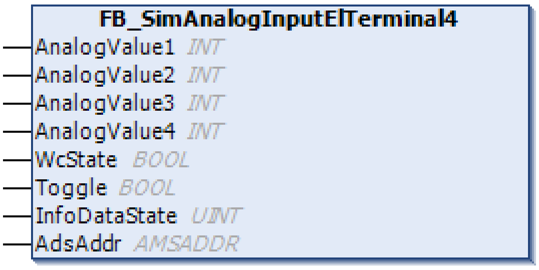
Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

7.4.4 I/O Simulation containers

To create a simulation, function blocks are needed that can replace an I/O device in a compatible way.

Name	Channels	Description
FB_SimAnalogInputElTerminal4 [▶ 246]	4	Simulation of an analog EtherCAT input terminal. Example EL3134
FB_SimAnalogOutputElTerminal4 [▶ 247]	4	Simulation of an analog EtherCAT output terminal. Example EL4134
FB_SimSsiInputElTerminal1 [▶ 248]	1	Simulation of an EtherCAT SSI input terminal. Example EL5001
FB_CoE402_ServoAdapt [▶ 247]	1	AX8000

7.4.4.1 Simulation of an analog input terminal



Simulation of a 4-channel analog input terminal. This FB only provides a mapping interface and does not contain an implementation.

Syntax:

```
FUNCTION_BLOCK FB_SimAnalogInputElTerminal4
```

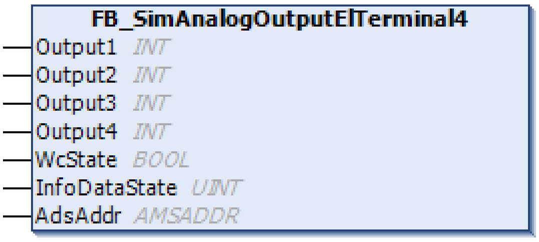
Inputs

Name	Type	Access	Initial value	Description
AnalogValue1	INT	IN	0	The simulated input value of the 1st channel.
AnalogValue2	INT	IN	0	The simulated input value of the 2nd channel.
AnalogValue3	INT	IN	0	The simulated input value of the 3rd channel.
AnalogValue4	INT	IN	0	The simulated input value of the 4th channel.
WcState	BOOL	OUT	FALSE	The simulated Working Counter State.
Toggle	BOOL	OUT	FALSE	The simulated toggle bit.
InfoDataState	UINT	OUT	8	The simulated Device State.
AdsAddr	AMSADDR	OUT		The simulated ADS address.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.4.2 Simulation of an analog output terminal



Simulation of a 4-channel analog output terminal. This FB only provides a mapping interface and does not contain an implementation.

Syntax:

```
FUNCTION_BLOCK FB_SimAnalogOutputElTerminal4
```

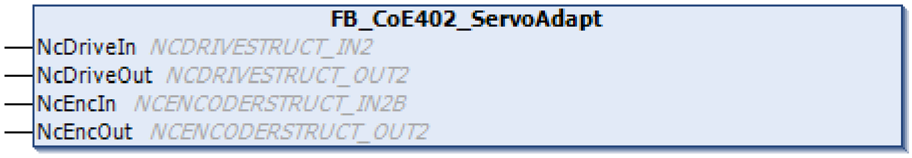
Inputs

Name	Type	Access	Initial value	Description
Output1	INT	IN	0	The simulated output value of the 1st channel.
Output2	INT	IN	0	The simulated output value of the 2nd channel.
Output3	INT	IN	0	The simulated output value of the 3rd channel.
Output4	INT	IN	0	The simulated output value of the 4th channel.
WcState	BOOL	OUT	FALSE	The simulated Working Counter State.
InfoDataState	UINT	OUT	8	The simulated Device State.
AdsAddr	AMSADDR	OUT		The simulated ADS address.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.4.3 Simulation of a CoE402 servo drive


FB_CoE402_ServoAdapt
NcDriveIn NCDRIVESTRUCT_IN2
NcDriveOut NCDRIVESTRUCT_OUT2
NcEncIn NCENCODERSTRUCT_IN2B
NcEncOut NCENCODERSTRUCT_OUT2

Simulation of a CoE408 servo drive. This FB only provides a mapping interface and does not contain an implementation.

Syntax:

```
FUNCTION_BLOCK FB_CoE402_ServoAdapt
```

Inputs

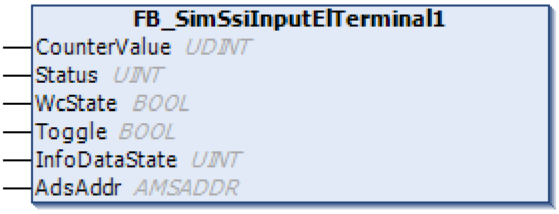
Name	Type	Access	Description
NcDriveIn	NCDRIVESTRUCT_IN2	OUT	For the DriveIn interface of the NC.

Name	Type	Access	Description
NcDriveOut	NCDRIVESTRUCT_OUT2	IN	For the DriveOut interface of the NC.
NcEncIn	NCENCODERSTRUCT_IN2B	OUT	For the EncoderIn interface of the NC.
NcEncOut	NCENCODERSTRUCT_OUT2	IN	For the EncoderOut interface of the NC.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.4.4 Simulation of an SSI input terminal



Simulation of a 1-channel SSI input terminal. This FB only provides a mapping interface and does not contain an implementation.

Syntax:

```
FUNCTION_BLOCK FB_SimSsiInputElTerminal1
```

Inputs

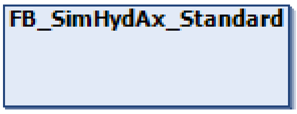
Name	Type	Access	Initial value	Description
AnalogValue1	INT	IN	0	The simulated input value of the 1st channel.
AnalogValue2	INT	IN	0	The simulated input value of the 2nd channel.
AnalogValue3	INT	IN	0	The simulated input value of the 3rd channel.
AnalogValue4	INT	IN	0	The simulated input value of the 4th channel.
WcState	BOOL	OUT	FALSE	The simulated Working Counter State.
Toggle	BOOL	OUT	FALSE	The simulated toggle bit.
InfoDataState	UINT	OUT	8	The simulated Device State.
AdsAddr	AMSADDR	OUT		The simulated ADS address.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.4.5 Components of the hydraulic simulation

7.4.5.1 FB_SimHydAx_Standard



This FB is a simplified simulation of a universal hydraulic axis. A number of typical effects are supported.

The implementation of this object is not a full-fledged and realistic simulation.

Syntax:

```
FUNCTION_BLOCK FB_SimHydAx_Standard EXTENDS FB_SimAxCommon
```

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	This runtime value provides information about the simulated axis position. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**
Cylinder	I_SimCylinder [▸ 253]	Get	An interface to the local `FB_SimCylinder` object.
Encoder	I_SimUniversalEncoder [▸ 256]	Get	An interface to the local `FB_SimUniversalEncoder` object.
ExternalForce	LREAL	Get, Set	This runtime value can be updated with pressure values required for process simulation. It is used to calculate the total pressure on the simulated cylinder and is part of the reported actual pressure.
Overlapp_PA	LREAL	Get, Set	Overlap on the A-side of the valve.
Overlapp_PB	LREAL	Get, Set	Overlap on the B-side of the valve.
PressureTransducerA	I_PressureTransducer [▸ 255]	Get	An interface to the local `FB_PressureTransducer` object on the A-side of the `FB_SimCylinder` object.
PressureTransducerB	I_PressureTransducer [▸ 255]	Get	An interface to the local `FB_PressureTransducer` object on the B-side of the `FB_SimCylinder` object.
Qnominal_PA	LREAL	Get, Set	Oil flow between the supply connection and the A-side connection of the valve.
Qnominal_PB	LREAL	Get, Set	Oil flow between the supply connection and the B-side connection of the valve.
SpoolFeedback	INT	Get	Feedback value of the valve coil position.
SupplyPressure	LREAL	Get, Set	Supply pressure of the valve.
Valve	I_SimValve [▸ 251]	Get	An interface to the local `FB_SimValve` object on the B-side of the `FB_SimCylinder` object.
ValveControlValueType	E_SimControlValueType	Get, Set	The valve type used for the axis.
Valve*Xyz*	*Diverse*	Get, (Set)	Forwarded values of the internal `FB_SimValve` instance.
AbsolutSwitch	BOOL	Get	This runtime value provides information about the simulated feedback signal as defined by AbsSwitchHighSelect.
AbsSwitchHighEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=TRUE: AbsolutSwitch will be TRUE if the simulated actuator position is >= AbsSwitchHighEnd. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**

Name	Type	Access	Description
AbsSwitchHighSelect	BOOL	Get, Set	This parameter defines the active AbsolutSwitch. A TRUE will select AbsSwitchHighEnd while a FALSE makes the AbsSwitchLowEnd the active AbsolutSwitch.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold in case AbsSwitchHighSelect=FALSE: AbsolutSwitch will be TRUE if the simulated actuator position is <= AbsSwitchLowEnd. *Notice* **The simulated position of the actuator can deviate from the reported axis position.**
CycleTime	LREAL	Get, Set	This parameter must be initialized with the update calling cycle of the Cyclic() method and will be forwarded to any sub-component if required. It is used to define any time behavior.
EncoderInterpolation	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor denominator • Hydraulics library: encoder, inc. Interpolation
EncoderNoiseLevel	LREAL	Get, Set	The simulated axis offers the possibility to disturb the reported position with a pseudo-random white noise. This is realized by a FB_Noise() function block. The parameter EncoderNoiseLevel is forwarded as SetWhiteNoiseLevel.
EncoderWeighting	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: encoder, parameter, scaling factor numerator • Hydraulics library: encoder, weighting factor
EncoderZeroShift	LREAL	Get, Set	This parameter must be initialized with the same value as the corresponding parameter in the motion technology: • NC: rotary encoder, parameter, position bias • Hydraulics library: encoder, zero offset
HighSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value less than or equal to this parameter.
HighSideEndswitch	LREAL	Get, Set	This parameter defines the UpperEndSwitch threshold that becomes TRUE when the simulated position of the actuator is >= this parameter.
HighSideSpringLengt	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the upper side of the actuator stroke.
LowerEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes TRUE if the position of the simulated actuator is >= HighSideEndswitch.

Name	Type	Access	Description
LowSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value above or equal to this parameter.
LowSideEndswitch	LREAL	Get, Set	This parameter defines the LowerEndSwitch threshold that becomes TRUE when the simulated position of the actuator is <= this parameter.
LowSideSpringLength	LREAL	Get, Set	This parameter defines the length of a simulated spring-like effect at the lower side of the actuator stroke.
MovingMass	LREAL	Get, Set	This parameter can be used to define a moving mass. It is used to calculate dynamic acceleration and deceleration forces, torques or pressures.
NoBlock	BOOL	Get, Set	If this parameter is set to TRUE, the use of the springs and blocks on the upper and lower sides will be disabled.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. It becomes TRUE if the position of the simulated actuator is <= LowSideEndswitch.

Methods

Name	Description
GoFaultState() [▶ 232]	This method can be used to simulate an error situation on the simulated axis.
Cyclic()	Cycle method

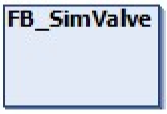
Interfaces

Type	Description
I_SimAxCommon	Standard interface for FB_SimAxCommon.
I_SimHydAx	Standard interface for FB_SimHydAx_Standard.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0) Tc3_PlasticHydraulic (>= v3.12.5.0)

7.4.5.2 FB_SimValve



This simplified simulation of an analog proportional or servo valve is used as part of a simulated hydraulic axis. It supports a range of typical variants of non-linearities. There is no dynamic behavior.

The implementation of this object is not a full-fledged and realistic simulation.

Objects of this type are typically used as local elements in an implementation of an axis simulation.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_SimValve
```

Properties

Name	Type	Access	Description
AdsAddr	AMSADDR	Get, Set	This variable is used to implement a mapping interface of an analog valve.
ControlValue	INT	Get, Set	This runtime value must be updated cyclically with an output property of a FB_SimAnalogOutputElTerminal4 instance. **This assignment can be understood as the cable connecting an output terminal to an input pin of the valve connector.**
ControlValueNorm	LREAL	Get	This runtime value provides information about the current control value in abstract units (e.g. %).
ControlValueType	E_SimControlValueType	Get, Set	This parameter must be initialized with a value that specifies the type of the control signal. It must match the type of the simulated output terminal.
InfoDataState	UINT	Get, Set	This variable is used to implement a mapping interface of an analog valve.
Orifice_PA	LREAL	Get	This runtime value provides information about the current opening between the supply port and the A-side port of the valve.
Orifice_PB	LREAL	Get	This runtime value provides information about the current opening between the supply port and the B-side port of the valve.
Overlapp_PA	LREAL	Get, Set	This parameter must be initialized with the overlap on the A-side of the valve. **Typical values can be found in the manufacturer's data sheets.**
Overlapp_PB	LREAL	Get, Set	This parameter must be initialized with the overlap on the B-side of the valve. **Typical values can be found in the manufacturer's data sheets.**
P_A	LREAL	Get, Set	This runtime value reports about the pressure at the A-side port of the valve.
P_B	LREAL	Get, Set	This runtime value reports about the pressure at the B-side port of the valve.
Q_PA	LREAL	Get	This runtime value reports about the oil flow between the supply port and the A-side port of the valve.
Q_PB	LREAL	Get	This runtime value reports about the oil flow between the supply port and the B-side port of the valve.
Qnominal_PA	LREAL	Get, Set	This parameter must be initialized with the capacity of the supply to the A-side opening of the simulated valve. **Typical values can be found in the manufacturer's data sheets.**
Qnominal_PB	LREAL	Get, Set	This parameter must be initialized with the capacity of the supply to the B-side opening of the simulated valve.

Name	Type	Access	Description
			Typical values can be found in the manufacturer's data sheets.
SpoolFeedback	INT	Get	This runtime value reports about the actual position of the valve piston. It is used to implement a mapping interface of an analog valve.
SupplyPressure	LREAL	Get, Set	This runtime value must be updated with the supply pressure. If the value is assumed to be constant, it can be updated once during initialization.
Toggle	BOOL	Get	This variable is used to implement a mapping interface of an analog valve.
ValveSpoolType	E_SimValveSpoolType	Get, Set	This parameter sets the transfer characteristic of the valve.
WcState	BOOL	Get, Set	This variable is used to implement a mapping interface of an analog valve.

Methods

Name	Description
Cyclic()	Cycle method

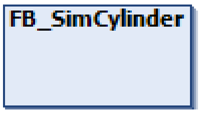
Interfaces

Type	Description
I_SimValve	Standard interface for FB_SimValve.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

7.4.5.3 FB_SimCylinder



This simplified simulation of a universal cylinder is used as part of a simulated hydraulic axis. A number of typical effects are supported.

The implementation of this object is not a full-fledged and realistic simulation.

Objects of this type are typically used as local elements in an implementation of an axis simulation.

Syntax:

```
FUNCTION_BLOCK FB_SimCylinder
```

Properties

Name	Type	Access	Description
CycleTime	LREAL	Get, Set	This parameter must be initialized with the call cycle time.
ExternalForce	LREAL	Get, Set	This runtime value can be used to update an external force in the simulation calculation. It is used for conversion to pressure values.
HighSideBlock	LREAL	Get, Set	The simulated upper mechanical stop.
HighSideSpringLength	LREAL	Get, Set	The simulated spring action before the upper mechanical stop.
LowSideBlock	LREAL	Get, Set	The simulated lower mechanical stop.
LowSideSpringLength	LREAL	Get, Set	The simulated spring action before the lower mechanical stop.
P_A	LREAL	Get, Set	The simulated pressure at the A-side cylinder port.
P_B	LREAL	Get, Set	The simulated pressure at the B-side cylinder port.
PistonDiameter	LREAL	Get, Set	The diameter of the cylinder piston. It is used to calculate the effective areas of the cylinder.
Position	LREAL	Get, Set	The current position of the cylinder.
Q_PA	LREAL	Get, Set	This runtime value provides information about the oil flow flowing into or out of the A-side cylinder port. **This value is usually determined by a valve simulation and passed on by a hydraulic axis simulation.**
Q_PB	LREAL	Get, Set	This runtime value provides information about the oil flow flowing into or out of the B-side cylinder port. **This value is usually determined by a valve simulation and passed on by a hydraulic axis simulation.**
RodDiameter	LREAL	Get, Set	The diameter of the rod of the cylinder. It is used to calculate the effective areas of the cylinder.
Stroke	LREAL	Get, Set	The stroke length (i.e. the distance between the lower and upper mechanical end positions) of the cylinder. It is used to limit the actual position.

Methods

Name	Description
Cyclic()	Cycle method

Interfaces

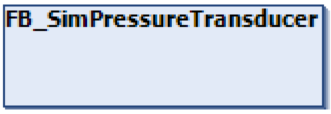
Type	Description
I_SimCylinder	Standard interface for FB_SimCylinder.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

Development environment	Target platform	PLC libraries to include
		Tc3_PlasticHydraulic v3.12.4.26 or higher

7.4.5.4 FB_SimPressureTransducer



FB_SimPressureTransducer

This simplified simulation of a universal pressure transducer is used as component of a simulated hydraulic axis. A number of typical effects are supported.

The implementation of this object is not a full-fledged and realistic simulation.

Objects of this type are typically used as local elements in an implementation of an axis simulation.

Syntax:

```
FUNCTION_BLOCK FB_SimPressureTransducer
```

Properties

Name	Type	Access	Description
AdcValue	INT	Get	This runtime value contains the input value of the simulated terminal.
CycleTime	LREA	Get, Set	This parameter must be initialized with the call cycle time.
NoiseLevel	LREAL	Get, Set	This parameter can be used to set the amplitude of a simulated noise signal that is added to the pressure at the sensor.
NominalPressure	LREAL	Get, Set	This parameter must be initialized with the rated pressure (i.e. full scale) of the simulated transducer.
Pressure	LREAL	Get, Set	This runtime value must be updated cyclically with the pressure of a specific point in a simulated hydraulic system.
Signal	LREAL	Get	This runtime value contains the electrical signal of a simulated 0 to 10 V pressure transducer.

Methods

Name	Description
Cyclic()	Cycle method

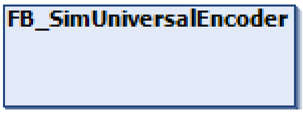
Interfaces

Type	Description
I_SimValve	Standard interface for FB_SimValve.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

7.4.5.5 FB_SimUniversalEncoder


FB_SimUniversalEncoder

This simplified simulation of a universal encoder is used as part of a simulated hydraulic axis. A number of typical effects are supported.

The implementation of this object is not a full-fledged and realistic simulation.

Objects of this type are typically used as local elements in an implementation of an axis simulation.

Syntax:

```
FUNCTION_BLOCK FB_SimUniversalEncoder
```

Properties

Name	Type	Access	Description
AdsAddr	AMSADDR	Get	This variable is used to implement a mapping interface of an analog valve.
Count_UDINT	UDINT	Get	This variable is used to implement a mapping interface of an analog valve.
Count_UINT	UINT	Get	This variable is used to implement a mapping interface of an analog valve.
InfoDataState	UINT	Get, Set	This variable is used to implement a mapping interface of an analog valve.
NoiseLevel	LREAL	Get, Set	This parameter must be initialized with the white noise amplitude to be simulated.
Offset	LREAL	Get, Set	This parameter must be initialized with the displacement between the cylinder zero point (e.g. the lower stop) and the encoder zero point.
Position	LREAL	Get, Set	This runtime value provides information about the actual position of the encoder.
Resolution	LREAL	Get, Set	This parameter must be initialized with the resolution (i.e. the path per increment) of the encoder.
Toggle	BOOL	Get	This variable is used to implement a mapping interface of an analog valve.
WcState	BOOL	Get, Set	This variable is used to implement a mapping interface of an analog valve.

Methods

Name	Description
Cyclic()	Cycle method

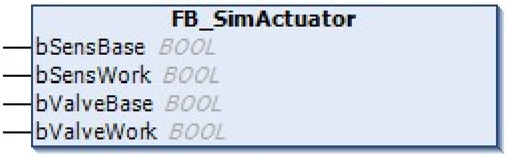
Interfaces

Type	Description
I_SimUniversalEncoder	Standard interface for FB_SimUniversalEncoder.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

7.4.6 SimActuator



FB_SimActuator
bSensBase BOOL
bSensWork BOOL
bValveBase BOOL
bValveWork BOOL

This class simulates a digital linear actuator.

Syntax:

```
FUNCTION_BLOCK FB_SimActuator
```

I/O variables

Name	Type	Direction	Description
bSensBase	BOOL	Output	Sensor for the base position of the actuator.
bSensWork	BOOL	Output	Sensor for the working position of the actuator.
bValveBase	BOOL	Input	Valve for the base position of the actuator.
bValveWork	BOOL	Input	Valve for the working position of the actuator.

Properties

Name	Type	Access	Description
DelayBase	LREAL	Get, Set	Delay between signal for base position and sensor signal.
DelayWork	LREAL	Get, Set	Delay between signal for the working position and sensor signal.
Enable	BOOL	Get, Set	Enable the simulation.
HasMidPosition	BOOL	Get, Set	The simulated actuator valve has a middle position.
SpringReturned	BOOL	Get, Set	The simulated actuator valve is spring-returned.

Methods

Name	Description
Cyclic()	Cycle method

Interfaces

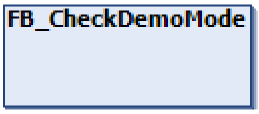
Type	Description
I_SimActuator	Standard interface on FB_SimActuator.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v3.12.5.0)

7.5 Debug

7.5.1 FB_CheckDemoMode



A FB of this type can be instantiated and used by the PLC application to scan the fieldbus configuration. It reports on fieldbuses with special features such as real-time performance or USB support.

Syntax:

```
FUNCTION_BLOCK FB_CheckDemoMode
```

Properties

Name	Type	Description
DemoMode	BOOL	A TRUE after scanning the fieldbus configuration has one of the following causes: There is no fieldbus configured with real-time performance. ForceDemoMode=TRUE and ForceNonDemoMode=FALSE.
EtC_detected	BOOL	A TRUE after scanning the fieldbus configuration signals that the configuration provides for an EtherCAT fieldbus.
ForceDemoMode	BOOL	With a TRUE the DemoMode can be forced.
ForceNonDemoMode	BOOL	With a TRUE the NonDemoMode can be forced.
NonDemoMode	BOOL	A TRUE after scanning the fieldbus configuration has one of the following causes: At least one fieldbus with real-time performance is configured. ForceDemoMode=FALSE and ForceNonDemoMode=TRUE.
ReScan	BOOL	A TRUE on this property triggers a rescan of the fieldbus configuration.
RtBus_detected	BOOL	A TRUE after scanning the fieldbus configuration signals that the configuration provides for a fieldbus with real-time performance.
USB_detected	BOOL	A TRUE after scanning the fieldbus configuration signals that the configuration provides for a USB connection. This typically indicates a control panel.

Methods

Name	Description
Cyclic	This method is to be called cyclically by the application.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.5.2 FB_ChangeDetector

This class can recognize sudden changes in a value.

Syntax:

```
FUNCTION_BLOCK FB_ChangeDetector
```

Properties

Name	Type	Access	Description
Cycletime	LREAL	Get, Set	Cycle time used for checking.
Filter	LREAL	Get, Set	Filter time over which a slight change in the value is adjusted.
Tolerance	LREAL	Get, Set	Tolerance above which a change is recognized.

Methods

Name	Description
Cyclic(`fValue`)	Cycle method ***Notice* Returns FALSE if `fValue` has changed.**

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.5.3 FB_CoreDebug

This class is used for better debugging of the core function chain.

Syntax:

```
FUNCTION_BLOCK FB_CoreDebug
```

Methods

Name	Description
AxisStateIncrement ()	Increments the internal counter value `nAxisStatecounter`. ***Notice* Is called by `FB_CoreFunction.AxisState.Set`.**
AxisStateReset()	Resets the internal counter value `nAxisStatecounter`.

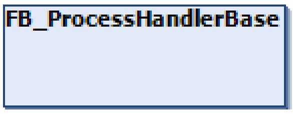
Name	Description
CyclicCounterIncrement()	Increments the internal counter value `nCyclicCounter`. ***Notice*** **Is called by** `FB_CoreFunction.Cyclic()` **.**
CyclicCounterReset()	Resets the internal counter value `nCyclicCounter`.
FunctionCounterIncrement()	Increments the internal counter value `nFunctionCounter`. ***Notice*** **Is called by** `FB_CoreFunction.NextCore.Set` **.**
FunctionCounterReset()	Resets the internal counter value `nFunctionCounter`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher

7.6 Pressure handling

7.6.1 FB_ProcessHandlerBase



This function block is used in injection molding machines to switch from injection pressure to holding pressure.

Syntax:

```
FUNCTION_BLOCK FB_ProcessHandlerBase IMPLEMENTS I_ProcessHandler
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
```

Properties

Name	Type	Description
Axis	I_AxisBase	An interface to the axis to be monitored.
BadSwitchOver	BOOL	TRUE if an undesired switchover occurred.
ProcessValue	LREAL	The process value to be monitored.
SwitchOver	BOOL	TRUE if an expected switchover occurred.

Methods

Name	Description
GetControlParameter [▶ 261]	This method can be used to get access to the parameter container of the axis pressure controller.
GetProcessValues [▶ 261]	This method is used to update the FB with a range of actual values (pressure, position, state) of the axis. **This method is called cyclically by the axis. The application is not intended to call this method directly.**
GetSwitchEnable [▶ 262]	This method is used to read back the rules for the switchover.

Name	Description
GetSwitchParameter [▶ 262]	This method may be used to read back parameters of the switchover rules.
SetControlParameter [▶ 263]	This method can be used to assign a parameter container to the axis pressure controller.
SetSwitchEnable [▶ 263]	This method is used to define the rules for the switchover. See below for more information.
SetSwitchParameter [▶ 263]	This method may be used to define parameters of the switchover rules.

If the monitored axis is in an inactive state (Init, Ready, Idle), the FB is inactive.

An active command of the axis also activates the monitoring.

The rules set with SetSwitchEnable determine the response:

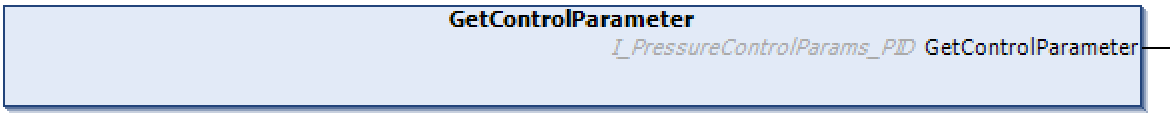
If no faulty switchover has been detected yet:

- If bSwitchOnPressure is TRUE and the process value exceeds ePressureThreshold and the axis was moved longer than eGardingTravel from the start point, an expected switchover is detected.
- If bSwitchOnTravel is TRUE and the axis has traveled longer than eTravelThreshold from the start point, an expected switchover is detected.
- If bSwitchOnTime is TRUE and the time since the start of monitoring exceeds the eTimeThreshold, an expected switchover is detected.
- If there is more than one active rule the first match will cause the expected switchover.

If there was still no switchover:

- If eTimeout is set to more than 0.0 and the time since the start of monitoring exceeds eTimeout, a faulty switchover is detected.
- If the axis position falls below eAlarmPositionLimit, a faulty switchover is detected.
- If eAlarmPressureLimit is set to more than 0.0 and the process value exceeds eAlarmPressureLimit, a faulty switchover is detected.

7.6.1.1 GetControlParameter

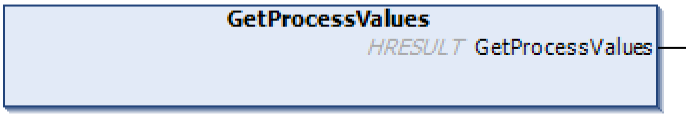


This method can be used to get access to the parameter container of the axis pressure controller.

Syntax:

```
METHOD GetControlParameter : I_PressureControlParams_PID
VAR_INPUT
END_VAR
```

7.6.1.2 GetProcessValue



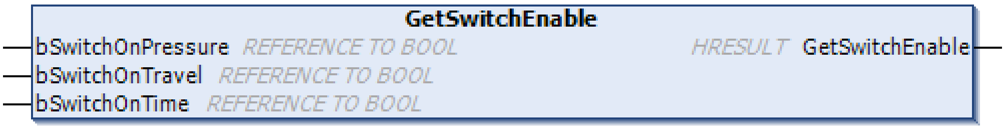
This method is used to update the FB with a range of actual values (pressure, position, state) of the axis.

This method is called cyclically by the axis. The application is not intended to call this method directly.

Syntax:

```
METHOD GetProcessValues : HRESULT
VAR
END_VAR
```

7.6.1.3 GetSwitchEnable


GetSwitchEnable
bSwitchOnPressure REFERENCE TO BOOL
bSwitchOnTravel REFERENCE TO BOOL
bSwitchOnTime REFERENCE TO BOOL
HRESULT GetSwitchEnable

This method is used to read back the rules for the switchover.

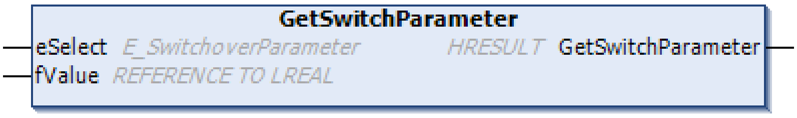
Syntax:

```
METHOD GetSwitchEnable : HRESULT
VAR_INPUT
bSwitchOnPressure: REFERENCE TO BOOL;
bSwitchOnTravel: REFERENCE TO BOOL;
bSwitchOnTime: REFERENCE TO BOOL;
END_VAR
```

Inputs

Name	Type	Description
bSwitchOnPressure	REFERENCE TO BOOL	A reference to the variable to be updated with the parameter.
bSwitchOnTravel	REFERENCE TO BOOL	A reference to the variable to be updated with the parameter.
bSwitchOnTime	REFERENCE TO BOOL	A reference to the variable to be updated with the parameter.

7.6.1.4 GetSwitchParameter


GetSwitchParameter
eSelect E_SwitchoverParameter
fValue REFERENCE TO LREAL
HRESULT GetSwitchParameter

This method may be used to read back parameters of the switchover rules.

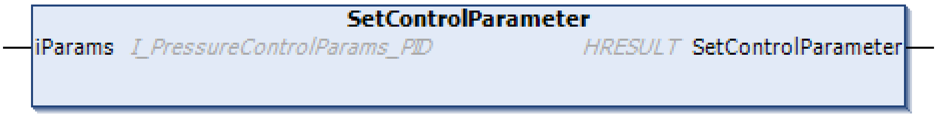
Syntax:

```
METHOD SetSwitchParameter : HRESULT
VAR_INPUT
eSelect: E_SwitchoverParameter;
fValue: REFERENCE TO LREAL;
END_VAR
```

Inputs

Name	Type	Description
eSelect	E_SwitchoverParameter [▶ 264]	The selection of the parameter.
fValue	REFERENCE TO LREAL	A reference to the variable to be updated with the parameter.

7.6.1.5 SetControlParameter


SetControlParameter
iParams I_PressureControlParams_PID
HRESULT SetControlParameter

This method can be used to assign a parameter container to the axis pressure controller.

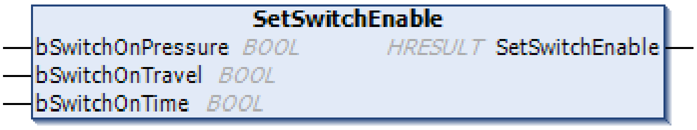
Syntax:

```
METHOD SetControlParameter : HRESULT
VAR_INPUT
iParams: I_PressureControlParams_PID;
END_VAR
```

Inputs

Name	Type	Description
iParams	I_PressureControlParams_PID	An interface to a FB with a parameter set for a pressure controller.

7.6.1.6 SetSwitchEnable



This method is used to define the rules for the switchover. See below for more information.

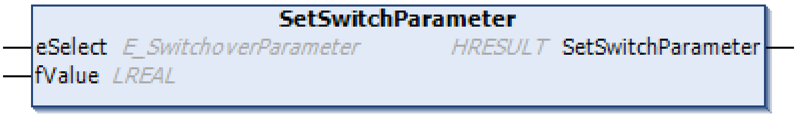
Syntax:

```
METHOD SetSwitchEnable : HRESULT
VAR_INPUT
bSwitchOnPressure: BOOL;
bSwitchOnTravel: BOOL;
bSwitchOnTime: BOOL;
END_VAR
```

Inputs

Name	Type	Description
bSwitchOnPressure	BOOL	A TRUE enables the switchover by exceeding the pressure threshold.
bSwitchOnTravel	BOOL	A TRUE enables the switchover by falling below a position threshold.
bSwitchOnTime	BOOL	A TRUE enables the switchover by reaching a time threshold.

7.6.1.7 SetSwitchParameter



This method may be used to define parameters of the switchover rules.

Syntax:

```
METHOD SetSwitchParameter : HRESULT
VAR_INPUT
eSelect: E_SwitchoverParameter;
fValue: LREAL;
END_VAR
```

Inputs

Name	Type	Description
eSelect	E_SwitchoverParameter [▶ 264]	The selection of the parameter.
fValue	LREAL	The value with which the parameter should be updated.

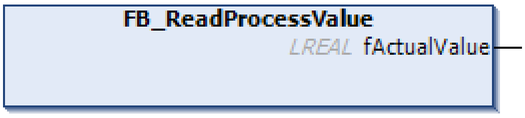
7.6.2 E_SwitchoverParameter

Values of this enumeration are use by GetSwitchParameter() and SetSwitchParameter() of FB_ProcessHandlerBase [▶ 260].

```
TYPE E_SwitchoverParameter :
(
ePressureThreshold := 0,
eGardingTravel,
eTravelThreshold,
eTimeThreshold,

eAlarmPressureLimit,
eAlarmPositionLimit,
eTimeout
);
END_TYPE
```

7.6.3 FB_ReadProcessValue



A function block of this type is used to supply a function block of type FB_ProcessHandlerBase [▶ 260] with actual values.

Syntax:

```
FUNCTION_BLOCK FB_ReadProcessValue EXTENDS FB_MessageBase IMPLEMENTS I_ReadProcessValue
VAR_INPUT
END_VAR
VAR_OUTPUT
fActualValue: LREAL;
END_VAR
```

Outputs

Name	Type	Description
fActualValue	LREAL	The current actual value.

Properties

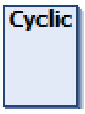
Name	Type	Access	Description
Offset	LREAL	Get, Set	The offset taken into account when calculating the actual value.
ProcessValue	LREAL	Get	The current process value calculated by the last call to Cyclic().

Name	Type	Access	Description
Weighting	LREAL	Get, Set	The scaling factor.

Methods

Name	Description
Cyclic [▶ 265]	This method must be called by the application once per cycle. It checks the EtherCAT connection and updates the ProcessValue with a filtered actual value.
DoReset [▶ 265]	A call of this method with bExecute=TRUE resets the connected sensor interface and clears all local errors.

7.6.3.1 Cyclic

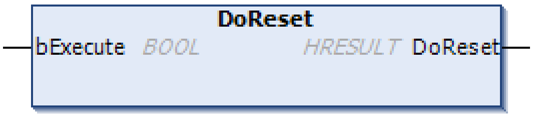


This method must be called by the application once per cycle. It checks the EtherCAT connection and updates the ProcessValue with a filtered actual value.

Syntax:

```
METHOD Cyclic
VAR_INPUT
END_VAR
```

7.6.3.2 DoReset



A call of this method with bExecute=TRUE resets the connected sensor interface and clears all local errors.

Syntax:

```
METHOD DoReset : HRESULT
VAR_INPUT
bExecute: BOOL;
END_VAR
```

7.7 PlcMcManager Support

This class establishes the connection between TwinCAT 3 Plastic Technology Functions hydraulic axes and the PlcMcManager.

Syntax:

```
FUNCTION_BLOCK FB_PlcMcManSupport
```

Initialization parameters of the FB_init constructor

Name	Type	Description
Path	STRING(40)	File path under which the hydraulic configuration is to be saved.

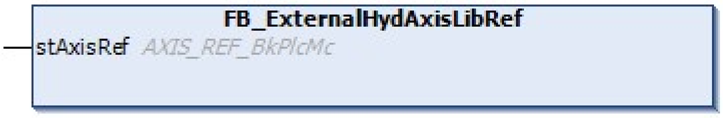
Methods

Name	Description
Append(`iAxis`)	Appends an axis from the internal list to be connected.
Cyclic()	Cycle method

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

7.8 ExternalHydAxisLibRef



This class represents a standard hydraulic axis of the Tc2_Hydraulic library (without using an FB_AxisHydraulicBase).

Syntax:

```
FUNCTION_BLOCK FB_ExternalHydAxisLibRef
VAR_INPUT
    stAxisRef : AXIS_REF_BkPlcMc;
END_VAR
```

Inputs

Name	Type	Description
stAxisRef	AXIS_REF_BkPlcMc	Reference to the hydraulic axis.

Properties

Name	Type	Access	Description
NextHydAxis	I_ExternalHydAxisLibRef	Get, Set	Interface to another hydraulic axis.
NextHydAxisChainlength	INT	Get	Length of the chain on hydraulic axes.
UseDatFile	BOOL	Get, Set	A TRUE here signals that the axis loads its parameters with function blocks of the hydraulics library from a file during startup. *Notice* **A TRUE must be set before the first Cyclic call.**

Methods

Name	Description
EnterCriticalSection()	[INTERNAL] Used internally.
GetHydAxisRef()	Returns a `POINTER` to the `AXIS_REF_BkPlcMc`.
LeaveCriticalSection()	[INTERNAL] Used internally.

Interfaces

Type	Description
I_ExternalHydAxisLibRef	Standard interface on FB_ExternalHydAxisLibRef.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.35	PC or CX (x64, x86)	Tc3_PlasticFunctions v3.12.4.26 or higher Tc3_PlasticHydraulic v3.12.4.26 or higher

More Information:
www.beckhoff.com/tf8560.html

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