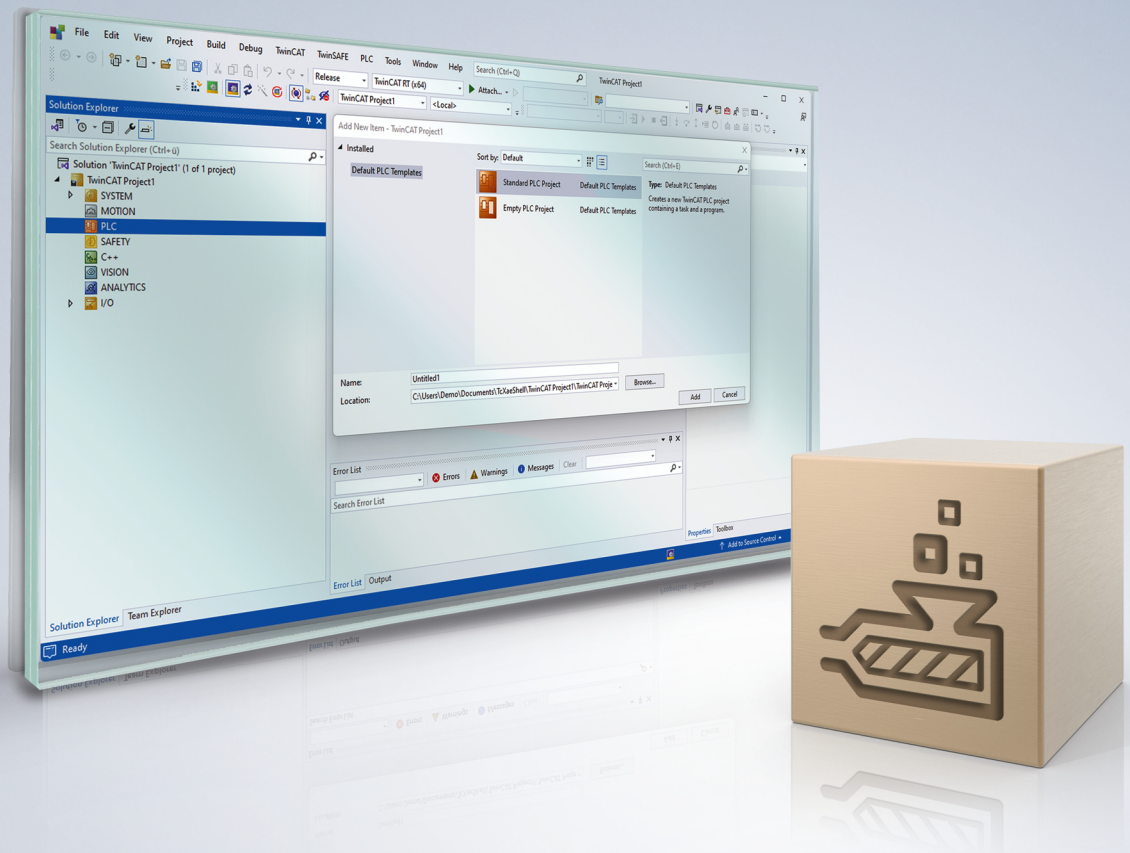
BECKHOFF New Automation Technology

Manual | EN

TF8560

TwinCAT 3 | Plastic Technology Functions



2026-08-17 | Version: 1.6.0

Table of contents

1 Foreword 9
1.1 Notes on the documentation 9
1.2 For your safety 10
1.3 Notes on information security 11
1.4 Documentation issue status 11
2 Installation 13
3 Concept of Libraries 14
4 Axes 16
4.1 FB_AxisBase - Virtual axes 16
4.2 Axis types 17
4.2.1 FB_AxisNcBase 17
4.2.2 FB_AxisNcTrafoBase 19
4.2.3 FB_AxisHydraulicBase 30
4.2.4 FB_AxisInvBase 32
4.2.5 FB_ActuatorBase 34
4.3 Embedding via interfaces 35
4.3.1 I_MotionBase 37
4.3.2 I_MotionBaseDev 37
4.3.3 I_AxisBase 39
4.3.4 I_AxisBaseDev 40
4.3.5 I_AxisNcBase 43
4.3.6 I_AxisNcBaseDev 44
4.3.7 I_AxisHydraulicBase 46
4.3.8 I_AxisHydraulicBaseDev 48
4.3.9 I_AxisInvBase 51
4.3.10 I_AxisInvBaseDev 52
4.3.11 I_AxisTrafoBase 54
4.3.12 I_AxisTrafoBaseDev 55
4.3.13 I_AxisNcTrafoBase 58
4.3.14 I_AxisNcTrafoBaseDev 60
4.3.15 I_ActuatorBase 62
4.3.16 I_ActuatorBaseDev 63
4.4 Properties and methods 65
4.5 States (state machine) 65
5 Core functions 68
5.1 Basic structure and variations 69
5.1.1 Core scope (FB_Corefunction) 70
5.1.2 Commandable variants (FB_CorefunctionFeedback) 71
5.1.3 Permanently active core functions 73
5.1.4 Statically controlled core functions 74
5.1.5 Edge-triggered core functions 76
5.2 Actuals 80
5.3 ActualsActuator 81

5.4 ActualsHydraulics ... 83
5.5 ActualsNc ... 84
5.6 AutoIdent ... 85
5.6.1 DoAutoIdent() ... 87
5.6.2 SetParameter() ... 88
5.7 Camming ... 90
5.7.1 DoCamming ... 93
5.7.2 FB_CammingLookUp ... 94
5.7.3 E_LookupMode ... 102
5.8 DirectOutput ... 103
5.8.1 DoActivate ... 105
5.9 DisableSoftEnd ... 106
5.9.1 DoDisable ... 107
5.9.2 ReEnable ... 108
5.10 ExternalGenerating ... 109
5.10.1 DoEnable() ... 110
5.11 Estop ... 111
5.11.1 DoEstop ... 113
5.12 Homing ... 114
5.12.1 Abort ... 116
5.12.2 AbsoluteSwitch ... 118
5.12.3 AbsoluteSwitchDetect ... 122
5.12.4 Block ... 125
5.12.5 BlockDetect ... 129
5.12.6 Finish ... 132
5.12.7 LimitSwitch ... 135
5.12.8 LimitSwitchDetect ... 138
5.13 Jog ... 141
5.13.1 DoJogM() ... 143
5.13.2 DoJogP() ... 144
5.13.3 SetParameter() ... 145
5.14 Limiting ... 145
5.14.1 Transition ... 147
5.14.2 Mode ... 148
5.15 MotionParams ... 150
5.15.1 ActuatorParamsH ... 155
5.15.2 MdMotionParams ... 156
5.16 MotionSetpoints ... 157
5.17 Power ... 158
5.17.1 DoPower() ... 160
5.17.2 FeedEnable() ... 161
5.18 PressureControl ... 161
5.18.1 PressureControl.PID ... 162
5.18.2 PressureControl.Supervisor ... 171
5.19 Ptp ... 173
5.19.1 Blending ... 176

5.19.2 ClampingPoint ... 176
5.19.3 DoMove() ... 179
5.19.4 Profiles ... 180
5.19.5 PtpPoint ... 182
5.20 PtpLookUp ... 185
5.20.1 GetPoint() ... 186
5.20.2 ReadMaster() ... 187
5.20.3 SetPoint() ... 188
5.20.4 UpdatePosition() ... 188
5.21 Reset ... 189
5.22 SetPosition ... 191
5.22.1 DoSetPosition() ... 192
5.22.2 SetParameter() ... 193
5.23 Stop ... 194
5.23.1 DoStop() ... 195
5.24 Teach ... 196
5.25 ToolAdaption ... 199
5.26 Transformation ... 200
5.26.1 TranslateFromDrive() ... 203
5.26.2 TranslateFromLoad() ... 203
5.26.3 TranslateToDrive() ... 204
5.26.4 TranslateToLoad() ... 205
5.26.5 UpdateToLinLoad() ... 206
5.27 Tuning ... 206
5.28 VelocityFeed ... 208
5.28.1 DoFeed() ... 210

6 I/O ... 211
6.1 Base (Abstract) ... 212
6.1.1 IoBase ... 212
6.1.2 IoEc ... 214
6.1.3 IoEcAds ... 217
6.1.4 InputBase ... 218
6.1.5 OutputBase ... 221
6.2 Analog ... 227
6.2.1 InputAnalog16 ... 227
6.2.2 InputAnalog16Status ... 229
6.2.3 InputAnalog32 ... 231
6.2.4 OutputAnalog16 ... 233
6.3 CoE ... 234
6.3.1 InputCoE_Polling ... 234
6.3.2 InputCoE406 ... 235
6.3.3 OutputCoE408 ... 237
6.3.4 OutputCoE408_P ... 240
6.4 Digital ... 242
6.4.1 IoDigital ... 242
6.4.2 InputDigital ... 243

6.4.3 InputDigitalCam ... 244
6.4.4 OutputDigital ... 245
6.5 Serial ... 246
6.5.1 InputSsi ... 246
6.6 Device (inverter) ... 247
6.6.1 InvDeviceBase ... 247
6.6.2 InvDeviceEc ... 248
6.6.3 InvDeviceCoE402 ... 250
6.6.4 InverterCoE402_CSVM ... 253
6.6.5 InverterCoE402_VM ... 255
6.6.6 InverterAnalog ... 256
6.7 LimitingOutput ... 258
6.7.1 LimitingOutputCoE ... 258
6.7.2 LimitingOutputSoE ... 259
6.8 ST_FlexValue ... 260
6.8.1 E_FlexValue ... 260
6.8.2 U_FlexValue ... 261

7 MachineData ... 263
7.1 FB_MdContainer ... 263
7.1.1 E_MdFileType ... 264
7.1.2 I_MdJsonOptions ... 264
7.2 FB_MdComponent ... 265
7.2.1 MdNextParameter() ... 266
7.2.2 MdSetParameter() ... 266
7.3 ST_Mdata ... 267

8 Utilities ... 268
8.1 Filter ... 268
8.1.1 FB_FilterBase ... 268
8.1.2 FB_FilterPt1 ... 269
8.1.3 FB_FilterSlewRateLimit ... 270
8.1.4 FB_FilterSrlPt1 ... 270
8.2 HRESULT ... 271
8.2.1 F_SucceededHr ... 272
8.2.2 F_HresultCreate ... 272
8.2.3 F_HresultFailure ... 273
8.2.4 F_HresultSuccess ... 274
8.3 Messages ... 274
8.3.1 FB_Message ... 274
8.3.2 FB_XyzMessage ... 275
8.3.3 FB_MessageBase ... 276
8.4 Services ... 277
8.4.1 FB_CoEService ... 277
8.4.2 FB_CoEParameter ... 278
8.4.3 FB_CheckIoMapping ... 279
8.5 Simulation ... 280

8.5.1 General simulation components ... 280
8.5.2 Simulation of an EtherCAT based servo drive axis ... 285
8.5.3 Simulation of an inverter axis ... 293
8.5.4 I/O Simulation containers ... 307
8.5.5 Components of the hydraulic simulation ... 310
8.5.6 SimActuator ... 318
8.5.7 Injection pressure simulation ... 319
8.6 Debug ... 321
8.6.1 FB_CheckDemoMode ... 321
8.6.2 FB_ChangeDetector ... 322
8.6.3 FB_CoreDebug ... 322
8.7 PlcMcManager Support ... 323
8.8 ExternalHydAxisLibRef ... 324

9 FAQ ... 326
9.1 Hydraulic ... 326

10 Appendix ... 327
10.1 Camming ... 327
10.1.1 SetGuidingValue() ... 327
10.1.2 SetLookupInterface() ... 328
10.2 PressureControl ... 329
10.2.1 I_AxisBase.SetProcessHandler() ... 329
10.2.2 PID.EnableSwitchOver() ... 329
10.2.3 PID.GetActual() ... 330
10.2.4 PID.GetParams() ... 330
10.2.5 PID.SetParams() ... 331
10.2.6 PID.Setpoint() ... 331
10.2.7 FB_ProcessHandlerBase ... 332
10.2.8 FB_ReadProcessValue ... 336
10.3 Ptp ... 337
10.3.1 CheckPoint() ... 338
10.3.2 GetClampPoint() ... 338
10.3.3 GetPoint() ... 339
10.3.4 GetUpdatedPoint() ... 340
10.3.5 SetClampPoint() ... 341
10.3.6 SetPoint() ... 342
10.3.7 UpdatePosition() ... 342
10.4 Stop ... 343
10.4.1 SetParameter() ... 343
10.5 TeachPosition ... 344
10.6 TeachUpdate ... 345
10.7 TorqueLimiting ... 346

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.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled 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.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.





EtherCAT® is a 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.

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

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.

1.4 Documentation issue status

Version	Change
1.6.x	**New:** LimitingOutputCoE [▶ 258] LimitingOutputSoE [▶ 259] FAQ [▶ 326]
1.5.x	**New:** How a transformation axis works [▶ 20]: Information on semi transformation and full transformation mode MdMotionParams [▶ 156] Blending [▶ 176] ClampingPoint [▶ 176] Profiles [▶ 180] PtpPoint [▶ 182] Teach [▶ 196] FB_CheckIoMapping [▶ 279] Injection pressure simulation [▶ 319]
1.4.x	Software v12.10.0 - new features added Properties and methods supplemented in several function blocks **New:** FB_CrankDoubleShaftTableGenerator [▶ 26] PressureControl [▶ 329]
1.3.x	**New:** FB_SimCoE402_Inverter [▶ 294]
1.2.x	Additions in the chapters: Axes [▶ 16], Core functions [▶ 68], I/O [▶ 211], MachineData [▶ 263], Utilities [▶ 268], Appendix [▶ 327]
1.1.x	**New:**

Version	Change
	I/O [▶ 211]
1.0.x	Initial creation

2 Installation

The TF8560 Plastic Technology Functions can be installed under TwinCAT 4026 via the TwinCAT Package Manager. A separate workload is available for this.



Use in TwinCAT 3.1.4024

You can also use the TwinCAT Package Manager to maintain a TwinCAT 4024 project with a TwinCAT 4026 Engineering installation and a corresponding Build 4024 Remote Manager version. If you are using a TwinCAT 4024 Engineering installation, please contact Plastics@beckhoff.com directly.

3 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.

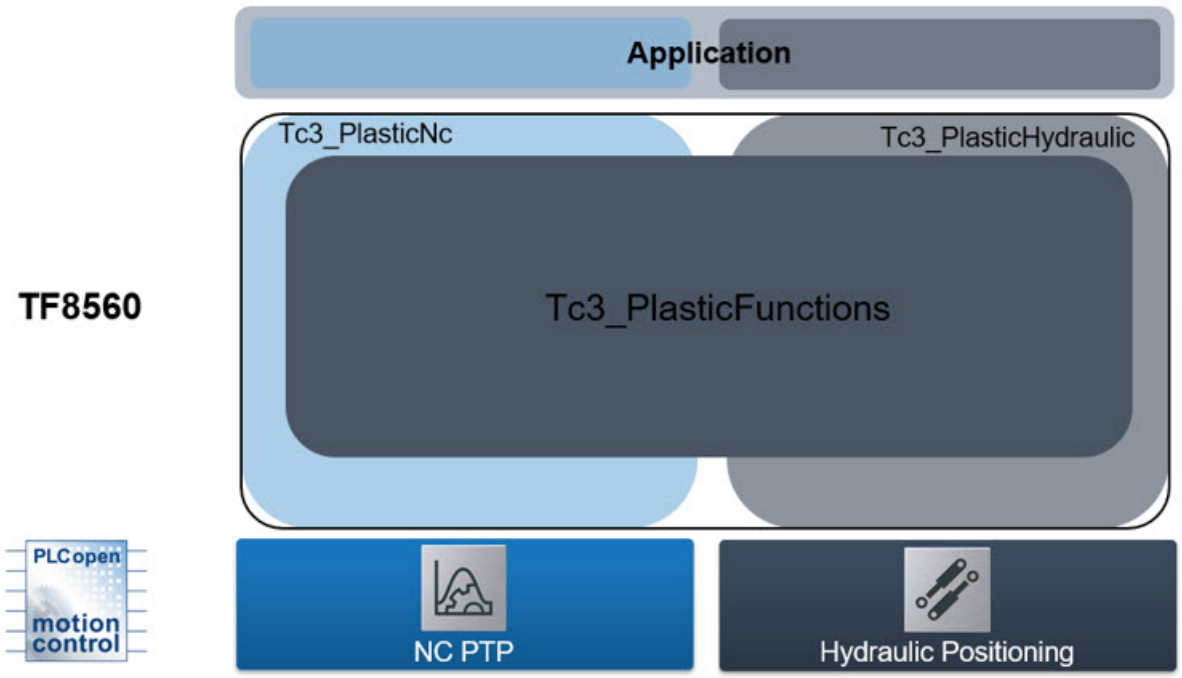
In addition, the elementary motion tasks commonly used in plastics processing, such as cam plates for wall thickness control, multi-segment PTP motion, and pressure control, have been implemented and fully tested in TC3 Plastic Technology Functions, encapsulated as core functions. Customers are freed 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
Tc3_PlasticFunctions
TF8560
PLCopen
motion control
NC PTP
Hydraulic Positioning

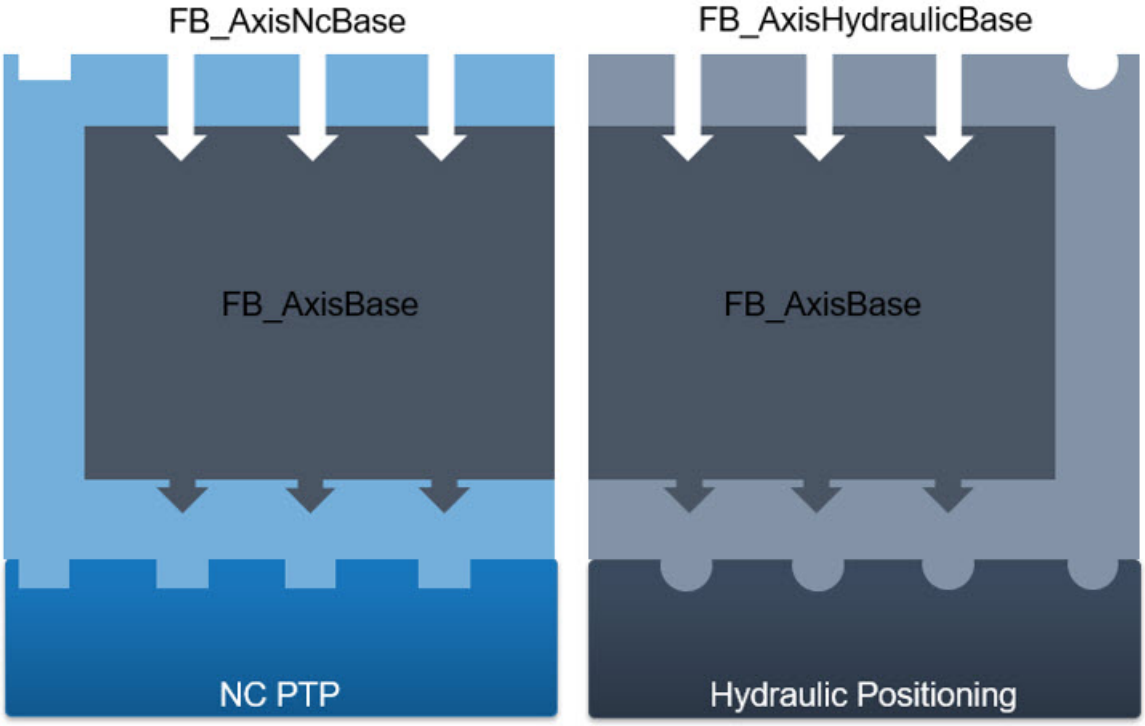
4 Axes

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".

4.1 FB_AxisBase - Virtual axes

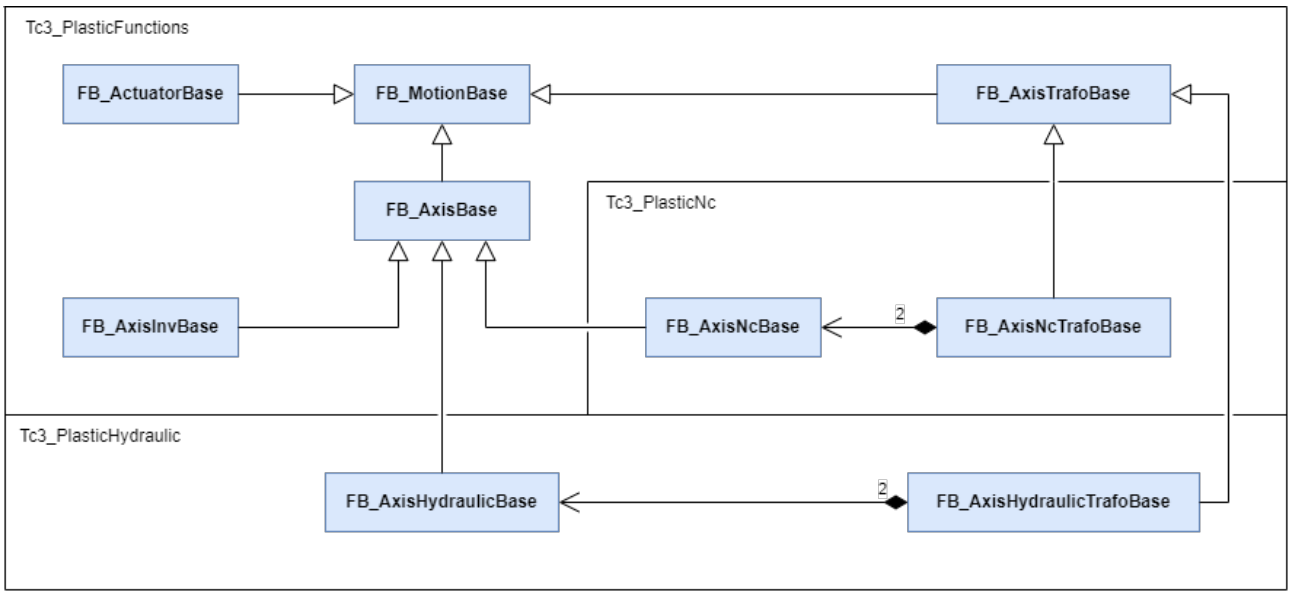
TwinCAT 3 Plastic Technology Functions defines `FB_AxisBase` as a virtual base axis. This provides the core functions required for both electric and hydraulic drives. The ABSTRACT definition means that `FB_AxisBase` cannot be instantiated. This is intentional, as the connection to specific Motion Control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



4.2 Axis types

Any of the axis types listed in the following table can be instantiated as desired. In each section, you will find sample code for instantiating these axis types, along with notes on the configurable hardware connections. Some of these connection interfaces are already defined by the FB_init() function. However, this approach is outdated in the context of this library and will be replaced by runtime assignment in future developments. Assignment is still supported for existing FB_init() inputs.

In addition, the Open-source project TF85xx – Plastic Application, which is based on TF8560 – TwinCAT 3 Plastic Technology Functions, provides common machine axes and their motion tasks for 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.

Name	Description
FB_AxisHydraulicBase [▶ 30]	Hydraulic axis, operated with the Tc2_Hydraulics library.
FB_AxisNcBase [▶ 17]	Servo axis, operated with the Tc2_MC2 library.
FB_AxisNcTrafoBase [▶ 19]	Transforming servo axis, operated with the Tc2_MC2 library.
FB_AxisInvBase [▶ 32]	Inverter axis
FB_ActuatorBase [▶ 34]	Digital linear actuator

4.2.1 FB_AxisNcBase



This FB creates an axis operated with the Tc2_MC2 library.

This axis type requires the TF50x0 NC PTP license.

Syntax:

- Runtime-based:

```
fbNcAxis:        FB_AxisNcBase('', 0, 0, 0, 0, 0);

// Initialization code
fbNcAxis.Name := 'NcAxis1'
fbNcAxis.PtpPoints := 10;
```

- Declarative (outdated):

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

Connection options (I/Os)

Name	Type	Description
AxisRef	AXIS_REF	Mapping interface to the TwinCAT NC axis.
PressureControl [▶ 161].Supervisor [▶ 171].Sensor	I_InputBase [▶ 218]	Interface to a pressure sensor for pressure control.
Limiting [▶ 145].Output	I_LimitingOutput [▶ 258]	Interface for the connection of limit outputs (e.g., torque or pressure).

AxisRef is not available as a property

AxisRef is listed here for the sake of completeness, but it cannot be accessed directly via a property.

Initialization parameters of the FB_init constructor

Name	Type	Description
AxisName	STRING	The text-based name of the axis. Used for messages and file names, among other things. **The name can also be set via `.Name`.**
nPtpPoints	INT	Number of usable points in the PTP function. **The number of points can also be set via `.PtpPoints`**
iProcessHandler	I_ProcessHandler	A function block for handling pressures and other process variables. **Outdated implementation! This input must be set to zero, since the availability of pressure control is already high in PressureControl [▶ 161].Supervisor [▶ 171] without requiring an external assignment.**
iPosCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a position cam plate. **Outdated implementation! This input must be set to zero, since a cam plate is already available at Camming [▶ 90].Pos.LookUp without an external assignment.**
iVeloCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a speed cam plate. **Outdated implementation! This entry must be set to "zero" because a cam plate is already available at Camming [▶ 90].Velo.LookUp without an external assignment.**

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

4.2.2 FB_AxisNcTrafoBase



This FB creates a transforming axis operated with the Tc2_MC2 and Tc2_MC2_Camming libraries. This requires two NC axes, each of which must be configured as either the drive side or the load side. A transformation table then converts, for example, a rotary drive position into a virtual linear position (load position) characterized by a calibration curve. During operation, this allows the axis to be controlled to the virtual linear position. For further details on how the transformation axes work, please refer to the Section on transformation [▶ 20].

This axis type requires the TF50x0 NC PTP license and the TF5050 NC Camming license.

Syntax:

- Runtime-based:

```
fbNcTrafoAxis:          FB_AxisNcTrafoBase('', 0, 0, 0, 0, 0, 0);

// Initialization code
fbNcTrafoAxis.Name := 'NcTrafoAxis'
fbNcTrafoAxis.PtpPoints := 10;
fbTrafoTable.SetPoints(181); // FB_CammingLookUp [▶ 94]
// ... (further init steps of the trafo table)
fbNcTrafoAxis.Transformation.Table := fbTrafoTable;
```

- Declarative (outdated):

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

Connection options (I/Os)

Name	Type	Description
fbDriveAxis.AxisRef	AXIS_REF	Mapping interface to the drive side as a TwinCAT NC axis.
fbLoadAxis.AxisRef	AXIS_REF	Mapping interface to the load side as a TwinCAT NC axis.
fbActualsNc.nDataIn1	UINTARR2	Mapping interface to the encoder input on the load side as a TwinCAT NC axis. This provides the simulated position of the load side. Tree path: Enc > Inputs > In > nDataIn1
fbActualsNc.nState4	USINT	Mapping interface to the encoder input on the load side as a TwinCAT NC axis. This provides the simulated status of the load side. Tree path: Enc > Inputs > In > nState4
PressureControl [▶ 161].Supervisor [▶ 171].Sensor	I_InputBase [▶ 218]	Interface to a pressure sensor for pressure control.
Limiting [▶ 145].Output	I_LimitingOutput [▶ 258]	Interface for the connection of limit outputs (e.g., torque or pressure).

Some of the connection options are not available as properties

The respective AxisRef values and the encoder mappings on the load side are listed here for the sake of completeness; however, they cannot be accessed directly via a property.

Initialization parameters of the FB_init constructor

Name	Type	Description
AxisName	STRING	The text-based name of the axis. Used for messages and file names, among other things. **The name can also be set via** `.Name`**.**
nPtpPoints	INT	Number of usable points in the PTP function. **The number of points can also be set via** `.PtpPoints`
nTrafoPoints	INT	Number of usable transformation points in the transformation table. **Outdated implementation! This value should be set to 0, since the number of points is automatically retrieved from the transformation table being used.**
iProcessHandler	I_ProcessHandler	A function block for handling pressures and other process variables. **Outdated implementation! This input must be set to zero, since the availability of pressure control is already high in** PressureControl [▶ 161]**.**Supervisor [▶ 171] **without requiring an external assignment.**
iPosCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a position cam plate. **Outdated implementation! This input must be set to zero, since a cam plate is already available at** Camming [▶ 90]**.Pos.LookUp without an external assignment.**
iVeloCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a speed cam plate. **Outdated implementation! This entry must be set to "zero" because a cam plate is already available at** Camming [▶ 90]**.Velo.LookUp without an external assignment.**

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

4.2.2.1 How a transformation axis works

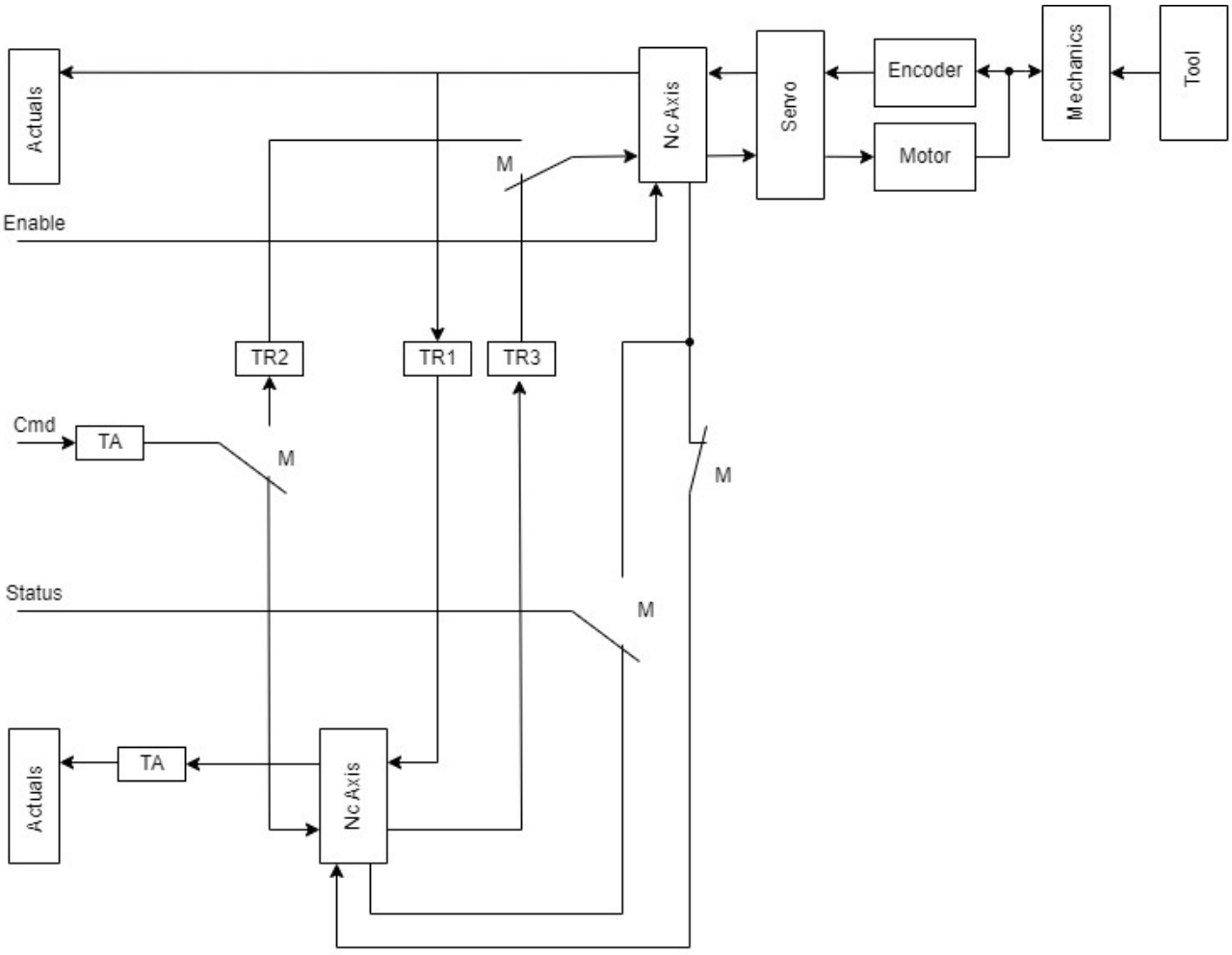
The mechanical solution for some axis types requires the design of a non-linear lever mechanism. The tool path does not follow the drive movement in a constant ratio. This enables the optimization of axis properties for varying requirements in individual areas of the travel. For the best possible embedding 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 [▶ 26]
- Scotch yoke [▶ 28]
- customized concepts. [▶ 23]
- Etc…

Abstracted circuit principle

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.

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 converted values from the drive side.

Any commanded motion will be executed by the drive side using converted 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 converted

The commanded velocity cannot be converted 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 move with constant velocity.

No position or velocity camming

Position or velocity camming is not supported.

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 converted 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 move with constant velocity.

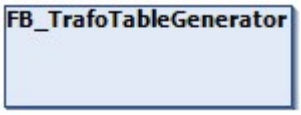
Excessive drive side velocity values required

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 conversion.

4.2.2.1.1 FB_TrafoTableGenerator



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

The following pre-implemented geometries are included:

1. FB_ClampTableGenerator [▶ 23] - Geometry of a typical clamp
2. FB_CrankTableGenerator [▶ 26] - Geometry of a crank drive
3. FB_ScotchYokeTableGenerator [▶ 28] - Geometry of a Scotch yoke
4. FB_CrankDoubleShaftTableGenerator [▶ 29] - Crank drive with drive lever on both sides

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_TrafoTableGenerator
```

Properties

Name	Type	Access	Description
LookUp	I_CammingLookUp [▶ 94]	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.

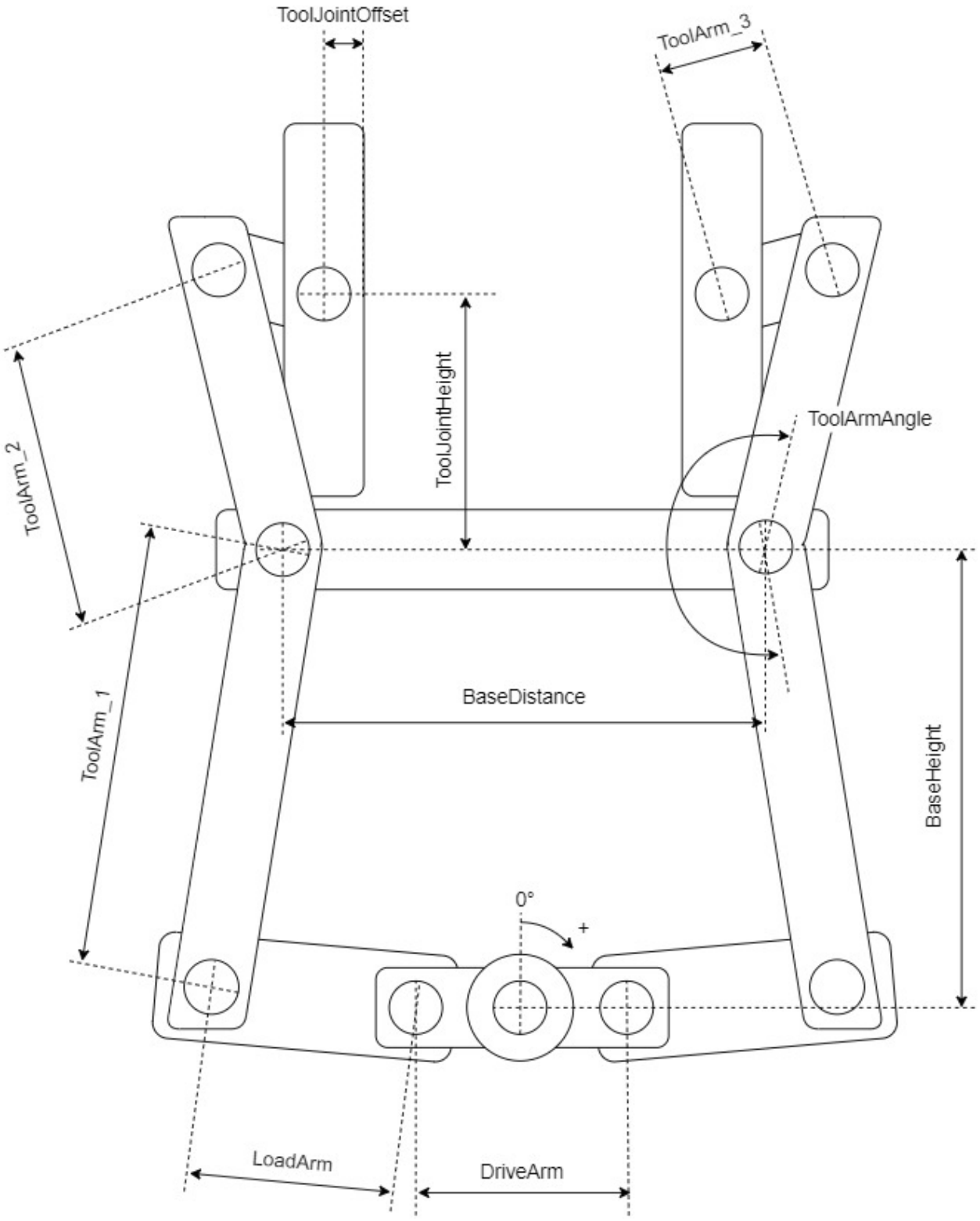
Requirements

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

4.2.2.1.2 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 [▶ 94]	Get, Set	Interface to the table to be written
AsymHalf	I_ClampTableGenerator_Half	Get	Interface to an asymmetrical half of the clamp. **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 from 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.

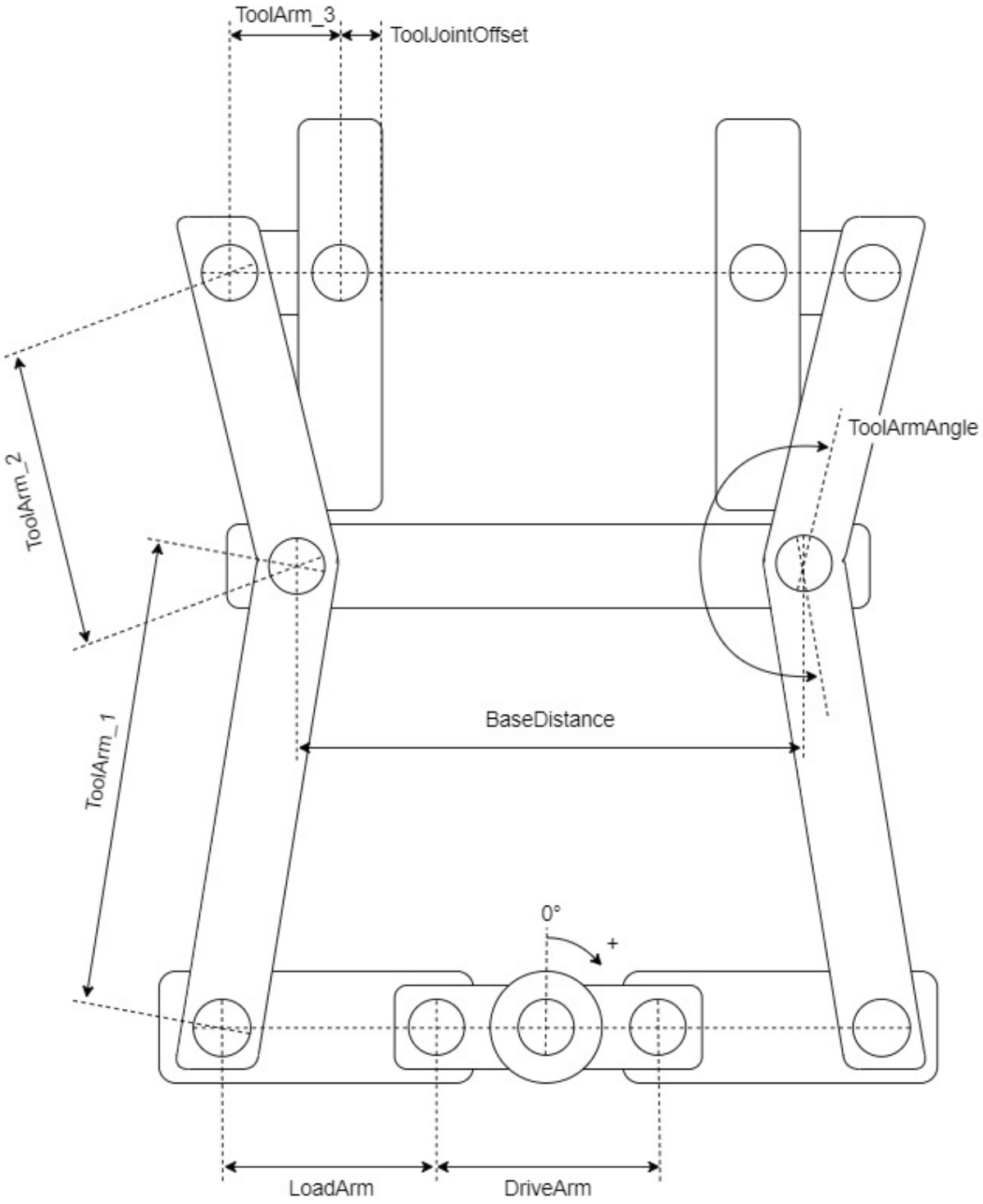
Methods

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.

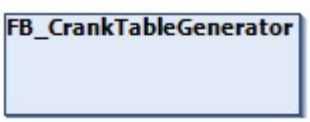
Drawing without optional parameters



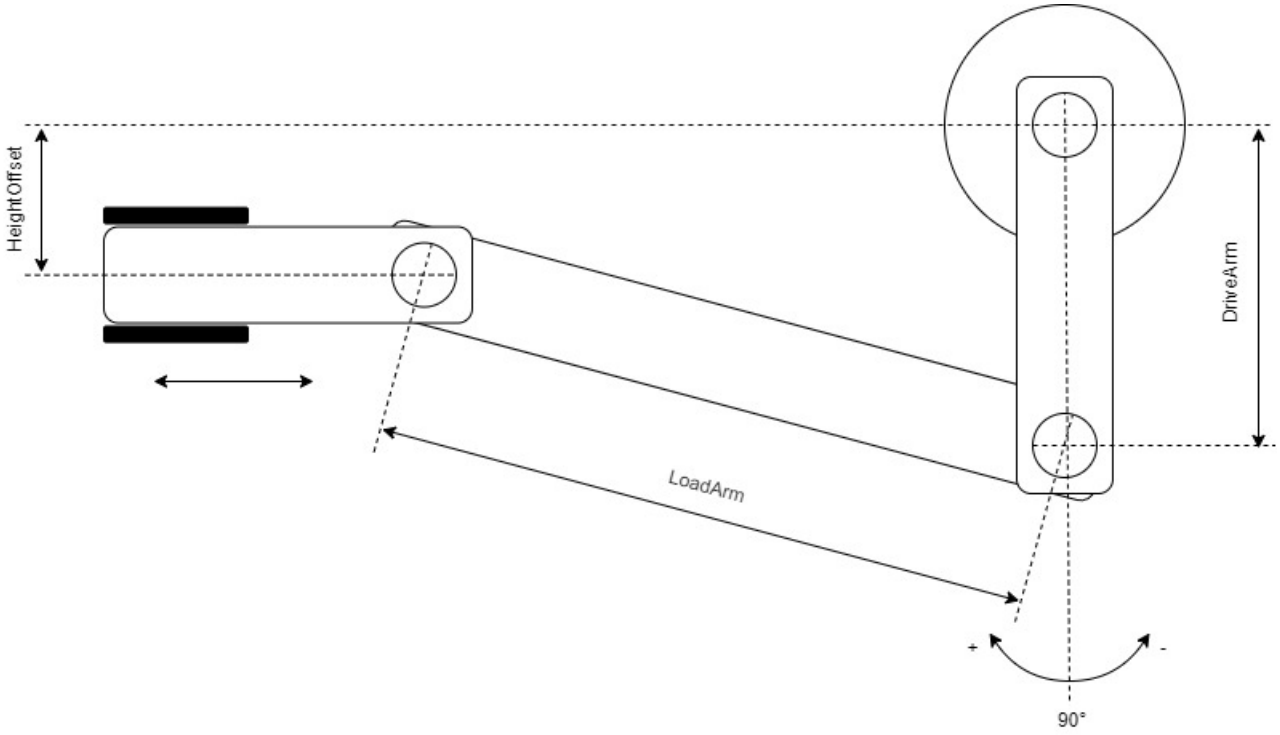
Requirements

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

4.2.2.1.3 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 [▶ 94]	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.

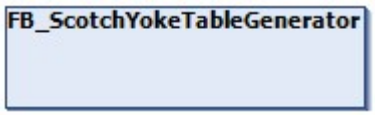
Interfaces

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

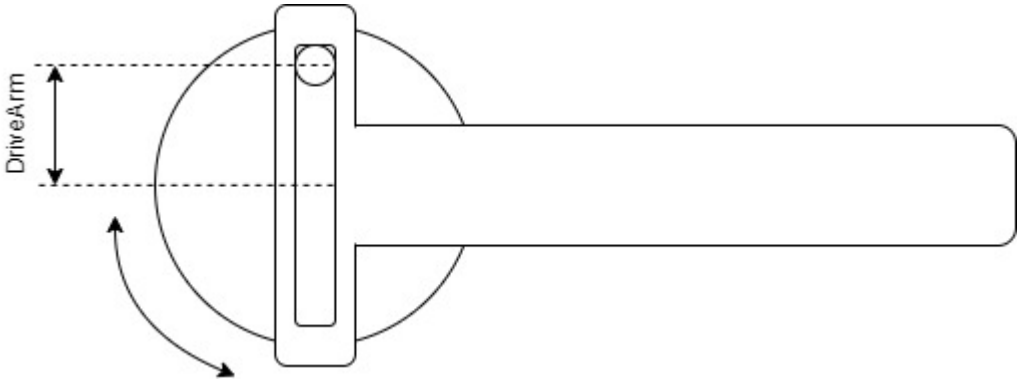
Requirements

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

4.2.2.1.4 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 [▶ 94]	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.
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.

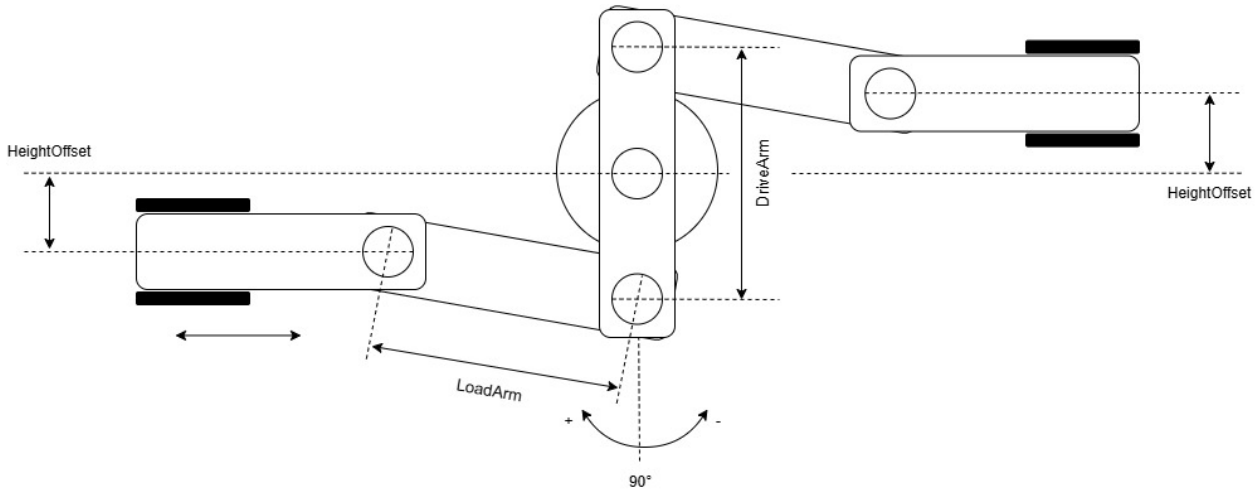
Requirements

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

4.2.2.1.5 FB_CrankDoubleShaftTableGenerator



Implements a calculation algorithm for a crank mechanism with drive levers on both sides.



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 [▶ 94]	Get, Set	Interface to the table to be written

Name	Type	Access	Description
AsymHalf	I_CrankDoubleShaftTableGenerator_Half	Get	Interface to an asymmetrical half of the crank drive. **If this property is called at least once, the class switches to asymmetrical operation mode.**
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.

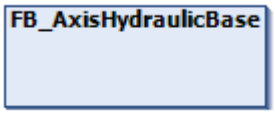
Interfaces

Type	Description
I_TrafoTableGenerator	Standard interface on FB_TrafoTableGenerator.
I_CrankDoubleShaftTableGenerator	Standard interfaces on FB_CrankDoubleShaftTableGenerator.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

4.2.3 FB_AxisHydraulicBase



This FB creates an axis operated with the Tc2_Hydraulics library. To perform commissioning using PlcMcManager, the axis must be connected to an FB_PlcMcManSupport [▶ 323] after instantiation.

This axis type requires the TF5810 Hydraulic Positioning license.

Syntax:

- Runtime-based:

```
fbHydAxis: FB_AxisHydraulicBase('', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);

// Initialization code
fbHydAxis.Name := 'HydAxis1'
fbHydAxis.PtpPoints := 10;
fbHydAxis.Encoder := fbEncoder;          // e.g. FB_InputSsi [▶ 246];
```

```
fbHydAxis.Drive := fbDrive;             // e.g. FB_OutputCoE408 [▶ 237];
fbHydAxis.PressureInputP := fbPressureP;// e.g. FB_InputAnalog16 [▶ 227]
fbHydAxis.PressureInputN := fbPressureN;// e.g. FB_InputAnalog16 [▶ 227]
fbHydAxis.VeloFilter := fbVeloFilter;   // e.g. FB_FilterPt1 [▶ 269]

fbPlcMcManSupport.Append(fbHydAxis);
```

- Declarative (outdated):

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

Connection options (I/Os)

Name	Type	Description
Encoder	I_InputBase [▶ 218]	I/O interface for the encoder. This interface is used to determine the actual position of the axis.
Drive	I_OutputBase [▶ 221]	I/O interface for the drive. This interface is used to output the target velocity of the axis.
PressureInputP	I_InputBase [▶ 218]	I/O interface for a pressure sensor. This interface is used to determine the pressure on the positive cylinder surface.
PressureInputM	I_InputBase [▶ 218]	I/O interface for a pressure sensor. This interface is used to determine the pressure on the negative cylinder surface.
PosFilter	I_Filter [▶ 268]	Interface for a filter of the actual position.
VeloFilter	I_Filter [▶ 268]	Interface for a filter of the actual velocity.
PressureControl [▶ 161].Supervisor [▶ 171].Sensor	I_InputBase [▶ 218]	Interface to a pressure sensor for pressure control.
Limiting [▶ 145].Output	I_LimitingOutput [▶ 258]	Interface for the connection of limit outputs (e.g., torque or pressure).

Strong filtering

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

Initialization parameters of the FB_init constructor

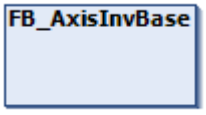
Name	Type	Description
AxisName	STRING	The text-based name of the axis. Used for messages and file names, among other things. **The name can also be set via `.Name`.**
nPtpPoints	INT	Number of usable points in the PTP function. **The number of points can also be set via `.PtpPoints`**
iProcessHandler	I_ProcessHandler	A function block for handling pressures and other process variables.

Name	Type	Description
		Outdated implementation! This input must be set to zero, since the availability of pressure control is already high in PressureControl [▶ 161].Supervisor [▶ 171] **without requiring an external assignment.**
iPosCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a position cam plate. **Outdated implementation! This input must be set to zero, since a cam plate is already available at** Camming [▶ 90]**.Pos.LookUp without an external assignment.**
iVeloCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a speed cam plate. **Outdated implementation! This entry must be set to "zero" because a cam plate is already available at** Camming [▶ 90]**.Velo.LookUp without an external assignment.**
iEncoder	I_InputBase [▶ 218]	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 [▶ 221]	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 [▶ 218]	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 [▶ 218]	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 [▶ 268]	Interface for a filter of the actual position. **Can also be assigned via a property**
iVeloFilter	I_Filter [▶ 268]	Interface for a filter of the actual velocity. **Can also be assigned via a property**

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

4.2.4 FB_AxisInvBase



This FB creates an axis operated with an inverter. This type of axis is designed for use with drive components whose functionality excludes the positioning tasks of a servo axis. Therefore, this type of axis is suitable for particularly simple speed-based tasks, such as rotating an extruder or feeding a conveyor belt. The axis does not require position feedback and does not support position-dependent Corefunctions.

Syntax:

- Runtime-based:

```
fbInvAxis:        FB_AxisInvBase('', 0, 0, 0, 0);

// Initialization code
fbInvAxis.Name := 'InverterAxis1';
fbInvAxis.Device := fbDeviceCoE; // e.g. FB_InverterCoE402_VM [▶ 255]
```

- Declarative (outdated):

```
fbInvAxis:  FB_AxisInvBase
            (
                AxisName        := 'InverterAxis1',
                nPtpPoints      := 0,
                iProcessHandler := 0,
                iPosCamLookup   := 0,
                iVeloCamLookup  := 0
            );
```

Connection options (I/Os)

Name	Type	Description
Device	I_InvDevice [▶ 247]	Field device used to control the axis
PressureControl [▶ 161].Supervisor [▶ 171].Sensor	I_InputBase [▶ 218]	Interface to a pressure sensor for pressure control.
Limiting [▶ 145].Output	I_LimitingOutput [▶ 258]	Interface for the connection of limit outputs (e.g., torque or pressure).

Initialization parameters of the FB_init constructor

Name	Type	Description
AxisName	STRING	The text-based name of the axis. Used for messages and file names, among other things. **The name can also be set via `.Name`.**
nPtpPoints	INT	Not supported for inverter axes
iProcessHandler	I_ProcessHandler	A function block for handling pressures and other process variables. **Outdated implementation! This input must be set to zero, since the availability of pressure control is already high in PressureControl [▶ 161].Supervisor [▶ 171] without requiring an external assignment.**
iPosCamLookup	I_CammingLookup [▶ 94]	Not supported for inverter axes
iVeloCamLookup	I_CammingLookup [▶ 94]	An interface to an external instance with a speed cam plate. **Outdated implementation! This entry must be set to "zero" because a cam plate is already available at Camming [▶ 90].Velo.LookUp without an external assignment.**

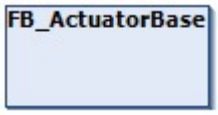
Requirements

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

Also see about this

- Device (inverter) [▶ 247]

4.2.5 FB_ActuatorBase



This FB creates an instance of a linear Boolean actuator. These include, for example, pneumatic or hydraulic valves with connected cylinders. For control, the actuator shares the common interface I_MotionBase [▶ 37] with the other axis types. This allows simple motion functions to be programmed regardless of the compact drive technology used: for example, a pneumatic handling device could be replaced by a compact servo axis via the interface.

Syntax:

```
fbActuator:              FB_ActuatorBase;

// Initialization code
fbActuator.FeedbackBase := fbActrFeedback; // FB_InputDigital [▶ 243]
fbActuator.OutputWork := fbActrOutput;     // FB_OutputDigital [▶ 245]
```

Connection options (I/Os)

Name	Type	Description
FeedbackBase	I_InputBase [▶ 243]	Feedback signal for reaching the base position
FeedbackWork	I_InputBase [▶ 243]	Feedback signal for reaching the working position
OutputBase	I_OutputBase [▶ 245]	Output signal for control in base position
OutputWork	I_OutputBase [▶ 245]	Output signal for control in working position
FeedbackPower	I_InputBase [▶ 243]	Feedback signal for the successful release of active operation. Can be connected to pressure monitoring of the supply, for example.
OutputPower	I_OutputBase [▶ 245]	Output signal for enabling the actuator. Can be connected to a supply shut-off valve, for example.

Availability of connection options

The connection options are available as properties of the axis and can be linked once to an I/O instance: e.g. `fbActuator.OutputWork := fbOutputWork;`

Necessity of individual connections

All linkable device connections (I/Os) of an actuator are optional,

- Missing outputs limit the possible commands
- Missing inputs are simulated with a parameterizable time [▶ 155]

Only digital I/O devices possible

Despite the use of general I/O interfaces for the device connections, only digital options can currently be used. An extension is planned for future versions.

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)

4.3 Embedding via interfaces

The entire virtualization principle of the TF8560 Plastic Technology Functions is based on the use of interfaces. Therefore, the library provides a corresponding `INTERFACE` for each axis type and common base types. For a general description of interfaces, see the INTERFACE concept in TwinCAT PLC programming.

Application-specific abstraction

When integrating a TF8560 axis into an application, it is recommended to select the interface with the largest overlap in variants. Specifically, this can be illustrated by the following use cases:

- For the carriage unit in a blow molding application, the appropriate interface, I_AxisBase [▶ 39], should be preferred.
  - This makes it possible to operate the carriage axis both **hydraulically** and **electrically**, regardless of the machine-specific configuration. In addition, it is possible to switch flexibly between an electrically **linear** axis and an electrically **transforming** axis.
- For the removal gripper of any plastics machine, the I_MotionBase [▶ 37] interface would be the preferred choice.
  - This makes it possible to use both digitally controllable **pneumatic** and **hydraulic cylinders** for control purposes. Furthermore, it is also possible to configure an **electric servo axis** that can use the same jog commands to move between the end positions.

Beyond their purpose of abstraction, the interfaces for each axis type offer a pre-filtered choice of axis functions, which significantly simplifies IntelliSense-guided programming in TwinCAT.

Therefore, do not access an axis directly in your program code; instead, use interfaces and initialize them with the appropriate axis instances.

```
fPosition := fbNcAxis1.Actuals.Position   // not preferred
fPosition := iAxisBase.Actuals.Position   // preferred
```

Querying the axis type and variant-specific handling

In some cases, despite the abstraction provided by, for example, I_AxisBase [▶ 39], the specific functionality of a particular axis type may be required. For example, this can occur with hydraulic axes when the characteristic curve measurement is about to begin. In this case, access to an I_AxisBase interface [▶ 39] is sufficient. Using the system function `QUERYINTERFACE`, you can run this query and gain access at the same time:

In the following example, the return value of the `QUERYINTERFACE` statement indicates whether the instance behind `iAxisBase` is compatible with the `iAxisHydraulicBase` interface (I_AxisHydraulicBase [▶ 46]). This applies, for example, to an FB_AxisHydraulicBase [▶ 30]. Accordingly, this statement can be used to check whether a hydraulic axis is defined in `iAxisBase`. If this is the case (`__QUERYINTERFACE(…) = TRUE`), the interface to be tested can be accessed directly, since the axis is already assigned during the check performed by `QUERYINTERFACE`.

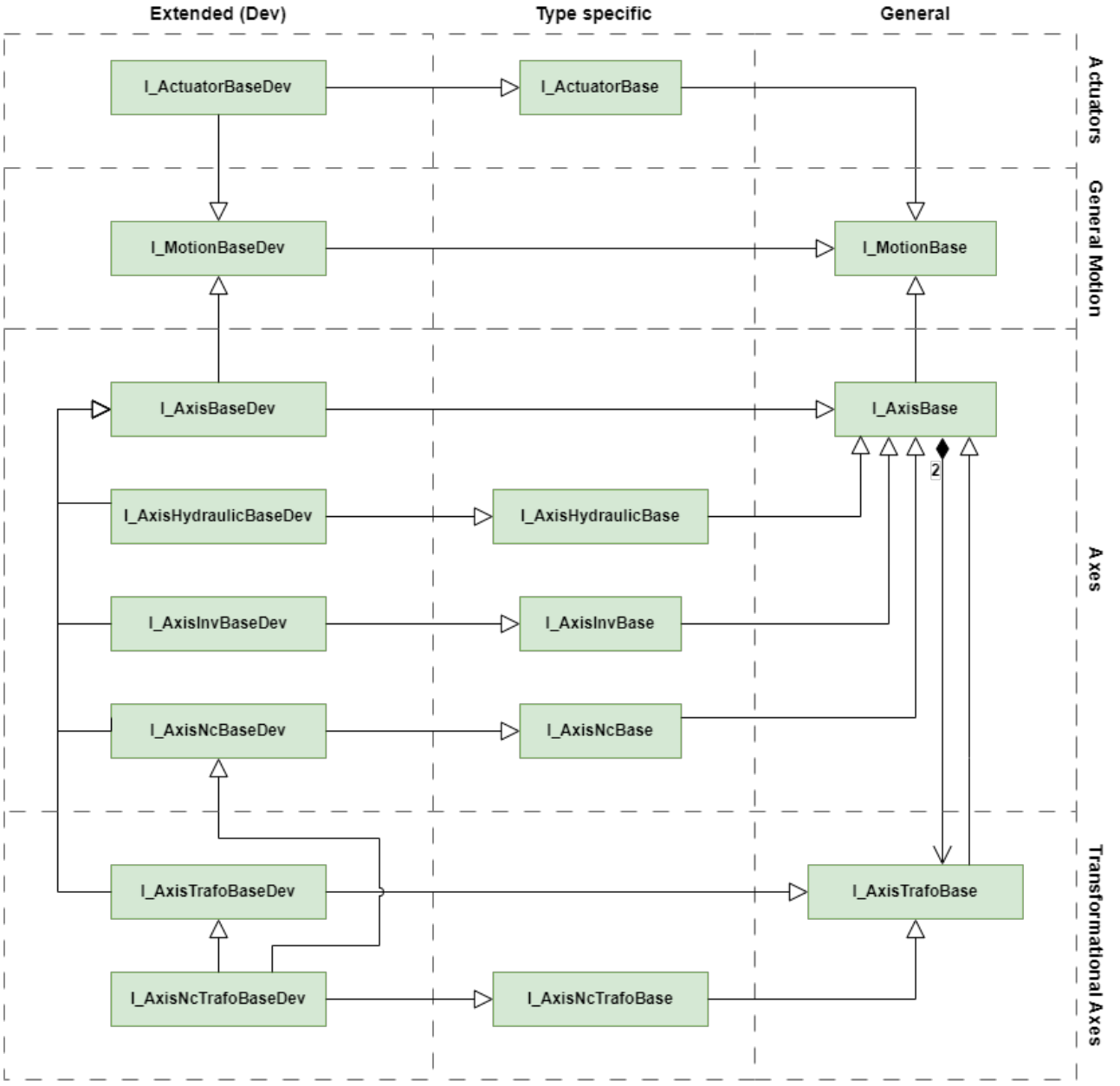
```
IF __QUERYINTERFACE(iAxisBase, iAxisHydraulicBase) THEN
    iAxisHydraulicBase.AutoIdent.DoAutoIdent(TRUE);
ELSE
    ;    // Axis is not hydraulic
END_IF
```

For more information on the `QUERYINTERFACE()` function, see the corresponding section in the TwinCAT documentation.

To provide full access to an axis, each axis interface provided also includes an extended `Dev` variant. This should generally not be used, as it makes all properties and methods of an axis available. Among these are also some that are not intended for regular use by the application.

Available interfaces overview

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



Name	Description
I_MotionBase [▶ 37]	General motion unit
I_MotionBaseDev [▶ 37]	
I_AxisBase [▶ 39]	General axis
I_AxisBaseDev [▶ 40]	
I_AxisNcBase [▶ 43]	NC-based axis
I_AxisNcBaseDev [▶ 44]	
I_AxisHydraulicBase [▶ 46]	Hydraulic-based axis.
I_AxisHydraulicBaseDev [▶ 48]	.
I_AxisInvBase [▶ 51]	Inverter-based axis
I_AxisInvBaseDev [▶ 52]	
I_AxisTrafoBase [▶ 54]	General transformation axis
I_AxisTrafoBaseDev [▶ 55]	
I_AxisNcTrafoBase [▶ 58]	NC-based transformation axis
I_AxisNcTrafoBaseDev [▶ 60]	

Name	Description
I_ActuatorBase [▶ 62]	Digital linear actuator
I_ActuatorBaseDev [▶ 63]	

4.3.1 I_MotionBase

Represents the general interface for a motion unit.

Syntax:

```
INTERFACE I_MotionBase EXTENDS I_MessageBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

4.3.2 I_MotionBaseDev

Represents the extended interface for a motion unit.

Syntax:

```
INTERFACE I_MotionBaseDev EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_MotionBaseDev			

Name	Type	Access	Description
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.
[INTERN] ReadCycleTime()	Used to determine the cycle time.
[INTERN] AppendCorefunction(iCoreFunc)	For internal use only. Replaced by _CheckCF()
I_MotionBase	
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)

4.3.3 I_AxisBase

Represents the general interface for an axis.

Syntax:

```
INTERFACE I_AxisBase EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
AutoTorqueLimitSelect[1]	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. Alternatively: `.AutoLimitingSelect`
AxisName[1]	STRING	Get, Set	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. Alternatively: `.Name`
BadSwitchOver[1]	BOOL	Get	Identifies whether a pressure switchover has occurred due to safety conditions. Alternatively: `.PressureControl.Supervisor.BadSwitchOver`

Name	Type	Access	Description
TeachPosition [▶ 344][1]	I_TeachPosition	Get	Commissioning of axes with analog position measuring systems. Alternatively: `.Teach`
TeachUpdate [▶ 345][1]	I_TeachUpdate	Get	
TorqueLimiting [▶ 346][1]	I_TorqueLimiting	Get	Torque limitation of the axis. Alternatively: `.Limiting`
SwitchOver[1]	BOOL	Get	The pressure switchover has been triggered. Alternatively: `.PressureControl.Supervisor.SwitchOver`
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

[1]Obsolete

Methods

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

I_AxisBase
Obsolete elements only

GetProcessHandler()[1]	Reserved for future extension.
SetProcessHandler() [▶ 329][1]	Reserved for future extension.
I_MotionBase	
DoReset(`bExecute`)	Triggers the reset of error states in the axis and in all local CoreFunctions.

[1]Obsolete

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

4.3.4 I_AxisBaseDev

Represents the extended interface for an axis.

Syntax:

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

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_MotionBaseDev			
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.

Name	Type	Access	Description
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_AxisBaseDev	
[INTERN] ConvertCountToPos	Only for internal use.
[INTERN] ConvertPosToCount	Only for internal use.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
SetLimiting(iLimiting)	This method connects a limiting block to the axis. Alternatively: Limiting.Output
SetTorqueLimiting(iTorqueLimiting)[1]	This method connects a function block for torque limitation to the axis. Alternatively: Limiting.Output
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.
[INTERN] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
DoReset(bExecute)	Triggers the reset of error states in the axis and in all local CoreFunctions.

[1] Obsolete

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

4.3.5 I_AxisNcBase

Represents the general interface for an NC axis.

Syntax:

```
INTERFACE I_AxisNcBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisNcBase			
ActualsNc [▶ 84]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			

Name	Type	Access	Description
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.6 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_MotionBaseDev			
[`INTERN`] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[`INTERN`] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.

Name	Type	Access	Description
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_AxisNcBase			
ActualsNc [▶ 84]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_AxisNcBaseDev	
GetNcAxisRef()	The address to the mapping interface (AXIS_REF) between NC and PLC.
I_AxisBaseDev	
[INTERN] ConvertCountToPos	Only for internal use.
[INTERN] ConvertPosToCount	Only for internal use.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.
[INTERN] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.7 I_AxisHydraulicBase

Represents the general interface for a hydraulic axis.

Syntax:

```
INTERFACE I_AxisHydraulicBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)

Name	Type	Access	Description
I_AxisHydraulicBase			
ActualsHydraulics [▶ 83]	I_ActualsHydraulic	Get	Provision of current actual values (positions, velocities, pressures, etc.). It is an extended version of Actuals.
AutoIdent [▶ 85]	I_AutoIdent	Get	Automatic characteristic measurement of the hydraulic axis.
DirectOutput [▶ 103]	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. **A TRUE must be set before the first Cyclic call.**
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.8 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisHydraulicBaseDev			
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.
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_MotionBaseDev			

Name	Type	Access	Description
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_AxisHydraulicBase			
ActualsHydraulics [▶ 83]	I_ActualsHydraulic	Get	Provision of current actual values (positions, velocities, pressures, etc.). It is an extended version of Actuals.
AutoIdent [▶ 85]	I_AutoIdent	Get	Automatic characteristic measurement of the hydraulic axis.
DirectOutput [▶ 103]	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. **A TRUE must be set before the first Cyclic call.**
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.

Name	Type	Access	Description
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_ExternalHydAxisLibRef	
GetHydAxisRef()	The address to the mapping interface (`AXIS_REF_BkPlcMc`) between hydraulics library and PLC.
I_AxisBaseDev	
[`INTERN`] ConvertCountToPos	Only for internal use.
[`INTERN`] ConvertPosToCount	Only for internal use.
GoErrorBase(`nErrorCode`)	This method can be used to set the axis to an error state.
I_MotionBaseDev	
[`PROTECTED`] _CheckCF(`iCoreFunc`)	Checks and configures a Corefunction for cycle processing.
[`INTERN`] EnterCriticalSection()	Only for internal use.
[`INTERN`] ForceState(`eNewState`)	This method changes the state of the axis with high priority.
[`INTERN`] GetRefFunction(`nId`)	Is used to initialize the Corefunctions.
[`INTERN`] LeaveCriticalSection()	Only for internal use.
[`INTERN`] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
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

Development environment	Target platform	PLC libraries to include
		Tc3_PlasticHydraulic v3.12.4.26 or higher

4.3.9 I_AxisInvBase

Represents the general interface for an inverter axis.

Syntax:

```
INTERFACE I_AxisInvBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.

Name	Type	Access	Description
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

I_AxisBase
Obsolete elements only

I_MotionBase	
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)

4.3.10 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisInvBaseDev			
Device	I_InvDevice	Get, Set	I/O interface for communication with the axis inverter
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	I_Estop	Get	Function for canceling an active motion command to bring the axis to a standstill. If available, increased dynamic parameters are applied.

Name	Type	Access	Description
ExtGenerated	BOOL	Get	TRUE if the setpoint generation runs via the ExternalGenerating [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_AxisBaseDev	
[INTERN] ConvertCountToPos	Only for internal use.
[INTERN] ConvertPosToCount	Only for internal use.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.

Name	Description
[INTERN] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
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)

4.3.11 I_AxisTrafoBase

Represents the general interface for a transformation axis.

Syntax:

```
INTERFACE I_AxisTrafoBase EXTENDS I_AxisBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisTrafoBase			
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 [▶ 200]	I_Transformation	Get	Transformation function of the axis.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.

Name	Type	Access	Description
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.12 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisTrafoBaseDev			
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.
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_MotionBaseDev			
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_AxisTrafoBase			
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 [▶ 200]	I_Transformation	Get	Transformation function of the axis.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.

Name	Type	Access	Description
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_AxisBaseDev	
[INTERN] ConvertCountToPos	Only for internal use.
[INTERN] ConvertPosToCount	Only for internal use.
GoErrorBase(nErrorCode)	This method can be used to set the axis to an error state.
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.

Name	Description
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.
[INTERN] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.13 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisNcTrafoBase			
ActualsDriveNc	I_NcActuals	Get	Provision of current actual values of the drive side (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisNcBase			
ActualsNc [▶ 84]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisTrafoBase			
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 [▶ 200]	I_Transformation	Get	Transformation function of the axis.
I_AxisBase			

Name	Type	Access	Description
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

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

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.14 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 [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_AxisTrafoBaseDev			
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.
I_AxisBaseDev			
AxisIsInverter	BOOL	Get	TRUE if the axis is of type FB_AxisInvBase [▶ 32].
AxisIsNc	BOOL	Get	TRUE if the axis is based on a TwinCAT NC axis.
PtpLookUp [▶ 185]	I_PtpLookup	Get	Storage of the multi-segment information of the PTP function.
PtpPoints	INT	Get, Set	The number of supported segments in the table supported by Ptp [▶ 173].
Reset [▶ 189]	I_Reset	Get	Internal function for processing an axis reset.
I_MotionBaseDev			
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_AxisNcTrafoBase			
ActualsDriveNc	I_NcActuals	Get	Provision of current actual values of the drive side (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisTrafoBase			
ActualsDrive	I_MotionActuals	Get	Provision of current actual values of the drive side (positions, velocities, etc.).

Name	Type	Access	Description
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 [▶ 200]	I_Transformation	Get	Transformation function of the axis.
I_AxisNcBase			
ActualsNc [▶ 84]	I_NcActuals	Get	Provision of current actual values (positions, velocities, torque, etc.). It is an extended version of Actuals.
I_AxisBase			
AutoLimitSelect	BOOL	Get, Set	A TRUE allows the axis to select an internal function block for limiting if the servo drive type (CoE, SoE) has been detected.
Camming [▶ 90]	I_CammingMain	Get	Functions for processing cam plates.
DisableSoftEnd [▶ 106]	I_DisableSoftEnd	Get	Possibility to temporarily disable the software limit switches of the axis and restore their original state.
Estop [▶ 111]	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 [▶ 109] core function.
Homing [▶ 114]	I_HomingMain	Get	Provision of a selection of homing procedures.
Limiting [▶ 145]	I_Limiting	Get	Limitation (torque, pressure, etc.) of the axis.
MotionSetpoints [▶ 157]	I_MotionSetpoints	Get	Provision of current setpoints (position, velocity, etc.).
Ptp [▶ 173]	I_Ptp	Get	Function for active movements. A table with a section-by-section definition is used.
SetPosition [▶ 191]	I_SetPosition	Get	Set a new actual position of the axis.
Stop [▶ 194]	I_Stop	Get	Canceling an active motion command to bring the axis to a standstill. The dynamics parameters of the motion command are used.
Teach [▶ 196]	I_Teach	Get	Commissioning of axes with analog position measuring systems.
ToolAdaption [▶ 199]	I_ToolAdaption	Get	Conversion of positions and velocities between an axis and a tool.
VelocityFeed [▶ 208]	I_VelocityFeed	Get	An interface to a local function block that triggers a motion without specifying a destination.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_AxisNcTrafoBaseDev	
GetNcDriveAxisRef()	The address on the mapping interface (`AXIS_REF`) of the drive side between NC and PLC.
I_AxisNcBaseDev	
GetNcAxisRef()	The address to the mapping interface (`AXIS_REF`) between NC and PLC.
I_AxisBaseDev	
[`INTERN`] ConvertCountToPos	Only for internal use.
[`INTERN`] ConvertPosToCount	Only for internal use.
GoErrorBase(`nErrorCode`)	This method can be used to set the axis to an error state.
I_MotionBaseDev	
[`PROTECTED`] _CheckCF(`iCoreFunc`)	Checks and configures a Corefunction for cycle processing.
[`INTERN`] EnterCriticalSection()	Only for internal use.
[`INTERN`] ForceState(`eNewState`)	This method changes the state of the axis with high priority.
[`INTERN`] GetRefFunction(`nId`)	Is used to initialize the Corefunctions.
[`INTERN`] LeaveCriticalSection()	Only for internal use.
[`INTERN`] ReadCycleTime()	Used to determine the cycle time.

I_AxisBase
Obsolete elements only

I_MotionBase	
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

4.3.15 I_ActuatorBase

Represents the general interface for an actuator.

Syntax:

```
INTERFACE I_ActuatorBase EXTENDS I_MotionBase
```

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_ActuatorBase			
ActualsActuator [▶ 81]	I_ActualsActuator	Get	Provision of current actual values (positions, velocities, motion times, etc.). It is an extended version of Actuals.
Tuning [▶ 206]	I_ActuatorTuning	Get	Tuning function for calibrating the movement time.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	E_AxiState	Get	The current state of the axis.

Methods

Name	Description
Cyclic()	This method implements all motion mechanisms that must be called cyclically.
I_ActuatorBase	
DoBase(`bEnable`)	Command for the movement to the base position (`Jog.DoJogM()`).
DoWork(`bEnable`)	Command to move to the working position (`Jog.DoJogP()`).
I_MotionBase	
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)

4.3.16 I_ActuatorBaseDev

Represents the extended interface for an actuator.

Syntax:

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

Properties

Name	Type	Access	Description
Actuals [▶ 80]	I_MotionActuals	Get	Provision of current actual values (positions, velocities, etc.)
I_ActuatorBaseDev			
FeedbackBase	I_InputBase	Get, Set	I/O interface to an end position sensor of the base position.

Name	Type	Access	Description
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.
I_MotionBaseDev			
[INTERN] CmdCurrent	UDINT	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdCurrentId	E_CurrentCmd	Get, Set	A counter that assigns an identification to each activated command.
[INTERN] CmdNext	UDINT	Get	The next value to be used as CmdCurrent.
[INTERN] CoreDebug	I_CoreDebug	Get	Debugging support
[INTERN] CycleTime	LREAL	Get	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.
[INTERN] CycleTimeValid	LREAL	Get	The internal query of the cycle time is valid
I_ActuatorBase			
ActualsActuator [▶ 81]	I_ActualsActuator	Get	Provision of current actual values (positions, velocities, motion times, etc.). It is an extended version of Actuals.
Tuning [▶ 206]	I_ActuatorTuning	Get	Tuning function for calibrating the movement time.
I_MotionBase			
Jog [▶ 141]	I_Jog	Get	Command motion without a defined destination (travel on revocation).
MotionParams [▶ 150]	I_MotionParams	Get	Compilation of motion parameters
Name	STRING	Get	The textual name of the motion unit.
Power [▶ 158]	I_Power	Get	Function for releasing the axis and its direction-related feed releases.
State [▶ 65]	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.
I_MotionBaseDev	
[PROTECTED] _CheckCF(iCoreFunc)	Checks and configures a Corefunction for cycle processing.
[INTERN] EnterCriticalSection()	Only for internal use.
[INTERN] ForceState(eNewState)	This method changes the state of the axis with high priority.

Name	Description
[INTERN] GetRefFunction(nId)	Is used to initialize the Corefunctions.
[INTERN] LeaveCriticalSection()	Only for internal use.
[INTERN] ReadCycleTime()	Used to determine the cycle time.
I_ActuatorBase	
DoBase(bEnable)	Command for the movement to the base position (Jog.DoJogM()).
DoWork(bEnable)	Command to move to the working position (Jog.DoJogP()).
I_MotionBase	
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)

4.4 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.

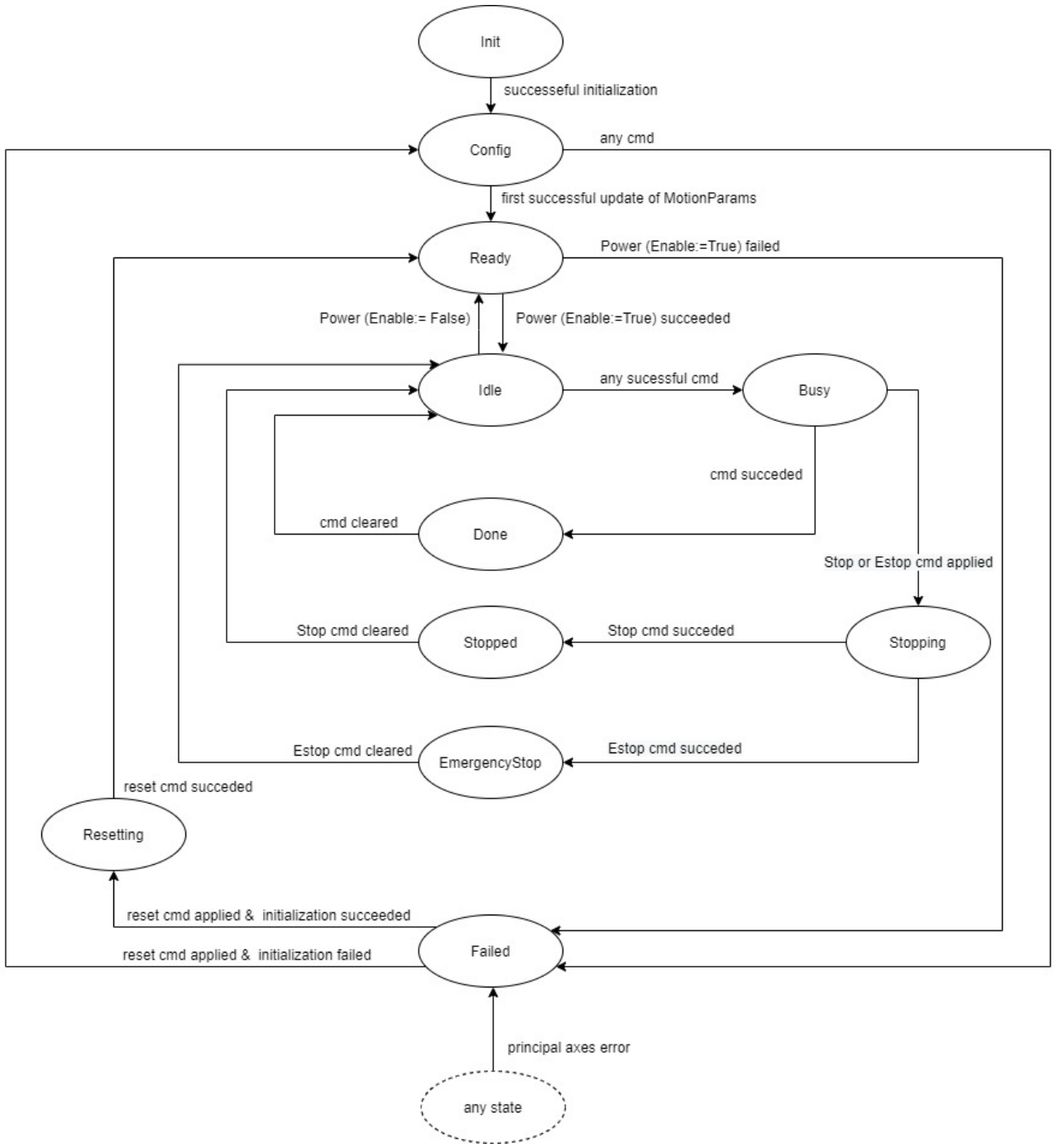
```
//  CoreFunction method calls
iNcSampleAxis.Power.DoPower(TRUE)
iNcSampleAxis.Power.FeedEnable(TRUE, TRUE);

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

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

4.5 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.

5 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. TF8560 Plastic Technology Functions implements and encapsulates each of these motion tasks in a separate FB called a Corefunction. All Corefunctions follow a standardized format based on PLCopen.

The following table shows the availability of the Corefunction for each axis type:

Corefunction	Type	Availability by axis type				
				(Transfor-mation)		
Actuals [▶ 80]	Permanently active [▶ 73]	✓	✓	✓	✓	✓
ActualsActuator [▶ 81]	Permanently active [▶ 73]	✗	✗	✗	✗	✓
ActualsHydraulics [▶ 83]	Permanently active [▶ 73]	✓	✗	✗	✗	✗
ActualsNc [▶ 84]	Permanently active [▶ 73]	✗	✓	✓	✗	✗
AutoIdent [▶ 85]	Statically controlled [▶ 74]	✓	✗	✗	✗	✗
Camming [▶ 90].Pos	Statically controlled [▶ 74]	✓	✓	✓	✗	✗
Camming [▶ 90].Velo	Statically controlled [▶ 74]	✓	✓	✓	✓	✗
DirectOutput [▶ 103]	Statically controlled [▶ 74]	✓	✗	✗	✗	✗
DisableSoftEnd [▶ 106]	Edge-triggered [▶ 76]	✓	✓	✓	✗	✗
ExternalGenerating [▶ 109]	Statically controlled [▶ 74]	(✓)	✗	✗	✗	✗
Estop [▶ 111]	Edge-triggered [▶ 76]	✓	✓	✓	✓	✗
Homing [▶ 114]	Edge-triggered [▶ 76]	✓	✓	✓	✗	✗
Jog [▶ 141]	Statically controlled [▶ 74][2]	✓	✓	✓	✓	✓
Limiting [▶ 145]	Permanently active [▶ 73]	✓	✓	✓	✓	✗
MotionParams [▶ 150]	Permanently active [▶ 73]	✓	✓	✓	✓	✓
MotionSetpoints [▶ 157]	Permanently active [▶ 73]	✓	✓	✓	✓	✓
Power [▶ 158]	Edge-triggered [▶ 76][1]	✓	✓	✓	✓	✓
PressureControl [▶ 161]	Statically controlled [▶ 74]	✓	✓	✓	✓	✗
Ptp [▶ 173]	Edge-triggered [▶ 76]	✓	✓	✓	✗	✗
PtpLookUp [▶ 185]	Permanently active [▶ 73]	(✓)	(✓)	(✓)	✗	✗
Reset [▶ 189]	Edge-triggered [▶ 76]	(✓)	(✓)	(✓)	(✓)	(✗)

Corefunction	Type	Availability by axis type				
				(Transfor-mation)		
SetPosition [▶ 191]	Edge-triggered [▶ 76]	✓	✓	✓	✗	✗
Stop [▶ 194]	Edge-triggered [▶ 76]	✓	✓	✓	✓	✗
Teach [▶ 196]	Edge-triggered [▶ 76]	✓	✓	✗	✗	✗
ToolAdaption [▶ 199]	Permanently active [▶ 73]	✓	✓	✓	✗	✗
Transformation [▶ 200]	Statically controlled [▶ 74]	✗	✗	✓	✗	✗
Tuning [▶ 206]	Edge-triggered [▶ 76]	✗	✗	✗	✗	✓
VelocityFeed [▶ 208]	Statically controlled [▶ 74]	✓	✓	✓	✓	✗

[1] Power [▶ 158] operates differently from other Edge-triggered [▶ 76] Corefunctions

[2] In the case of Digital actuators [▶ 34], the jog is Edge-triggered. [▶ 76]

Legend

- TF5810 Hydraulic Positioning
  FB_AxisHydraulicBase
- TF5000 NC PTP
  FB_AxisNcBase
  FB_AxisNcTrafoBase
- TF8560 Inverter Axis
  FB_AxisInvBase
- TF8560 Digital Actuator
  FB_ActuatorBase

✓ Available
(✓) Available internally
✗ Not available

5.1 Basic structure and variations

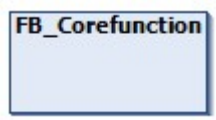
A core function cannot operate independently because it contains only the code implementation of the motion task, but is not connected to the Motion Control library. When an axis is initialized, the core function receives the interface of the axis, while the axis contains an interface of the core function.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

5.1.1 Core scope (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 [▶ 71] 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 technology specific adaption.

Syntax:

```
FUNCTION_BLOCK ABSTRACT FB_Corefunction EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Corefunction			
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.
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.
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.
[INTERN] AxisState	E_AxisState [▶ 65]	Get, Set	The current state of the axis state machine.
[INTERN] Connected	BOOL	Get	The Corefunction is connected to an axis.
[INTERN] CycleTime	LREAL	Get, Set	The cycle time of the PLC task from which the method `Cyclic()` of the Corefunction is executed.
[INTERN] CycleTimeValid	BOOL	Get, Set	A TRUE indicates that the `CycleTime` of the Corefunction is defined.
[INTERN] NextCore	I_Corefunction	Get, Set	This property is part of the execution chain and must not be affected by the application task.
[INTERN] OnlineChangeMark	BOOL	Get, Set	The axis uses this property to signal an Online Change to the Corefunction.

Methods

Name	Description
[INTERN] Cyclic()	This method is called cyclically by the axis.

Name	Description
FB_Corefunction	
[INTERN] ApplyCommand()	Applies the command to the axis.
[INTERN] ClearStates()	Clears up some of the possible states.
[INTERN] DoReset(bExecute)	This method is called when the axis is reset.
[INTERN] Init(iMBase)	Initializes the Corefunction.
[INTERN] RemoveCommand()	Removes the command from the axis.

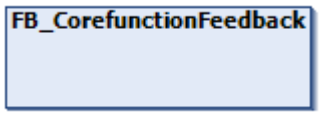
Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.1.2 Commandable variants (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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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.
[INTERN] AxisState	E_AxisState [▶ 65]	Get, Set	The current state of the axis state machine.
[INTERN] Connected	BOOL	Get	The Corefunction is connected to an axis.
[INTERN] CycleTime	LREAL	Get, Set	The cycle time of the PLC task from which the method `Cyclic()` of the Corefunction is executed.
[INTERN] CycleTimeValid	BOOL	Get, Set	A TRUE indicates that the `CycleTime` of the Corefunction is defined.

Name	Type	Access	Description
[INTERN] NextCore	I_Corefunction	Get, Set	This property is part of the execution chain and must not be affected by the application task.
[INTERN] OnlineChangeMark	BOOL	Get, Set	The axis uses this property to signal an Online Change to the Corefunction.

Methods

Name	Description
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Corefunction	
[INTERN] ApplyCommand()	Applies the command to the axis.
[INTERN] ClearStates()	Clears up some of the possible states.
[INTERN] DoReset(bExecute)	This method is called when the axis is reset.
[INTERN] Init(iMBase)	Initializes the Corefunction.
[INTERN] RemoveCommand()	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

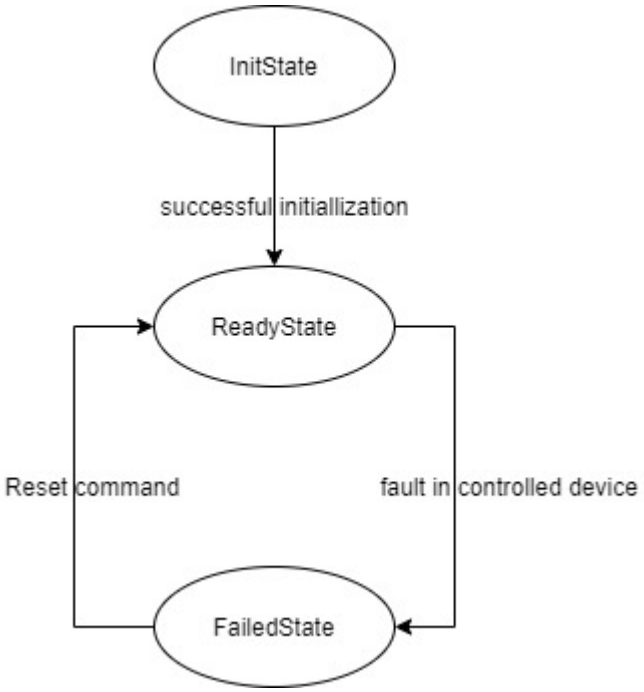
5.1.3 Permanently active core functions

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 Corefunctions	Description
Actuals [▶ 80]	Contains the state of the axis.
ActualsActuator [▶ 81]	Is derived from Actuals and interprets position and velocity as Boolean.
ActualsHydraulics [▶ 83]	Derived from Actuals and extends it to include hydraulic-specific elements.
ActualsNc [▶ 84]	Is derived from Actuals and extends it with NC-specific elements.
Limiting [▶ 145]	Control of power limitation, e.g., in the form of torque or pressure.
MotionParams [▶ 150]	Provides access to a range of axis parameters.
MotionSetpoints [▶ 157]	Contains the current setpoints of the axis.
PtpLookUp [▶ 185]	Contains the multi-segment PTP move command.
ToolAdaption [▶ 199]	Contains the parameters of a tool adaptation.

State machine



5.1.4 Statically controlled core functions

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

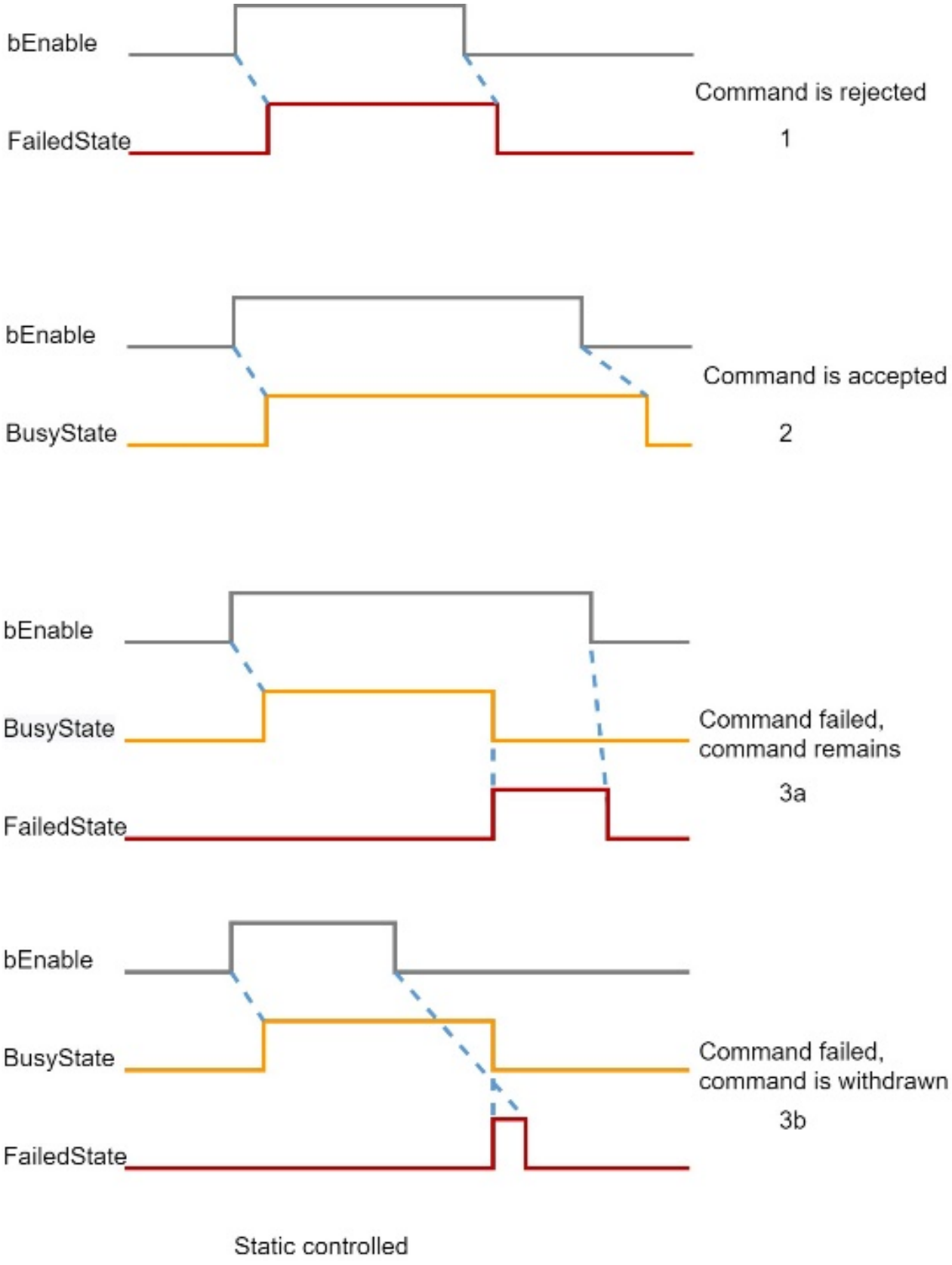
The rising edge of `bEnable` triggers a series of checks to determine whether the execution conditions are met. If the execution conditions are not met, the command is rejected and the core function enters the `FailedState` (case 1 in the figure below). If the execution conditions are met, the command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

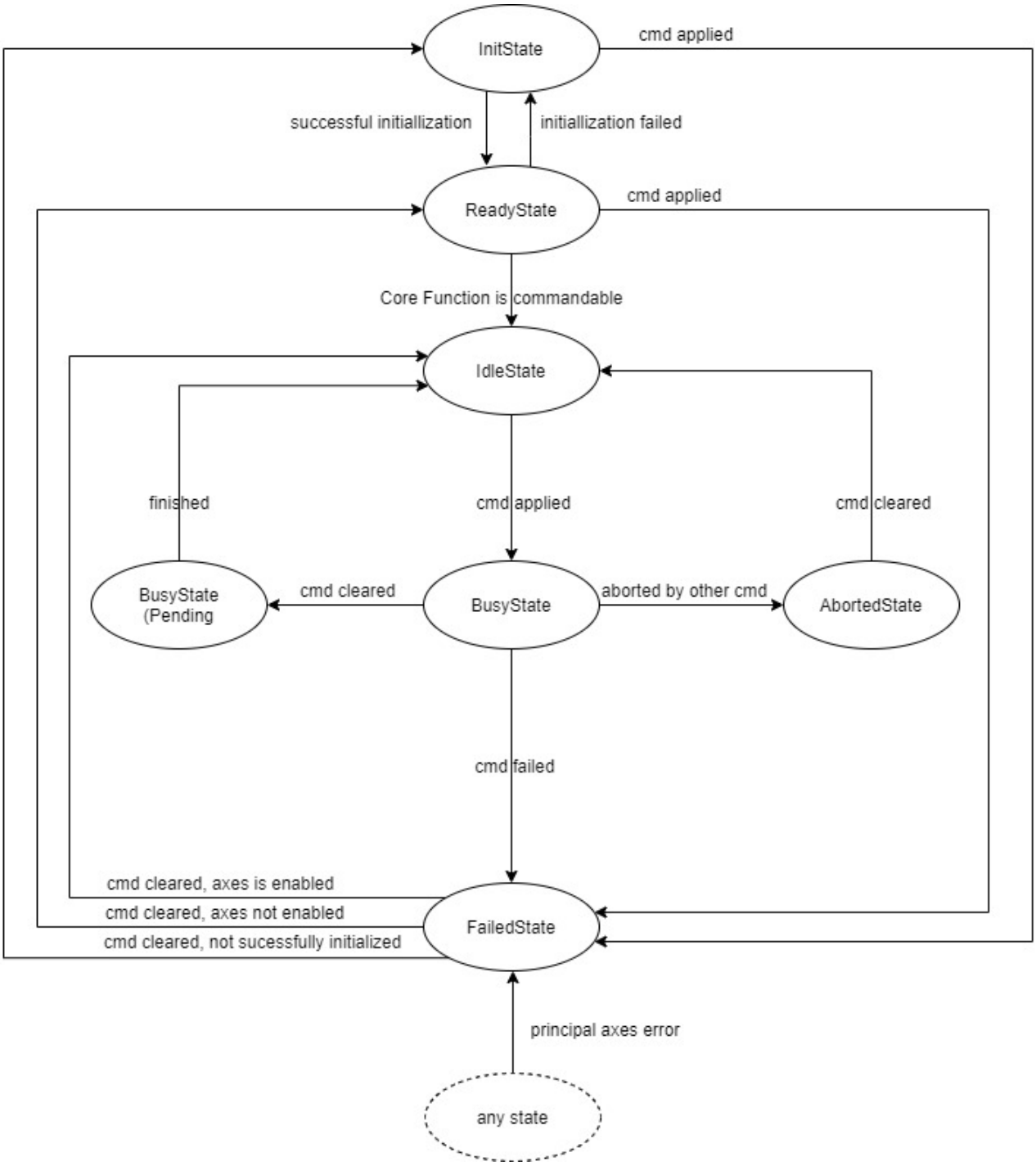
The following core functions are statically controlled:

Corefunction	Description
Camming [▶ 90]	Is used to activate setpoint generation that is controlled by a guide value.
ExternalGenerating [▶ 109]	Is used internally for setpoint generation by TwinCAT 3 Plastic Technology Functions.
Jog [▶ 141]	Is used to command a movement without a specified target.
PressureControl [▶ 161]	Used to implement pressure control using a sensor of your choice.
VelocityFeed [▶ 208]	Is used to activate a movement without a specified target and a velocity that is controlled by a guide value.
AutoIdent [▶ 85]	A special core function for hydraulic axes. It is used to identify the characteristic velocity behavior of the axis.
DirectOutput [▶ 103]	A special core function for hydraulic axes. It is used to send output signals directly to the control device.
Transformation [▶ 200]	A special core function for transforming axes. It is used for coupling the two internal axes of a transforming axis.

Timing Diagram



State machine



5.1.5 Edge-triggered core functions

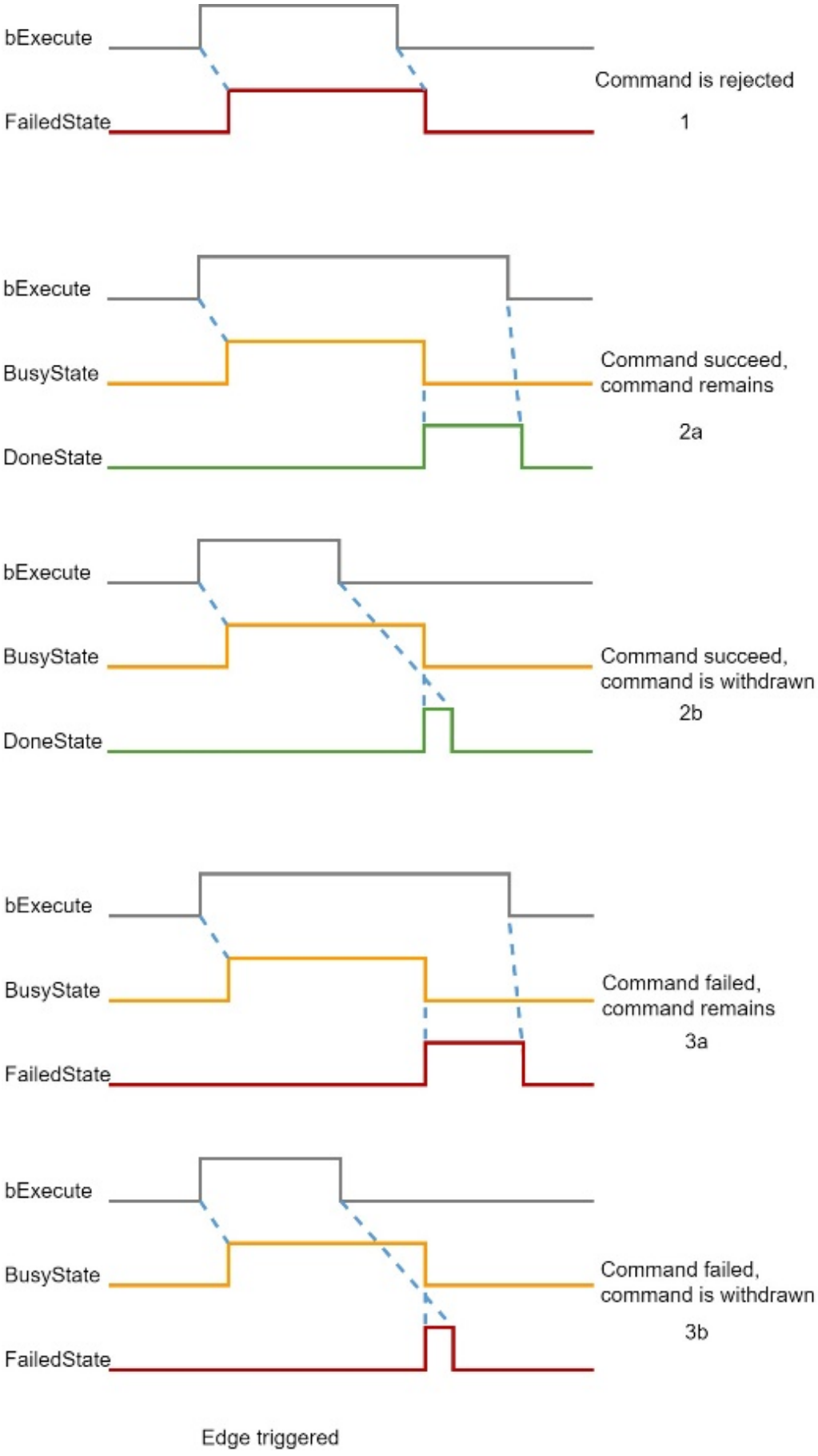
The command accepted by these core functions is usually called `bExecute` (with the exception of `Power`, which is described below).

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 state `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.

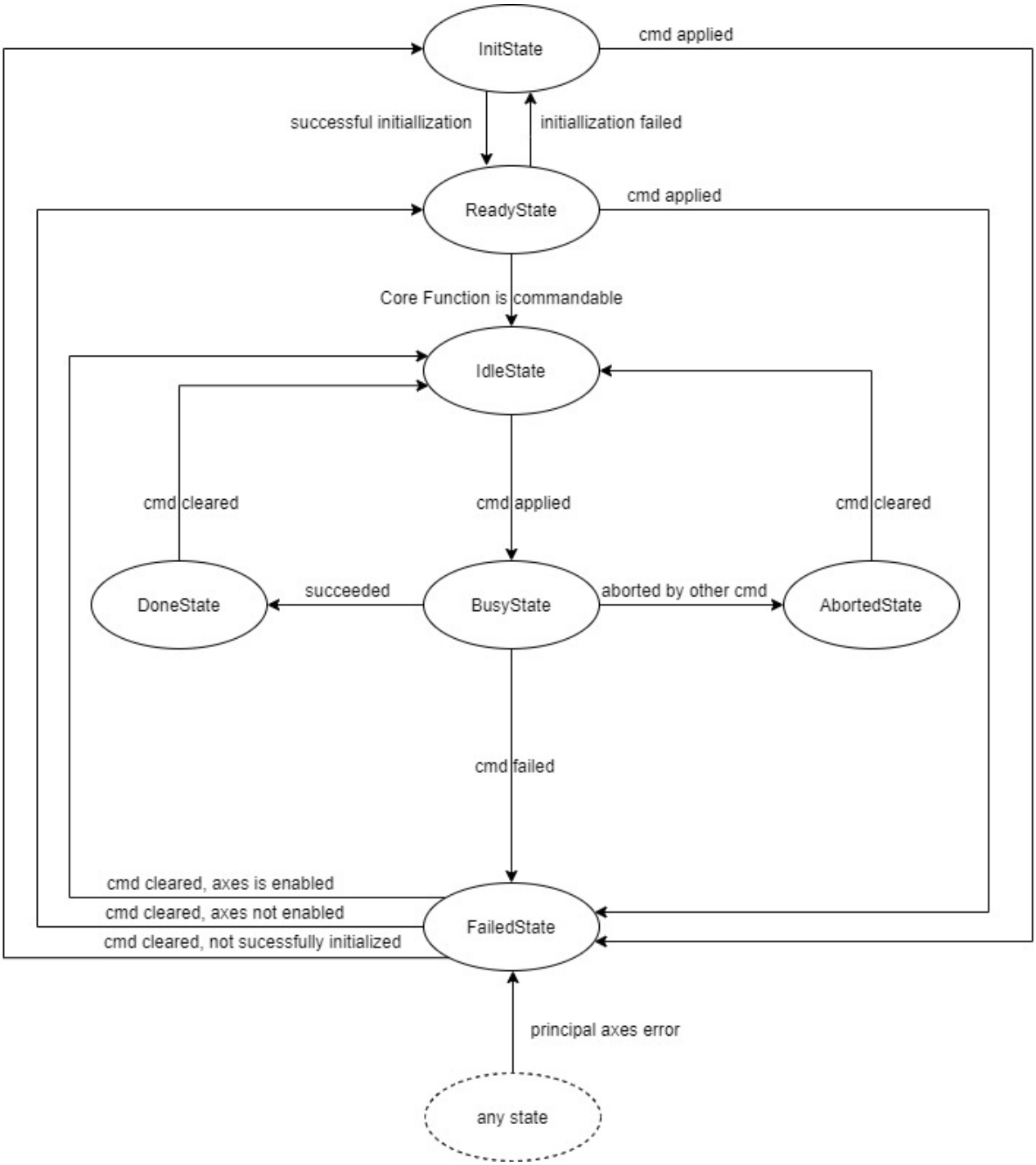
The following functions are edge-triggered core functions:

Corefunction	Description
DisableSoftEnd [▶ 106]	Is used to temporarily disable and re-enable the soft limit switches of the axis.
Homing [▶ 114]	Provides a range of homing methods.
Power [▶ 158]	Used to activate and deactivate the axis.
Ptp [▶ 173]	Is used to perform multi-segment PTP movements.
Reset [▶ 189]	Is used to reset the error state of axes and devices.
SetPosition [▶ 191]	Is used to change the actual position of the axis.
Estop [▶ 111]	Command for emergency shutdown using the maximum dynamic parameters to stop the axis.
Stop [▶ 194]	Command for stop with the standard dynamic parameters for stopping the axis.
Teach [▶ 196]	Used for referencing analog encoder systems.
Tuning [▶ 206]	A special core function for actuators. It is used for automatic optimization such as motion time measurement.

Timing Diagram



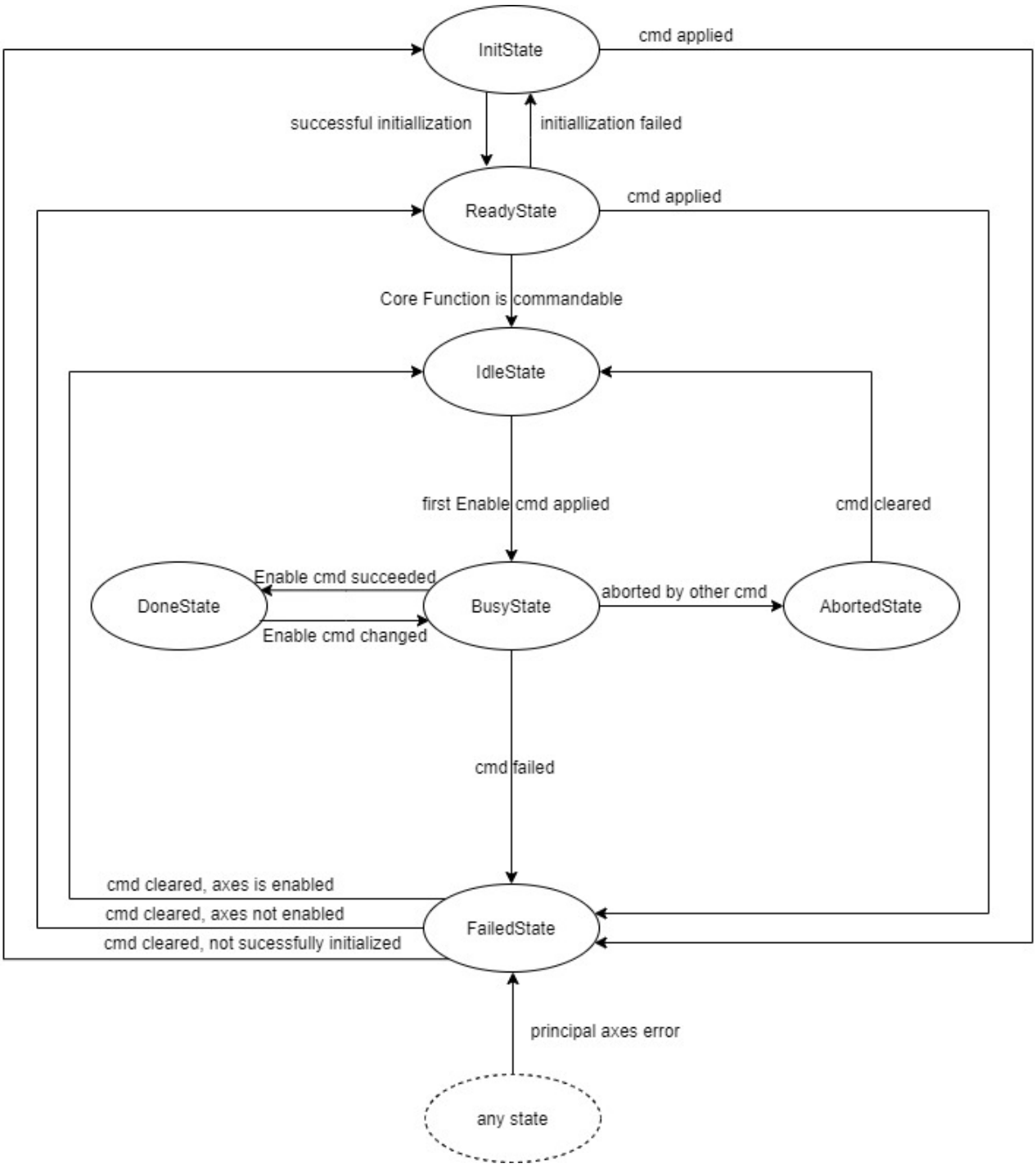
State machine



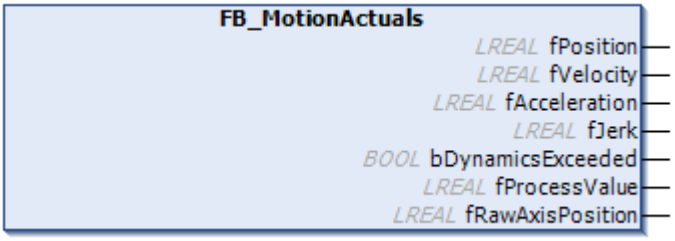
Exceptional case: Power

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.



5.2 Actuals



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.

Example:

```
IF iAxisBase.Actuals.Position >= fTargetPosition THEN
    bInPos := TRUE;
END_IF
```

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 ActualsActuator



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.

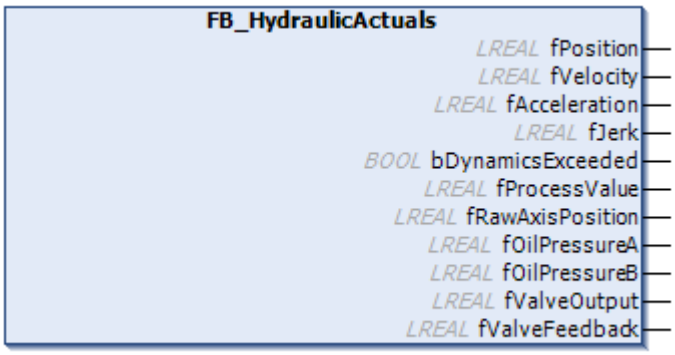
Example:

```
// Get specific interface I_ActuatorBase from general interface I_MotionBase
IF __QUERYINTERFACE(iMotionBase, iActuatorBase) THEN
    bInBasePos := iActuatorBase.ActualsActuator.IsBase
ELSE
    // interface is not set to an instance of FB_ActuatorBase
    bIncompatible := TRUE
END_IF
```

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 ActualsHydraulics



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 Corefunction belongs to the group of permanently active Corefunction.

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.
I_HydraulicActuals	Standard interface on FB_HydraulicActuals.

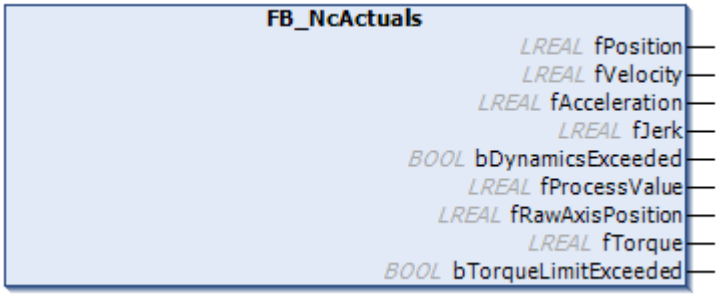
Example:

```
// Get specific interface I_AxisHydraulicBase from general interface I_AxisBase
IF __QUERYINTERFACE(iAxisBase, iAxisHydraulic) THEN
    bClampingForceReached := iAxisHydraulic.ActualsHydraulic.OilPressureA > fClampingPressure;
ELSE
    // interface is not set to an instance of FB_AxisHydraulicBase
    bIncompatible := TRUE
END_IF
```

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

5.5 ActualsNc



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.

Name	Type	Access	Description
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.

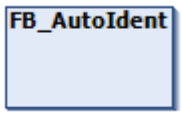
Example:

```
// Get specific interface I_AxisNcBase from general interface I_AxisBase
IF __QUERYINTERFACE(iAxisBase, iAxisNc) THEN
    bClampingForceReached := iAxisNc.ActualsNc.Torque > fClampingTorque;
ELSE
    // interface is not set to an instance of FB_AxisNcBase
    bIncompatible := TRUE
END_IF
```

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

5.6 AutoIdent



This core function is used to analyze the non-linear transfer characteristic of hydraulic axes. It belongs to the group of statically controlled Corefunctions.

Syntax:

```
FUNCTION_BLOCK FB_AutoIdent EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_CorefunctionFeedback			
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.

Name	Type	Access	Description
IdleState	BOOL	Get, Set	The Corefunction is ready for operation and commandable.
IsCommanded	BOOL	Get	Signals the pending of a command.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_AutoIdent	
DoAutoIdent() [▶ 87]	Activates and terminates the measurement procedure.
SetParameter() [▶ 88]	Sets the parameters for the measurement procedure.

Interfaces

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

Example:

```
// This sample requires the Autoident parameters set by the PlcMcManager in 'LinDef'

// Get specific interface I_AxisHydraulicBase from general interface I_AxisBase
IF __QUERYINTERFACE(iAxisBase, iAxisHydraulic) THEN

    // abort or done
    IF bAbort OR iAxisHydraulic.AutoIdent.DoneState THEN

        bAutoIdent := FALSE;
        iAxisHydraulic.AutoIdent.DoAutoIdent(FALSE);

    // start the command
    ELSIF bAutoIdent THEN

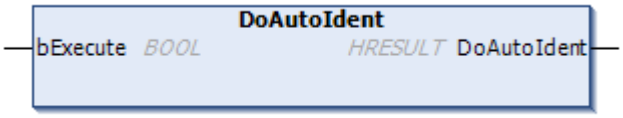
        hresult := iAxisHydraulic.AutoIdent.DoAutoIdent(TRUE);
    END_IF

ELSE
    // interface is not set to an instance of FB_AxisHydraulicBase
    bIncompatible := TRUE;
END_IF
```

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

5.6.1 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.

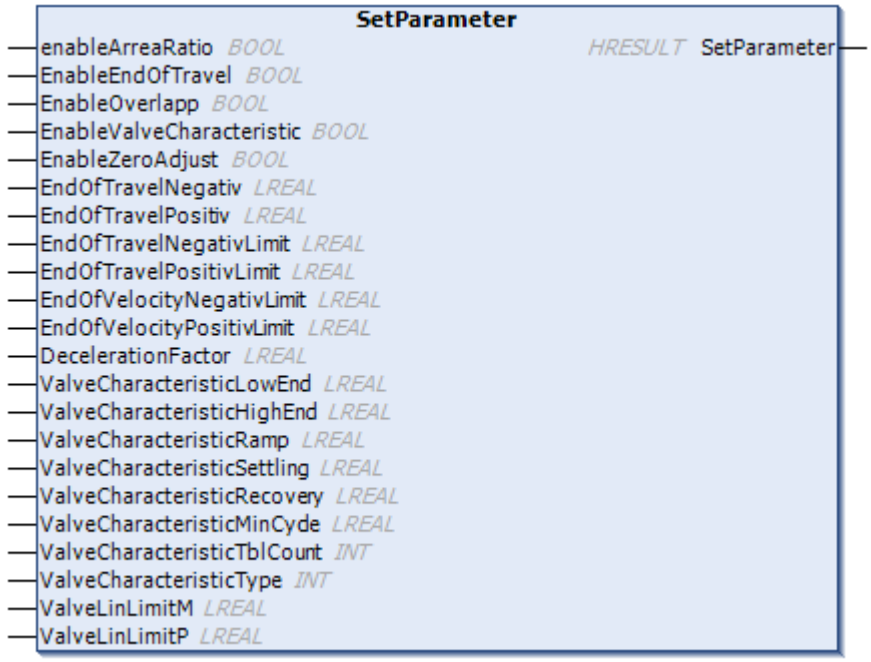
Inputs

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

5.6.2 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_INVALIDDATA	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

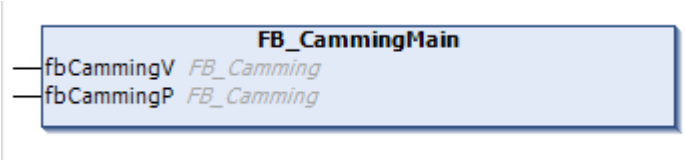
Name	Type	Description
EnableArreaRatio	BOOL	A TRUE will request a separate identification of effects caused by cylinder asymmetry.
EnableEndOfTravel	BOOL	A TRUE will request a separate identification of mechanical travel limits.
EnableOverlapp	BOOL	A TRUE will request a separate identification of effects caused by valve overlap.
EnableValveCharacteristic	BOOL	A TRUE will request the identification of the valve transfer characteristics.
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. **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.

Name	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.

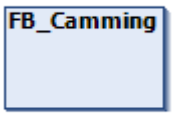
Requirements

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

5.7 Camming



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.



Both types of camming use a guiding value to identify a control value within a camming lookup table. For more information, see CammingLookUp [▶ 94].

Any useful information can serve as a guiding value. Common options are a time (`LREAL` variable that starts with `0` 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 guiding values or table points can lead to unexpected reactions of the controlled axis. Consequently, these can lead to accidents or damage.

Using camming requires several steps:

- The camming lookup table must be initialized. If you use the internal table (recommended), you must specify at least the interpolation type and define the number of points in the table. You can find an example of initialization either here or in the example below.
- The table must be defined by specifying all table points. Make sure that the point data corresponds to the camming type (position, velocity) for which they are to be used.
- In the case of equidistant points, a minimum and maximum value must also be assigned to the `GuidingValue` in the table.
- An initial reference value should be specified using the `GuidingValue` property before enabling camming.
- At the right time use `DoCamming(TRUE)` to enable the function. Make sure that the axis is in a state (position, velocity) that matches the guidance value of the camming table.
- Make sure that the guiding value is updated cyclically.
- At the right time use `DoCamming(FALSE)` to disable the function. Pay attention to the state (position, velocity) of the axis.

Syntax:

```
FUNCTION_BLOCK FB_Camming EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Camming			
GuidingValue	LREAL	Get, Set	The current guiding value of the camming. This value should be updated by the application at the same frequency as the axis cycle time.
LookUp [▶ 94]	I_CammingLookUp	Get, Set	The camming lookup table used for the design. After initialization, or as soon as `Null` is assigned, the internal table becomes active. An external instance can also be used by assigning it.
Overrun	BOOL	Get	TRUE, if the current guiding value is not in 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.
[`INTERN`] UseAsPosition	BOOL	Get, Set	This property defines the camming rule. A TRUE causes the Corefunction to act as position camming. A FALSE will cause it to act as velocity camming. This property is set during startup. The application must not change its setting.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Camming	
DoCamming() [▶ 93]	This method is used to enable and disable the camming.
SetGuidingValue() [▶ 327][1]	This method is used to update the guiding value. Alternative: GuidingValue
SetLookupInterface() [▶ 328][1]	This method can be used to connect a camming table. Alternative: LookUp

[1] Obsolete

Interfaces

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

Example:

```
// For this example, the axis should currently be in the fCenter position
CASE nState OF
0:  // Define camming curve
    fCenter := 100.0;       // [mm]
    fAmplitude := 100.0;    // [mm]
    fCycleTime := 0.002;    // [s]

    // parameterise the lookup table
    iAxisBase.Camming.Pos.LookUp.Equidistance := TRUE;
    iAxisBase.Camming.Pos.LookUp.Interpolation := E_LookupMode.eSecondOrder;
    iAxisBase.Camming.Pos.LookUp.SetPoints(401);
    iAxisBase.Camming.Pos.LookUp.MinGuidingValue := 0.0;
    iAxisBase.Camming.Pos.LookUp.MaxGuidingValue := 15.0;

    // Create a sinusoidal curve over 15 seconds
    FOR i := 1 TO 401 DO
        fCamTime := 15.0 * (i - 1) / 400.0;
        fPoint := fCenter + fAmplitude * SIN(2.0 * PI * fCamTime / 15.0);
        IF FAILED(iAxisBase.Camming.Pos.LookUp.SetPoint(i, fPoint, 35.0)) THEN
            nState := -1;
        END_IF;
    END_FOR

    // check table validation
    IF NOT iAxisBase.Camming.Pos.LookUp.Valid THEN
        nState := -1;
    END_IF

    // assign initial camming value
    iAxisBase.Camming.Pos.GuidingValue := 0.0;
    fTime := 0.0;

    IF nState >= 0 THEN
        nState := nState + 1;
    END_IF

1:  // start camming
    IF FAILED(iAxisBase.Camming.Pos.DoCamming(TRUE)) THEN
        nState := -1;
    END_IF

    IF nState > 0 THEN
        nState := nState + 1;
    END_IF

2:  // operate camming
    fTime := MIN(fTime + fCycleTime, 15.0);
```

```
    iAxisBase.Camming.Pos.GuidingValue := fTime;

    IF iAxisBase.Camming.Pos.FailedState THEN
        nState := -1;
    ELSIF fTime >= 15.0 THEN
        nState := nState + 1;;
    END_IF

3,  // camming done
-1: // error

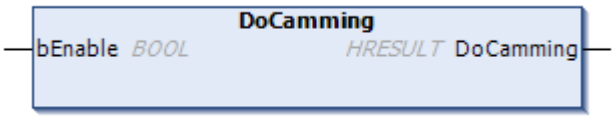
    fTime := 0.0;
    iAxisBase.Camming.Pos.DoCamming(FALSE);

END_CASE
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.7.1 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

5.7.2 FB_CammingLookUp



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

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 [▶ 102]	Defines the operation mode of the LookUp table = `eUndefined` enables subsequent assignment via `Interpolation` and `Equdistance`

Properties

Name	Type	Access	Description
Equidistance	BOOL	Get, Set[1]	In equidistant mode, the table assumes equal distances between the `MinGuidingValue` and the `MaxGuidingValue`.
Interpolation	E_LookupMode [▶ 102]	Get, Set[1]	Interpolation mode of the table
InvalidPoint	BOOL	Get	Is set by a call to `LookUp()` or `LookDown()` if the requested point is outside the valid range.
MinGuidingValue	LREAL	Get, Set	Smallest possible value for the `LookUp()` and `LookDown()` function
MaxGuidingValue	LREAL	Get, Set	Largest possible value for the `LookUp()` and `LookDown()` function
Overrun	BOOL	Get	Additional information on InvalidPoint: TRUE -> InvalidPoint is above the definition range. FALSE -> InvalidPoint is below the definition range.
Points	LREAL	Get	Number of available points in the table
Valid	BOOL	Get	The configuration of the table is validated and ready for use.
OpMode [2]	E_LookupMode [▶ 102]	Get	Operation mode of the table

[1] Must be set before determining the number of points

[2] Obsolete

Methods

Name	Description
GetPoint() [▶ 95]	Returns an equidistant point in the table.
GetPointNonEqui() [▶ 96]	Returns any point consisting of a master and a slave value.
LookDown() [▶ 98]	Translates a slave value to a matching master value.
LookUp() [▶ 99]	Translates a master value to a matching slave value.
SetPoint() [▶ 99]	Sets an equidistant point in the table.
SetPointNonEquidistant() [▶ 100]	Sets any point consisting of a master and a slave value.
SetPoints() [▶ 101]	Defines the number of assignable points once.
SetProperties() [▶ 101][2]	Sets the master limit values of the table
GetProperties() [▶ 97][2]	Returns the master limit values of the table
GetPoints() [▶ 96][2]	Returns the number of points available in the table
GetValid()[2]	Checks the table for validity

[2]Obsolete

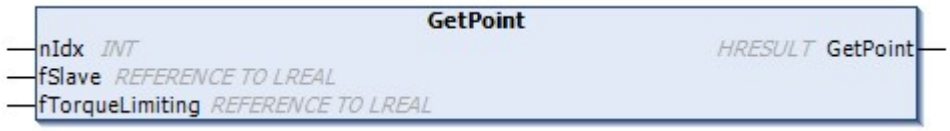
Interfaces

Type	Description
I_CammingLookUp	Standard interface on FB_CammingLookUp

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

5.7.2.1 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

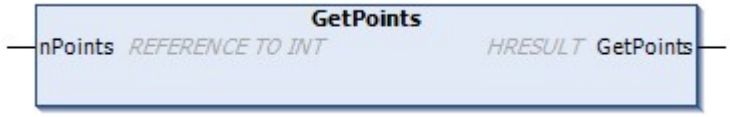
Name	Type	Description
nIdx	INT	Index of the requested table point

Name	Type	Description
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

5.7.2.2 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

5.7.2.3 GetPointNonEqui()



This method reads a point in the table addressed by `nIdx` that has a specific master position. If the equidistant operation mode (`Equidistant= FALSE`) is active, this function returns a 0 as the master value.

Syntax:

```
METHOD GetPointNonEqui : HRESULT
VAR_INPUT
    nIdx:           INT;
    fMaster:        REFERENCE TO LREAL := 0;    // optional
    fSlave:         REFERENCE TO LREAL;
    fLimiting:      REFERENCE TO LREAL := 0;    // optional
END_VAR
```

Return value

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

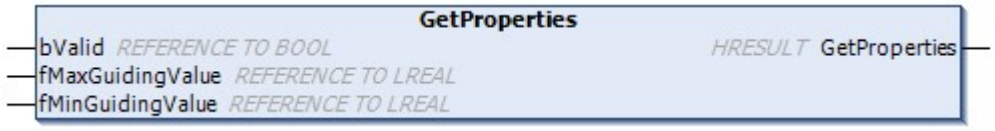
Inputs

Name	Type	Description
nIdx	INT	Index of the table point to be read.
fMaster	REFERNCE TO LREAL	Master value of the table point.
fSlave	REFERNCE TO LREAL	Slave value of the table point.
fLimiting	REFERNCE TO LREAL	Limiting 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

5.7.2.4 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

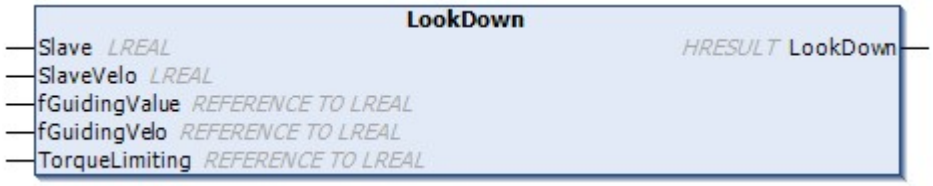
Inputs

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

5.7.2.5 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

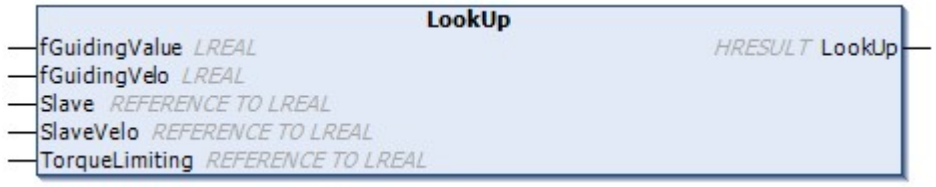
Inputs

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

5.7.2.6 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.

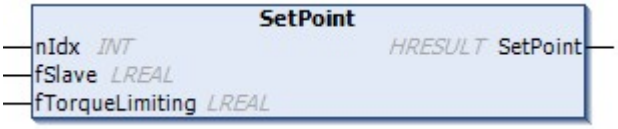
Inputs

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

5.7.2.7 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.

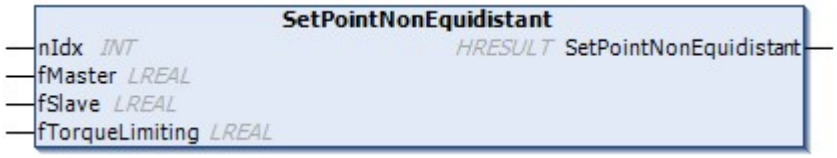
Inputs

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

5.7.2.8 SetPointNonEquidistant()



This method sets a point in the table addressed via `nIdx` that has a specific master position. If the equidistant operation mode (`Equidistant= FALSE`) is active, the master value is ignored.

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.

Name	Type	Description
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

5.7.2.9 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.

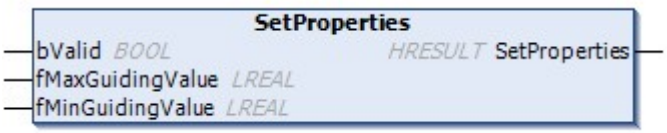
Inputs

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

5.7.2.10 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

5.7.3 E_LookupMode

Operation mode of the FB_CammingLookUp class.

Syntax:

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_LookupMode :
(
    // bad configuration
    eUndefined := 0,

    // obsolete ones (hidden)
    eNoneEqui := 1,
    eNoneEquiLinIpol,
    eEquiLinIpol,
    eEquiSecondOrder,

    // interpolation modes
    eStep := 16#10,
    eLinear := 16#20,
    eSecondOrder := 16#30,

    // equidistant modes are not actively shown
(hidden), because they're determined by the MSB = TRUE/FALSE
    eStep_Equi := 16#8010, // not implemented yet
    eLinear_Equi := 16#8020,
    eSecondOrder_Equi := 16#8030
);
END_TYPE
```

Values

Name	Number	Interpolation
eStep	0x0010	No interpolation, sectional definition
eLinear	0x0020	Linear
eSecondOrder	0x0030	Second order (parabola)
Hidden [`INTERNAL` **]**		

Name	Number	Interpolation
eStep_Equi	0x8010	Reserved - not implemented yet
eLinear_Equi	0x8020	Linear (equidistant)
eSecondOrder_Equi	0x8020	Second order (equidistant)

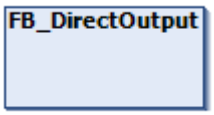
Obsolete

eNoneEqui	1	No	/
eNoneEquiLinIpol	2		Linear
eEquiLinIpol	3	Yes	Linear
eEquiSecondOrder	4		Second order (parabola)

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

5.8 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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_DirectOutput			
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.
FB_CorefunctionFeedback			
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.

Name	Type	Access	Description
IdleState	BOOL	Get, Set	The Corefunction is ready for operation and commandable.
IsCommanded	BOOL	Get	Signals the pending of a command.
FB_Corefunction			
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.
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.
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.

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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_DirectOutput	
DoActivate() [▶ 105]	This method enables and disables the direct output.

Interfaces

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

Example:

```
// Get specific interface I_AxisHydraulicBase from general interface I_AxisBase
IF __QUERYINTERFACE(iAxisBase, iAxisHydraulic) THEN

    IF bDirectOutput THEN

        // in this example, the reference is the electrical unit used
        iAxisHydraulic.DirectOutput.OutputReference := 10.0;    // [V]
        iAxisHydraulic.DirectOutput.Ramptime := 1.0;            // [s]
        iAxisHydraulic.DirectOutput.DoActivate(TRUE);

        // the set value will switch after 2 seconds
        fbTon.Pt := T#2S;
        IF fbTon.Q THEN
            iAxisHydraulic.DirectOutput.OutputValue := 3.0;    // [V]
        ELSE
            iAxisHydraulic.DirectOutput.OutputValue := 1.0;    // [V]
        END_IF
```

```
    ELSIF iAxisHydraulic.DirectOutput.IsCommanded THEN

        iAxisHydraulic.DirectOutput.DoActivate(FALSE);
    END_IF

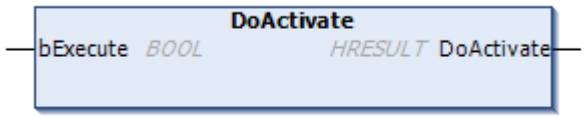
    fbTon.In := bDirectOutput;
    fbTon();

ELSE
    // interface is not set to an instance of FB_AxisHydraulicBase
    bIncompatible := TRUE;
END_IF
```

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

5.8.1 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

5.9 DisableSoftEnd



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.

Syntax:

```
FUNCTION_BLOCK FB_DisableSoftEnd EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_DisableSoftEnd			
Disabled	BOOL	Get	A TRUE signal reports active disabling.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_DisableSoftEnd	
DoDisable() [▶ 107]	This method disables the set software position limits.
ReEnable() [▶ 108]	This method re-enables the set software position limits.

Interfaces

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

Example:

```
// disable on entering setup mode
IF bEnterSetupMode THEN
    IF NOT iAxisBase.DisableSoftEnd.Disabled THEN

        iAxisBase.DisableSoftEnd.DoDisable(TRUE);

    ELSIF iAxisBase.DisableSoftEnd.IsCommanded THEN

        iAxisBase.DisableSoftEnd.DoDisable(FALSE);
    END_IF

// reenable on leaving setup mode
ELSIF bLeaveSetupMode THEN
    IF NOT iAxisBase.DisableSoftEnd.Disabled THEN

        iAxisBase.DisableSoftEnd.ReEnable(TRUE);

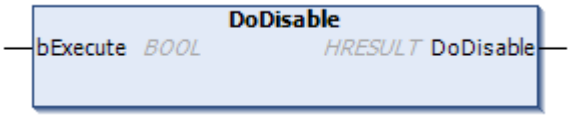
    ELSIF iAxisBase.DisableSoftEnd.IsCommanded THEN

        iAxisBase.DisableSoftEnd.ReEnable(FALSE);
    END_IF
END_IF
```

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

5.9.1 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.

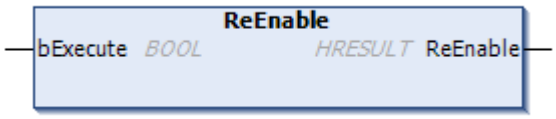
Inputs

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

5.9.2 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.

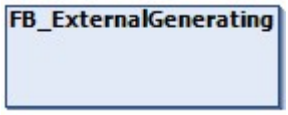
Requirements

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

5.10 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.



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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_ExternalGenerating			
Position	LREAL	Get, Set	Transfer of the set position.
Limiting	LREAL	Get, Set	Transfer of the active limitation value
Velocity	LREAL	Get, Set	Transfer of the target velocity.
TorqueLimiting[1]	LREAL	Get, Set	Transfer of the applied torque limit.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_ExternalGenerating	
DoEnable() [▶ 110]	Enables/disables external setpoint generation.
FB_Corefunction	
[INTERN] ApplyCommand()	Applies the command to the axis.
[INTERN] ClearStates()	Clears up some of the possible states.
[INTERN] DoReset(bExecute)	This method is called when the axis is reset.
[INTERN] Init(iMBase)	Initializes the Corefunction.
[INTERN] RemoveCommand()	Removes the command from the axis.

Interfaces

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

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

5.10.1 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

5.11 Estop



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

Syntax:

```
FUNCTION_BLOCK FB_Estop EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Estop			
NoCreeping	BOOL	Get, Set	A TRUE in this property avoids the creep phase at the end of the stopping process of a hydraulic axis.

Name	Type	Access	Description
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Estop	
DoEstop() [▶ 113]	A rising edge triggers the stop.

Interfaces

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

Example:

```
// Execute estop on pressing the emergency stop button
IF bEmergencyStopPressed THEN

    iAxisBase.Estop.DoEstop(TRUE);

// revert command
ELSIF iAxisBase.Estop.IsCommanded THEN

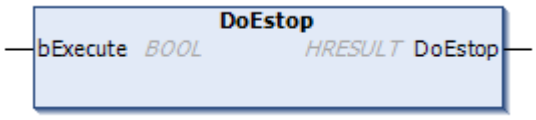
    iAxisBase.Estop.DoEstop(FALSE);

END_IF
```

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

5.11.1 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

5.12 Homing



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.

Syntax:

```
FUNCTION_BLOCK FB_HomingMain
```

Properties

Name	Type	Access	Description
Abort [▶ 116]	I_HomingAbort	Get	An active homing is aborted without success.
AbsoluteSwitch [▶ 118]	I_HomingAbsoluteSwitch	Get	Homing is performed at a position reported by a binary sensor.
AbsoluteSwitchDetect [▶ 122]	I_HomingAbsoluteSwitchDetection	Get	The actual position is recorded at a position reported by a binary sensor.
Block [▶ 125]	I_HomingBlock	Get	Homing is performed on a mechanical stop.
BlockDetect [▶ 129]	I_HomingBlockDetection	Get	The actual position is recorded at a mechanical stop.
Finish [▶ 132]	I_HomingFinish	Get	An active homing is completed successfully.
LimitSwitch [▶ 135]	I_HomingLimitSwitch	Get	Homing is performed at a position reported by a hardware limit switch.
LimitSwitchDetect [▶ 138]	I_HomingLimitSwitchDetection	Get	The actual position is recorded at a position reported by a hardware limit switch.

Interfaces

Type	Description
I_HomingMain	Standard interface on FB_HomingMain.

Example:

```
CASE nState OF
0:  // parameterise homing functions

    // homing block
    iAxisBase.Homing.Block.SetParameterGeneral(
        eDirection := E_AdaptableHomingDirection.eNegative,
        fVelocity := 20.0,
        fAcceleration := 0.0,
        fDeceleration := 0.0,
        fJerk := 0.0,
        tTimeLimit := T#60S,
        fDistanceLimit := 800.0,
        fTorqueLimit := 0.0
    );
    iAxisBase.Homing.Block.SetParameter(
        fSetPosition:=0.0,
        fDetectionVelocityLimit:=5.0,
        tDetectionVelocityTime:=T#0.5S,
        fTorqueTolerance:=0.0,
        bOptionsDisableDriveAccess := TRUE,
        bOptionsInstantLagReduction := TRUE,
        bOptionsTorquePolarityInverted := FALSE
    );

    // homing block detect
    iAxisBase.Homing.BlockDetect.SetParameterGeneral(
        eDirection:=E_AdaptableHomingDirection.ePositive,
        fVelocity:=50.0,
        fAcceleration:=0.0,
        fDeceleration:=0.0,
        fJerk:=0.0,
        tTimeLimit:=T#60S,
        fDistanceLimit:=800.0,
        fTorqueLimit:=0.0
    );
    iAxisBase.Homing.BlockDetect.SetParameter(
        fDetectionVelocityLimit:=5.0,
        tDetectionVelocityTime:=T#0.25S,
        fTorqueTolerance:=0.0,
        bOptionsDisableDriveAccess := FALSE,
        bOptionsInstantLagReduction := TRUE,
        bOptionsTorquePolarityInverted := FALSE
    );

    // homing finish
    iAxisBase.Homing.Finish.SetParameter(
        fDistance:=-20.0,
        fVelocity:=50.0,
        fAcceleration:=0.0,
        fDeceleration:=0.0,
        fJerk:=0.0,
        bOptionsDisableDriveAccess := FALSE
    );

    // start with homing on block
    IF FAILED(iAxisBase.Homing.Block.DoHoming(TRUE)) THEN
        nState := -1;
    ELSE
        nState := nState + 1;
    END_IF

1:  // after block, start block detect

    IF iAxisBase.Homing.Block.FailedState THEN
        nState := -1;
    ELSIF iAxisBase.Homing.Block.DoneState THEN

        iAxisBase.Homing.Block.DoHoming(FALSE);

        IF FAILED(iAxisBase.Homing.BlockDetect.DoHoming(TRUE)) THEN
            nState := -1;
        ELSE
            nState := nState + 1;
        END_IF
    END_IF

2:  // after blcok detect, finish
```

```
    IF iAxisBase.Homing.BlockDetect.FailedState THEN
        nState := -1;
    ELSIF iAxisBase.Homing.BlockDetect.DoneState THEN

        iAxisBase.Homing.BlockDetect.DoHoming(FALSE);

        IF FAILED(iAxisBase.Homing.Finish.DoFinish(TRUE)) THEN
            nState := -1;
        ELSE
            nState := nState + 1;
        END_IF
    END_IF

3:  // waiting for finish

    IF iAxisBase.Homing.Finish.FailedState THEN
        nState := -1;
    ELSIF iAxisBase.Homing.Finish.DoneState THEN

        iAxisBase.Homing.Finish.DoFinish(FALSE);

        nState := nState + 1;
    END_IF

4,  // done
-1: // error

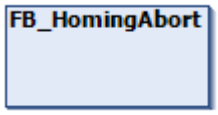
    iAxisBase.Homing.Block.DoHoming(FALSE);
    iAxisBase.Homing.BlockDetect.DoHoming(FALSE);
    iAxisBase.Homing.Finish.DoFinish(FALSE);

END_CASE
```

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

5.12.1 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() [▶ 117]	A rising edge triggers the abort.
SetParameter() [▶ 117]	The procedure for returning to normal operation can be specified.

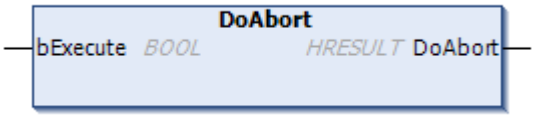
Interfaces

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

5.12.1.1 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_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 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

5.12.1.2 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.

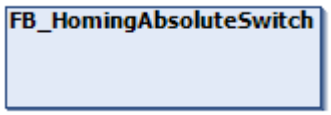
Non-Beckhoff servo drives

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

5.12.2 AbsoluteSwitch



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() [▶ 119]	This method triggers the homing.

Name	Description
SetParameter() [▶ 120]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 121]	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

5.12.2.1 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

5.12.2.2 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.

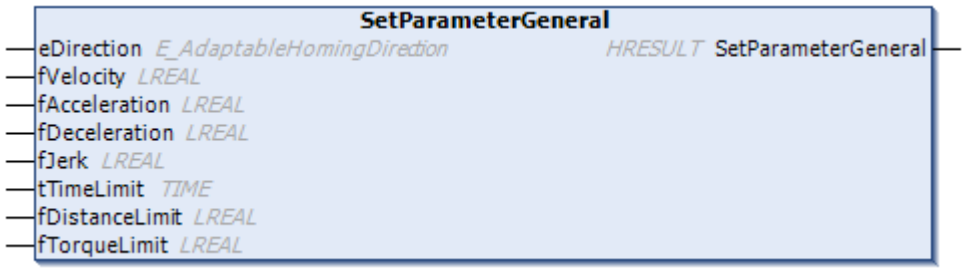
Inputs

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 following error monitoring 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

5.12.2.3 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.

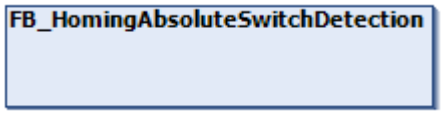
Torque limitation in the event of a mechanical blockage

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

5.12.3 AbsoluteSwitchDetect



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() [▶ 123]	This method triggers the homing.
SetParameter() [▶ 123]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 124]	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

5.12.3.1 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

5.12.3.2 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.

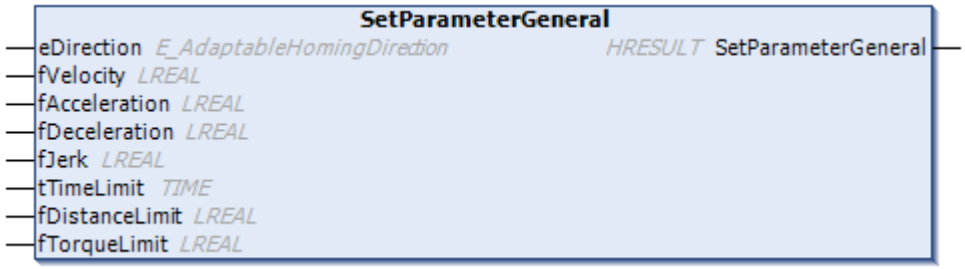
Inputs

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 following error monitoring 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

5.12.3.3 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

5.12.4 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() [▶ 126]	This method triggers the homing.
SetParameter() [▶ 127]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 128]	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

5.12.4.1 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.

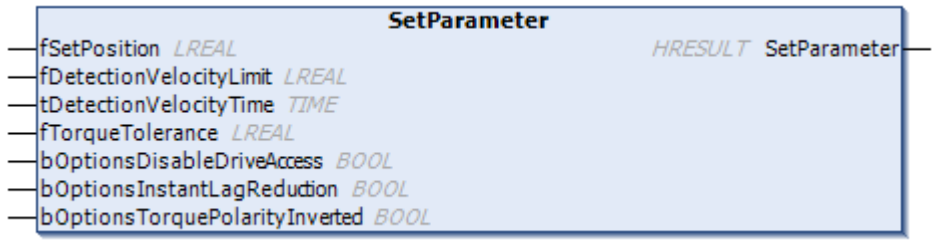
Inputs

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

5.12.4.2 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. **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. **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 (gear units, etc.).**

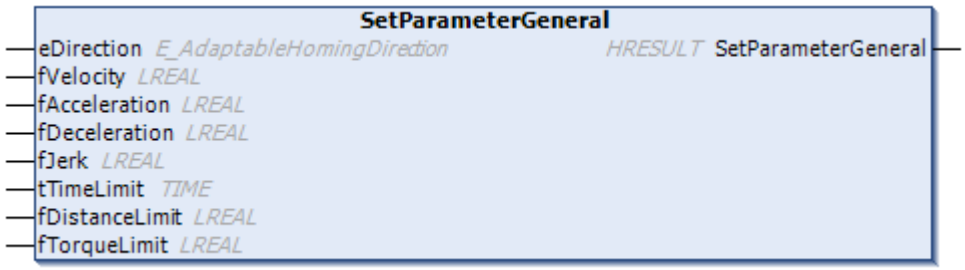
The homing event

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

5.12.4.3 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

5.12.5 BlockDetect



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() [▶ 129]	This method triggers the homing.
SetParameter() [▶ 130]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 131]	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

5.12.5.1 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.

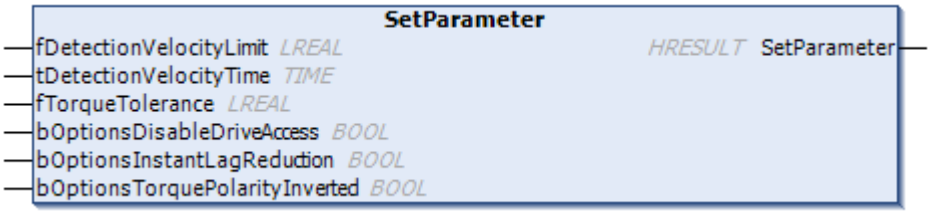
Inputs

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

5.12.5.2 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. **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. **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 (gear units, etc.).**

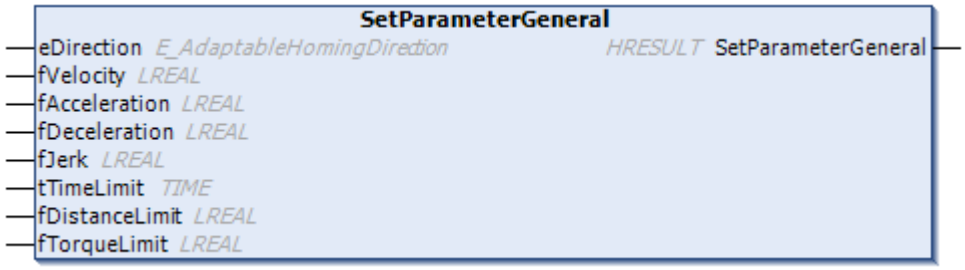
The homing event

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

5.12.5.3 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.
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

5.12.6 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() [▶ 133]	A rising edge triggers the termination.
SetParameter() [▶ 134]	The procedure for returning to normal operation can be specified.

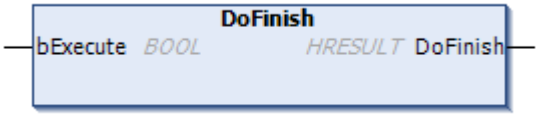
Interfaces

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

5.12.6.1 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.

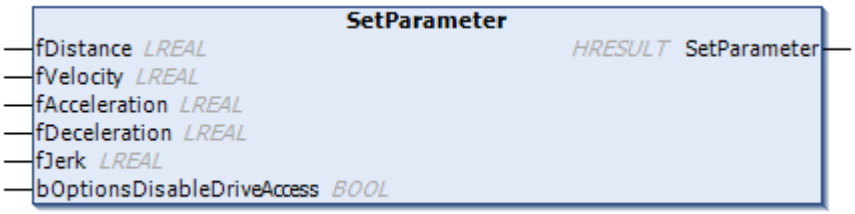
Inputs

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

5.12.6.2 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.

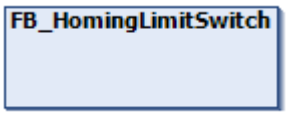
Inputs

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. **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

5.12.7 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() [▶ 135]	This method triggers the homing.
SetParameter() [▶ 136]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 137]	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

5.12.7.1 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

5.12.7.2 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

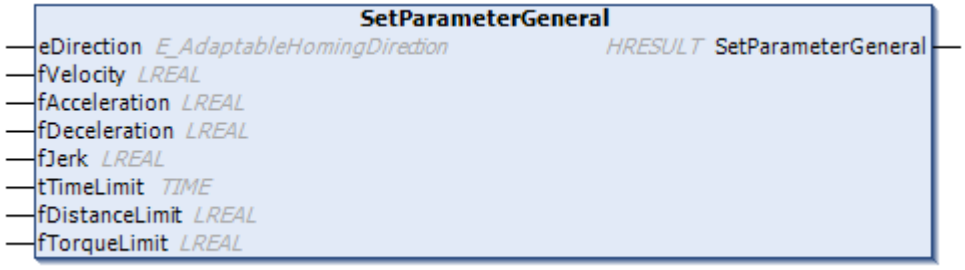
Name	Description
fSetPosition	Specifies the value that is applied to the actual position at the homing event.

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

5.12.7.3 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.

Name	Type	Description
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.

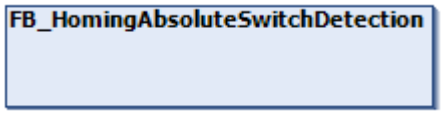
Torque limitation in the event of a mechanical blockage

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

5.12.8 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() [▶ 139]	This method triggers the homing.
SetParameter() [▶ 140]	A set of parameters specific to this homing procedure is set.
SetParameterGeneral() [▶ 140]	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

5.12.8.1 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_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

5.12.8.2 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.

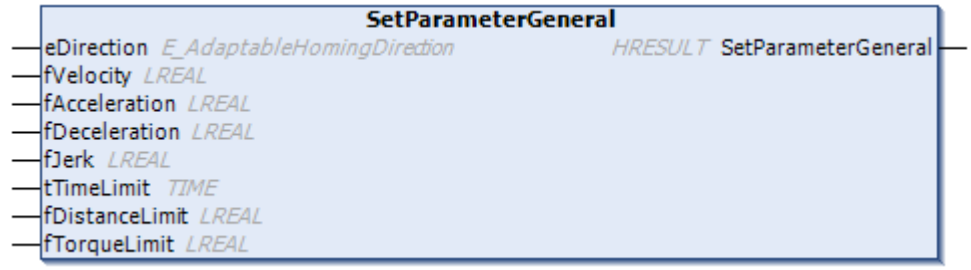
Inputs

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

5.12.8.3 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.
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

5.13 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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Jog			
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.
Limiting	LREAL	Get, Set	Setpoint for limiting (e.g. torque, pressure) during jogging.
TorqueLimiting[1]	LREAL	Get, Set	This sets the torque limit during jogging.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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.

[1] Obsolete

Methods

Name	Description
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Jog	
DoJogM() [▶ 143]	This method performs the start and stop of the axis in the negative direction.
DoJogP() [▶ 144]	This method performs the start and stop of the axis in the positive direction.
SetParameter() [▶ 145]	This method is used to set the parameters of a movement in jogging mode.

Interfaces

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

Example:

```
IF bJogPos THEN

    hresult := iAxisBase.Jog.SetParameter(
        fAcceleration := 0.0,    // default
        fDeceleration := 0.0,    // default
        fJerk := 0.0,            // default
        fVelocity := 20.0        // [mm/s]
    );
    iAxisBase.Jog.DoJogP(TRUE);

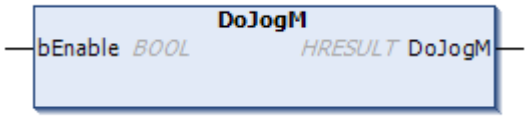
ELSIF iAxisBase.Jog.IsCommanded THEN

    iAxisBase.Jog.DoJogP(FALSE);
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

5.13.1 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.

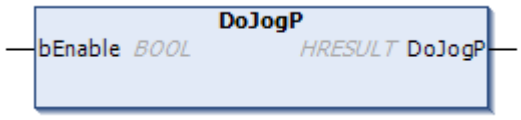
Inputs

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

5.13.2 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_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 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

5.13.3 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.

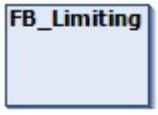
Inputs

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

5.14 Limiting



This Corefunction is responsible for limiting all axis functions. The type of limitation depends on the drive technology. For example, the torque of electric axes and the pressure of hydraulic axes can be limited. The following function blocks are automatically available internally for axes of type `FB_AxisNcBase`:

- FB_LimitingOutputCoE [▶ 258] – DS402 (AX8000)-based TorqueLimiting
- FB_LimitingOutputSoE [▶ 259] – AX5000-Based torque limiting

This Corefunction is one of the permanently active Corefunctions.

Syntax:

```
FUNCTION_BLOCK FB_Limiting EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Limiting			
Actual	LREAL	Get	Current actual value of the limited variable (torque/ pressure)
CmdReference	LREAL	Get, Set	Reference value for scaling the value commanded in a Corefunction. Typically, 100% is specified here.
EffectiveSetpoint	LREAL	Get	The effective value currently being passed to the output.
ExternalMode	BOOL	Get	TRUE as soon as external limiting has been assigned according to the outdated scheme using I_LimitingFb.
IdleValue	LREAL	Get, Set	The idle value for the limitation to which the axis returns after a command.
MaxValue	LREAL	Get, Set	The effective limit of the limit for each command.
Mode [▶ 148]	I_LimitingMode	Get	Choice of the operation mode for the limiting function.
Output	I_LimitingOutput [▶ 258]	Get, Set	Interface to the output element for the limit.
SetpointCmded	LREAL	Get, (Set)	Setpoint currently specified by the active Corefunction.
SetpointDirection	E_AdaptableDirection	Get, (Set)	The direction currently specified by the active core function.
Transition [▶ 147]	I_LimitingTransition	Get	Configuration of the transition between command-based and idle-mode limiting.
ActualTorque[1]	LREAL	Get, Set	Current actual torque value of the axis. Alternatively: `Actual`
Activate[1]	BOOL	Get, Set	Activates the use of the ActiveValue as an active limit. Alternatively: `Cmd()`
ActiveValue[1]	LREAL	Get	Limitation value currently available Alternatively: `EffectiveSetpoint`
ConnectedToDrive[1]	BOOL	Get	An internal connection to a drive unit has been established. Alternatively: `MotionParams.HasAutoLimiting`
DefaultFb[1]	I_LimitingFb	(Get, Set)	Access to the standard function block for torque limitation if specific handling is used. Alternatively: `SetExternal()`
Direction[1]	E_AdaptableDirection	Get, Set	Direction in which the torque limit should act. Alternatively: `SetpointDirection`
NominalValue[1]	LREAL	Get, Set	The value specified by a core function Alternatively: `SetpointCmded`
ReferenceValue[1]	LREAL	Get, Set	Reference value for the maximum command value of the limit. Alternatively: `CmdReference`
FB_Corefunction			
FailedState	BOOL	Get, Set	If IsActivated is TRUE at the same time: Signals the failure of an accepted command.

Name	Type	Access	Description
			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.
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.
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.

[1]Obsolete

Methods

Name	Description
[INTERN] Cmd(bActivate)	Enables/disables the default limit set by a core function. **This method should not be called from within the application.**

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_Limiting	Standard interface to FB_Limiting.
I_LimitingDev	Extended interface for integration with Corefunction.
I_LimitingOutput	Internal implementation to the output interface
I_LimitingFb[1]	Interface for the implementation of an external limiting function Alternatively: /

[1]Obsolete

Example:

```
// configuration of some settings
iAxisBase.Limiting.CmdReference := 100.0; // %
iAxisBase.Limiting.Mode.Adaptive := TRUE;
iAxisBase.Limiting.Transition.HoldLow := TRUE;
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.14.1 Transition

This interface defines the transition between the limiting of a command and idle-mode limiting. The interface works as a selection function that is activated by setting one of the properties. If you make a choice of mutually exclusive options at the same time, your current choice may change.

Syntax:

```
INTERFACE I_LimitingTransition
```

Properties

Name	Type	Access	Description
HoldHigh	BOOL	Get, Set	When a command ends and the axis continues to apply the limit set in the command (e.g., 80%), the transition back to the idle value is delayed until the `Actual` value has dropped to the idle value.
HoldLow	BOOL	Get, Set	When a command ends and the axis continues to apply the reduced limit specified in the command (e.g., 10%), the transition back to the idle value is delayed until the value `Actual` has decreased to the last limit value multiplied by `LowTolerance`.
LowTolerance	LREAL	Get, Set	Tolerance for the factor-based value of `Actual` required to perform a transition using the `HoldLow` option.
NoHold	BOOL	Get, Set	If TRUE, no delay is applied to idle transitions.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.14.2 Mode

This interface defines the mode in which direction-based limiting is performed. The interface works as a selection function that is activated by setting one of the properties. The previously set choice is automatically reset.

Syntax:

```
INTERFACE I_LimitingMode
```

Properties

Name	Type	Access	Description
Adaptive	BOOL	Get, Set	The direction of the limit takes into account the set acceleration of the axis.
DirectionalFixed	BOOL	Get, Set	The directional application is performed statically in the intended direction of travel.
NonDirectional	BOOL	Get, Set	Direction is not taken into account.

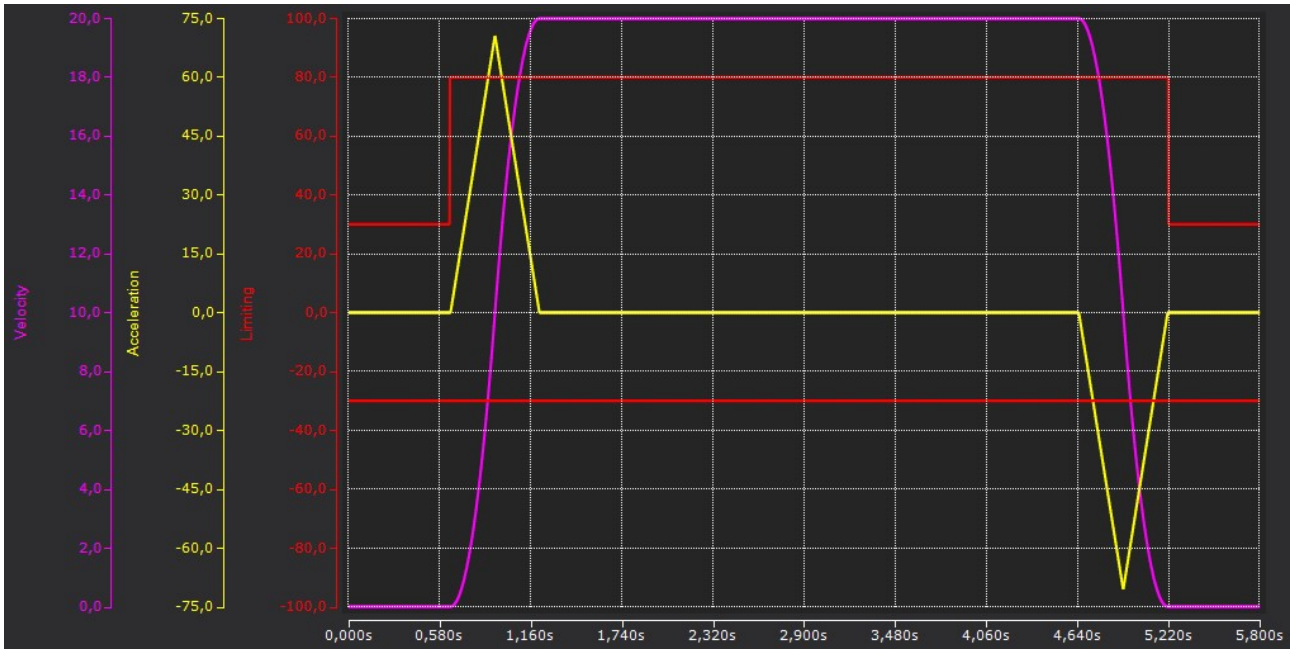
Dynamically adaptive (Adaptive):

The limit adjusts to the current dynamics. When accelerating or traveling at a constant speed, the directional limit is applied in the specified direction. In the event of a deceleration, the directional restriction is also applied in the opposite direction.



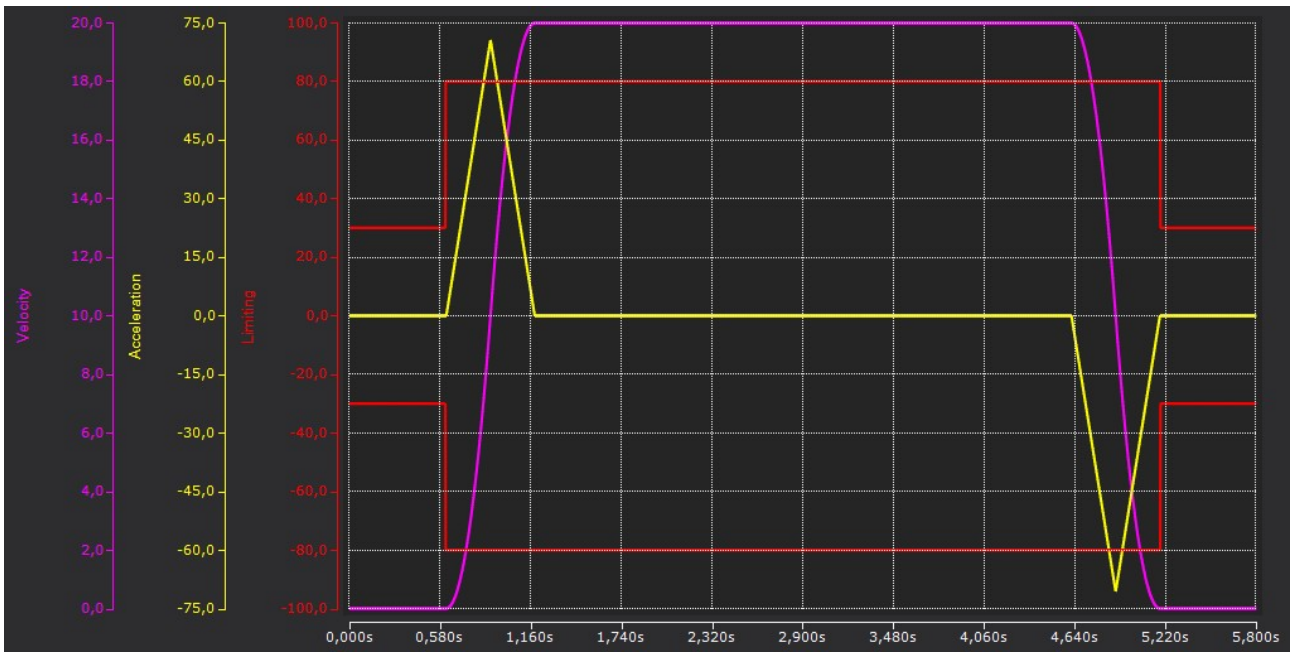
Directional (DirectionalFixed):

The restriction applies only to the direction of travel.



Direction-independent (NonDirectional):

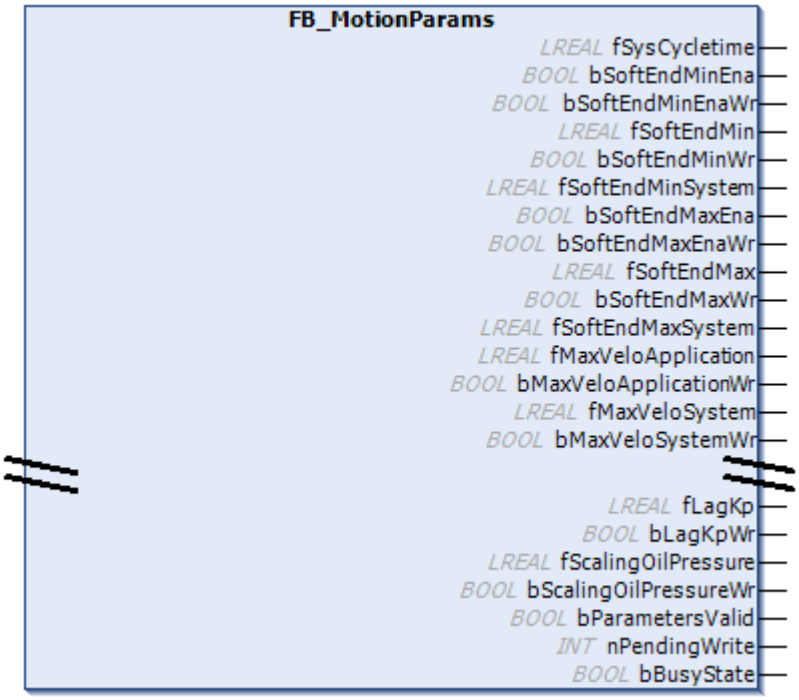
Directions are not taken into account and are always applied simultaneously.



Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.15 MotionParams



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

The writeable parameters can be saved using the `Persist` option with the functionality provided by NC (TF5000) or Hydraulic Library (TF5810). Alternatively, Plastic Technology Functions also offers a storage option via the machine data [▶ 263]. For this purpose, `FB_MotionParams` already implements a fully-fledged machine data component [▶ 156]. You will find an example of the setup in the lower part of this description.

Support of the parameters depends on the technology

Some parameters are only available for specific drive technologies. Please note the comments in the “Selection” 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

Axis may need to be disabled

Some parameters cannot be written while an axis is enabled.

Syntax:

```
FUNCTION_BLOCK FB_MotionParams EXTENDS FB_MdMotionParams
```

Properties

Name	Type	Access	Selection	Description
ActuatorBase [▶ 155]	I_Actuator ParamsH	Get		Access to parameters for the base position of an actuator.
ActuatorWork [▶ 155]	I_Actuator ParamsH	Get		Access to parameters for the working position of an actuator.
AsymTargeting	BOOL	Get, Set		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		TRUE if the braking distance is determined automatically.
AxisIsNc	BOOL	Get	/	TRUE, if the axis is operated with TwinCAT NC.
BrakeDistanceM	LREAL	Get, Set		The braking distance in negative direction. See also AsymTargeting.
BrakeDistanceP	LREAL	Get, Set		The braking distance in positive direction.
CreepDistanceM	LREAL	Get, Set		The creep distance in negative direction. See also AsymTargeting.
CreepDistanceP	LREAL	Get, Set		The creep distance in positive direction.
CreepSpeedM	LREAL	Get, Set		The creep speed in negative direction.
CreepSpeedP	LREAL	Get, Set		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.
CylinderAreaA	LREAL	Get, Set		Cylinder area of the A side of the activated hydraulic cylinder (positive direction).
CylinderAreaB	LREAL	Get, Set		Cylinder area of the B side of the activated hydraulic cylinder (negative direction).

Name	Type	Access	Selection	Description
DefaultAcceleration	LREAL	Get, Set		The acceleration used when a command is executed with an acceleration of 0.
DefaultDeceleration	LREAL	Get, Set		The deceleration applied when a command is executed with an acceleration of 0.
DefaultJerk	LREAL	Get, Set	/	Reserved for future use.
DriveAcceptBlockedSignal	BOOL	Get, Set		If TRUE, the axis does not immediately enter the error state as soon as the drive stops following the setpoint due to a stop signal (e.g., limit switch).
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		TRUE if the drive of the axis is inverted. **The outputs are swapped for actuators.**
DriveTorqueWeighting	LREAL	Get	/	Scaling factor of the actual torque value of the drive.
DriveType	UDINT	Get	/	A numeric identifier for the type of the connected drive. **The meaning of the constants is defined in the Tc2 NC or the Tc2_Hydraulics library.**
EncoderInterpolation	LREAL	Get, Set		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.
EncoderMask	UDINT	Get	/	Encoder mask for encoder value overflow
EncoderMode	UDINT	Get, Set		Operation mode of the encoder
EncoderReversed	BOOL	Get, Set		TRUE if the encoder / feedback of the axis is inverted. **The feedback inputs are swapped for actuators.**
EncoderType	UDINT	Get	/	A numeric identifier for the type of the connected encoder. **The meaning of the constants is defined in the Tc2 NC or the Tc2_Hydraulics library.**
EncoderWeighting	LREAL	Get, Set		The factor for encoder evaluation. It specifies the distance corresponding to a number of increments specified by EncoderInterpolation.
EncoderZeroShift	LREAL	Get, Set		The zero point shift of the encoder.
HasAutoLimiting	BOOL	Get	/	TRUE if the axis uses an internal automatic limiting function.
LagControlled	BOOL	Get	/	TRUE if the axis has a position controller.
LagCtrlKp	LREAL	Get, Set		The kP factor of the position controller.
LagFilter	LREAL	Get, Set		The filter time of the following error monitoring.

Name	Type	Access	Selection	Description
LagLimit	LREAL	Get, Set		The threshold value of the following error monitoring.
LagMonitored	BOOL	Get, Set		TRUE, if the following error monitoring of the axis is active.
MaxAcceleration	LREAL	Get, Set		The maximum permissible acceleration. **A write operation on this parameter is always also applied to DefaultAcceleration.**
MaxDeceleration	LREAL	Get, Set		The maximum permissible deceleration. **A write operation on this parameter is always also applied to DefaultDeceleration.**
MaxJerk	LREAL	Get, Set		The maximum permissible jerk.
MaxVeloApplication	LREAL	Get, Set		An additional velocity restriction.
MaxVeloSystem	LREAL	Get, Set		The maximum velocity of the axis. **This parameter acts as a reference value for the maximum velocity output of the drive interface.**
MinVeloApplication	LREAL	Get, Set		The minimum velocity of the axis.
Persist	BOOL	Get, Set		Enables the persistent saving of parameter write operations of this function block on the target system.
ScalingOilPressure	LREAL	Get, Set		The scaling factor for the actual pressure acquisition.
SoftEndMax	LREAL	Get, Set		The maximum software limit switch.
SoftEndMaxEna	BOOL	Get, Set		Enable for the maximum software limit switch.
SoftEndMaxSystem	LREAL	Get, Set		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		The minimum software limit switch.
SoftEndMinEna	BOOL	Get, Set		Enable for the minimum software limit switch.
SoftEndMinSystem	LREAL	Get, Set		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.
VeloMonitored	BOOL	Get, Set		TRUE if the velocity monitoring of the axis is active.
VeloLagLimit	LREAL	Get, Set		The tolerance window for triggering the velocity monitoring. **For actuators, the limit is interpreted as time.**

Name	Type	Access	Selection	Description
BusyState	BOOL	Get	/	TRUE, if the function block writes changed parameters.
ReadyState	BOOL	Get	/	TRUE if the function block has loaded all parameters at least once.
MaxAccApplication[1]	LREAL	Get, Set	Obsolete	Returns the value of DefaultAcceleration
MaxDecApplication[1]	LREAL	Get, Set	Obsolete	Returns the value of DefaultDeceleration
HasTorqueLimiting[1]	BOOL	Get	Obsolete	Returns the value of HasAutoLimiting.
MinVeloJog[1]	LREAL	Get, Set	Obsolete	Returns the value of MinVeloApplication.

[1]Obsolete

Legend

- TF5810 Hydraulic Positioning
  FB_AxisHydraulicBase
- TF5000 NC PTP
  FB_AxisNcBase
  FB_AxisNcTrafoBase
- TF8560 Inverter Axis
  FB_AxisInvBase
- TF8560 Digital Actuator
  FB_ActuatorBase

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_MotionParams	Standard interface on FB_MotionParams.
I_MdComponent	Interface for use as machine data component

Example:

```
iAxisBase.MotionParams.SoftEndMax := fAxisStroke;
```

Example | Configure machine data:

```
VAR
    iComponent:         I_MdComponent;
    fbAxisData:         FB_MdContainer;
END_VAR

// query the compatible component interface from the axis motion params
IF __QUERYINTERFACE(iAxisBase.MotionParams, iComponent) AND NOT bAppend THEN
    bAppend := fbAxisData.AddComponent(iComponent);
END_IF

// A saving commands is received
IF bSave THEN
    fbAxisData.Save := TRUE;
    IF fbAxisData.Done THEN
        fbAxisData.Save := FALSE;
        bSave := FALSE;
    END_IF
END_IF

fbAxisData.Cyclic();
```

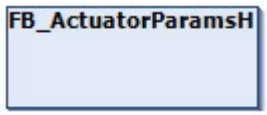
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.15.1 ActuatorParamsH

Exclusive function for actuators

This core function is only available for actuators.



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. **The output remains pending only if the end position has been reached.**
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.
ReturnEarly	BOOL	Get, Set	Allows the time-based reset command to be triggered without the end position having 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 v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.15.2 MdMotionParams



This class implements a machine data component to store the writable MotionParams [▶ 150] of an axis.

Syntax:

```
FUNCTION_BLOCK FB_MdMotionParams EXTENDS FB_Corefunction
```

Machine data

Index (Identifier)	Parameter
0x0000 03E8	SoftEndMinEna
0x0000 03E9	SoftEndMin
0x0000 03EA	SotEndMaxEna
0x0000 03EB	SoftEndMax
0x0000 044C	VeloMaxApplication
0x0000 044D	VeloMaxSystem
0x0000 044F	VeloMinApplication
0x0000 04B0	MaxAcceleration
0x0000 04B1	MaxDeceleration
0x0000 04B2	MaxJerk
0x0000 04B3	DefaultAcceleration
0x0000 04B4	DefaultDeceleration
0x0000 04B5	DefaultJerk
0x0000 07D1	EncoderReversed
0x0000 07D2	EncoderWeighting
0x0000 07D3	EncoderInterpolation
0x0000 07D4	EncoderZeroShift
0x0000 07D5	EncoderMode
0x0000 0BB8	DriveReversed
0x0000 0BBB	DriveAcceptBlockedSignal
0x0000 0FA1	LagKp
0x0000 0FA2	CreepDistanceP
0x0000 0FA3	CreepDistanceM
0x0000 0FA4	CreepSpeedP
0x0000 0FA5	CreepSpeedM
0x0000 0FA6	BrakeDistanceP
0x0000 0FA7	BrakeDistanceM
0x0000 0FA8	AutoBrakeCalculation
0x0000 0FA9	AsymTargeting
0x0000 1388	LagMonitored
0x0000 1389	LagLimit
0x0000 138A	LagFilter
0x0000 138B	VeloLagMonitored
0x0000 138C	VeloLagLimit
0x0000 1770	ScalingOilPressure

Index (Identifier)	Parameter
0x0000 1771	CylinderAreaA
0x0000 1772	CylinderAreaB

Event-driven methods (callback methods)

Name	Description
MdNextParameter(...) [▶ 266]	Container call to save the parameters to a file.
MdSetParameter(...) [▶ 266]	Container call to load the parameters into the runtime.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_MdComponent	Interface for use as machine data component

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

5.16 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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_MotionSetpoints			
Acceleration	LREAL	Get	The current acceleration setpoint.
Jerk	LREAL	Get	The current jerk setpoint.
Position	LREAL	Get	The current position setpoint.
Limiting	LREAL	Get	The current setpoint of the limiting function (e.g. torque, pressure).
Velocity	LREAL	Get	The current velocity setpoint.
TorqueLimiting[1]	LREAL	Get	The current setpoint for torque limitation.
FB_Corefunction			

Name	Type	Access	Description
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.
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.
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.

[1]Obsolete

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_MotionSetpoints	Standard interface on FB_MotionSetpoints.

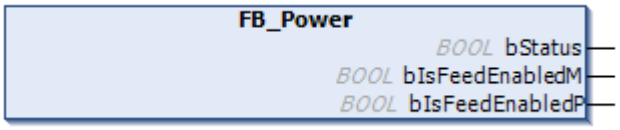
Example:

```
// calculating the difference between target and actual velocity
fVeloLag := ABS(iAxisBase.Actuals.Velocity) - ABS(iAxisBase.MotionSetpoints.Velocity);
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

5.17 Power



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

Syntax:

```
FUNCTION_BLOCK FB_Power EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Power			

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. **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. **For an active movement, the direction-related enable is also required.**
Timeout	LREAL	Get, Set	If there is no response, an axis error is triggered after the time-out period.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Power	
DoPower() [▶ 160]	Enable for active operation of the axis.
FeedEnable() [▶ 161]	Directional enables for commanding active axis movements.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_Power	Standard interface on FB_Power.

Type	Description
I_PowerDev	Extended interface with access to the `Timeout` property.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

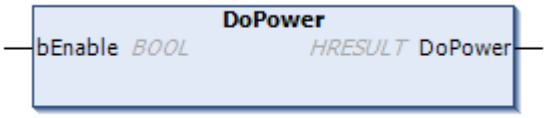
Example:

```
iAxisBase.Power.DoPower(bEnable);
iAXisBase.Power.FeedEnable(bEnable, bEnable);
```

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

5.17.1 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

5.17.2 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

5.18 PressureControl



This Corefunction is provided to allow access to a number of controller types. There is currently an extended PID controller [▶ 162] and a supervisor for reading the pressure value and switching to pressure control.

Syntax:

```
FUNCTION_BLOCK FB_PressureControlMain
```

Properties

Name	Type	Access	Description
PID [▶ 162]	I_PressureControl	Get	Extended PID controller.
Supervisor [▶ 171]	I_ProcessSupervisor	Get	Infrastructure and organization of pressure control.

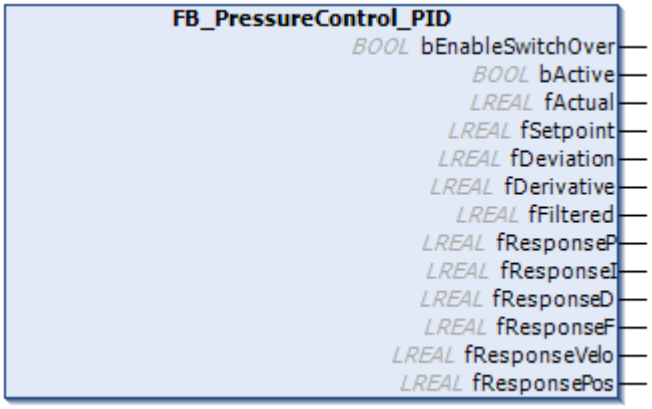
Interfaces

Type	Description
I_PressureControlMain	Standard interface on FB_PressureControlMain.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.18.1 PressureControl.PID



This core function implements an extended PID controller functionality. The pressure value read by the Supervisor [▶ 171] is used as the controlled variable. The velocity of the axis is controlled as the manipulated variable. Accordingly, this control structure can be used for a range of tasks:

- Holding pressure control in injection molding machines
- Back pressure control in injection molding machines
- Melt pressure control for extruders
- Others

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_PressureControl_PID			

Name	Type	Access	Description
Actual	BOOL	Get	Current actual pressure value.
AllowSwitchOver	BOOL	Get, Set	Allows switching from another active Corefunction to pressure control. **Currently, only `Ptp` is supported for switching.**
Limiting	LREAL	Get, Set	Setpoint for the limitation (e.g., torque, pressure) during pressure control.
Parameters [▶ 166]	I_PressureControlParams_PID	Get, Set	Parameter set that is used for control. The parameter sets can be switched by assignment. If a 0 is assigned, the internal parameter set is used.
PressureLookUp	I_PressureLookUp	Get, Set	Pressure setpoint table for sequential processing of Pressure setpoints. **Preliminary implementation**
Target	LREAL	Get, Set	Command variable (setpoint) of the controller.
IsEnabled[1]	BOOL	Get	A TRUE signals the active state of the pressure switchover. Alternatively: `.AllowSwitchOver`
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
Activate() [▶ 164]	Activate / deactivate the controller.
SwitchOver() [▶ 165]	Activate the controller by switching another command.
EnableSwitchOver() [▶ 329][1]	Enable for automatic activation by a PressureHandler. Alternatively: `AllowSwitchOver`
GetActual() [▶ 330][1]	The actual value of the controller is determined. Alternatively: `Actual`

Name	Description
GetParams() [▶ 330][1]	An interface to the connected parameter set is determined. Alternatively: `Parameters`
SetParams() [▶ 331][1]	A parameter set is connected to the controller. Alternatively: `Parameters`
Setpoint() [▶ 331][1]	Set the command variable (setpoint) of the controller. Alternatively: `Target`

[1]Obsolete

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_PressureControl	Standard interfaces on FB_PressureControl_PID.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Example:

```
// Requires connected pressure input, e.g. via PressureControl.Supervisor.Sensor
// Start pressure control on application start signal
IF bStart THEN

    // Assign pressure setpoint
    iAxisBase.PressureControl.PID.Target := 20.0;

    // parameterize once before activation and activate
    IF iAxisBase.PressureControl.PID.IdleState THEN

        iAxisBase.PressureControl.PID.Limiting := 80.0;
        iAxisBase.PressureControl.PID.Parameters.Kp := 0.5;
        iAxisBase.PressureControl.PID.Parameters.EnableI := TRUE;
        iAxisBase.PressureControl.PID.Parameters.Tn := 1.0;

        iAxisBase.PressureControl.PID.Activate(TRUE);

    // wait until function is terminated
    ELSIF bStop THEN

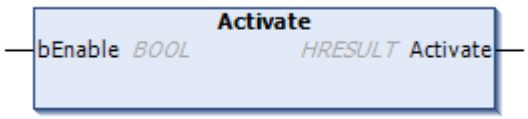
        bStart := FALSE;
        iAxisBase.PressureControl.PID.Activate(FALSE);

    END_IF
END_IF
```

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.18.1.1 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.

5.18.1.2 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

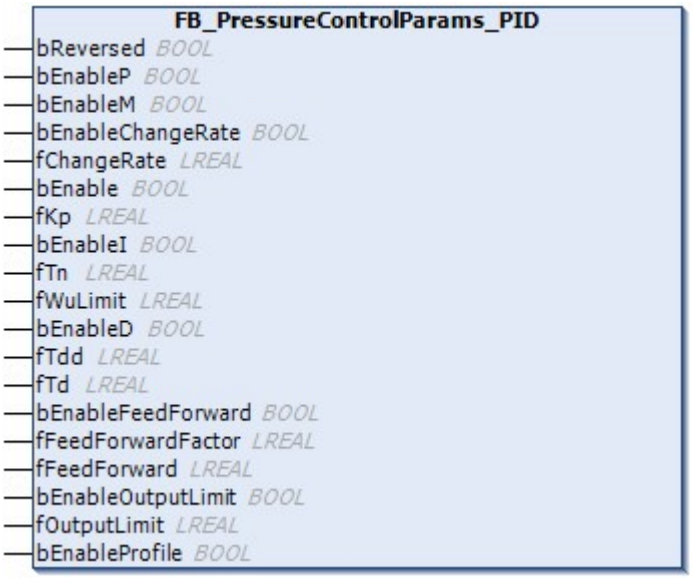
The return value of the method is of type HRESULT. The following return values are to be expected.

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.

5.18.1.3 FB_PressureControlParams_PID



This function block contains a parameter set to be used by a FB_PressureControl_PID [▶ 162] function block.

Syntax:

```
FUNCTION_BLOCK FB_PressureControlParams_PID
```

Properties

Name	Type	Access	Properties
ChangeRate	LREAL	Get, Set	The ramp rate used in pressure units per second (typ. [bar/s]). **Further notes at** `EnableChangeRate`**.**
Enable	BOOL	Get, Set	This property is used to activate and disable the proportional component of the PID controller. **Further notes at** `Kp`**.**
EnableChangeRate	BOOL	Get, Set	Enables a setpoint ramp function. **The internally used setpoint for the control can be updated to the specified setpoint at a limited rate, specified as** `ChangeRate`**.** **A setting of** `ChangeRate=0.0` **or** `EnableChangeRate=FALSE` **disables the ramp function and causes the internal setpoint to immediately follow the specified setpoint.**
EnableOutputLimit	BOOL	Get, Set	Enables a limiting function for the output of the controller (manipulated variable). **Further notes at** `OutputLimit`**.**
EnableD	BOOL	Get, Set	Enables the differential component of the PID controller. **Further notes at** `Td`**.**
EnableFeedForward	BOOL	Get, Set	Enables the velocity feedforward control. **Further notes at** `FeedForward`**.**
EnableI	BOOL	Get, Set	Enables the integrating part of the PID controller. **Further notes at** `Tn`**.**
EnableNeg	BOOL	Get, Set	Enables a negative output of the controller.
EnablePos	BOOL	Get, Set	Enables a positive output of the controller.

Name	Type	Access	Properties
EnableProfile	BOOL	Get, Set	Enables the use of the setpoint profile via `PID.PressureLookUp` **Preliminary implementation**
FeedForward	LREAL	Get, Set	Feedforward values for velocity feedforward control. **If `EnableFeedForward` is set to `TRUE`, the value of `FeedForward` is multiplied by `FeedForwardFactor` and added to the PID response output.** **In the case of back pressure control, the reverse velocity of the injector can be adapted to the effect of the dosing axis using feedforward control. This can make a positive contribution to the control quality when dynamically adjusting the dosing speed.**
FeedForwardFactor	LREAL	Get, Set	Gain factor for the velocity feedforward control. **Further notes at `FeedForward`.**
Kp	LREAL	Get, Set	The proportional gain of the PID controller. `Enable` must be `TRUE` to release the effect. The value is measured in velocity unit per pressure unit (typically [mm/s / bar].
OutputLimit	LREAL	Get, Set	Limitation for the output of the controller.
Reversed	BOOL	Get, Set	Reverses the output of the PID controller. **In some use cases, the axis must move in the positive direction to relieve pressure. A typical example is holding pressure and back pressure control in injection molding machines.**
Td	LREAL	Get, Set	The differential component of the PID controller. The value is measured in velocity units multiplied by seconds per pressure unit (typically [mm/s*s/bar]) **The response is calculated if `EnableD=TRUE` and `Td` and `Tdd>=CycleTime`, otherwise it is `0`.**
Tdd	LREAL	Get, Set	Attenuation of the differential component of the PID controller. **Further notes at `Td`.**
Tn	LREAL	Get, Set	The integrating component of the PID controller. The value is calculated in velocity units per pressure unit and seconds (typically [mm/s / (bar * s)]) **The response is calculated if `EnableI=TRUE` and `Tn>=CycleTime`, otherwise it is `0`. The output is limited to `WuLimit`.**
WuLimit	LREAL	Get, Set	Effective limitation of the integrating component of the PID controller. **Further notes at `Tn`.**
EnableM[1]	BOOL	Get, Set	Obsolete variant of `.EnableNeg`
EnableP[1]	BOOL	Get, Set	Obsolete variant of `.EnablePos`
EnableClipping[1]	BOOL	Get, Set	Obsolete variant of `.EnableOutputLimit`

[1]Obsolete

Methods

Name	Description
GetBoolParameter [▶ 168]	This method is used to read BOOL parameters of the controller. For details see E_PressureControlParam [▶ 170].
GetFloatParameter [▶ 169]	This method is used to read LREAL parameters of the controller. For details see E_PressureControlParam [▶ 170].
SetBoolParameter [▶ 169]	This method is used to define BOOL parameters of the controller. For details see E_PressureControlParam [▶ 170].
SetFloatParameter [▶ 170]	This method is used to define LREAL parameters of the controller. For details see E_PressureControlParam [▶ 170].

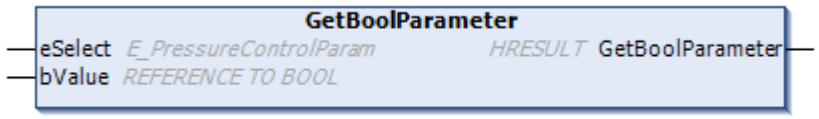
Interfaces

Name	Description
I_PressureControlParams_PID	Standard interface to FB_PressureControlParams_PID

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.18.1.3.1 GetBoolParameter



This method is used to read BOOL parameters of the controller. For details see E_PressureControlParam [▶ 170].

Syntax:

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

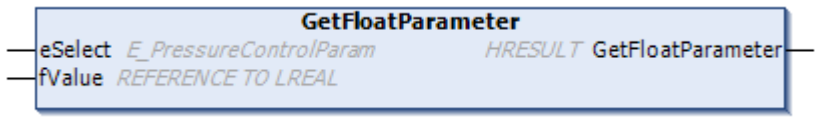
Return value

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.

5.18.1.3.2 GetFloatParameter



This method is used to read LREAL parameters of the controller. For details see E_PressureControlParam [▶ 170].

Syntax:

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

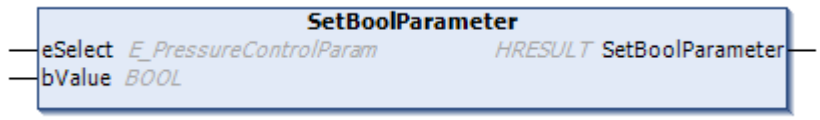
Return value

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.

5.18.1.3.3 SetBoolParameter



This method is used to define BOOL parameters of the controller. For details see E_PressureControlParam [▶ 170].

Syntax:

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

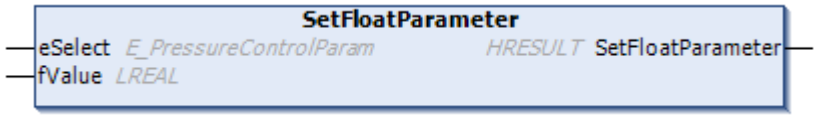
Return value

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.

5.18.1.3.4 SetFloatParameter



This method is used to define LREAL parameters of the controller. For details see E_PressureControlParam [▶ 170].

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.

5.18.1.3.5 E_PressureControlParam

The values of this enumeration are used by GetBoolParameter(), GetFloatParameter(), SetBoolParameter() and SetFloatParameter() of FB_PressureControlParams_PID [▶ 162].

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

eEnable,
eEnableP,
eEnableM,
//
eReversed,
```

```
 //
eEnableFF,
eEnableI,
eEnableD,
eEnableClipping,
eEnableChangeRate
);
END_TYPE
```

5.18.2 PressureControl.Supervisor



This Corefunction organizes the pressure control of the axis and provides the infrastructure interfaces. This is reflected in an interface for a pressure sensor and the switching function for pressure switchover.

Pressure switchover only for certain core functions

The pressure switchover can only be carried out with the following core functions:

- PTP [▶ 173]

Syntax:

```
FUNCTION_BLOCK FB_ProcessSupervisor EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_ProcessSupervisor			
BadSwitchOver	BOOL	Get	Identifies whether a pressure switchover occurred due to safety conditions.
ExternalMode	BOOL	Get	TRUE as soon as an external ProcessHandler has been assigned.
Filter	I_Filter [▶ 268]	Get, Set	Interface for a filter to be applied to the actual pressure value.
MinCushion	LREAL	Get, Set	Minimum distance that the axis must travel before the pressure-based switchover can be triggered.
PositionLimit	LREAL	Get, Set	Safety-limiting position from which a `BadSwitchOver` is identified.
PressureLimit	LREAL	Get, Set	Safety-limiting pressure above which a `BadSwitchOver` is identified.
PressureThreshold	LREAL	Get, Set	Pressure threshold value above which a switchover should take place. **Requirement is** `SwitchOnPressure`
Sensor	I_InputBase [▶ 218]	Get, Set	Interface for an actual pressure value sensor
SwitchOnPressure	BOOL	Get, Set	Activates the switchover based on a pressure threshold value.
SwitchOnTime	BOOL	Get, Set	Activates the switchover based on the time of the previous command.
SwitchOnTravel	BOOL	Get, Set	Activates the switchover based on the value set by a previous command.
SwitchOver	BOOL	Get	If TRUE, the Supervisor has triggered the pressure switchover.

Name	Type	Access	Description
Timeout	LREAL	Get, Set	Safety-limiting time from which a `BadSwitchOver` is identified.
TimeThreshold	LREAL	Get, Set	Time threshold from which a switchover should take place. **Requirement is** `SwitchOnTime`
TravelThreshold	LREAL	Get, Set	Threshold value from which a switchover should take place. **Requirement is** `SwitchOnTravel`
ValueOffset	LREAL	Get, Set	Zero offset shift for the read actual pressure value.
ValueWeighting	LREAL	Get, Set	Weighting factor for the maximum value of the read actual pressure value.
FB_Corefunction			
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.
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.
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
GetExternal()	Returns the associated external supervisor (which implements I_ProcessHandler [▶ 173]). **Can return `NULL` if it has not been assigned previously.**
SetExternal(`iHandler`)	Assigns an external supervisor (implements I_ProcessHandler [▶ 173]). If `0` is transferred, the internal supervisor is reactivated.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_ProcessSupervisor	Standard interface on FB_ProcessSupervisor.

Example:

```
iAxisBase.PressureControl.Supervisor.Sensor := fbInputMainExtruderMeltPressure;
iAxisBase.PressureControl.Supervisor.ValueOffset := 0.0;
iAxisBase.PressureControl.Supervisor.ValueWeighting := 200.0;    // bar
```

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.18.2.1 I_ProcessHandler

Interface for an external supervisor.

Syntax:

```
INTERFACE I_ProcessHandler
```

Properties

Name	Type	Access	Description
Axis	I_AxisBase [▶ 39]	Get, Set	Interface to the connected axis. **Assigned by** `Supervisor.SetExternal()`
BadSwitchOver	BOOL	Get	Reports a safety-related pressure switchover to the axis.
ProcessValue	LREAL	Get	Pressure value with which the pressure controller should operate.
SwitchOver	BOOL	Get	Triggers the switching over to holding pressure.

Methods

Name	Description
GetProcessValues()	Method that implements the read process of the pressure value and can be called at a suitable point by the axis.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.19 Ptp



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
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Ptp			

Name	Type	Access	Description
ActiveSegment	INT	Get	The number of the point that is currently being approached.
Blending [▶ 176]	I_PtpBlendingType	Get	Setting the blending mode between individual PTP points.
ClampingPoint [▶ 176]	I_PtpClampingPoint	Get	Parameterization of a clamping point.
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.
Point [▶ 182]	I_PtpPoint	Get	Parameterization of a PTP point.
Profile [▶ 180]	I_PtpProfileType	Get	Configuration of the motion profile to be used.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Ptp	
DoMove() [▶ 179]	The execution is triggered.
[INTERN] CheckPoint() [▶ 338]	The specified segment is compared with the axis parameters.

Name	Description
[INTERN] GetClampPoint() [▶ 338]	A segment of the clamping table is read back.
[INTERN] GetPoint() [▶ 339]	A segment of the lookup table is read back.
[INTERN] GetUpdatedPoint() [▶ 340]	A segment of the lookup table is read back.
[INTERN] InvalidateClampPoint()	All segments in the clamping table are marked as invalid.
[INTERN] InvalidateTable()	All segments in the lookup table are marked as invalid.
[INTERN] SetClampPoint() [▶ 341]	A segment of the clamping table is defined.
[INTERN] SetPoint() [▶ 342]	A segment of the lookup table is defined.
[INTERN] UpdatePosition() [▶ 342]	The target position of the segment is changed after it is defined.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_Ptp	Standard interface on FB_Ptp.
I_PtpDev	Internal interface on FB_Ptp.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Example:

```
CASE nState OF
0:  // define 2 example points

    iAxisBase.Ptp.Point.Position := 100.0;       // [mm]
    iAxisBase.Ptp.Point.Velocity := 20.0;        // [mm/s]

    // assign first point
    IF FAILED(iAxisBase.Ptp.Point.Set(1)) THEN
        nState := -1;
    END_IF

    iAxisBase.Ptp.Point.Position := 20.0;        // [mm]
    iAxisBase.Ptp.Point.Velocity := 30.0;        // [mm/s]

    // assign second point
    IF FAILED(iAxisBase.Ptp.Point.Set(2)) THEN
        nState := -1;
    END_IF

    // send the axis to travel
    IF FAILED(iAxisBase.Ptp.DoMove(TRUE)) THEN
        nState := -1;
    ELSIF nState >= 0 THEN
        nState := nState + 1;
    END_IF

1:  // observe the movement

    IF iAxisBase.Ptp.FailedState THEN
        nState := -1;
    ELSIF iAxisBase.Ptp.DoneState THEN
        nState := nState + 1;
    END_IF

2,  // done
-1: // error

    iAxisBase.Ptp.DoMove(FALSE);

END_CASE
```

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

5.19.1 Blending

This interface defines the blending of multi-segment movements in FB_Ptp [▶ 173]. The interface works as a selection function that is activated by setting one of the properties. The previously set choice is automatically reset.

Syntax:

```
INTERFACE I_PtpBlendingType
```

Properties

Name	Type	Access	Description
Buffered	BOOL	Get, Set	The PTP segments are processed in a buffered manner. This means that each point is reached with a standstill and no blending takes place.
High	BOOL	Get, Set	Between two segments, the higher velocity is reached at the end point of the first segment.
Low	BOOL	Get, Set	Between two segments, the lower velocity is reached at the end point of the first segment.
Next	BOOL	Get, Set	Between two segments, the velocity of the second segment is reached at the end point of the first segment.
Previous	BOOL	Get, Set	Between two segments, the ramp to the velocity of the second segment is started at the end point of the first segment.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.2 ClampingPoint

This interface is used in FB_Ptp [▶ 173] for the configuration of a clamping point. It can be called up and configured directly.

Syntax:

```
INTERFACE I_PtpClampingPoint EXTENDS I_PtpPointBase
```

Properties

Name	Type	Access	Description
Position	LREAL	Get, Set	Target position of the PTP point.
Duration	LREAL	Get, Set	Duration over which the clamping point is to be executed. **Can be shortened by reaching the position of a subsequent clamping point.**
LimitingRamp	LREAL	Get, Set	Ramp to the limitation value [limitation unit / s].

Name	Type	Access	Description
Acceleration	LREAL	Get, Set	Acceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Limiting	LREAL	Get, Set	Limitation (torque or pressure) with which the point is to be approached. **If 0 is transferred, the idle value of the axis is used.**
NegativeOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in the negative direction.
PositiveOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in a positive direction.
Velocity	LREAL	Get, Set	Velocity with which the point is to be approached.

Methods

Name	Description
Set(`nIdx`)	Assigns the configured properties of the internal lookup table to position `nIdx`. This activates the point for the next movement. The activation status can be queried after assignment via `Fetch().Activated`.
AsStruct() [▶ 177]	Read and write a point as ST_LookUpClamping [▶ 178] structure.
Fetch(`x`)	Loads point `x` from the internal lookup table into this interface so that the values can be retrieved via the properties. The method returns an interface to the `ClampingPoint` and can be called in a chain. Information on the activation status of the point is only available in the return interface: `Ptp.ClampingPoint.Fetch(2).Activated`
DeleteAll()	All segments in the lookup table are marked as invalid.
Reset()	Resets all set properties of this point to the default values.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.2.1 AsStruct()



A clamping point of the lookup table can be retrieved or written as a structure using this method.

Syntax:

```
METHOD AsStruct : ST_LookUpClamping
VAR_INPUT
    nIdx:            INT; // target index in the point table
    refWrite:        REFERENCE TO ST_LookUpClamping; // if = 0, the structure is read instead of wri
tten
END_VAR
```

Return value

Name	Type	Description
AsStruct	ST_LookUpClamping [▶ 178]	If the method is used for reading, the requested point is returned.

Inputs

Name	Type	Meeting
nIdx	INT	The index of the interpolation point in the table.
refWrite	REFERENCE TO ST_LookUpClamping [▶ 178]	Reference to a variable that is to be written to the lookup table. If 0 is passed, the method is executed in read mode.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.2.2 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

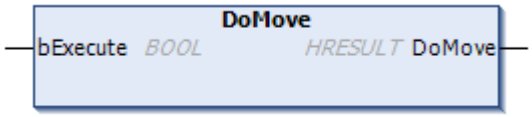
Name	Type	Description
Position	LREAL	Target position of the clamping point.
Velocity	LREAL	Velocity at which the clamping feed is to be executed.
Acceleration	LREAL	Acceleration with which the clamping velocity should be moved. **If 0 is transferred, the default values of the axis are used.**
Limiting	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].

Name	Type	Description
Duration	LREAL	Duration over which the clamping point is to be executed. **Can be shortened by reaching the position of a subsequent clamping point.**
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 v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

5.19.3 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

5.19.4 Profiles

This interface defines the profile that is used for the multi-segment movements in FB_Ptp [▶ 173]. The interface works as a selection function that is activated by setting one of the properties. The previously set choice is automatically reset.

Syntax:

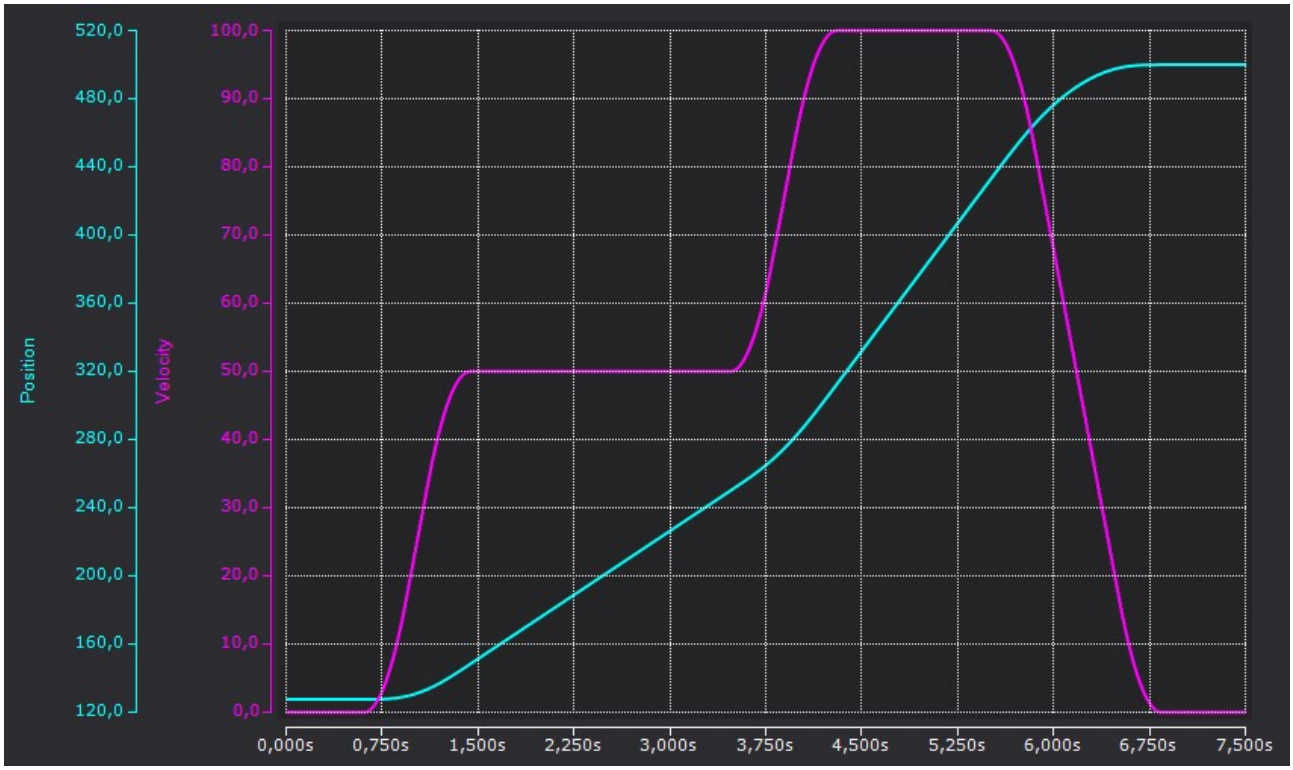
```
INTERFACE I_PtpProfileType
```

Properties

Name	Type	Access	Description
SevenPhase	BOOL	Get, Set	7-phase profile, typically used with NC axes (see also TF50x0 Setpoint Generator).
Sine	BOOL	Get, Set	Sine profile in which a Ptp segment is driven at the velocity of a sinusoidal oscillation.
ThreePhase	BOOL	Get, Set	3-phase profile, typically used for hydraulic axes (see also iTcMc_ProfileCtrlBased).

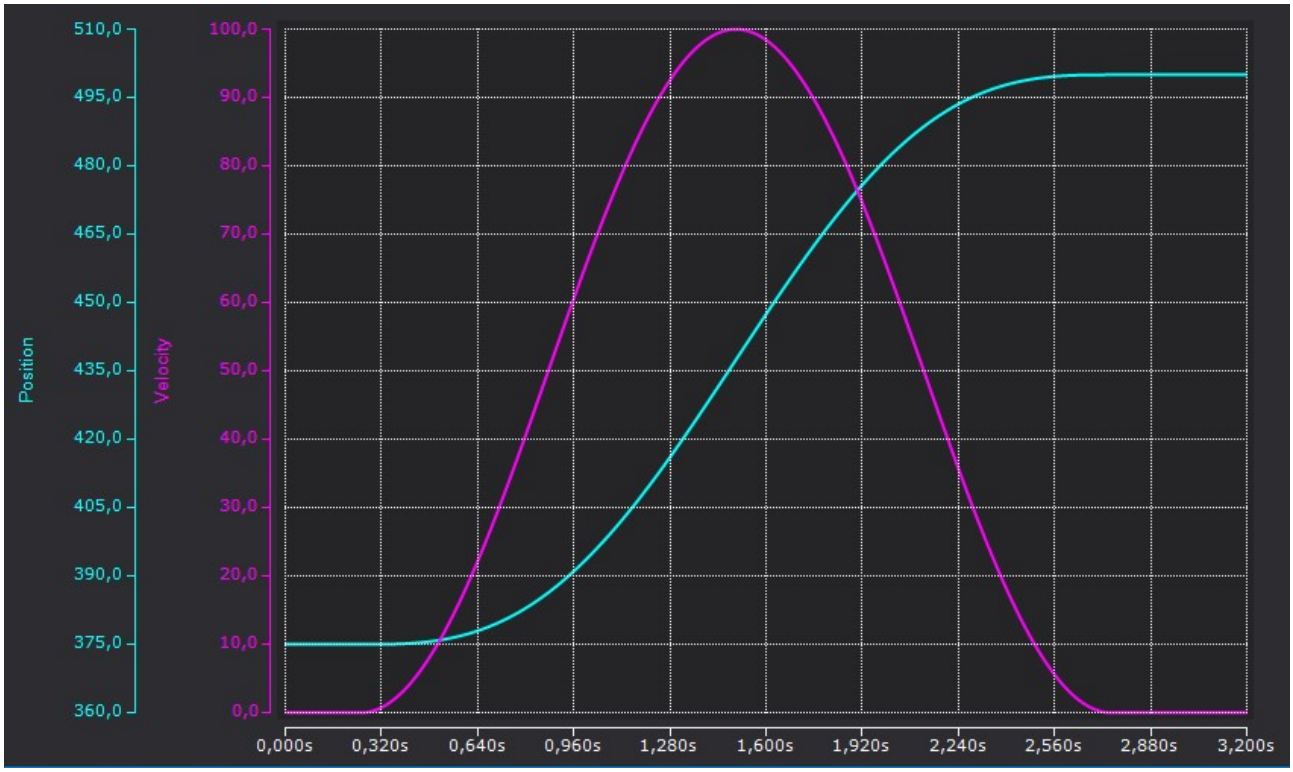
7-phase profile (SevenPhase):

The 7-phase profile corresponds to the standard profile of the TwinCAT NC axes. It consists of a jerk-limited acceleration phase (1), which is followed by a constant acceleration (2) and a jerk-limited reduction of the acceleration (3) to a constant velocity (4). If blending takes place, the following segment begins with its first phase from this section. Otherwise, the deceleration is started with a limited jerk (5), changes to a constant deceleration (6) and is finally reduced again with a limited jerk (7) until it comes to a standstill at the end point. The following illustration shows a sequence with blending of two segments.



Sine profile (Sine):

The sine profile is available as an alternative profile for the phase-based profiles. Instead of multiple phases, the velocity of the sine movement consists of a harmonic sine curve. The velocity and acceleration of the segment are taken into account as limiting factors so that the vertex is not higher than the specified velocity and the maximum gradient is not higher than the specified acceleration.

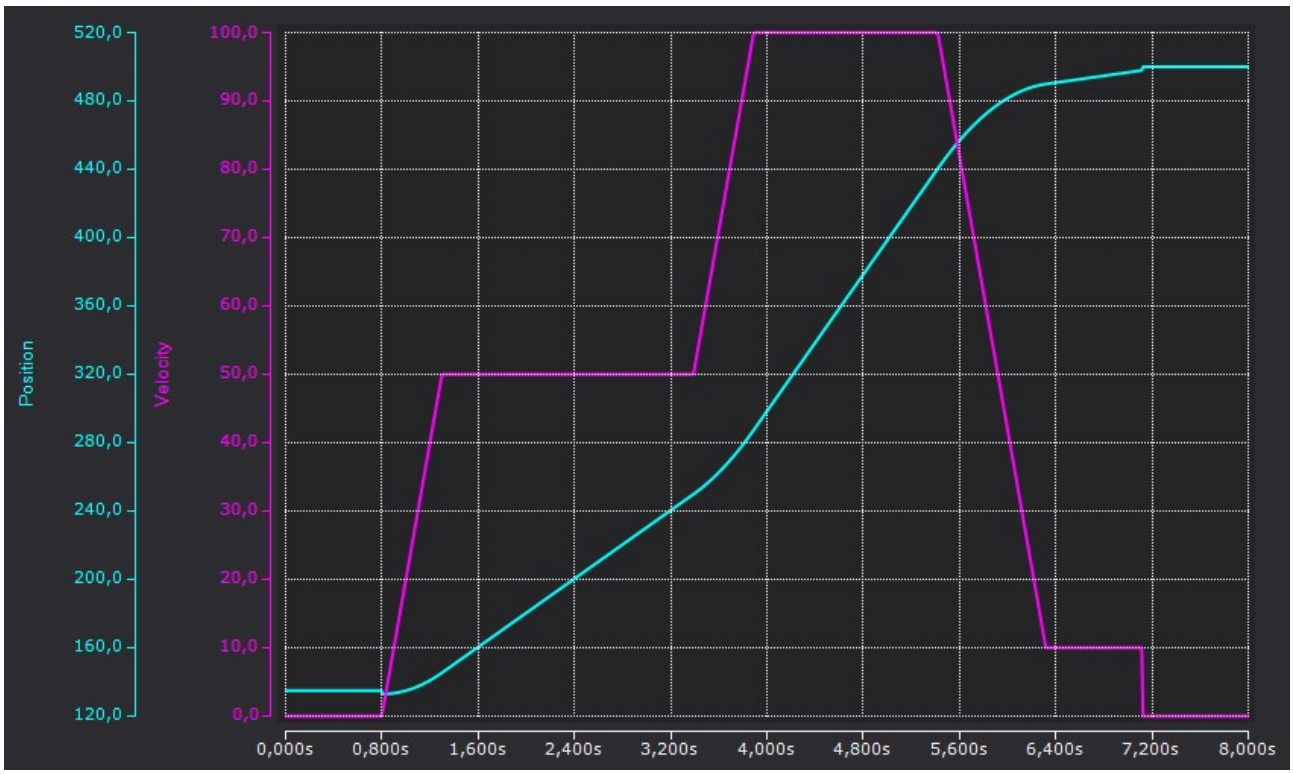


Currently no blending with sine profile possible

The sine profile does not currently support blending [▶ 176]. However, multi-segment movements can be buffered (Buffered).

3-phase profile (ThreePhase):

The 3-phase profile differs from the 7-phase profile only in that the jerk-limited phases are omitted. In the case of a hydraulic axis, this is supplemented by a creep phase (a fourth phase) (see also iTcMc_ProfileCtrlBased).



Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.5 PtpPoint

This interface is used in FB_Ptp [▶ 173] for the configuration of a PTP point. It can be called up and configured directly.

Syntax:

```
INTERFACE I_PtpPoint EXTENDS I_PtpPointBase
```

Properties

Name	Type	Access	Description
Position	LREAL	Get, Set	Target position of the PTP point.
Deceleration	LREAL	Get, Set	Deceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Jerk	LREAL	Get, Set	Jerk with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Acceleration	LREAL	Get, Set	Acceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Limiting	LREAL	Get, Set	Limitation (torque or pressure) with which the point is to be approached.

Name	Type	Access	Description
			If 0 is transferred, the idle value of the axis is used.
NegativeOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in the negative direction.
PositiveOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in a positive direction.
Velocity	LREAL	Get, Set	Velocity with which the point is to be approached.

Methods

Name	Description
Set(`nIdx`)	Assigns the configured properties of the internal lookup table to position `nIdx`. This activates the point for the next movement. The activation status can be queried after assignment via `Fetch().Activated`.
AsStruct()	Read and write a point as ST_LookUpPtpPoint [▶ 184] structure.
Fetch(`x`)	Loads point `x` from the internal lookup table into this interface so that the values can be retrieved via the properties. The method returns an interface to the PtpPoint and can be called in a chain. Information on the activation status of the point is only available in the return interface: `Ptp.Point.Fetch(2).Activated`
DeleteAll()	All segments in the lookup table are marked as invalid.
Reset()	Resets all set properties of this point to the default values.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.5.1 AsStruct()



A point in the lookup table can be retrieved or written as a structure using this method.

Syntax:

```
METHOD AsStruct : ST_LookUpPtpPoint
VAR_INPUT
    nIdx:            INT; // target index in the point table
    refWrite:        REFERENCE TO ST_LookUpPtpPoint; // if = 0, the structure is read instead of wri
tten
END_VAR
```

Return value

Name	Type	Description
AsStruct	ST_LookUpPtpPoint [▶ 184]	If the method is used for reading, the requested point is returned.

Inputs

Name	Type	Meeting
nIdx	INT	The index of the interpolation point in the table.

Name	Type	Meeting
refWrite	REFERENCE TO ST_LookUpPtpPoint [▶ 184]	Reference to a variable that is to be written to the lookup table. If 0 is passed, the method is executed in read mode.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.5.2 PtpPointBase

This interface provides the basis for ClampingPoints and Points that are used to configure the FB_Ptp [▶ 173].

Syntax:

```
INTERFACE I_PtpPointBase
```

Properties

Name	Type	Access	Description
Position	LREAL	Get, Set	Target position of the PTP point.
Acceleration	LREAL	Get, Set	Acceleration with which the point is to be approached. **If 0 is transferred, the default values of the axis are used.**
Limiting	LREAL	Get, Set	Limitation (torque or pressure) with which the point is to be approached. **If 0 is transferred, the idle value of the axis is used.**
NegativeOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in the negative direction.
PositiveOnly	BOOL	Get, Set	The PTP movement should only be executed if the resulting movement will be in a positive direction.
Velocity	LREAL	Get, Set	Velocity with which the point is to be approached.

Methods

Name	Description
Set(`nIdx`)	Assigns the configured properties of the internal lookup table to position `nIdx`. This activates the point for the next movement. The activation status can be queried after assignment via `Fetch().Activated`.
DeleteAll()	All segments in the lookup table are marked as invalid.
Reset()	Resets all set properties of this point to the default values.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.19.5.3 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.

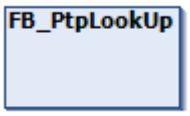
Requirements

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

5.20 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 Axis types [▶ 17].

Methods

Name	Description
GetPoint() [▶ 186]	A point from the lookup table is read back.
Invalidate()	All points in the lookup table are marked as invalid.
ReadMaster() [▶ 187]	A master value is determined that corresponds to the specified slave value.
SetPoint() [▶ 188]	A point of the lookup table is defined.
UpdatePosition() [▶ 188]	In some applications, the target position of the segments must be changed after definition. **This method is used, for example, with transforming axes.**

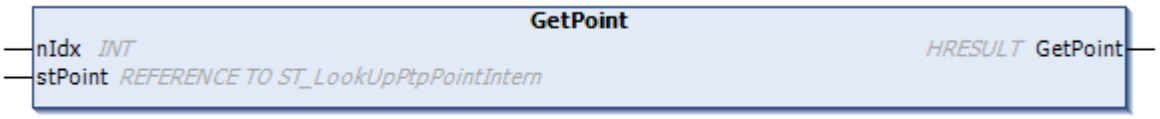
Interfaces

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

5.20.1 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.

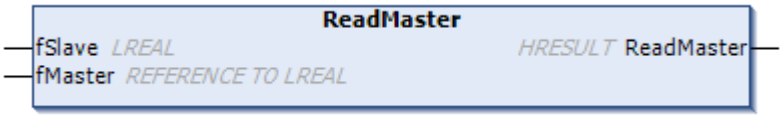
Inputs

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

5.20.2 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.

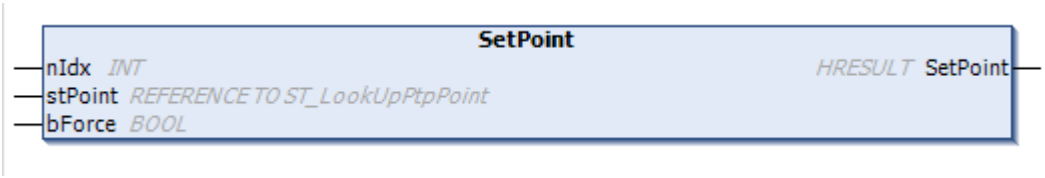
Inputs

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

5.20.3 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 [▶ 184]	A reference to the variable to be updated with the point data.
bForce	BOOL	If the value is TRUE, the segment is updated even when the axis is executing a command. **Care must be taken to ensure 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

5.20.4 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 the value is TRUE, the point is updated even when the axis is executing a command. **Care must be taken to ensure 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

5.21 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
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Reset			

Name	Type	Access	Description
StoppingDelay	LREAL	Get, Set	Maximum delay when calling the reset. Takes effect if the reset command is issued while the axis is still braking due to the error status.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_Reset	
DoReset()	Resets the error state of an axis.

Interfaces

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

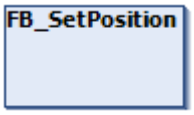
Example:

```
iAxisBase.DoReset(bReset);
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.22 SetPosition



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

Syntax:

```
FUNCTION_BLOCK FB_SetPosition EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_SetPosition			
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.**
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_SetPosition	
DoSetPosition() [▶ 192]	A rising edge at the input bExecute triggers the setting of the position.
SetParameter() [▶ 193]	Here the new position and the operation mode of the Corefunction are defined.

Interfaces

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

Example:

```
IF bSetPosition THEN

    iAxisBase.SetPosition.Mode := TRUE;           // TRUE => Add relative
    iAxisBase.SetPosition.Target := 12.34;        // [mm]
    iAxisBase.SetPosition.DoSetPosition(TRUE);

// revert command
ELSIF iAxisBase.SetPosition.IsCommanded THEN

    iAxisBase.SetPosition.DoSetPosition(FALSE);

END_IF
```

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

5.22.1 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.
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

5.22.2 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

5.23 Stop



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

Syntax:

```
FUNCTION_BLOCK FB_Stop EXTENDS FB_StopBase
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_StopBase			
Deceleration	LREAL	Get, Set	Deceleration of the stop operation. If `0` is passed, the `MaxDecApplication` of the MotionParams [▶ 150] is used.
Jerk	LREAL	Get, Set	Jerk of the stop operation. If `0` is passed, the `MaxJerk` of the MotionParams [▶ 150] is used.
NoCreeping	BOOL	Get, Set	A `TRUE` in this property avoids the creep phase at the end of the stopping process of the hydraulics library.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_StopBase	
DoStop() [▶ 195]	A rising edge triggers the stop.
SetParameter() [▶ 343][1]	This method is used to define the dynamic parameters of the operation. Alternatively: `.Deceleration` and `.Jerk`

[1]Obsolete

Interfaces

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

Example:

```
IF bStopAxis THEN

    iAxisBase.Stop.DoStop(TRUE);

// revert command
ELSIF iAxisBase.Stop.IsCommanded THEN

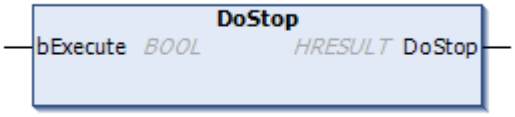
    iAxisBase.Stop.DoStop(FALSE);

END_IF
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.23.1 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

5.24 Teach



This core function is used for referencing analog encoder systems.

Syntax:

```
FUNCTION_BLOCK FB_Teach EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_Teach			
AllowAxisDisable	BOOL	Get, Set	If `TRUE`, the axis is automatically switched off when `DoUpdate()` is called up so that the scaling determined can be applied. **WARNING Only use this option if it is safe to switch off the axis spontaneously.**
IncrNeg	DINT	Get, (Set)	The recorded value when executing `TeachNeg()` in analog increments.
IncrPos	DINT	Get, (Set)	The recorded value when executing `TeachPos()` in analog increments.
MinResolution	UDINT	Get, Set	The minimum resolution (between `IncrNeg` and `IncrPos`) required to achieve a usable result
MinStroke	LREAL	Get, Set	The minimum stroke required for a meaningful result.
Position	LREAL	Get, Set	The position setpoint to be set at the taught (teach) position. If `SetUpperPosition` is set, this position is taken into account at the top end.

Name	Type	Access	Description
SetUpperPosition	BOOL	Get, Set	If `TRUE`, the position setpoint defined at `Position` is set at the upper end of the measured scale.
Stroke	LREAL	Get, Set	The distance (or stroke) that is spanned between `IncrNeg` and `IncrPos` by the axis.
Valid	BOOL	Get	Two valid values were recorded.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[`INTERN`] Cyclic()	This method is called cyclically by the axis.
FB_Teach	
DoTeach(`bExecute`)	Sets the axis to teaching mode. Teaching mode is ended prematurely on a falling edge.
DoUpdate(`bExecute`)	Triggers the final calculation of the encoder parameters.
TeachNeg(`bExecute`)	Triggers the recording of the lower position.
TeachPos (`bExecute`)	Triggers the recording of the upper position.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
FB_Teach	
I_Teach	Standard interface on FB_Teach.
I_TeachDev	Extended interface with access to the inaccessible properties.
I_TeachPosition	Compatibility interface to the obsolete FB_TeachPosition function.
I_TeachUpdate	Compatibility interface to the obsolete FB_TeachUpdate function.

Type	Description
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Example:

```
CASE nState OF

0:  // activate the teach mode
    IF bStartTeaching AND iAxisBase.Teach.IdleState THEN
        iAxisBase.Teach.DoTeach(TRUE);

    ELSIF iAxisBase.Teach.BusyState THEN
        nState := nState + 1;
    ELSIF iAxisBase.Teach.FailedState THEN
        nState := -1;
    END_IF

1:  // move to the first teach position

    // approaching the upper position by a jog command
    // e.g. by pressing a push button
    iAxisBase.Jog.DoJogP(bButtonFwd);

    // on reaching final position, use a second button to teach the value
    iAxisBase.Teach.TeachPos(bTeach);

    IF bTeach THEN
        iAxisBase.Jog.DoJogP(FALSE);
        nState := nState + 1;
    END_IF

2:  // wait for teach button release
    IF NOT bTeach THEN
        nState := nState + 1;
    END_IF

3:  // move in the opposite direction for the second teach position

    // approaching the lower position by a jog command
    // e.g. by pressing a push button
    iAxisBase.Jog.DoJogM(bButtonBwd);

    // on reaching final position, use the second button again to teach the value
    iAxisBase.Teach.TeachNeg(bTeach);

    IF bTeach THEN
        iAxisBase.Jog.DoJogM(FALSE);
        nState := nState + 1;
    END_IF

4:  // configure teach parameters and update the axis by the taught positions

    iAxisBase.Teach.AllowAxisDisable := TRUE;
    iAxisBase.Teach.Position := 0.0;    // mm
    iAxisBase.Teach.Stroke := 200.0;    // mm

    IF iAxisBase.Teach.Valid THEN
        iAxisBase.Teach.DoUpdate(TRUE);
        nState := nState + 1;
    ELSE
        nState := -1;
    END_IF

5:  // wait for the update procedure getting done

    IF iAxisBase.Teach.DoneState THEN
        nState := nState + 1;
    ELSIF iAxisBase.Teach.FailedState THEN
        nState := -1;
    END_IF

6,  // teaching done
-1: // error

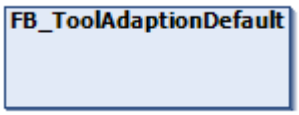
    iAxisBase.Teach.DoTeach(FALSE);
    iAxisBase.Teach.DoUpdate(FALSE);
```

```
END_CASE
```

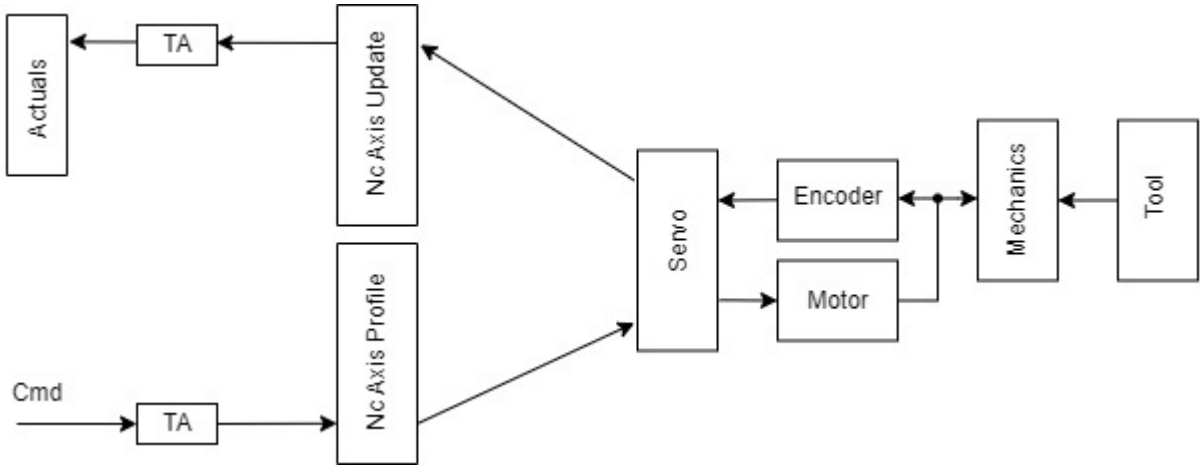
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.25 ToolAdaption



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



Syntax:

```
FUNCTION_BLOCK FB_ToolAdaptionDefault EXTENDS FB_Corefunction
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_ToolAdaptionDefault			
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.
FB_Corefunction			
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.
IsActivated	BOOL	Get	The core function has an accepted command pending.

Name	Type	Access	Description
IsLocalCmd	BOOL	Get	Signals that the axis is assigned with a command of this core function.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_ToolAdaptionDefault	
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_Corefunction	Standard interface on FB_Corefunction.
I_ToolAdaption	Standard interface on FB_ToolAdaptionDefault.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Example:

```
iAxisBase.ToolAdaption.Inverting := FALSE;
iAxisBase.ToolAdaption.Offset := 100.0;
```

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

5.26 Transformation



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

To this end, the transformation works internally with three translation paths:

1. Drive table - Drive position in a load position.
   - The drive table itself consists of the points in the transferred table.
   - Is used as the basis for generating the virtual encoder position of the load side.
2. Load table - load position in a drive position.
   - Is the inverse function of the transferred load table.
   - Is used in the full transformation for synchronization of the drive side.

- Used in the half transformation to convert the command positions
- Is linearized if `LowerLimit` and/or `UpperLimit` are within the load positions

3. Drive table - Reverse translation of an actuator position into a load position.
   - Is used to check the validity of a target position without linearization.

Syntax:

```
FUNCTION_BLOCK FB_NcTransformation EXTENDS FB_TrafoCorefunction
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_NcTransformation			
ApplyLimitsToLoad	BOOL	Get, Set	If TRUE, `LowerLimit` and `UpperLimit` refer to the movement range of the load side. Otherwise, the limits are interpreted in the movement range of the drive side.
IsUpdating	BOOL	Get	TRUE as long as a previously passed table is being processed internally.
LowerLimit	LREAL	Get, Set	Lower limit from which the load table is linearized.
Offset	LREAL	Get, Set	Offset by which the transformation table is shifted on the drive side.
Table	I_CammingLookUp [▶ 94]	Get, Set	Interface for the transformation table to be transferred.
UpperLimit	LREAL	Get, Set	Upper limit from which the load table is linearized.
Valid	BOOL	Get	The transferred table was loaded successfully.
Reversed	LREAL	Get, Set	Reserved for future use.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_NcTransformation	
GetSlot()	Reserves the index of a slot for asynchronous translation function and returns it.

Name	Description
ReleaseSlot(nSlot)	Releases a reserved slot.
TranslateFromDrive() [▶ 203]	Translates a drive position into a load position using the load table.
TranslateFromLoad() [▶ 203]	Reserved for future use.
TranslateToDrive() [▶ 204]	Translates a load position into a drive position using the load table.
TranslateToLoad() [▶ 205]	Translates a drive position into a load position using the drive table.
UpdateToLinLoad() [▶ 206]	Translates a position in a linearized way, depending on the set LowerLimit properties.
[INTERN] TranslateToDrivePrio()	Prioritized execution of the TranslateToDrive() method.
[INTERN] TranslateToLoadPrio()	Prioritized execution of the TranslateToLoad() method.

Interfaces

Type	Description
I_Corefunction	Standard interface on FB_Corefunction.
I_Transformation	Standard interface on FB_NcTransformation.
I_TransformationDev	Extended interface on FB_NcTransformation.
I_CorefunctionFeedback	Standard interface on FB_CorefunctionFeedback.

Example:

```
// Get specific interface I_AxisTrafoBase from general interface I_AxisBase
IF __QUERYINTERFACE(iAxisBase, iAxisTrafoBase) THEN

    CASE nState OF
    0:  // define transformation mechanics (e.g. a crank shaft)
        fbCrankTableGenerator.DriveArm := 50.0;         // [mm]
        fbCrankTableGenerator.LoadArm := 200.0;         // [mm]
        fbCrankTableGenerator.HeightOffset := 0.0;      // [mm]
        fbCrankTableGenerator.DriveLowEnd := 0.0;       // [°]
        fbCrankTableGenerator.DriveHighEnd := 180.0;    // [°]

        // configure a table in which the generated characteristic curve is stored
        fbCammingLookup.Equidistance := TRUE;
        fbCammingLookup.Interpolation := E_LookupMode.eSecondOrder;
        fbCammingLookup.SetPoints(181);         // Resultion => 1 point per degree [°]
        fbCrankTableGenerator.LookUp := fbCammingLookup;

        // Start table generation
        IF fbCrankTableGenerator.DefineTable() THEN
            nState := nState + 1;
        ELSE
            nState := -1;
        END_IF

    1:  // assign the caracteristics curve to the axis
        iAxisTrafoBase.Transformation.Table := fbCammingLookup;

        IF iAxisTrafoBase.Transformation.Valid THEN
            nState := nState + 1;
        ELSIF NOT iAxisTrafoBase.Transformation.IsUpdating THEN
            nState := -1;
        END_IF

    2:  // transformation is ready to use.

        // Activate full transformation to linearise load side behavior
        iAxisTrafoBase.Synchronise := TRUE;

    -1: // error

    END_CASE
ELSE
```

```
    // interface is not set to an instance deriving FB_AxisTrafoBase (e.g. FB_AxisNcTrafoBase)
    bIncompatible := TRUE;
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0) Tc3_PlasticNc (>= v12.12.0.0)

5.26.1 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

5.26.2 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

5.26.3 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.

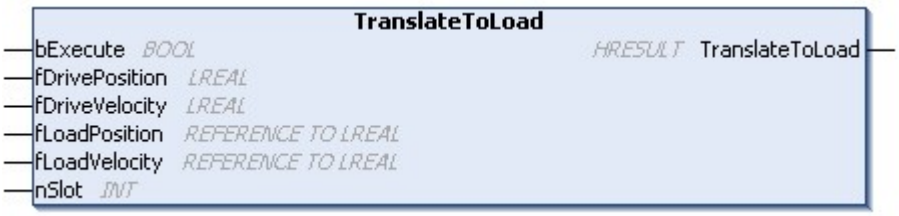
Inputs

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.
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

5.26.4 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

5.26.5 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.

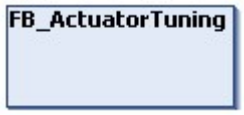
Inputs

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

5.27 Tuning



This core function is used to automatically optimize an actuator.

Syntax:

```
FUNCTION_BLOCK FB_ActuatorTuning EXTENDS FB_CorefunctionFeedback
```

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_ActuatorTuning			

Name	Type	Access	Description
TimeoutFactor	LREAL	Get, Set	Factor (multiplier) for determining the timeout of the movement time.
FB_CorefunctionFeedback			
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.
FB_Corefunction			
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.
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.
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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_ActuatorTuning	
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. **Make sure that the connected end positions can be reached mechanically.**

Interfaces

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

Example:

```
// Get specific interface I_ActuatorBase from general interface I_MotionBase
IF __QUERYINTERFACE(iMotionBase, iActuatorBase) THEN

    CASE nState OF
    0:  // Start the tuning
        iActuatorBase.Tuning.DoTuneTimeout(TRUE);
```

```
    1:  // observe or abort
        IF iActuatorBase.Tuning.FailedState THEN
            nState := -1;
        ELSIF iActuatorBase.Tuning.DoneState THEN
            nState := nState + 1;
        ELSIF bAbortTuning THEN
            iActuatorBase.Tuning.DoAbort(TRUE);
            nState := nState + 2;
        END_IF

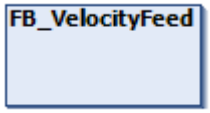
    2,  // done
    3,  // aborted
    -1: // error
        iActuatorBase.Tuning.DoAbort(FALSE);
        iActuatorBase.Tuning.DoTuneTimeout(FALSE);

    END_CASE
ELSE
    // interface is not set to an instance of FB_ActuatorBase
    bIncompatible := TRUE;
END_IF
```

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)

5.28 VelocityFeed



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

Properties

Name	Type	Access	Description
InitState	BOOL	Get, Set	The core function is not completely and successfully initialized.
FB_VelocityFeed			
Acceleration	LREAL	Get, Set	Acceleration with which the velocity change of the axis is generated. If `0` is transferred, the `MaxAccApplication` from the MotionParams [▶ 150] is used.
Deceleration	LREAL	Get, Set	Deceleration with which the velocity change of the axis is generated. If `0` is transferred, the `MaxDecApplication` from the MotionParams [▶ 150] is used.
GearFactor	LREAL	Get, Set	The factor with which to respond to `GuidingValue`.
GuidingValue	LREAL	Get, Set	The commanded velocity.
Limiting	LREAL	Get, Set	Setpoint for the limitation (e.g., torque, pressure) during command execution.
FB_CorefunctionFeedback			
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.

Name	Type	Access	Description
IdleState	BOOL	Get, Set	The Corefunction is ready for operation and commandable.
IsCommanded	BOOL	Get	Signals the pending of a command.
FB_Corefunction			
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.
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.
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.

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
[INTERN] Cyclic()	This method is called cyclically by the axis.
FB_VelocityFeed	
DoFeed() [▶ 210]	A rising edge triggers the core function.

Interfaces

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

Example:

```
IF bCmdVeloFeed THEN

    iAxisBase.VelocityFeed.GearFactor := 1.0;
    iAxisBase.VelocityFeed.GuidingValue := 20.0;    // [mm/s]
    iAxisBase.VelocityFeed.DoFeed(TRUE);

ELSIF iAxisBase.VelocityFeed.IsCommanded THEN

    iAxisBase.VelocityFeed.DoFeed(FALSE);
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

5.28.1 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

6 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.

The following tables list the compatible I/O elements available for each available standard I/O interface.

I_InputBase

Name	Description
InputAnalog16 [▶ 227]	Input module for (up to) 16-bit analog input terminals.
InputAnalog16Status [▶ 229]	Expanded input module for (up to) 16-bit analog input terminals with additional status information.
InputAnalog32 [▶ 231]	Input module for (up to) 32-bit analog input terminals.
InputCoE406 [▶ 235]	Input module for DS406-based encoder systems via EtherCAT.
InputCoE_Polling [▶ 234]	Input element for values from an EtherCAT participant that can be read asynchronously via CoE.
InputDigital [▶ 243]	Input element for a digital (Boolean) input signal.
InputDigitalCam [▶ 244]	Input element for digital feedback from a hydraulic axis.
InputSsi [▶ 246]	Input module for SSI-based encoder systems.

I_OutputBase

Name	Description
OutputAnalog16 [▶ 233]	Output module for (up to) 16-bit analog output terminals.
OutputCoE408 [▶ 237]	Output module for DS408-based hydraulic valves via EtherCAT.
OutputCoE408_P [▶ 240]	An advanced version of the CoE408 input with integrated pressure measurement.
OutputDigital [▶ 245]	Output element for a digital (Boolean) output signal.

I_InvDevice

Name	Description
InverterCoE402_VM [▶ 255]	Inverter with a DS402-based interface in *VelocityMode*.
InverterCoE402_CSVM [▶ 253]	Inverter with a DS402-based interface in *CyclicSynchronousVelocityMode*.
InverterAnalog [▶ 256]	Inverter with an analog interface.

I_LimitingOutput

Name	Description
LimitingOutputCoE [▶ 258]	Limitation output of a DS402-based servo axis.
LimitingOutputSoE [▶ 259]	Limitation output of a Sercos III-based servo axis.

6.1 Base (Abstract)

6.1.1 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() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.1.1.1 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. **This return value is not marked as an error.**
DEVICE_NOERR	0x1B00 000	Execution was successful.

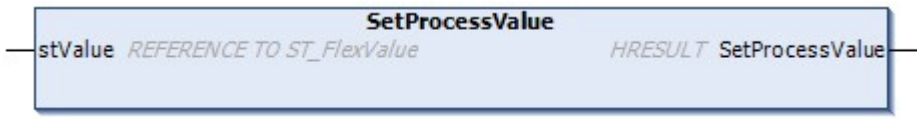
Inputs

Name	Type	Description
stValue	REFERENCE TO ST_FlexValue [▶ 260]	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)

6.1.1.2 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_SYMBOLNOTFOUND	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. **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 [▶ 260]	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)

6.1.2 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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Properties

Name	Type	Access	Description
EcMisses	UINT	Get	Number of EtherCAT diagnostic errors in a row

Methods

Name	Description
Cyclic()	Cycle method
GetProcessIoStatus() [▶ 215]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 216]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 215]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 216]	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)

6.1.2.1 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.

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. **This return value is not marked as an error.**
DEVICE_NOERR	0x1B00 000	Execution was successful.

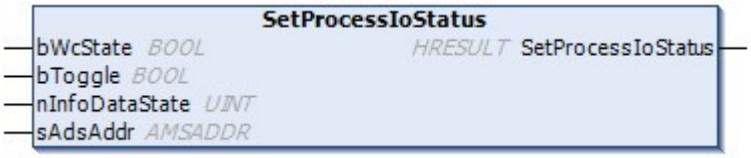
Inputs

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. **Only available with** IoEcAds [▶ 217]**.**

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)

6.1.2.2 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.

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. **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. **Only available with** IoEcAds [▶ 217].

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)

6.1.3 IoEcAds



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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.1.4 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.

Variants already implemented:

Name	Description
InputAnalog16 [▶ 227]	Input module for (up to) 16-bit analog input terminals.
InputAnalog16Status [▶ 229]	Expanded input module for (up to) 16-bit analog input terminals with additional status information.
InputAnalog32 [▶ 231]	Input module for (up to) 32-bit analog input terminals.
InputCoE406 [▶ 235]	Input module for DS406-based encoder systems via EtherCAT.
InputCoE_Polling [▶ 234]	Input element for values from an EtherCAT participant that can be read asynchronously via CoE.
InputDigital [▶ 243]	Input element for a digital (Boolean) input signal.
InputDigitalCam [▶ 244]	Input element for digital feedback from a hydraulic axis.
InputSsi [▶ 246]	Input module for SSI-based encoder systems.

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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.1.4.1 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_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. **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 [▶ 260]	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

6.1.4.2 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
SetChannelStat e	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. **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 [▶ 260]	Variable with the state to be written to the input variable.

Requirements

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

6.1.5 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.

Variants already implemented:

Name	Description
OutputAnalog16 [▶ 233]	Output module for (up to) 16-bit analog output terminals.
OutputCoE408 [▶ 237]	Output module for DS408-based hydraulic valves via EtherCAT.
OutputCoE408_P [▶ 240]	An advanced version of the CoE408 input with integrated pressure measurement.
OutputDigital [▶ 245]	Output element for a digital (Boolean) output signal.

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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 223]	Reads the status word of the output element.
GetControlword() [▶ 223]	Reads the control word of the output element.
GetProcessFeedback() [▶ 224]	Reads the check signal of the output element.
SetChannelState() [▶ 225]	Writes the channel state of the output element.
SetControlword() [▶ 226]	Writes the control word of the output element.
SetProcessFeedback() [▶ 227]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.1.5.1 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.
DEVICE_SRVNOTSUPP	0x1B00 0701	Function is not supported by the I/O element. **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 [▶ 260]	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

6.1.5.2 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. **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

6.1.5.3 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.

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. **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 [▶ 260]	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

6.1.5.4 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. **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 [▶ 260]	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

6.1.5.5 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. **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

6.1.5.6 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. **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 [▶ 260]	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

6.2 Analog

6.2.1 InputAnalog16



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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.2.2 InputAnalog16Status



This class can be used for a 16-bit analog input element with additional status information.

The following are currently explicitly supported:

- EL3255

Syntax:

```
FUNCTION_BLOCK FB_InputAnalog16Status EXTENDS FB_InputAnalog16
```

I/O variables

Name	Type	Direction	Description
WcState	BIT	Input	Working Counter State
Status	U_InputAnalog16_Status [▶ 230]	Input	Detailed status value for a single analog input.
AnalogValue	INT	Input	Numerical signed analog input value.
AdsAddr	AMSADDR	Input	ADS address of the I/O element for asynchronous communication.
Toggle	BIT	Input	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.

Name	Description
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	Writes the terminal-specific input/output value.

Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputAnalog16Status	Standard interface on FB_InputAnalog16Status.
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 v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

6.2.2.1 U_InputAnalog16_Status

Union for decoding an analog input status word.

Syntax:

```
TYPE U_InputAnalog16_Status :
UNION
    {attribute 'tc_no_symbol'}
    w:              WORD;
    {attribute 'tc_no_symbol'}
    el3255:         ST_InputAnalog16_Status_EL3255;
END_UNION
END_TYPE
```

Values

Name	Type	Description
w	WORD	Word as a complete data type.
el3255	ST_InputAnalog16_Status_EL3255 [▶ 230]	Breakdown of the word into bits of an EL3255.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

6.2.2.2 ST_InputAnalog16_Status_EL3255

Structure for the breakdown of an analog input channel status of an EL3255.

Syntax:

```
TYPE ST_InputAnalog16_Status_EL3255 :
STRUCT
    Underrange:       BIT;
    Overrange:        BIT;
    Limit1:           ST_InputAnalog16_Status_Limit;
    Limit2:           ST_InputAnalog16_Status_Limit;
    Error:            BIT;
    Reserve_7:        BIT;
    Reserve_8:        BIT;
    Reserve_9:        BIT;
    Reserve_10:       BIT;
    Reserve_11:       BIT;
    Reserve_12:       BIT;
    SyncError:        BIT;
    TxPDOState:       BIT;
    TxPDOToggle:      BIT;
END_STRUCT
END_TYPE
```

Bits

Name	Offset	Description
Underrange	0	The electrical measuring range was undershot.
Overrange	1	The electrical measuring range has been exceeded.
Limit 1	2	Evaluation of Limit 1 (see EL3255 documentation).
Limit 2	4	Evaluation of Limit 2 (see EL3255 documentation).
Error	6	General measurement error
SyncError	13	Synchronization error with "Distributed Clocks".
TxPDOState	14	Validity of the process data, FALSE = valid.
TxPDOToggle	15	Changing bit for each process data exchange.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

6.2.3 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.

Name	Type	Direction	Description
Toggle	BIT	Input	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.2.4 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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 220]	Reads the status word of the output element.
GetControlword() [▶ 223]	Reads the control word of the output element.
GetProcessFeedback() [▶ 224]	Reads the check signal of the output element.
SetChannelState() [▶ 220]	Writes the channel state of the output element.
SetControlword() [▶ 226]	Writes the control word of the output element.
SetProcessFeedback() [▶ 227]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.

Name	Description
SetProcessValue() [▶ 213]	Writes the terminal-specific input/output value.

Interfaces

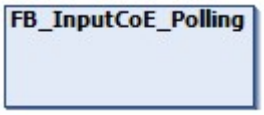
Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputAnalog16	Standard interface on FB_OutputAnalog16.
I_OutputBase	Standard interface on FB_OutputBase.
I_IoEcAds	Standard interface on FB_IoEcAds.
I_IoEc	Standard interface on FB_IoEc.

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)

6.3 CoE

6.3.1 InputCoE_Polling



This class implements a virtual input for an asynchronous CoE value from an EtherCAT device that is read via ADS. This function can be used for cyclic reading of internal terminal information (such as the internal temperature).

Syntax:

```
FUNCTION_BLOCK FB_InputCoE_Polling EXTENDS FB_IoBase
```

Properties

Name	Type	Access	Description
InputScale	DINT	Get, Set	Reserved (compatibility with `I_InputBase`)
FB_InputCoE_Polling			
Config	I_CoEParameter [▶ 278]	Get	Information on the parameter to be read: address, data type, memory type, validation
Device	I_IoEcAds [▶ 217]	Get, Set	The ADS address information is used from the device transferred here.
PollingInterval	LREAL	Get, Set	Time interval at which the asynchronous value is updated.

Methods

Name	Description
Cyclic()	Cycle method
FB_InputCoE_Polling	

Name	Description
GetChannelState() [▶ 220]	Reserved (compatibility with `I_InputBase`)
SetChannelState() [▶ 220]	Reserved (compatibility with `I_InputBase`)
GetProcessIoStatus() [▶ 215]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 216]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	Writes the terminal-specific input/output value.

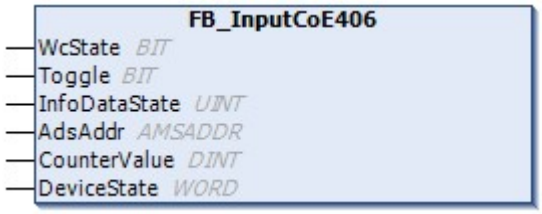
Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputCoE_Polling	Standard interface on FB_InputCoE_Polling.
I_InputBase	Compatible interface for general inputs.
I_IoEcAds	Standard interfaces for ADS-capable I/O devices.
I_IoEc	Standard interfaces for EtherCAT-capable I/O devices.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

6.3.2 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.

Name	Type	Direction	Description
Toggle	BIT	Input	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.3.3 OutputCoE408



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 [▶ 238]	Output	Union for the control word of the valve.
SpoolSetValue	INT	Output	Setpoint for the valve coil.
StatusWord	U_CoE408_StatusWord [▶ 239]	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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
InfoDataState	UINT	Input	Numerical information on the state of the EtherCAT slave (see State).

Methods

Name	Description
Cyclic()	Cycle method
GetChannelState() [▶ 220]	Reads the status word of the output element.
GetControlword() [▶ 223]	Reads the control word of the output element.
GetProcessFeedback() [▶ 224]	Reads the check signal of the output element.
SetChannelState() [▶ 220]	Writes the channel state of the output element.
SetControlword() [▶ 226]	Writes the control word of the output element.
SetProcessFeedback() [▶ 227]	Writes the check signal of the output element.

Name	Description
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.3.3.1 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 [▶ 238]	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)

6.3.3.2 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

6.3.3.3 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 [▶ 239]	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)

6.3.3.4 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

6.3.4 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.
ControlWord	U_CoE408_ControlWord [▶ 238]	Output	Union for the control word of the valve.
SpoolSetValue	INT	Output	Setpoint for the valve coil.
StatusWord	U_CoE408_StatusWord [▶ 239]	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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the output element.
GetControlword() [▶ 223]	Reads the control word of the output element.
GetProcessFeedback() [▶ 224]	Reads the check signal of the output element.
SetChannelState() [▶ 220]	Writes the channel state of the output element.
SetControlword() [▶ 226]	Writes the control word of the output element.
SetProcessFeedback() [▶ 227]	Writes the check signal of the output element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.4 Digital

6.4.1 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() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.4.2 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() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.4.3 InputDigitalCam



This class is used for the digital feedback of an FB_AxisHydraulicBase [▶ 30].

Syntax:

```
FUNCTION_BLOCK FB_InputDigitalCam EXTENDS FB_IoBase
```

Properties

Name	Type	Access	Description
BreakM	I_InputDigital [▶ 243]	Get, Set	Digital input element for a brake sensor in the negative direction on the axis track.
BreakP	I_InputDigital [▶ 243]	Get, Set	Digital input element for a brake sensor in the positive direction on the axis track.
TargetM	I_InputDigital [▶ 243]	Get, Set	Digital input element for the end position in the negative direction on the axis track.
TargetP	I_InputDigital [▶ 243]	Get, Set	Digital input element for the end position in the positive direction on the axis track.

Methods

Name	Description
Cyclic()	Cycle method
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	Writes the terminal-specific input/output value.

Coding of the `stValue` for FB_InputDigitalCam.`Get|SetProcessValue()`

The value for the `ST_FlexValue` of the InputDigitalCam input element is coded bit by bit as `BYTE`:

Bit 0: TargetP
Bit 1: BreakP
Bit 2: BreakM
Bit 3: TargetP
Bit 4: TargetP Connected
Bit 5: BreakP Connected
Bit 6: BreakM Connected
Bit 7 TargetM Connected

Interfaces

Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_InputDigitalCam	Standard interface on FB_InputDigitalCam.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

6.4.4 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() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	Writes the terminal-specific input/output value.

Interfaces

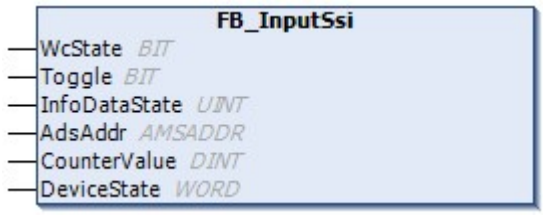
Type	Description
I_IoBase	Standard interface on FB_IoBase.
I_OutputBase	Compatible standard interface for outputs. **This class implements hidden placeholders for compatibility.**
I_OutputDigital	Standard interface on FB_OutputDigital.
I_IoEc	Standard interface on FB_IoEc.

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)

6.5 Serial

6.5.1 InputSsi



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	Cyclic switching bit that is synchronized with the update rate of the mapping. As a rule, the InputToggle of the WcState structure is mapped for this purpose.
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() [▶ 220]	Reads the status word of the input element.
SetChannelState() [▶ 220]	Writes the status word of the input element.
GetProcessIoStatus() [▶ 212]	Reads the EtherCAT status of the input/output element.
SetProcessIoStatus() [▶ 213]	Writes the EtherCAT status of the input/output element.
GetProcessValue() [▶ 212]	Reads the terminal-specific input/output value.
SetProcessValue() [▶ 213]	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)

6.6 Device (inverter)

6.6.1 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.

Variants already implemented:

Name	Description
InverterCoE402_VM [▶ 255]	Inverter with a DS402-based interface in *VelocityMode*.
InverterCoE402_CSVM [▶ 253]	Inverter with a DS402-based interface in *CyclicSynchronousVelocityMode*.
InverterAnalog [▶ 256]	Inverter with an analog interface.

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.

Name	Type	Access	Description
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_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)

6.6.2 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 [▶ 217]	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)

6.6.3 InvDeviceCoE402



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 [▶ 217]	Input	EtherCAT status of the inverter.
ControlWord	U_CoE402_ControlWord [▶ 251]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 252]	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)

6.6.3.1 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 [▶ 251]	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)

6.6.3.2 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)

6.6.3.3 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 [▶ 252]	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)

6.6.3.4 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)

6.6.4 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 [▶ 217]	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 [▶ 251]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 252]	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)

6.6.5 InverterCoE402_VM



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 [▶ 217]	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 [▶ 251]	Output	Control word to the inverter.
StatusWord	U_CoE402_StatusWord [▶ 252]	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)

6.6.6 InverterAnalog



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 [▶ 233]	Output	Velocity output
QPowerEnable	FB_OutputDigital [▶ 245]	Output	Power enable
QPositiveEnable	FB_OutputDigital [▶ 245]	Output	Positive directional enable
QNegativeEnable	FB_OutputDigital [▶ 245]	Output	Negative directional enable
QReset	FB_OutputDigital [▶ 245]	Output	Reset the inverter error state.
IActualVelocity	FB_InputAnalog16 [▶ 227]	Input	Message of the current velocity.

Name	Type	Direction	Description
IActualLoad	FB_InputAnalog16 [▶ 227]	Input	Message of the current load.
IEnableFeedback	FB_InputDigital [▶ 243]	Input	Message of power enable.
IFault	FB_InputDigital [▶ 243]	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)

6.7 LimitingOutput

This interface defines the output values of the limiting function. By implementing this in an FB, device-specific scaling can be implemented. An FB implemented in this way can be connected to the axis via the Limiting [▶ 145].`Output`. The following automatically available limitation functions are already pre-implemented, for example, in the `FB_AxisNcBase`:

- FB_LimitingOutputCoE [▶ 258] – DS402 (AX8000)-based TorqueLimiting
- FB_LimitingOutputSoE [▶ 259] – AX5000-Based torque limiting

Syntax:

```
INTERFACE I_LimitingOutput EXTENDS __SYSTEM.IQueryInterface
```

Properties

Name	Type	Access	Description
LimitingN	LREAL	Get, Set	Normalized value [0,1] for the lower bound.
LimitingP	LREAL	Get, Set	Normalized value [0,1] for the upper bound.

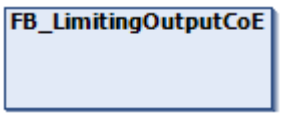
Methods

Name	Description
Init(`iMBase`)	Called when an axis is assigned to the limiting [▶ 145]-Corefunction. Allows you to retrieve axis information, for example, to verify the compatibility of the limit function and the axis type.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0) Tc3_PlasticNc (>= v12.14.0.0)

6.7.1 LimitingOutputCoE



This class is part of the `FB_AxisNcBase` and transmits the current limit to a DS402-compatible CoE field device, such as an AX8000.

Syntax:

```
FUNCTION_BLOCK FB_LimitingOutputCoE EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
LimitingN	LREAL	Get, Set	Normalized value [0,1] for the lower bound.
LimitingP	LREAL	Get, Set	Normalized value [0,1] for the upper bound.
OutputScale	INT	Get, Set	The scaling of the value sent to the CoE device. Corresponds to the value to be sent for Limiting [▶ 145].`EffectiveSetpoint = .ReferenceValue`. By default, the value is set to 1000 [‰].

Methods

Name	Description
Init(`iMBase`)	Called when an axis is assigned to the limiting [▶ 145]-Corefunction. Allows you to retrieve axis information, for example, to verify the compatibility of the limit function and the axis type.
[`INTERN`] Cyclic()	Cyclic method for processing communication.

Interfaces

Type	Description
I_LimitingOutput	Standard interface to a limit output.
I_LimitingOutputCoE	Standard interface on FB_LimitingOutputCoE

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0) Tc3_PlasticNc (>= v12.14.0.0)

6.7.2 LimitingOutputSoE



This class is part of the `FB_AxisNcBase` and transmits the current limit to an SoE-capable field device, such as an AX5000.

Syntax:

```
FUNCTION_BLOCK FB_LimitingOutputCoE EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
LimitingN	LREAL	Get, Set	Normalized value [0,1] for the lower bound.
LimitingP	LREAL	Get, Set	Normalized value [0,1] for the upper bound.

Methods

Name	Description
Init(iMBase)	Called when an axis is assigned to the limiting [▶ 145]-Corefunction. Allows you to retrieve axis information, for example, to verify the compatibility of the limit function and the axis type.
[INTERN] Cyclic()	Cyclic method for processing communication.

Interfaces

Type	Description
I_LimitingOutput	Standard interface to a limit output.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0) Tc3_PlasticNc (>= v12.14.0.0)

6.8 ST_FlexValue

Structure for transferring information without specifying the data type.

Syntax:

```
TYPE ST_FlexValue :
STRUCT
    VarType:    E_FlexValue;    // 4 byte
    Value:      U_FlexValue;    // 8 byte
END_STRUCT
END_TYPE
```

Values

Name	Type	Description
VarType	E_FlexValue [▶ 260]	Enumeration for identifying the value to be read.
Value	U_FlexValue [▶ 261]	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

6.8.1 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)

6.8.2 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

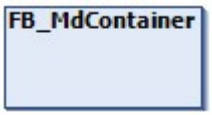
Name	Type
b	BOOL
si	SINT
usi	USINT
bt	BYTE
i	INT
ui	UINT
w	WORD
di	DINT
udi	UDINT
dw	DWORD
re	REAL
li	LINT
uli	ULINT
lw	LWORD
lr	LREAL

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)

7 MachineData

7.1 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.
FileType	E_MdFileType [▶ 264]	Get, Set	File encoding type
IgnoreComponentType	BOOL	Get, Set	The component type is not taken into account for calling the Callback methods.
IgnoreCrc	BOOL	Get, Set	A CRC checksum that does not match will not result in an abortion of the loading process **Always** `TRUE` **for FileType** `eJson`
JsonOptions	I_MdJsonOptions [▶ 264]	Get	Formatting options for saving in Json format
Load	BOOL	Get, Set	Command to read the file.
Name	STRING	Get, Set	Name of the machine data container.
NetId	T_AmsNetID	Get, Set	AMS (ADS) network ID of the target system. **Not available for the FileType** `eJson`
Path	T_MaxString	Get, Set	Path under which the machine data is to be saved.
Save	BOOL	Get, Set	Command for writing the file.
Timeout	LREAL	Get, Set	Time-out period of the system response of this function.
FilePath[1]	STRING	Get, Set	Path under which the machine data is to be saved.

[1]Obsolete

Methods

Name	Description
Abort()	Aborts the current loading or saving process.
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.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

7.1.1 E_MdFileType

Describes the supported file formatting of an FB_MdContainer [▶ 263].

```
TYPE E_MdFileType :
(
eBinary := 0,
eJson
);
END_TYPE
```

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

7.1.2 I_MdJsonOptions

Interface to the options available for Json formatting.

Syntax:

```
INTERFACE I_MdJsonOptions
```

Properties

Name	Type	Access	Description
GroupComponents	BOOL	Get, Set	Groups the parameters within the file into sub-nodes per component. **When using `IngoreComponentType`, only the index is grouped.**
ParamIndexAsHex	BOOL	Get, Set	All parameter index numbers are saved as hexadecimal coded `STRING`.
WordAsHex	BOOL	Get, Set	All parameters of the word data type (`WORD`, `DWORD`, `LWORD`, `BYTE`) are saved as hexadecimal coded `STRING`.
WriteDescriptors	BOOL	Get, Set	Component names and parameter descriptions are also written to the file.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

7.2 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
ComponentName	T_MaxString	Get, Set	Component name. **Is written to the file if the Json option `WriteDescriptors` is set.**
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.
MdParamDescriptor	T_MaxString	Get, Set	Name of the next parameter to be written. **Is written to the file if the Json option `WriteDescriptors` is set.**

Methods

Name	Description
ClearChain()	Resets the interface to the next component and calls its `ClearChain()` method.

Event-driven methods (callback methods)

Name	Description
MdNextParameter() [▶ 266]	Gets parameters from the application.
MdSetParameter() [▶ 266]	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

7.2.1 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 [▶ 267]	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

7.2.2 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 [▶ 267]	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

7.3 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;     // 12 byte
END_STRUCT                             //-------------
END_TYPE                               // 18 byte
```

Values

Name	Type	Description
nParamIndex	UDINT	Individual index of the written parameter.
nComponentType	USINT	Components Type [▶ 265] of the written parameter.
nComponentIndex	USINT	Components Index [▶ 265] of the written parameter.
stFlex	ST_FlexValue [▶ 260]	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

8 Utilities

8.1 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.

8.1.1 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 [▶ 268]	Cyclic transfer of a new input value.

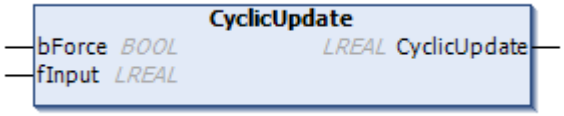
Interfaces

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

8.1.1.1 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. **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

8.1.2 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

Name	Description
CyclicUpdate [▶ 268]	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

8.1.3 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.

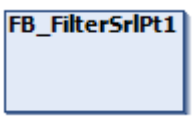
Methods

Name	Description
Cyclic	Cycle method
CyclicUpdate [▶ 268]	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

8.1.4 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 [▶ 268]	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

8.2 HRESULT

Most methods of the TwinCAT 3 Plastic Technology Functions return information about the success of the execution by means of a `HRESULT` value. This is formed from individual 16 bits of information and a 16-bit error code. This results in the 32-bit data type. The basic structure is as follows:

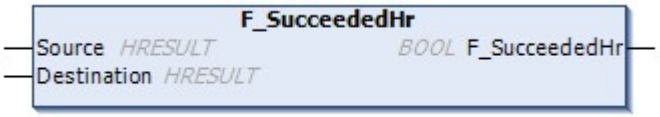
Bits	31	30-16	15-0
Meaning	0 = Success, 1 = Error	Characteristic identifier of the Plastic Technology Functions: 0x1B00	Error code: Typically matching the enumeration E_AdsErr

There are several ways to evaluate an `HRESULT` error code:

- Basically, the Tc3_Module library offers simple functions for checking.
  - SUCCEEDED() --> Success
  - FAILED() --> Failure
- As the `HRESULT` data type is signed, bit 31 can also be checked by a greater than/less than comparison.
  - >= 0 --> Success
  - < 0 --> Failure
- The F_SucceededHr(...) [▶ 272] function combines two error code evaluation processes.
  - The function itself returns a boolean `TRUE` if the `HRESULT` value indicates a success.
  - If the `HRESULT` indicates a failure, the information is copied to the `Destination` variable passed.

See also PLC Programming Conventions - Error Codes

8.2.1 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

8.2.2 F_HresultCreate



This method is used to create a HRESULT value. Whether the value is interpreted as a success depends on the input parameter bSuccess.

Syntax:

```
FUNCTION F_HresultSuccess : HRESULT
VAR_INPUT
    bSuccess:           BOOL;
    nFeedback:          UDINT;
END_VAR
```

Return value

Name	Type	Description
F_HresultCreate	HRESULT	Created `HRESULT` value.

Inputs

Name	Type	Description
nFeedback	UDINT	Error code to be integrated into HRESULT. **Only accepted if** `bSuccess = FALSE`**.**
bSuccess	BOOL	Indicator for the success.

Requirements

Development Environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.66	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.10.0.0)

8.2.3 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.

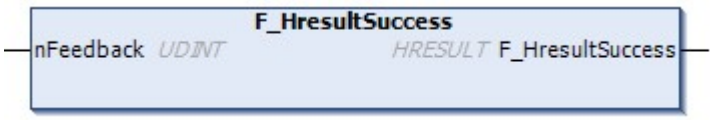
Inputs

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

8.2.4 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

8.3 Messages

8.3.1 FB_Message



This class is able to send messages via the EventLogger.

Syntax:

```
FUNCTION_BLOCK FB_Message EXTENDS FB_TcMessage
```

Methods

Name	Description
SetTraceLevel() [▶ 275]	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

8.3.1.1 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

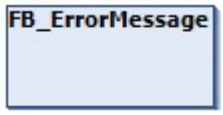
Name	Type	Description
bOwnTraceLevel	BOOL	Activates the fixed severity level.
eTraceLevel	TcEventSeverity	Severity level

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

8.3.2 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

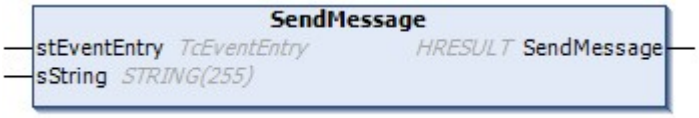
Name	Description
SetTraceLevel() [▶ 275]	Fixes the severity of the events thrown by this instance.

Name	Description
SendMessage() [▶ 276]	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

8.3.2.1 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.

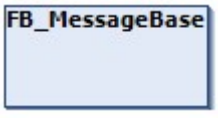
Inputs

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

8.3.3 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

8.4 Services

8.4.1 FB_CoEService



This class provides a read and write service for CoE-addressable EtherCAT devices. This can be used to read and/or write internal information from a field device, for example.

Syntax:

```
FUNCTION_BLOCK FB_CoEService EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
AdsAddr	AMSADDR	Get, Set	ADS address of the target device. **Is set automatically when an existing field device is linked via** EcDevice**.**
Busy	BOOL	Get	The service is currently performing a read or write operation.
Config	I_CoEParameter [▶ 278]	Get	Information on the parameter to be read: address, data type, memory type, validation
Done	BOOL	Get	The read or write process has been successfully completed.
EcDevice	I_IoEcAds [▶ 217]	Get, Set	Transfer of an existing I/O element with mapping of the ADS address
Error	BOOL	Get	An error has occurred during the read or write process.
ErrorId	UDINT	Get	Error code of the last error that occurred.

Methods

Name	Description
Cyclic()	Cycle method
Read(bExecute)	Commands a read process.
Write(bExecute)	Commands a write process.

Interfaces

Type	Description
I_CoEService	Standard interface on FB_CoEService.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

8.4.2 FB_CoEParameter



This class bundles the information of a CoE value in a field device.

Syntax:

```
FUNCTION_BLOCK FB_CoEParameter
```

Properties

Name	Type	Access	Description
Index	WORD	Get, Set	Identifies the main index of the CoE value.
SubIndex	BYTE	Get, Set	Addresses a sub-information in a structured CoE value.
Valid	BOOL	Get, Set	The value contained in Value was read by an FB_CoEService [▶ 277], for example, and is valid.
Value	REFERENCE TO U_FlexValue [▶ 261]	Get, Set	Union, which saves the read value (regardless of the set data type).
VarType	E_FlexValue [▶ 260]	Get, Set	Defines the data type of the parameter to be read. **The correct data type must be configured for the execution of a FB_CoEService service.**

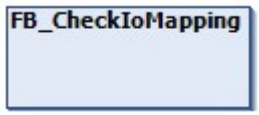
Interfaces

Type	Description
I_CoEParameter	Standard interface to FB_CoEParameter.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

8.4.3 FB_CheckIoMapping



This class checks for an I/O element that supports I_InputBase [▶ 218] or I_OutputBase [▶ 221] whether the `ProcessValue` (or, in the case of outputs, the `ProcessFeedback`) is cyclically synchronized via a TwinCAT mapping (e.g. with I/O).

Mapping analysis generates artificial noise

For analysis purposes, the function block describes the values to be interpreted with an artificial noise. If other parts of the program also read out the signal, this noise can lead to misinterpretations!

- The effects of synchronized mappings can be minimized by placing the I/O check as the last access to the I/O element within the PLC cycle.

Syntax:

```
FUNCTION_BLOCK FB_CheckIoMapping EXTENDS FB_MessageBase
```

Properties

Name	Type	Access	Description
Busy	BOOL	Get	A check is currently being carried out.
Done	BOOL	Get	The mappings have been checked.
Error	BOOL	Get	An error has occurred while checking the mappings.
ErrorId	UDINT	Get	Error code of the last error that occurred.
IoDevice	I_IoBase [▶ 212]	Get, Set	I/O object to be checked.
IsMapped	BOOL	Get	TRUE if an active mapping was detected during the check.

Methods

Name	Description
Cyclic()	Cycle method
DoCheck(`bExecute`)	Starts the check of the assigned I/O object.

Interfaces

Type	Description
I_CheckIoMapping	Standard interfaces on FB_CheckIoMapping.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

8.5 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

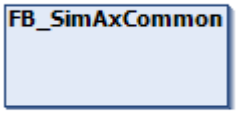
The most efficient way to connect a process simulation with a control implementation such as an application based on TC3 Plastic Functions is to replace the real I/O devices with compatible container objects.

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

8.5.1 General simulation components

8.5.1.1 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	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`.

Name	Type	Access	Description
			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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`. **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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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.

Name	Type	Access	Description
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	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

8.5.1.1.1 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
```

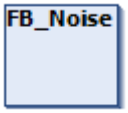
Return value

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

8.5.1.2 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 [▶ 283]	Cyclic call to generate a new value.
SetLineNoiseLevel [▶ 284]	Determination of the level of an influence by a supply network.
SetSparkNoiseLevel [▶ 284]	Determination of the level of an influence by static discharges.
SetWhiteNoiseLevel [▶ 284]	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

8.5.1.2.1 Cyclic

Method to be called cyclically to generate a new value.

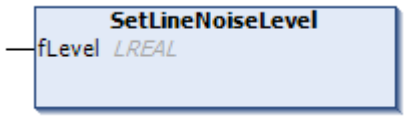
Syntax:

```
METHOD Cyclic : LREAL
```

Outputs

Name	Type	Description
Cyclic	LREAL	The new value of the noise signal

8.5.1.2.2 SetLineNoiseLevel



Determination of the level of an influence by a supply network.

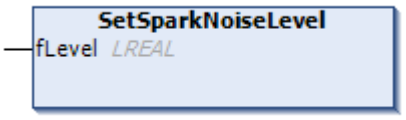
Syntax:

```
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.**

8.5.1.2.3 SetSparkNoiseLevel



Determination of the level of an influence by static discharges.

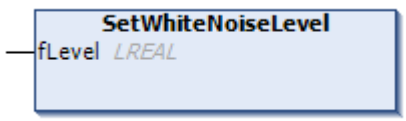
Syntax:

```
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.**

8.5.1.2.4 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.**

8.5.2 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 [▶ 286]	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 [▶ 292]	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 [▶ 286]	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.

Designation	Type	Description
fbNcAxisSimLimiter	FB_SimTorqueLimitingCoE402_Mapped [▶ 292]	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.

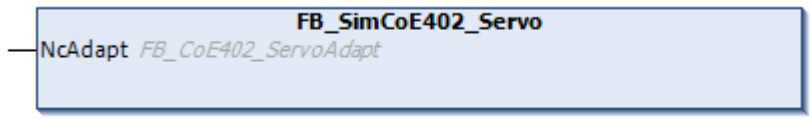
Functionality of the CoE402 simulation

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.

8.5.2.1 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	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimCoE402_Servo			
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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`. **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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.

Name	Type	Access	Description
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`. **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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. 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 [▶ 292] or FB_SimTorqueLimitingCoE402_Mapped [▶ 292] must be used to perform the data exchange between TwinCAT 3 Plastic Functions and the simulation.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimCoE402_Servo	
Cyclic()	This method must be called cyclically.
Gear() [▶ 290]	Starts a chaining of gear factors.
GearBox() [▶ 290]	Rotary transmission (gearbox).
GearRackNPinion() [▶ 291]	Linear transmission by rack and pinion.
GearScrew() [▶ 291]	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)

8.5.2.1.1 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)

8.5.2.1.2 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)

8.5.2.1.3 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)

8.5.2.1.4 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
```

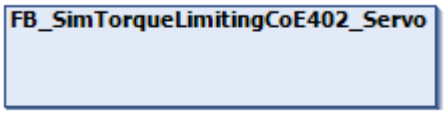
Inputs

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)

8.5.2.2 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() [▶ 293]	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.

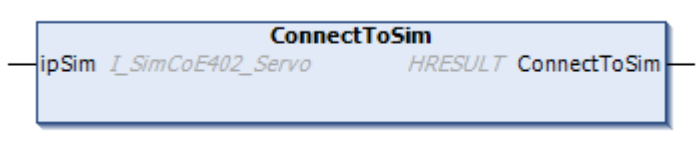
Interfaces

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

8.5.2.2.1 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

8.5.3 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:

Name	Type	Description
{attribute 'TcContextName':='FastTask'}		Instructs FB_SimCoE402_Inverter 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 [▶ 297]	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
    fbInvAxisSim.CycleTime := 0.002;
    fbInvAxisSim.MaxTurnRate := 1380.0;
    fbInvAxisSim.MinTurnrate := 45.0;
    fbInvAxisSim.RampTime := 2.0;
    fbInvAxisSim.OutputFactor := 1.0;
    fbInvAxisSim.ReferenceLoad := 100.0;
END_IF
```

The parameters must correspond to the behavior of the simulated device, not to the intended use.

8.5.3.1 FB_SimCoE402_Inverter



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 ABSTRACT FB_SimCoE402_Inverter EXTENDS FB_SimAxCommon
VAR_INPUT
    EtC_Device AT %Q* : FB_EtC_SimDevice;
END_VAR
```

I/O variables

Name	Type	Direction	Description
EtC_Device	FB_EtC_SimDevice	Input/Output	Diagnostic interface for an EtherCAT-capable device.

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimCoE402_Inverter			
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. Alias for `.MovingMass`
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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`. **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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`. **The simulated position of the actuator can deviate from the reported axis position.**

Name	Type	Access	Description
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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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`.
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.

Name	Type	Access	Description
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimCoE402_Inverter	
Cyclic()	Cycle method
TriggerError()[1]	A call to this method places the simulated drive into the error state. Alternatively: `GoFaultState()`

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

8.5.3.2 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	Input/Output	Diagnostic interface for an EtherCAT-capable device.

Name	Type	Direction	Description
I/O	FB_CoE402_InverterAdapt	Input/Output	The mapping interface for the inverter axis.

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimCoE402_Inverter			
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. Alias for `.MovingMass`
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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`. **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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`.

Name	Type	Access	Description
			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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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`.
LowSideBlock	LREAL	Get, Set	The simulated position of the actuator is limited to a value above or equal to this parameter.

Name	Type	Access	Description
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimCoE402_Inverter	
Cyclic()	Cycle method

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

8.5.3.3 FB_SimCoE402_Inverter_CSVM



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	Input/Output	Diagnostic interface for an EtherCAT-capable device.
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
ActualPosition	LREAL	Get, Set	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimCoE402_Inverter			
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. Alias for `.MovingMass`
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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`. **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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`. **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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimCoE402_Inverter	
Cyclic()	Cycle method

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0)

8.5.3.4 FB_SimAnalog_Inverter



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 offers mapping elements for an analog interface of basic frequency inverters (FU) 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_SimAnalog_Inverter EXTENDS FB_SimAxCommon
VAR_INPUT
    Velocity               : FB_SimAnalogOutputElTerminal4;   // mandatory
    Actuals                : FB_SimAnalogInputElTerminal4;
    PowerEnable AT %I*     : BOOL;   // mandatory
    PositiveEnable AT %I*  : BOOL;   // optional: will be TRUE if no mapping
    NegativeEnable AT %I*  : BOOL;   // optional: will be TRUE if no mapping
    Reset AT %I*           : BOOL;   // mandatory
    EnableFeedback AT %Q*  : BOOL;
    Fault AT %Q*           : BOOL;
END_VAR
```

I/O variables

Name	Type	Direction	Description
Fault	BOOL	Output	Output of a simulated error.
Velocity	FB_SimAnalogOutputElTerminal4 [▶ 308]	Input	Velocity input for the setpoint from the control.
Actuals	FB_SimAnalogInputElTerminal4 [▶ 307]	Output	Velocity feedback (actual value) to the control. The load value is returned on channel 2.
PowerEnable	BOOL	Input	Power enable for the inverter.
PositiveEnable	BOOL	Input	Positive power enable.
NegativeEnable	BOOL	Input	Negative power enable.
Reset	BOOL	Input	Resetting a simulated error.
EnableFeedback	BOOL	Output	Feedback of successful power enable.

Properties

Name	Type	Access	Description
ActualPosition	LREAL	Get, Set	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_Analog_Inverter			
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. Alias for `.MovingMass`

Name	Type	Access	Description
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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case `AbsSwitchHighSelect=TRUE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is >= `AbsSwitchHighEnd`. **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` selects `AbsSwitchHighEnd`, while a `FALSE` makes `AbsSwitchLowEnd` the active `AbsolutSwitch`.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case `AbsSwitchHighSelect=FALSE`: `AbsolutSwitch` is `TRUE` if the simulated actuator position is <= `AbsSwitchLowEnd`. **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.
DenyEnable	BOOL	Get, Set	A `TRUE` blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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:

Name	Type	Access	Description
			• 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`.
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimAnalog_Inverter	
Cyclic()	Cycle method

Interfaces

Type	Description
I_SimAxCommon	Standard interface for `FB_SimAxCommon`.
I_SimAnalog_Inverter	Standard interface for `FB_SimAnalog_Inverter`.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.6.1.0)

Also see about this

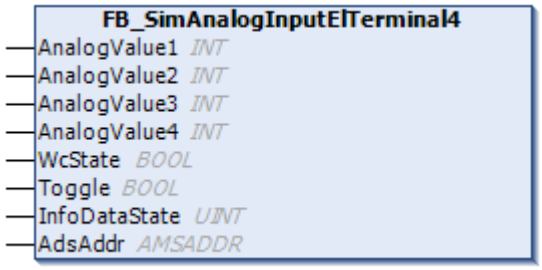
- Simulation of an analog output terminal [▶ 308]
- Simulation of an analog input terminal [▶ 307]

8.5.4 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 [▶ 307]	4	Simulation of an analog EtherCAT input terminal. Example EL3134
FB_SimAnalogOutputElTerminal4 [▶ 308]	4	Simulation of an analog EtherCAT output terminal. Example EL4134
FB_SimSsiInputElTerminal1 [▶ 309]	1	Simulation of an EtherCAT SSI input terminal. Example EL5001
FB_CoE402_ServoAdapt [▶ 309]	1	AX8000

8.5.4.1 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

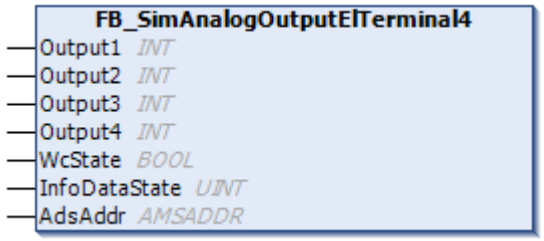
Name	Type	Access	Initial value	Description
AnalogValue1	INT	IN	0	The simulated input value of the 1st channel.
AnalogValue2	INT	IN	0	The simulated input value of the 2nd channel.
AnalogValue3	INT	IN	0	The simulated input value of the 3rd channel.
AnalogValue4	INT	IN	0	The simulated input value of the 4th channel.

Name	Type	Access	Initial value	Description
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

8.5.4.2 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
```

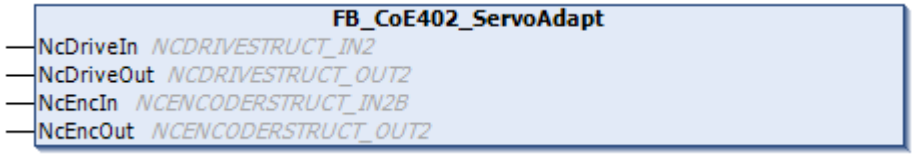
Inputs

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

8.5.4.3 Simulation of a CoE402 servo drive



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.
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

8.5.4.4 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

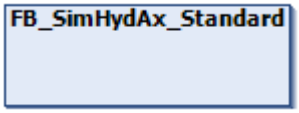
Name	Type	Access	Initial value	Description
AnalogValue1	INT	IN	0	The simulated input value of the 1st channel.
AnalogValue2	INT	IN	0	The simulated input value of the 2nd channel.
AnalogValue3	INT	IN	0	The simulated input value of the 3rd channel.
AnalogValue4	INT	IN	0	The simulated input value of the 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

8.5.5 Components of the hydraulic simulation

8.5.5.1 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	Simulated axis position. **The simulated position of the actuator can deviate from the reported axis position.**
FB_SimHydAx_Standard			
Cylinder	I_SimCylinder [▶ 315]	Get	An interface to the local `FB_SimCylinder` object.
Encoder	I_SimUniversalEncoder [▶ 317]	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 [▶ 316]	Get	An interface to the local `FB_PressureTransducer` object on the A-side of the `FB_SimCylinder` object.
PressureTransducerB	I_PressureTransducer [▶ 316]	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.

Name	Type	Access	Description
Valve	I_SimValve [▶ 313]	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.
FB_SimAxCommon			
ActualVelocity	LREAL	Get, Set	Simulated axis velocity. **The simulated velocity of the actuator may 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 value for the case AbsSwitchHighSelect=TRUE: AbsolutSwitch is TRUE if the simulated actuator position is >= AbsSwitchHighEnd. **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 selects AbsSwitchHighEnd, while a FALSE makes AbsSwitchLowEnd the active AbsolutSwitch.
AbsSwitchLowEnd	LREAL	Get, Set	This parameter defines the threshold value for the case AbsSwitchHighSelect=FALSE: AbsolutSwitch is TRUE if the simulated actuator position is <= AbsSwitchLowEnd. **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.
DenyEnable	BOOL	Get, Set	A TRUE blocks the feedback of the controller enable. This can be used to simulate a lack of response from the field device.
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:

Name	Type	Access	Description
			• 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`.
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.
Reversed	BOOL	Get, Set	The simulated axis works inverted.
UpperEndSwitch	BOOL	Get	This runtime value provides information about the state of a simulated sensor. Becomes `TRUE` if the position of the simulated actuator is <= `LowSideEndswitch`.

Methods

Name	Description
GoFaultState() [▶ 282]	This method can be used to simulate an error situation on the simulated axis.
FB_SimHydAx_Standard	
Cyclic()	Cycle method

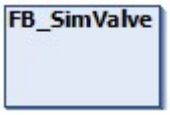
Interfaces

Type	Description
I_SimAxCommon	Standard interface for `FB_SimAxCommon`.
I_SimHydAx	Standard interface for FB_SimHydAx_Standard.

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0) Tc3_PlasticHydraulic (>= v12.8.2.0)

8.5.5.2 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.

Name	Type	Access	Description
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. **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

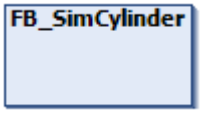
Interfaces

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

8.5.5.3 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.**

Name	Type	Access	Description
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

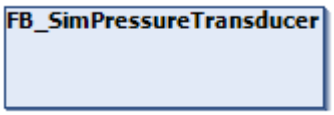
Interfaces

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 Tc3_PlasticHydraulic v3.12.4.26 or higher

8.5.5.4 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

8.5.5.5 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.
Reversed	BOOL	Get, Set	The encoder operates inverted to the working direction of the cylinder.
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
[PROTECTED] _Normalized()	Returns the normalized encoder value.

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

8.5.6 SimActuator



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)

8.5.7 Injection pressure simulation



This class implements a rudimentary simulation of the melt pressure of an injection process. For this purpose, a pressure sensor is simulated, which is synchronized into the controller via an analog interface (preferably FB_InputAnalog16 [▶ 227]).

Internally, a distinction is made in the class between the metering phase and the injection phase. For this purpose, both the rotary drive of the metering screw and the linear thrust drive must be provided in the simulation by an FB_SimAxCommon [▶ 280]-based axis.

Syntax:

```
FUNCTION_BLOCK FB_SimInjectionMeltPressure
```

I/O variables

Name	Type	Direction	Description
AnalogValue	INT	Output	The simulated pressure as an analog value.
EC.WcState	BOOL	Output	The simulated Working Counter State.
EC.Toggle	BOOL	Output	The simulated toggle bit.
EC.InfoDataState	UINT	Output	The simulated Device State.
EC.AdsAddr	AMSADDR	Output	The simulated ADS address.

Properties

Name	Type	Access	Description
AutoAdjustMeteringFactor	BOOL	Get, Set	If TRUE, the `MeteringFactor` is adjusted to the difference between the metered plastic and the linear thrust axis position after the metering process. This option is used to quickly calibrate the metering factor when activated for a short time. **If activated permanently, this option tends to incrementally reduce the `MeteringFactor`.**
Cavity	LREAL	Get, Set	Simulated cavity in length units of the linear thrust axis (typ. [mm]).
CycleTime	LREAL	Get, Set	Cycle time with which the simulation is called.
Enable	BOOL	Get, Set	If not TRUE, the sensor does not respond to any movements of the linked axes.
InjectionResistance	LREAL	Get, Set	Resistance in pressure units (typ. [bar]) during injection, for example, through an upstream injection nozzle.
Injector	I_SimAxCommon [▶ 280]	Get, Set	Interface to the linear thrust axis. **The assignment of a simulated axis is mandatory.**
IsInjecting	BOOL	Get	TRUE if the simulation has detected an injection process and is actively simulating it.
IsMetering	BOOL	Get	TRUE if the simulation has detected a metering process and is actively simulating it.
Metering	I_SimAxCommon [▶ 280]	Get, Set	Interface to the rotary metering screw. **The assignment of a simulated axis is mandatory.**
MeteringFactor	LREAL	Get, Set	Specifies the increase in metered plastic during the metering process. The value is measured in length units of the linear thrust axis per revolution of the rotary metering screw (typ. [mm/360°]).
Pressure	LREAL	Get	Current simulated pressure value.
PressureFactor	LREAL	Get, Set	Melt pressure generated per length unit of the linear thrust axis (typ. [bar/mm]). This value represents the compressibility of the liquid injection mass.
ShrinkingFactor	LREAL	Get, Set	The larger this factor is, the faster the plastic shrinks during simulated cooling. Cooling begins as soon as the filling volume of the cavity is reached.
WipeAfterMetering	BOOL	Get, Set	If TRUE, the metered volume is synchronized to the current position of the linear thrust axis after the metering process has been completed.

Name	Type	Access	Description
			This option improves the stability of the simulation process. This prevents overpressure arising from **in**correctly set back pressure control from influencing the start of the injection process.

Methods

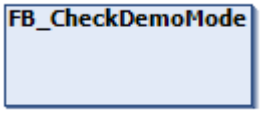
Name	Description
Cyclic()	Cycle method

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.12.0.0)

8.6 Debug

8.6.1 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.

Name	Type	Description
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

8.6.2 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.

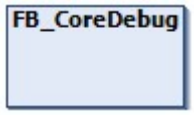
Methods

Name	Description
Cyclic(`fValue`)	Cycle method **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

8.6.3 FB_CoreDebug



This class is used for better debugging of the core function chain.

Syntax:

```
FUNCTION_BLOCK FB_CoreDebug
```

Properties

Name	Type	Access	Description
FunctionMatchesCounter	BOOL	Get	TRUE if CyclicCounter and FunctionCounter have the same value.

Methods

Name	Description
AxisStateIncrement()	Increments the internal counter value `nAxisStatecounter`. **Is called by** `FB_CoreFunction.AxisState.Set`.
AxisStateReset()	Resets the internal counter value `nAxisStatecounter`.
CyclicCounterIncrement()	Increments the internal counter value `nCyclicCounter`. **Is called by** `FB_CoreFunction.Cyclic()`.
CyclicCounterReset()	Resets the internal counter value `nCyclicCounter`.
FunctionCounterIncrement()	Increments the internal counter value `nFunctionCounter`. **Is called by** `FB_CoreFunction.NextCore.Set`.
FunctionCounterReset()	Resets the internal counter value `nFunctionCounter`.
SendDebugMessage()	Sends the transferred event as a debug message.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.71 TwinCAT v3.1.4026.19	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.14.0.0)

8.7 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(172)	File path under which the hydraulic configuration is to be saved.

Properties

Name	Type	Access	Description
Path	STRING(172)	Get, Set	File path under which the hydraulic configuration is to be saved.

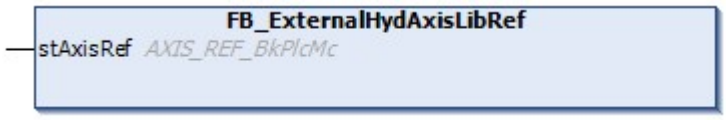
Methods

Name	Description
Append(`iAxis`)	Appends an axis from the internal list to be connected.
Cyclic()	Cycle method

Prerequisites

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.56	PC or CX (x64, x86)	Tc3_PlasticFunctions (>= v12.8.2.0) Tc3_PlasticHydraulic (>= v12.8.2.0)

8.8 ExternalHydAxisLibRef



This class represents a standard hydraulic axis of the Tc2_Hydraulic library (without using an FB_AxisHydraulicBase). This allows standard hydraulic axes and TF8560-based hydraulic axes to communicate with the PlcMcManager in one application.

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 of 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. **A TRUE must be set before the first Cyclic call.**

Methods

Name	Description
GetHydAxisRef()	Returns a `POINTER` to the `AXIS_REF_BkPlcMc`.
[`INTERNAL`] LeaveCriticalSection()	Used internally.
[`INTERNAL`] EnterCriticalSection()	Used internally.

Interfaces

Type	Description
I_ExternalHydAxisLibRef	Standard interface to 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

9 FAQ

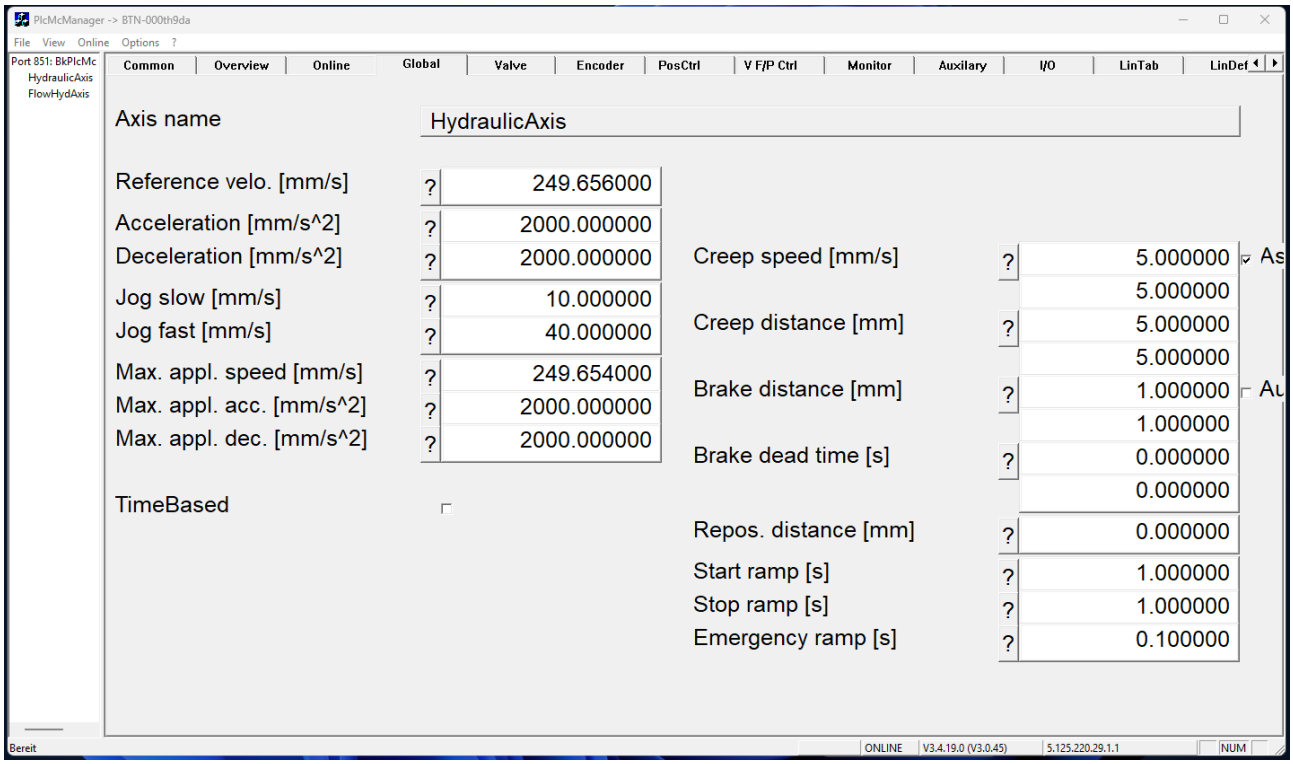
9.1 Hydraulic

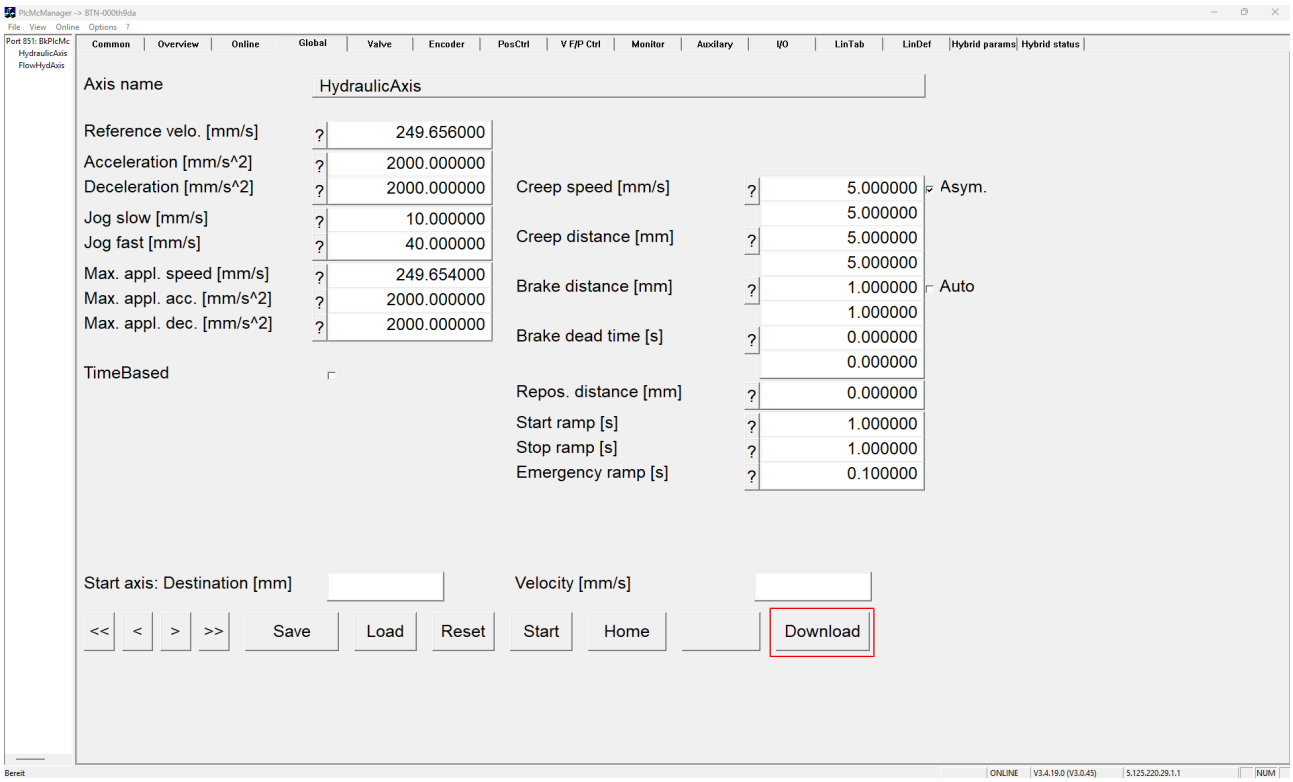
The values I enter in PlcMcManager are not being applied. What can I do?

Your changes will be applied to the PLC when you click the **Download** button.

If you enter new values and switch tabs before clicking **Download**, your changes will be ignored.

If you cannot see the download button, maximize the program window.





10 Appendix

10.1 Camming

This section summarizes the obsolete elements of the camming function.

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

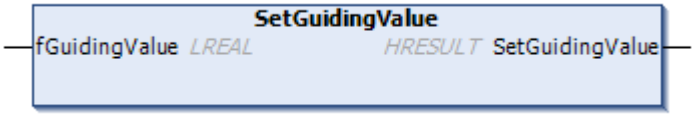
10.1.1 SetGuidingValue()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Camming [▶ 90].Pozna|Velo.GuidingValue

Original version:



This method must be used cyclically to update the guiding 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 guiding value that will be used 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

10.1.2 SetLookupInterface()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Camming [▶ 90].Pozna|Velo.LookUp

Original version:



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 [▶ 94]	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

10.2 PressureControl

This section summarizes the obsolete elements of pressure control.

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

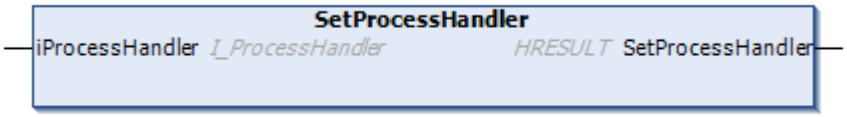
10.2.1 I_AxisBase.SetProcessHandler()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: .PressureControl.Supervisor.SetExternal()

Original version:



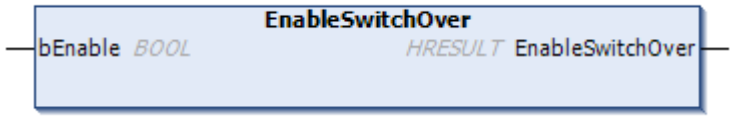
Reserved for extension: An interface to a FB for automatic pressure controller activation is entered.

10.2.2 PID.EnableSwitchOver()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PID [▶ 162].AllowSwitchOver



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.

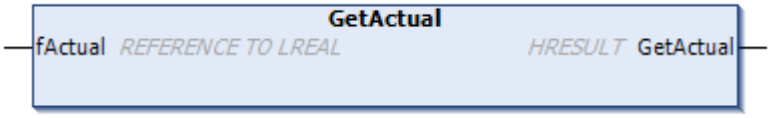
10.2.3 PID.GetActual()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PID [▶ 162].Actual

Original version:



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.

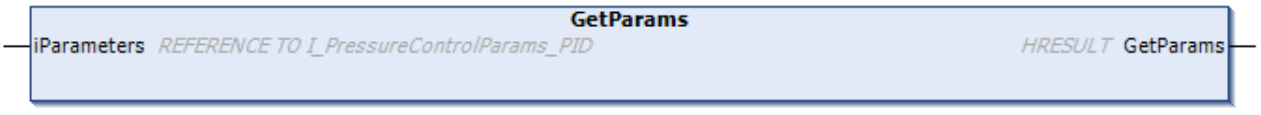
10.2.4 PID.GetParams()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PID [▶ 162].Parameters [▶ 166]

Original version:



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.

10.2.5 PID.SetParams()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PID [▶ 162].Parameters [▶ 166]

Original version:



A parameter set is connected to the controller.

Syntax:

```
METHOD SetParams: HRESULT
VAR_INPUT
    iParams: I_PressureControlParams_PID [▶ 166];
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.

10.2.6 PID.Setpoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PID [▶ 162].Target

Original version:



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.

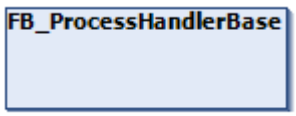
10.2.7 FB_ProcessHandlerBase

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PressureControl.Supervisor [▶ 171]

Original version:



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.

Name	Type	Description
ProcessValue	LREAL	The process value to be monitored.
SwitchOver	BOOL	TRUE if an expected switchover occurred.

Methods

Name	Description
GetControlParameter [▶ 333]	This method can be used to get access to the parameter container of the axis pressure controller.
GetProcessValues [▶ 334]	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 [▶ 334]	This method is used to read back the rules for the switchover.
GetSwitchParameter [▶ 334]	This method may be used to read back parameters of the switchover rules.
SetControlParameter [▶ 335]	This method can be used to assign a parameter container to the axis pressure controller.
SetSwitchEnable [▶ 335]	This method is used to define the rules for the switchover. See below for more information.
SetSwitchParameter [▶ 335]	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:

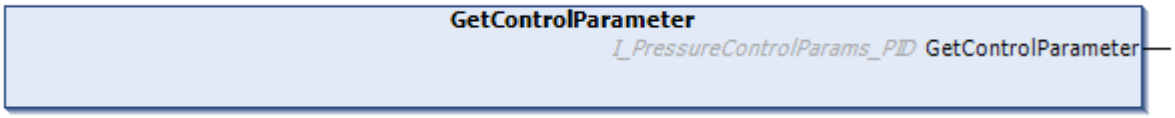
If no faulty switchover has been detected yet:

- 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.

10.2.7.1 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
```

10.2.7.2 GetProcessValue



This method is used to update the FB with a range of actual values (pressure, position, state) of the axis.

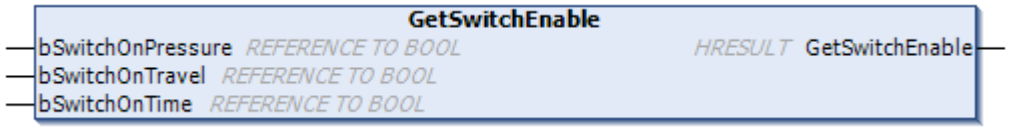


This method is called cyclically by the axis. The application is not intended to call this method directly.

Syntax:

```
METHOD GetProcessValues : HRESULT
VAR
END_VAR
```

10.2.7.3 GetSwitchEnable



This method is used to read back the rules for the switchover.

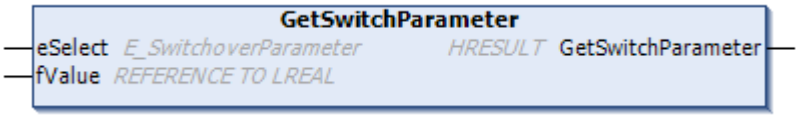
Syntax:

```
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.

10.2.7.4 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 [▶ 336]	The selection of the parameter.
fValue	REFERENCE TO LREAL	A reference to the variable to be updated with the parameter.

10.2.7.5 SetControlParameter



This method can be used to assign a parameter container to the axis pressure controller.

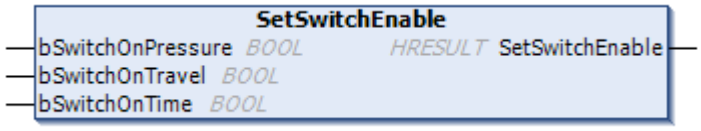
Syntax:

```
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.

10.2.7.6 SetSwitchEnable



This method is used to define the rules for the switchover. See below for more information.

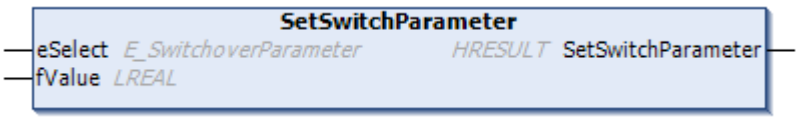
Syntax:

```
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.

10.2.7.7 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 [▶ 336]	The selection of the parameter.
fValue	LREAL	The value with which the parameter should be updated.

10.2.7.8 E_SwitchoverParameter

Values of this enumeration are use by GetSwitchParameter() and SetSwitchParameter() of FB_ProcessHandlerBase [▶ 332].

```
TYPE E_SwitchoverParameter :
(
ePressureThreshold := 0,
eGardingTravel,
eTravelThreshold,
eTimeThreshold,

eAlarmPressureLimit,
eAlarmPositionLimit,
eTimeout
);
END_TYPE
```

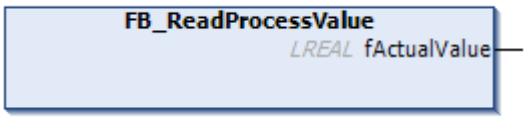
10.2.8 FB_ReadProcessValue

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: PressureControl.Supervisor [▶ 171]

Original version:



A function block of this type is used to supply a function block of type FB_ProcessHandlerBase [▶ 332] 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.

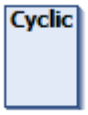
Properties

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().
Weighting	LREAL	Get, Set	The scaling factor.

Methods

Name	Description
Cyclic [▶ 337]	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 [▶ 337]	A call of this method with bExecute=TRUE resets the connected sensor interface and clears all local errors.

10.2.8.1 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
```

10.2.8.2 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
```

10.3 Ptp

This section summarizes the obsolete elements of the PTP function.

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

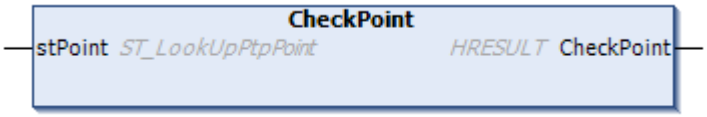
10.3.1 CheckPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: / (`INTERNAL`)

Original version:



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 [▶ 184]	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

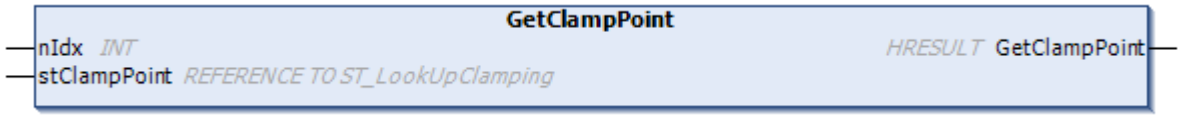
10.3.2 GetClampPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Ptp [▶ 173].ClampingPoint [▶ 176].Fetch(x)

Original version:



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 [▶ 178]	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

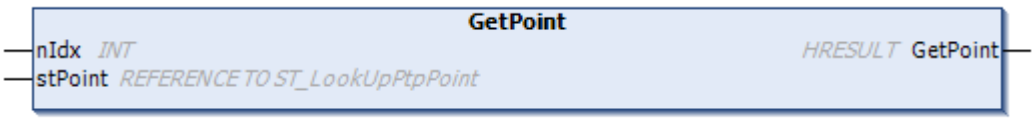
10.3.3 GetPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Ptp [▶ 173].Point [▶ 182].Fetch(x)

Original version:



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 [▶ 184]	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

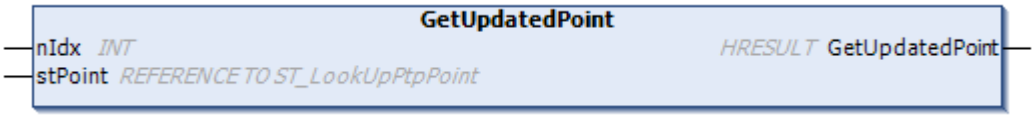
10.3.4 GetUpdatedPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: / (`INTERNAL`)

Original version:



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 [▶ 184]	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

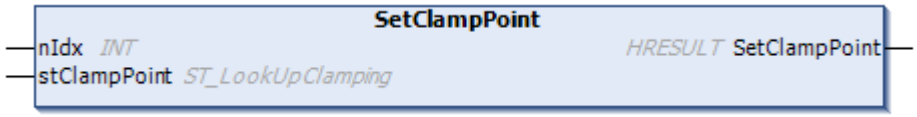
10.3.5 SetClampPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Ptp [▶ 173].ClampingPoint [▶ 176].Set(x)

Original version:



A segment of the clamping table is defined.

Syntax:

```
METHOD SetClampPoint: HRESULT
VAR_INPUT
    Idx:          INT;
    stClampPoint: ST_LookUpClamping;
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 [▶ 178]	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

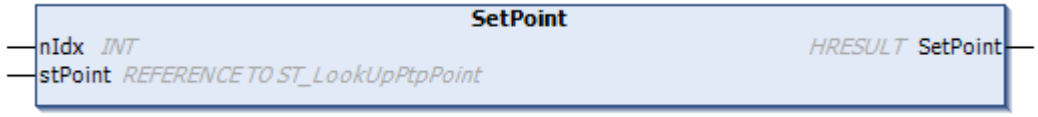
10.3.6 SetPoint()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Ptp [▶ 173].Point [▶ 182].Set(x)

Original version:



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 [▶ 184]	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

10.3.7 UpdatePosition()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: / (`INTERNAL`)

Original version:



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

10.4 Stop

This section summarizes the obsolete elements of the stop function.

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

10.4.1 SetParameter()

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Stop [▶ 194].Deceleration & Stop [▶ 194].Jerk

Original version:



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

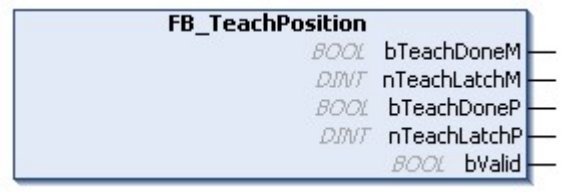
10.5 TeachPosition

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Teach [▶ 196]

Original version:



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.
TeachLatchP	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)

10.6 TeachUpdate

Class is 'obsolete'

This class is marked as `obsolete` and should not be used anymore!

Alternative: Teach [▶ 196]

Original version:



This core function is used for referencing analog encoder systems. It is used after TeachPosition [▶ 344] 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

10.7 TorqueLimiting

Class is 'obsolete'

This class is marked as obsolete and should not be used anymore!

Alternative: Limiting [▶ 145]

Original version:



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.

Name	Type	Access	Description
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 the maximum command value of the 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

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

MAX®, Stratix®, Cyclone®, Altera®, Agilex™, Arria®, Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

Sercos is a registered trademark of Sercos International e.V.

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