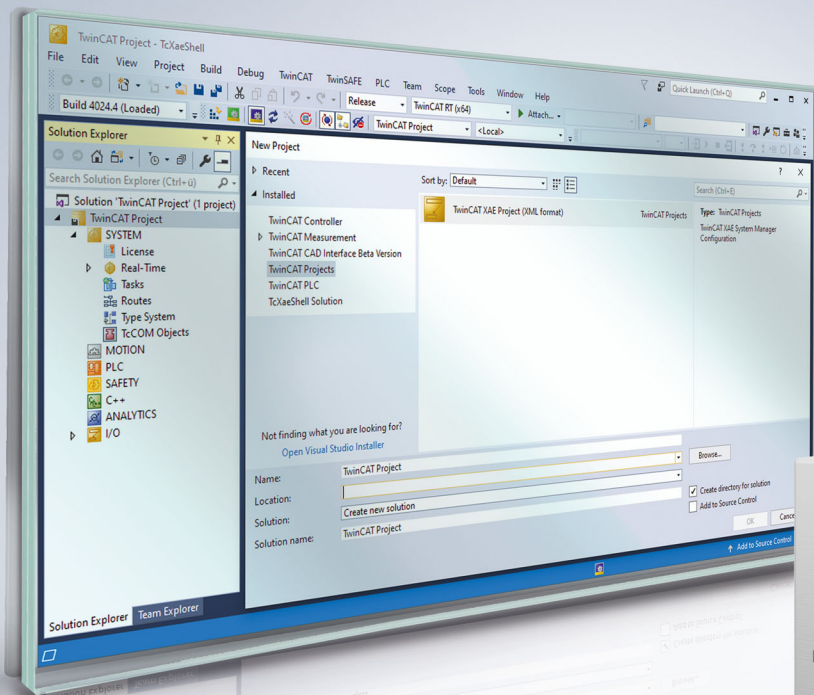


# BECKHOFF New Automation Technology

Manual | EN

# TE1000

TwinCAT 3 | PLC Library: Tc2\_MC2\_Drive





# Table of contents

<b>1</b>	<b>Foreword</b> .....	<b>5</b>
1.1	Notes on the documentation .....	5
1.2	For your safety .....	6
1.3	Notes on information security.....	7
<b>2</b>	<b>Overview</b> .....	<b>8</b>
<b>3</b>	<b>Functions</b> .....	<b>11</b>
3.1	F_GetVersionTcMc2Drive .....	11
<b>4</b>	<b>Function blocks</b> .....	<b>12</b>
4.1	General Beckhoff .....	12
4.1.1	FB_DeletePositionOffset.....	12
4.1.2	FB_BrakeControl.....	13
4.1.3	FB_SetPositionOffset.....	14
4.1.4	FB_ReadDriveInfo.....	15
4.1.5	FB_ParkAxis .....	16
4.2	General SoE.....	17
4.2.1	FB_SoERead .....	17
4.2.2	FB_SoEWrite .....	21
4.2.3	FB_SoEReset .....	24
4.2.4	FB_SoEWritePassword.....	25
4.2.5	FB_SoESetDataAccessMode .....	26
4.2.6	Function blocks for command .....	27
4.2.7	Function blocks for diagnostics .....	32
4.2.8	Function blocks for determining current values.....	38
4.3	General CoE .....	43
4.3.1	FB_CoERead .....	43
4.3.2	FB_CoEWrite .....	45
4.3.3	FB_CoEExecuteCommand.....	47
4.4	AX5000 SoE.....	48
4.4.1	FB_SoEAX5000ReadActMainVoltage .....	48
4.4.2	FB_SoEAX5000SetMotorCtrlWord .....	49
4.4.3	FB_SoEAX5000FirmwareUpdate .....	51
4.4.4	FB_SoEAX5000SetPositionOffset .....	55
4.4.5	FB_SoEAX5000DeletePositionOffset .....	56
4.4.6	FB_SoEAX5000ParkAxis.....	57
4.5	AX8000 CoE .....	58
4.5.1	FB_CoEAX8000BrakeControl.....	58
4.5.2	FB_CoEAX8000BrakeTest .....	59
4.5.3	FB_CoEAX8000SetPositionOffset.....	61
4.5.4	FB_CoEAX8000DeletePositionOffset .....	62
4.5.5	FB_CoEAX8000ParkAxis.....	63
4.6	EL72xx CoE .....	64
4.6.1	FB_CoEEL72xxBrakeControl.....	64
4.6.2	FB_CoEEL72xxSetPositionOffset.....	65

4.6.3	FB_CoEEL72xxDeletePositionOffset.....	66
<b>5</b>	<b>Data types .....</b>	<b>68</b>
5.1	E_FwUpdateState .....	68
5.2	E_DriveErrorCodes .....	69
5.3	E_PositionOffsetMemory .....	69
5.4	E_PositionOffsetFeedback.....	70
5.5	E_BrakeMode .....	70
5.6	E_DeviceType.....	70
5.7	E_ParkMode .....	71
5.8	ST_DriveInfo .....	71
5.9	E_SoEDataAccessMode.....	72
5.10	E_AX5000ParkMode.....	72
<b>6</b>	<b>Global constants .....</b>	<b>73</b>
6.1	SoE Parameter Access .....	73
<b>7</b>	<b>Support and Service .....</b>	<b>74</b>

# 1 Foreword

## 1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

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

### Disclaimer

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

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

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

### Trademarks

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

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

### Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702  
and similar applications and registrations in several other countries.

**EtherCAT** 

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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The distribution and reproduction of this document as well as the use and communication of its contents without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

## 1.2 For your safety

### Safety regulations

Read the following explanations for your safety.

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

### Exclusion of liability

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

### Personnel qualification

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

### Signal words

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

#### Personal injury warnings

##### **DANGER**

Hazard with high risk of death or serious injury.

##### **WARNING**

Hazard with medium risk of death or serious injury.

##### **CAUTION**

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

#### Warning of damage to property or environment

##### **NOTICE**

The environment, equipment, or data may be damaged.

#### Information on handling the product



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

## 1.3 Notes on information security

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

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

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

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

## 2 Overview

The Tc2\_MC2\_Drive library includes functions and function blocks for SoE drives that access the drive by MC2 axis structure (AXIS\_REF).

### Drive libraries

The three drive libraries Tc2\_Drive, Tc2\_NcDrive and Tc2\_MC2\_Drive were developed for different functional purposes, but are almost identical in their functionality. The function blocks of the libraries Tc2\_NcDrive and Tc2\_MC2\_Drive form wrapper function blocks around the function blocks of the Tc2\_Drive library.

Drive library	Use	Access to the drive	Remarks
<b>Tc2_Drive</b> See documentation <a href="#">TwinCAT 3 PLC Lib: Tc2_Drive</a>	Use the Tc2_Drive library if you use the drive entirely from the PLC (i.e. without NC).	The drive is accessed via a drive reference. Within the library, the ST_DriveRef structure is used for this with the NetID as a string. For linking purposes, a structure called ST_PlcDriveRef is also provided with the NetID as a byte array.  (See Drive reference ST_DriveRef)	If you want to access parameters in the drive for which no special function block has been implemented, use the function blocks FB_SoERead_ByDriveRef and FB_SoEWrite_ByDriveRef. These function blocks are implemented in the PLC Lib Tc2_EtherCAT in the SoE Interface folder.
<b>Tc2_NcDrive</b> See documentation <a href="#">TwinCAT 3 PLC Lib: Tc2_NcDrive</a>	Use the Tc2_NcDrive library if you are using the drive via the NC with the Tc2_Nc libraries.	The drive is accessed via the NC axis structure (NC_TO_PLC). The function blocks of the Tc2_NcDrive library independently determine the access data to the drive (NetID, address and channel number) via the NC axis ID from the NC axis structure.	If you want to access parameters in the drive for which no special function block has been implemented, use the function blocks FB_SoERead and FB_SoEWrite.
<b>Tc2_MC2_Drive</b> See: <a href="#">TwinCAT 3 PLC Lib Tc2_MC2_Drive</a> documentation	Use the Tc2_MC2_Drive library if you are using the drive via the NC with the Tc2_MC2 library.	The drive is accessed via the MC2 axis reference (AXIS_REF). The function blocks of the Tc2_MC2_Drive library independently determine the access data to the drive (NetID, address and channel number) via the NC axis ID from the MC2 axis reference.	If you want to access parameters in the drive for which no special function block has been implemented, use the function blocks FB_SoERead and FB_SoEWrite.



Note the differences when using the drive libraries with AX5000 and Bosch Rexroth IndraDrive CS (see Samples)

### Functions

Name	Description
F_GetVersionTcMc2Drive  ▶  11]	Reads version information from the PLC library. The function has been replaced by the global structure stLibVersion_Tc2_MC2_Drive.
F_ConvWordToSTAX5000C1D	Converts the C1D word (S-0-0011) of the AX5000 to an ST_AX5000_C1D structure.  See: <a href="#">TwinCAT 3 PLC Library:Tc2_Drive</a> documentation.



**Function blocks**

General Beckhoff

Name	Description
<a href="#">FB_BrakeControl [▶ 13]</a>	For opening and closing the motor brake.
<a href="#">FB_SetPositionOffset [▶ 14]</a>	Saves a position offset.
<a href="#">FB_DeletePositionOffset [▶ 12]</a>	Deletes a position offset.
<a href="#">FB_ReadDriveInfo [▶ 15]</a>	Reads basic information.
<a href="#">FB_ParkAxis [▶ 16]</a>	For parking and releasing the axis

Allgemein SoE

Name	Description
<a href="#">FB_SoERead [▶ 17]</a>	Reads a parameter.
<a href="#">FB_SoEWrite [▶ 21]</a>	Writes a parameter.
<a href="#">FB_SoEReset [▶ 24]</a>	Resets the drive (S-0-0099).
<a href="#">FB_SoEWritePassword [▶ 25]</a>	Sets the drive password (S-0-0267).
<a href="#">FB_SoESetDataAccessMode [▶ 26]</a>	Sets the access mode of SoE parameters.
<a href="#">FB_SoEExecuteCommand [▶ 27]</a>	Executes a command.
<a href="#">FB_SoEWriteCommandControl [▶ 29]</a>	Sets the Command Control.
<a href="#">FB_SoEReadCommandState [▶ 30]</a>	Checks the command state.
<a href="#">FB_SoEReadDiagMessage [▶ 32]</a>	Reads the diagnostic message (S-0-0095).
<a href="#">FB_SoEReadDiagNumber [▶ 33]</a>	Reads the diagnostic number (S-0-0390).
<a href="#">FB_SoEReadDiagNumberList [▶ 35]</a>	Reads the diagnostic number list (up to 30 entries) (S-0-0375).
<a href="#">FB_SoEReadClassXDiag [▶ 36]</a>	Reads the Class 1 diagnosis (S-0-0011) and the Class 3 diagnosis (S-0-0013).
<a href="#">FB_SoEReadAmplifierTemperature [▶ 38]</a>	Reads the drive temperature (S-0-0384).
<a href="#">FB_SoEReadMotorTemperature [▶ 39]</a>	Reads the motor temperature (S-0-0383).
<a href="#">FB_SoEReadDcBusCurrent [▶ 40]</a>	Reads the DC bus current (S-0-0381).
<a href="#">FB_SoEReadDcBusVoltage [▶ 42]</a>	Reads the DC bus voltage (S-0-0380).

General CoE

Name	Description
<a href="#">FB_CoERead [▶ 43]</a>	Reads a parameter.
<a href="#">FB_CoEWrite [▶ 45]</a>	Writes a parameter.
<a href="#">FB_CoEExecuteCommand [▶ 47]</a>	Executes a command.

AX5000 SoE

Name	Description
<a href="#">FB_SoEAX5000ReadActMainVoltage [▶ 48]</a>	Reads the mains voltage (P-0-0200).
<a href="#">FB_SoEAX5000SetMotorCtrlWord [▶ 49]</a>	Sets the Motor Control Word (P-0-0096).
<a href="#">FB_SoEAX5000FirmwareUpdate [▶ 51]</a>	Executes an automatic firmware update for the AX5000.
<a href="#">FB_SoEAX5000SetPositionOffset [▶ 55]</a>	Saves a position offset.

FB_SoEAX5000DeletePositionOffset [ <a href="#">▶ 56</a> ]	Deletes a position offset.
FB_SoEAX5000ParkAxis [ <a href="#">▶ 57</a> ]	For parking and releasing the axis.

## AX8000 CoE

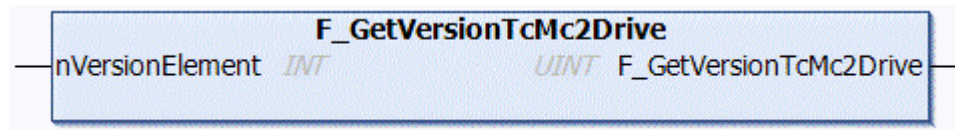
Name	Description
FB_CoEAX8000BrakeControl [ <a href="#">▶ 58</a> ]	For opening and closing the motor brake.
FB_CoEAX8000BrakeTest [ <a href="#">▶ 59</a> ]	Performs a brake test.
FB_CoEAX80000SetPositionOffset [ <a href="#">▶ 61</a> ]	Saves a position offset.
FB_CoEAX80000DeletePositionOffset [ <a href="#">▶ 62</a> ]	Deletes a position offset.
FB_CoEAX8000ParkAxis [ <a href="#">▶ 63</a> ]	For parking and releasing the axis.

## EL72xx CoE

Name	Description
FB_CoEEL72xxBrakeControl [ <a href="#">▶ 64</a> ]	For opening and closing the motor brake.
FB_CoEEL72xxSetPositionOffset [ <a href="#">▶ 65</a> ]	Saves a position offset.
FB_CoEEL72xxDeletePositionOffset [ <a href="#">▶ 66</a> ]	Deletes a position offset.

## 3 Functions

### 3.1 F\_GetVersionTcMc2Drive



This function can be used to read PLC library version information.

#### FUNCTION F\_GetVersionTcMc2Drive: UINT

```
VAR_INPUT
  nVersionElement : INT;
END_VAR
```

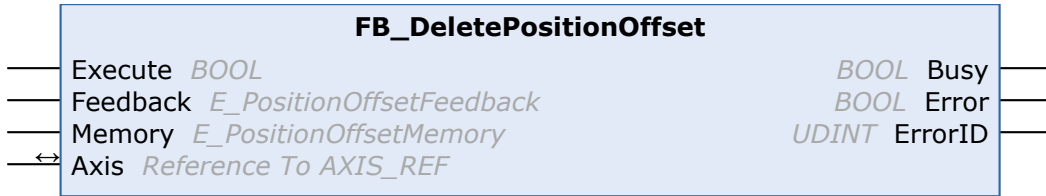
**nVersionElement:** Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

## 4 Function blocks

### 4.1 General Beckhoff

#### 4.1.1 FB\_DeletePositionOffset



The FB\_DeletePositionOffset function block can be used to delete a position offset in the memory of a Beckhoff servo hardware.

Supported Beckhoff hardware		
AX5xxx	AX8xxx / AMP8xxx / MD8xxx	Compact drive technology (servo)
✓ from FW v2.11 b0001	✓ from FW v1.04 b0001	✓ from FW v01 (AMI8xxx only Drive)

#### Inputs

```
VAR_INPUT
  Execute : BOOL;
  Feedback : E_PositionOffsetFeedback;
  Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Feedback	<a href="#">E_PositionOffsetFeedback</a> <a href="#"> ▶ 70 </a>	Specifies which feedback is considered.
Memory	<a href="#">E_PositionOffsetMemory</a> <a href="#"> ▶ 69 </a>	Specifies the memory from which the position offset is to be deleted.

#### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<a href="#">AXIS_REF</a>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

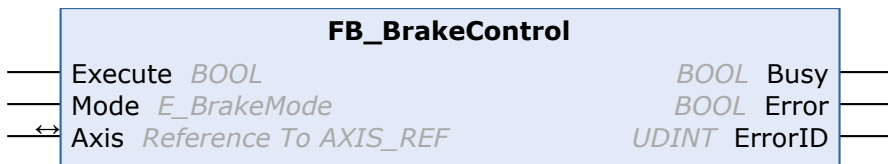
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.

Name	Type	Description
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library
TwinCAT ≥4024.48	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.34.0




**4.1.2 FB\_BrakeControl**



The FB\_BrakeControl function block can be used to manually open or close the brake of a motor that is operated on Beckhoff servo hardware.

It should be noted that the brake is permanently closed or opened via "Lock" or "Unlock".

It is recommended to open or close the brake manually only for the necessary duration. The brake should then be set back to "Automatic" mode using the function block.

Supported Beckhoff hardware		
AX5xxx	AX8xxx / AMP8xxx / MD8xxx	Compact drive technology (servo)
 from FW v1.07 b0001	 from FW v1.04 b0001	 from FW v01 (EL72xx only with OCT and from FW v16)

 **Inputs**

```
VAR_INPUT
Execute : BOOL;
Mode : E_BrakeMode;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	E_BrakeMode [ <a href="#">▶ 70</a> ]	The mode with which the brake is controlled.

 **Inputs/outputs**

```
VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.




**Requirements**

Development environment	Target platform type	PLC library
TwinCAT ≥4024.48	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.34.0

**4.1.3 FB\_SetPositionOffset**



The FB\_SetPositionOffset function block can be used to store a position offset in the memory of a Beckhoff servo hardware. Saving in the digital nameplate of the connected motor encoder is also possible. To do this, the offset must first be configured with the DriveManager and then the same memory location (encoder / drive) must be used in the function block.

Supported Beckhoff hardware		
AX5xxx	AX8xxx / AMP8xxx / MD8xxx	Compact drive technology (servo)
 from FW v2.11 b0001	 from FW v1.04 b0001	 from FW v01 (AMI8xxx only Drive)

 **Inputs**

```
VAR_INPUT
Execute : BOOL;
Position : LREAL;
Relative : BOOL;
Feedback : E_PositionOffsetFeedback;
Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New actual position of the NC axis
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	E_PositionOffsetFeedback [▶ 70]	Specifies which feedback is considered.

Name	Type	Description
Memory	E_PositionOffsetMemory [▶_69]	Specifies which memory the newly calculated position offset should be stored in.

Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

Outputs

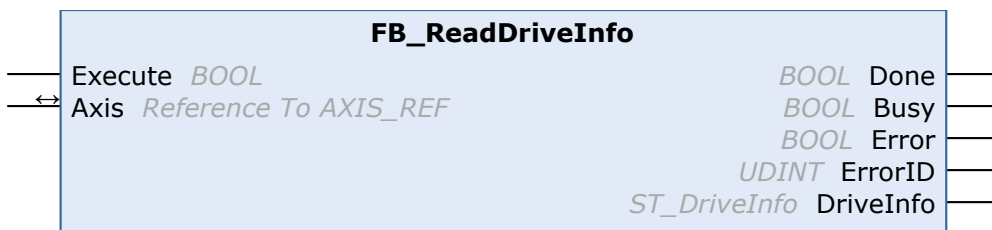
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

Requirements

Development environment	Target platform type	PLC library
TwinCAT ≥4024.48	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.34.0

### 4.1.4 FB\_ReadDriveInfo



The FB\_ReadDriveInfo function block can be used to read basic information for communication with a Beckhoff servo hardware.

Supported Beckhoff hardware		
AX5xxx	AX8xxx / AMP8xxx / MD8xxx	Compact drive technology
✔	✔	✔

Inputs

```
VAR_INPUT
  Execute : BOOL;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

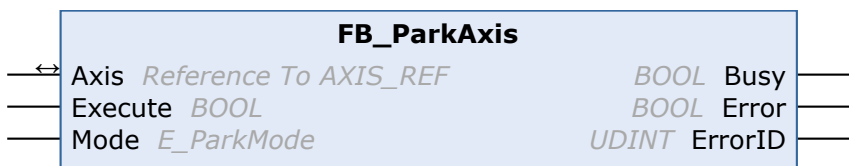
```
VAR_OUTPUT
  Done      : BOOL;
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
  DriveInfo : ST_DriveInfo;
END_VAR
```

Name	Type	Description
Done	BOOL	TRUE if the command was executed successfully.
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.
DriveInfo	<u>ST_DriveInfo</u> <a href="#"> ▶ 71</a>	Data structure with basic information to communicate with a Beckhoff servo hardware.

**Requirements**

Development environment	Target platform type	PLC library
TwinCAT ≥4024.48	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.34.0

**4.1.5 FB\_ParkAxis**



The function block enables the parking functionality of a supported Beckhoff drive. A parked channel of a drive is temporarily disabled.

In the case of a modular machine concept, certain motors may not be available. So that this does not lead directly to errors (e.g. feedback errors), the channels can be disabled via this.

Supported Beckhoff hardware		
AX5xxx	AX8xxx / MD8xxx	Compact drive technology
✓ from FW v2.12 b0001	✓ from FW v1.06 b0003	✗



**Inputs**

```
VAR_INPUT
Execute : BOOL;
Mode : E_ParkMode;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	E_ParkMode_[]72]	Mode for selecting whether the FB parks or releases the axis.

**Inputs/outputs**

```
VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**Outputs**

```
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
```

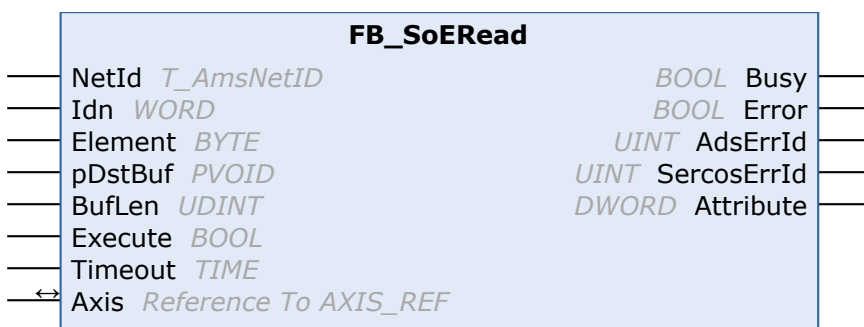
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library
TwinCAT ≥4024.60 TwinCAT ≥4026.8	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.41.0

## 4.2 General SoE

### 4.2.1 FB\_SoERead



With the FB\_SoERead function block a parameter can be read.

By default, attributes and values are read in parallel.

Should the call generate an ADS error on a third-party device, the lack of support for this faster parameter access method may be the cause.

In this case, the [FB\\_SoESetDataAccessMode](#) [▶ 26] function block can be used to switch to the slower sequential access method, if required.

### Inputs

```
VAR_INPUT
  NetId      : T_AmsNetID := '';
  Idn       : WORD;
  Element    : BYTE;
  pDstBuf    : PVOID;
  BufLen     : UDINT;
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Idn	WORD	Parameter number that FB_SoERead refers to, e.g. "S_0_IDNs + 33" for S-0-0033
Element	BYTE	Specifies which part of the parameter should be accessed, e.g. 16#40 is the value (Value) of the parameter.  EC_SOE_ELEMENT_DATASTATE :BYTE :=16#01;  EC_SOE_ELEMENT_NAME :BYTE :=16#02;  EC_SOE_ELEMENT_ATTRIBUTE :BYTE :=16#04;  EC_SOE_ELEMENT_UNIT :BYTE :=16#08;  EC_SOE_ELEMENT_MIN :BYTE :=16#10;  EC_SOE_ELEMENT_MAX :BYTE :=16#20;  EC_SOE_ELEMENT_VALUE :BYTE :=16#40;  EC_SOE_ELEMENT_DEFAULT :BYTE :=16#80;
pDstBuf	PVOID	ADR() of the variables that the value should be read to.
BufLen	UDINT	SIZEOF() of the variables that the value should be read to.
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.



The Idns can be taken from the corresponding drive documentation. For the AX5000 from Beckhoff see [AX5000 Idn description](#).

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
  Attribute : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

```
fbRead : FB_SoERead;
Read   : BOOL;
Idn    : WORD;
ReadValue : UINT;
(* NcAxis *)
Axis : AXIS_REF;

IF Read THEN
  Idn := S_0_IDNs + 33;
  fbRead(
    Axis := Axis,
    Idn := Idn,
    Element := 16#40,
    pDstBuf := ADR(ReadValue),
    BufLen := SIZEOF(ReadValue),
    Execute := TRUE,
    Timeout := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbRead.Busy THEN
    fbRead(Axis := Axis, Execute := FALSE);
    Read := FALSE;
  END_IF
END_IF
```

**Parameter access**

Note that for some parameters, the additional items of information "ActualLength" & "MaxLength" are necessary for the parameter access.

In order to determine the parameters for which this is necessary, the attributes can be displayed in the parameter list of the Drive Manager via the context menu of the header.

Index	Name	Actual value	Set value
S-0-0007	Feedback acquisition capture point (#4)	us	0 us
S-0-0011	Class 1 diagnostic (C1D)		0b0000 0000 0000 0000
S-0-0012	Class 2 diagnostic (C2D)		0b0000 0000 0000 0000
S-0-0013	Class 3 diagnostic (C3D)		0b0000 0000 0000 0000
S-0-0015	Telegram type		7
S-0-0016	Configuration list of AT		Show in list...
S-0-0017	IDN-list of all operation data		Show in list...
S-0-0018	IDN-list of operation data for CP2		Show in list...
S-0-0019	IDN-list of operation data for CP3		Show in list...
S-0-0020	IDN-list of operation data for CP4		Show in list...
S-0-0021	IDN-list of invalid operation data for CP2		Show in list...
S-0-0022	IDN-list of invalid operation data for CP3		Show in list...
S-0-0024	Configuration list of MDT		Show in list...
S-0-0025	IDN-list of all procedure commands		Show in list...
S-0-0029	MDT error counter		0
S-0-0032	Primary operation mode		pos ctrl feedback 1 la
S-0-0033	Secondary operation mode 1		no mode of operation
S-0-0034	Secondary operation mode 2		no mode of operation
S-0-0035	Secondary operation mode 3		no mode of operation
S-0-0036	Velocity command value	rev/(2^30 ms)	0 rev/(2^30 ms)
S-0-0037	Additive velocity command value	rev/(2^30 ms)	0 rev/(2^30 ms)
S-0-0040	Velocity feedback 1 value	rev/(2^30 ms)	0 rev/(2^30 ms)
S-0-0043	Velocity polarity parameter		0b0000 0000 0000 0000
S-0-0044	Velocity data scaling type		0b0000 0000 0000 1010
S-0-0045	Velocity data scaling factor		55 879
S-0-0046	Velocity data scaling exponent		-9
S-0-0047	Position command value	inc	0 inc
S-0-0048	Additive position command value	inc	0 inc
S-0-0051	Position feedback 1 value (motor feedback)	inc	0 inc
S-0-0053	Position feedback 2 value (external feedback)	inc	0 inc

If the fifth place from the right contains the value 4, 5, 6 or 7, then the additional items of information are also transmitted.

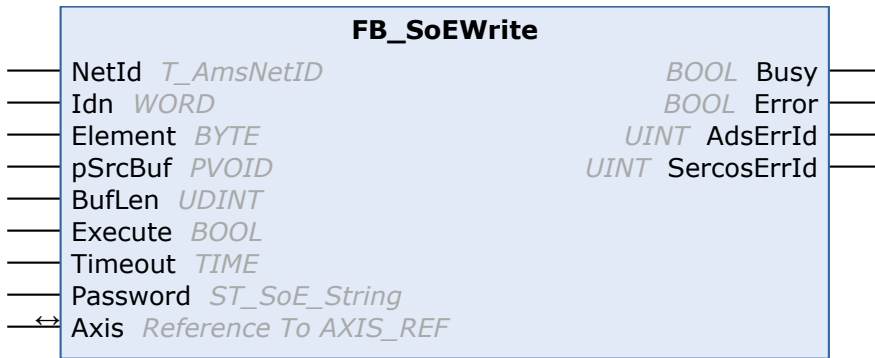
Index	Name	Actual value	Set value	Attribute
S-0-0007	Feedback acquisition capture point (#4)	us	0 us	0x60110001
S-0-0011	Class 1 diagnostic (C1D)		0b0000 0000 0000 0000	0x70010001
S-0-0012	Class 2 diagnostic (C2D)		0b0000 0000 0000 0000	0x70010001
S-0-0013	Class 3 diagnostic (C3D)		0b0000 0000 0000 0000	0x70010001
S-0-0015	Telegram type		7	0x60010001
S-0-0016	Configuration list of AT		Show in list...	0x60550001
S-0-0017	IDN-list of all operation data		Show in list...	0x70550001
S-0-0018	IDN-list of operation data for CP2		Show in list...	0x70550001
S-0-0019	IDN-list of operation data for CP3		Show in list...	0x70550001
S-0-0020	IDN-list of operation data for CP4		Show in list...	0x70550001
S-0-0021	IDN-list of invalid operation data for CP2		Show in list...	0x70550001
S-0-0022	IDN-list of invalid operation data for CP3		Show in list...	0x70550001
S-0-0024	Configuration list of MDT		Show in list...	0x60550001
S-0-0025	IDN-list of all procedure commands		Show in list...	0x70550001
S-0-0029	MDT error counter		0	0x70110001
S-0-0032	Primary operation mode		pos ctrl feedback 1 la	0x60010001
S-0-0033	Secondary operation mode 1		no mode of operation	0x00010001
S-0-0034	Secondary operation mode 2		no mode of operation	0x00010001
S-0-0035	Secondary operation mode 3		no mode of operation	0x00010001
S-0-0036	Velocity command value	rev/(2^30 ms)	0 rev/(2^30 ms)	0x70220001
S-0-0037	Additive velocity command value	rev/(2^30 ms)	0 rev/(2^30 ms)	0x70220001
S-0-0040	Velocity feedback 1 value	rev/(2^30 ms)	0 rev/(2^30 ms)	0x70220001
S-0-0043	Velocity polarity parameter		0b0000 0000 0000 0000	0x60010001
S-0-0044	Velocity data scaling type		0b0000 0000 0000 1010	0x70010001
S-0-0045	Velocity data scaling factor		55 879	0x70110001
S-0-0046	Velocity data scaling exponent		-9	0x70210001
S-0-0047	Position command value	inc	0 inc	0x70220001
S-0-0048	Additive position command value	inc	0 inc	0x70220001
S-0-0051	Position feedback 1 value (motor feedback)	inc	0 inc	0x70220001
S-0-0053	Position feedback 2 value (external feedback)	inc	0 inc	0x70220001

The structure to read out the serial number (S-0-0432) then looks like this:

```

TYPE ST_SerialNumber :
STRUCT
    ActualLength : UINT;
    MaxLength : UINT;
    SerialNumber : T_MaxString;
END_STRUCT
END_TYPE
    
```

### 4.2.2 FB\_SoEWrite



With the FB\_SoEWrite function block a parameter can be written.

#### Inputs

```

VAR_INPUT
    NetId      : T_AmsNetID := '';
    Idn        : WORD;
    Element    : BYTE;
    SrcBuf     : PVOID;
    BufLen     : UDINT;
    Execute    : BOOL;
    Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
    Password   : ST_SoE_String;
END_VAR
    
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Idn	WORD	Parameter number that FB_SoERead refers to, e.g. "S_0_IDNs + 47" for S-0-0047, see <a href="#">AX5000 Idn description</a> .
Element	BYTE	Specifies which part of the parameter should be accessed, e.g. 16#40 is the value (Value) of the parameter. Usually there is only write access to the value, other components of the parameter are read-only.  EC_SOE_ELEMENT_DATASTATE :BYTE :=16#01; EC_SOE_ELEMENT_NAME :BYTE :=16#02; EC_SOE_ELEMENT_ATTRIBUTE :BYTE :=16#04; EC_SOE_ELEMENT_UNIT :BYTE :=16#08; EC_SOE_ELEMENT_MIN :BYTE :=16#10; EC_SOE_ELEMENT_MAX :BYTE :=16#20; EC_SOE_ELEMENT_VALUE :BYTE :=16#40; EC_SOE_ELEMENT_DEFAULT :BYTE :=16#80;
SrcBuf	PVOID	ADR() of the variable containing the value to be written.
BufLen	UDINT	SIZEOF() of the variables that the value should be read to.
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.
Password	ST_SoE_String	Password as Sercos string. Currently not used. The password must be written with FB_SoEWritePassword.



The Idns can be taken from the corresponding drive documentation. For the AX5000 from Beckhoff see [AX5000 Idn description](#).

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UDINT	In the case of a set Error output returns the Sercos error of the last executed command.

### Sample

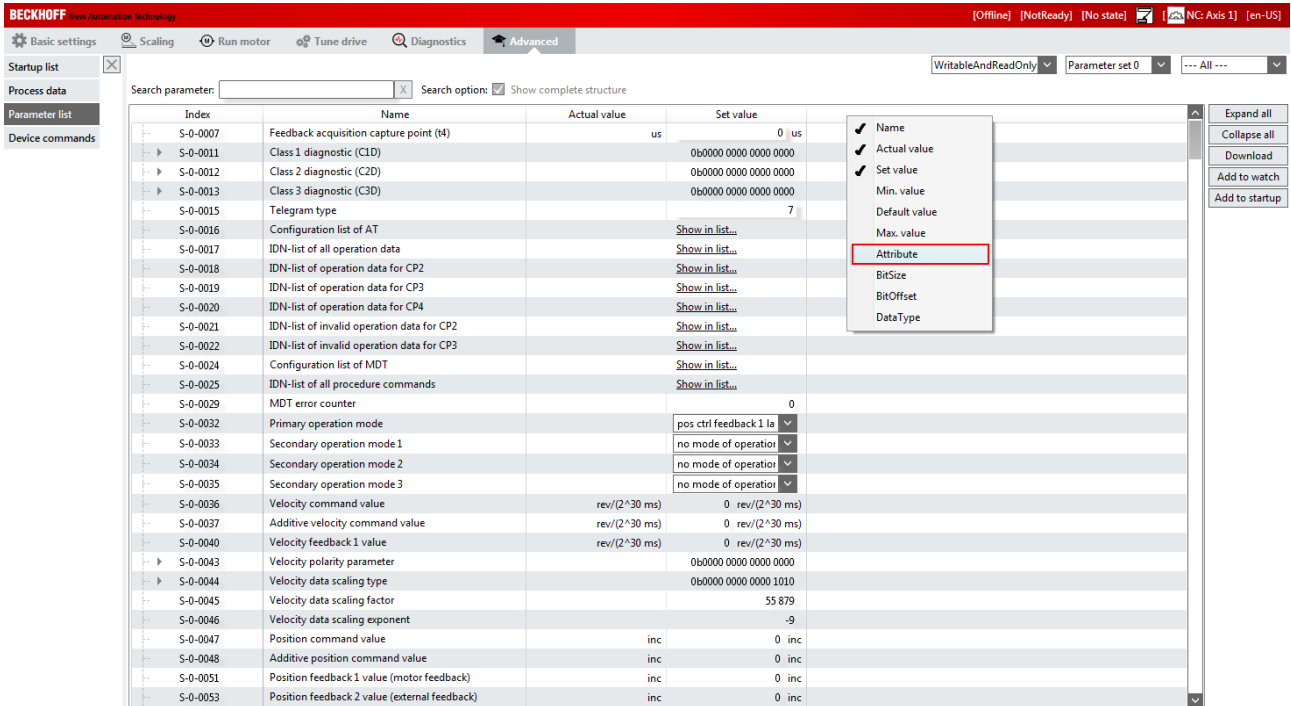
```
fbWrite      : FB_SoEWrite;
Idn          : WORD;
Write        : BOOL;
WriteValue   : UINT;
Password     : ST_SoE_String;
(* NoAxis *)
Axis        : AXIS_REF;

IF Write THEN
  Idn := S_0_IDNs + 33;
  fbWrite(
    Axis      := Axis,
    Idn       := Idn,
    Element   := 16#40,
    pSrcBuf   := ADR(WriteValue),
    BufLen    := SIZEOF(WriteValue),
    Password  := Password,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbWrite.Busy THEN
    fbWrite(Axis := Axis, Execute := FALSE);
    Write := FALSE;
  END_IF
END_IF
```

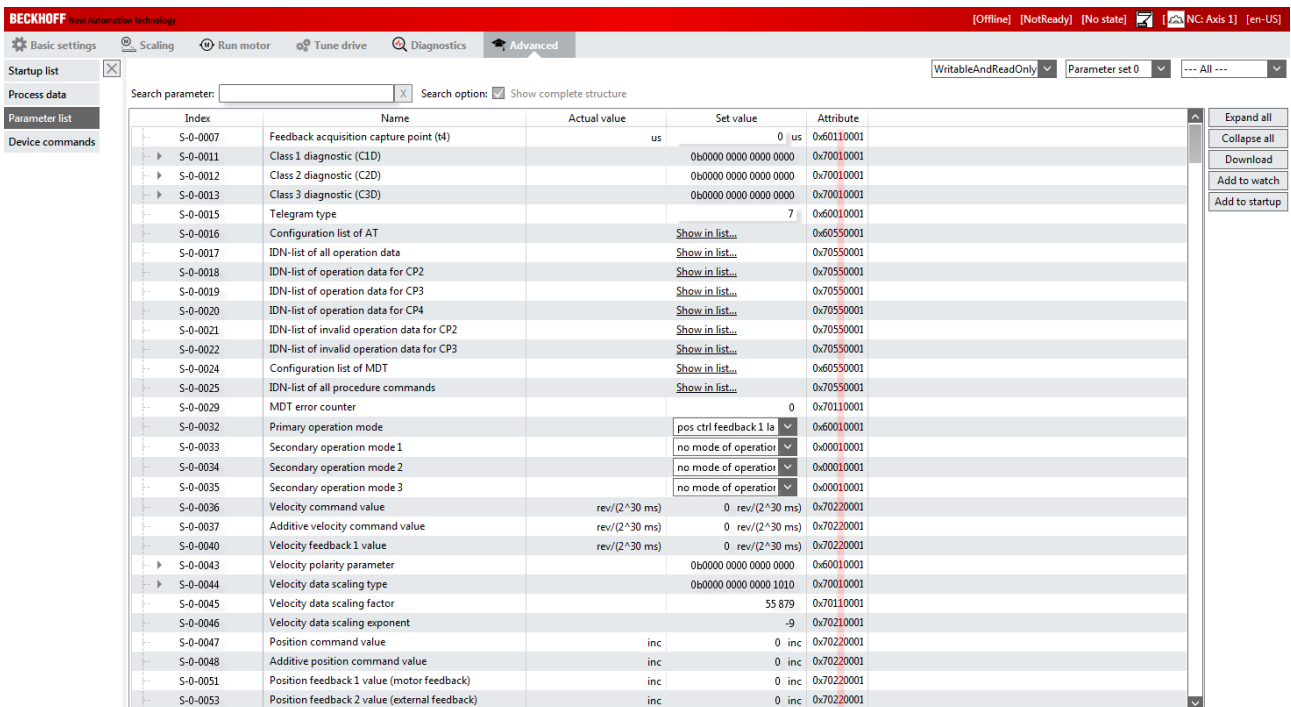
### Parameter access

Note that for some parameters, the additional items of information "ActualLength" & "MaxLength" are necessary for the parameter access.

In order to determine the parameters for which this is necessary, the attributes can be displayed in the parameter list of the Drive Manager via the context menu of the header.



If the fifth place from the right contains the value 4, 5, 6 or 7, then the additional items of information are also transmitted.



The structure to read out the serial number (S-0-0432) then looks like this:

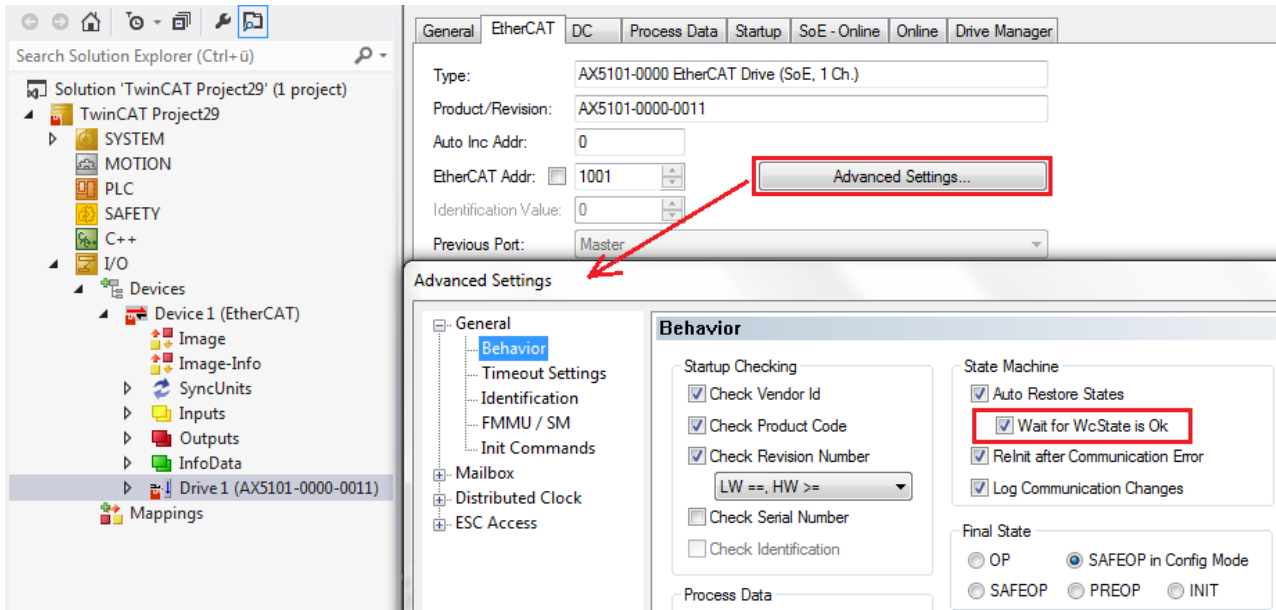
```

TYPE ST_SerialNumber :
STRUCT
    ActualLength : UINT;
    MaxLength : UINT;
    SerialNumber : T_MaxString;
END_STRUCT
END_TYPE
    
```

### 4.2.3 FB\_SoEReset



The drive (S-0-0099) can be reset with the function block FB\_SoEReset. In the case of multiple-channel devices if necessary, both channels will have to perform a reset. The timeout time must be 10 s, as the reset can take up to 10 s depending on the error. The flag "Wait For WcState is OK" must be enabled in the advanced EtherCAT settings for the AX5000.



An NC reset will not be performed. If an NC reset is necessary, it can be executed via the function block MC\_Reset from the Tc2\_MC2 PLC library.

#### Inputs

```
VAR_INPUT
    NetId : T_AmsNetID := '';
    Execute : BOOL;
    Timeout : TIME := T#10s;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```



Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
    Busy      : BOOL;
    Error     : BOOL;
    AdsErrId  : UINT;
    SercosErrId : UINT;
END_VAR
```

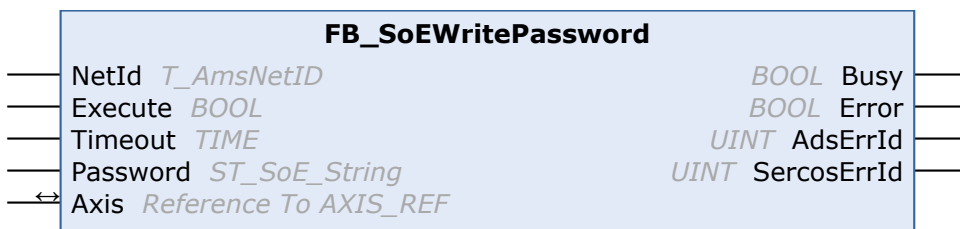
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

**Sample**

```
fbSoEReset : FB_SoEReset;
SoEReset   : BOOL;
(* NcAxis *)
Axis       : AXIS_REF;

IF SoEReset THEN
    fbSoEReset(
        Axis := Axis,
        Execute := TRUE,
    );
IF NOT fbSoEReset.Busy THEN
    fbSoEReset(Axis := Axis, Execute := FALSE);
    SoEReset := FALSE;
END_IF
END_IF
```

**4.2.4 FB\_SoEWritePassword**



With the FB\_SoEWritePassword function block (S-0-0267) the drive password can be set.

 **Inputs**

```
VAR_INPUT
    NetId      : T_AmsNetID := '';
    Execute    : BOOL;
    Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
    Password   : ST_SoE_String;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.
Password	ST_SoE_String	Password as a Sercos string.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

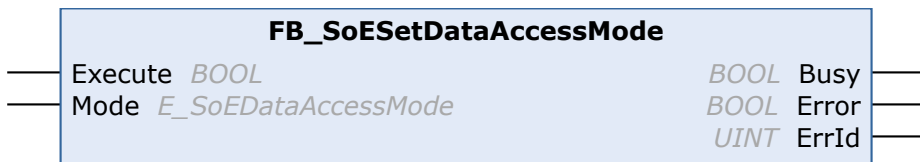
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

**Sample**

```
fbWritePassword : FB_SoEWritePassword;
WritePassword   : BOOL;
Password        : ST_SoE_String;
(* NoAxis *)
Axis            : AXIS_REF;

IF WritePassword THEN
  fbWritePassword(
    Axis      := Axis,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
    Password  := Password
  );
  IF NOT fbWritePassword.Busy THEN
    fbWritePassword(Axis := Axis, Execute := FALSE);
    WritePassword := FALSE;
  END_IF
END_IF
```

**4.2.5 FB\_SoESetDataAccessMode**



The SoE parameter access mode is set with the function block FB\_SoESetDataAccessMode. Attributes & Value are accessed in parallel by default. If this access mode is not supported by the device manufacturer, sequential access can be enforced. Generally, however, parallel access is faster.

**Inputs**

```
VAR_INPUT
  Execute : BOOL;
  Mode    : E_SoEDataAccessMode := E_SoEDataAccessMode.eSoEDataAccessMode_Parallel;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	E_SoEDataAccessMode [▶ 72]	Defines the access mode.

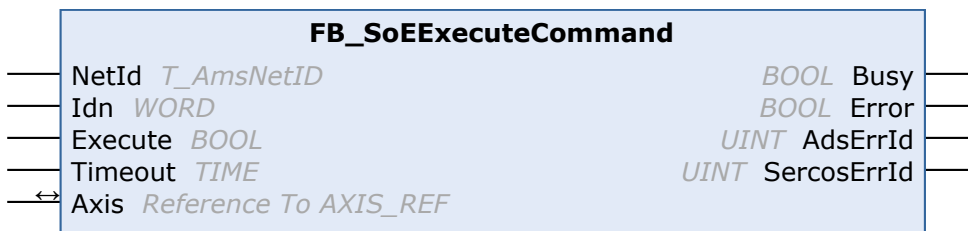
**Outputs**

```
VAR_OUTPUT
  Busy    : BOOL;
  Error   : BOOL;
  ErrId   : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrId	UINT	In the event of a set error output returns the ADS error code.

## 4.2.6 Function blocks for command

### 4.2.6.1 FB\_SoEExecuteCommand



With the FB\_SoEExecuteCommand function block a command can be executed.

**Inputs**

```
VAR_INPUT
  NetId : T_AmsNetID := '';
  Idn   : WORD;
  Execute : BOOL;
  Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Idn	WORD	Parameter number that FB_SoEExecuteCommand refers to, e.g. "P_0_IDNs + 160" for P-0-0160.

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

### Sample

```
fbExecuteCommand : FB_SoEExecuteCommand;
ExecuteCommand   : BOOL;
Idn              : WORD;
(* NcAxis *)
Axis            : AXIS_REF;

IF ExecuteCommand THEN
  Idn := P_0_IDNs + 160;
  fbExecuteCommand(
    Axis      := Axis,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
    Idn       := Idn,
  );
  IF NOT fbExecuteCommand.Busy THEN
    fbExecuteCommand(Axis := Axis, Execute := FALSE);
    ExecuteCommand := FALSE;
  END_IF
END_IF
```

### 4.2.6.2 FB\_SoEWriteCommandControl



With the FB\_SoEWriteCommandControl function block a command can be prepared, started or canceled.

#### Inputs

```
VAR_INPUT
  NetId      : T_AmsNetID := '';
  Idn        : WORD;
  CmdControl : E_SoE_CmdControl;
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Idn	WORD	Parameter number that FB_SoEWriteCommandControl refers to, e.g. "P_0_IDNs + 160" for P-0-0160.
CmdControl	E_SoE_CmdControl	Indicates whether a command should be prepared (eSoE_CmdControl_Set := 1), executed (eSoE_CmdControl_SetAndEnable := 3), or aborted (eSoE_CmdControl_Cancel := 0).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.

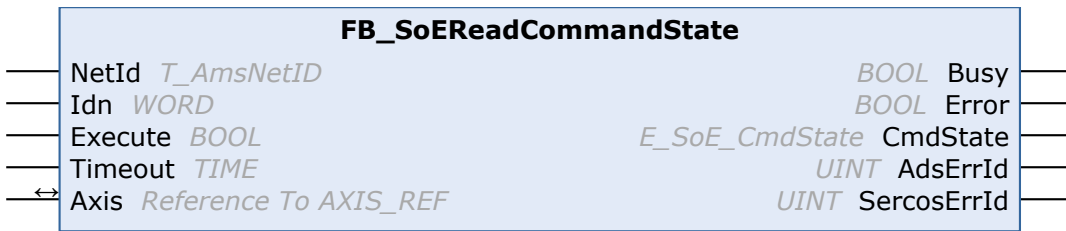
Name	Type	Description
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

**Sample**

```
fbWriteCommandControl : FB_SoEWriteCommandControl;
WriteCommandControl   : BOOL;
Idn                   : WORD;
CmdControl            : E_SoE_CmdControl;
(* NcAxis *)
Axis                  : AXIS_REF;

IF WriteCommandControl THEN
  Idn := P_0_IDNs + 160;
  fbWriteCommandControl(
    Axis      := Axis,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
    Idn       := Idn,
    CmdControl := CmdControl
  );
  IF NOT fbWriteCommandControl.Busy THEN
    fbWriteCommandControl(Axis := Axis, Execute := FALSE);
    WriteCommandControl := FALSE;
  END_IF
END_IF
```

**4.2.6.3 FB\_SoEReadCommandState**



With the FB\_SoEReadCommandState function block the command execution can be checked.

**Inputs**

```
VAR_INPUT
  NetId   : T_AmsNetID := '';
  Idn     : WORD;
  Execute : BOOL;
  Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Idn	WORD	Parameter number that FB_SoEReadCommandState_ByDriveRef refers to, e.g. "P_0_IDNs + 160" for P-0-0160
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

**Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  CmdState  : E_SoE_CmdState;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
CmdState	E_SoE_CmdState	Returns the command status: eSoE_CmdState_NotSet = 0 - kein Kommando aktiv  eSoE_CmdState_Set = 1 - Kommando gesetzt (vorbereitet) aber (noch) nicht ausgeführt  eSoE_CmdState_Executed = 2 - Kommando wurde ausgeführt  eSoE_CmdState_SetEnabledExecuted = 3 - Kommando gesetzt (vorbereitet) und ausgeführt  eSoE_CmdState_SetAndInterrupted = 5 - Kommando wurde gesetzt aber unterbrochen  eSoE_CmdState_SetEnabledNotExecuted = 7 - Kommandoausführung ist noch aktiv  eSoE_CmdState_Error = 15 - Fehler bei der Kommandoausführung, es wurde in den Fehlerstate gewechselt
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

**Sample**

```
fbReadCommandState : FB_SoEReadCommandState;
ReadCommandState   : BOOL;
Idn                 : WORD;
CmdState            : E_SoE_CmdState;
(* NcAxis *)
Axis                : AXIS_REF;

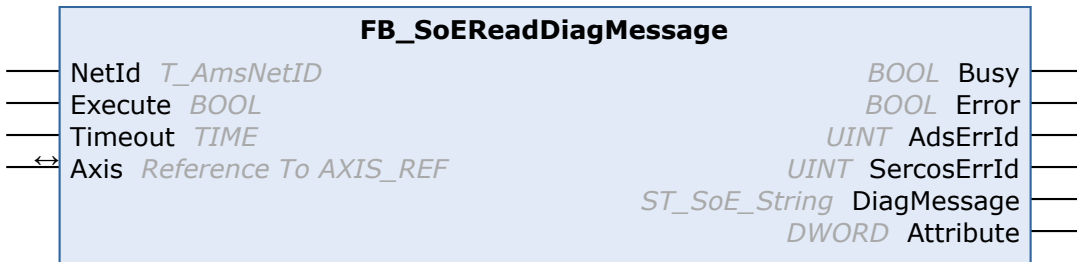
IF ReadCommandState THEN
```

```

Idn := P_0_IDNs + 160;
fbReadCommandState(
  Axis      := Axis,
  Execute   := TRUE,
  Timeout   := DEFAULT_ADS_TIMEOUT,
  Idn       := Idn,
  CmdState => CmdState
);
IF NOT fbReadCommandState.Busy THEN
  fbReadCommandState(Axis := Axis, Execute := FALSE);
  ReadCommandState := FALSE;
END_IF
END_IF
    
```

## 4.2.7 Function blocks for diagnostics

### 4.2.7.1 FB\_SoEReadDiagMessage



With the FB\_SoEReadDiagMessage function block the diagnosis message can be read out as a Sercos string (S-0-0095).

#### Inputs

```

VAR_INPUT
  NetId   : T_AmsNetID := '';
  Execute : BOOL;
  Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```

VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
    
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```

VAR_OUTPUT
  Busy       : BOOL;
  Error      : BOOL;
  AdsErrId   : UINT;
  SercosErrId : UINT;
    
```



```

    DiagMessage : ST_SoE_String;
    Attribute   : DWORD;
END_VAR

```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
DiagMessage	ST_SoE_String	Returns the diagnosis message.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

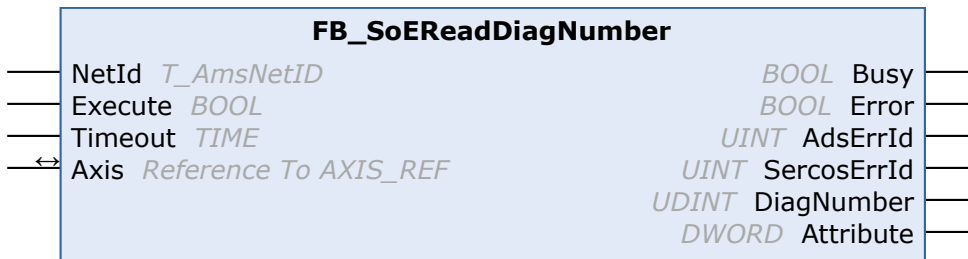
```

fbDiagMessage : FB_SoEReadDiagMessage;
bDiagMessage  : BOOL;
DiagMessage   : ST_SoE_String;
(* NcAxis *)
Axis          : AXIS_REF;

IF bDiagMessage THEN
  fbDiagMessage(
    Axis      := Axis,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
    DiagMessage => DiagMessage
  );
IF NOT fbDiagMessage.Busy THEN
  fbDiagMessage(Axis := Axis, Execute := FALSE);
  bDiagMessage := FALSE;
END_IF
END_IF

```

**4.2.7.2 FB\_SoEReadDiagNumber**



With the FB\_SoEReadDiagNumber function block the current diagnostic number can be read out as UDINT (S-0-0390).

**Inputs**

```

VAR_INPUT
  NetId   : T_AmsNetID := '';
  Execute : BOOL;
  Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.

Name	Type	Description
Timeout	TIME	Maximum time allowed for the execution of the function block.

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
  DiagNumber : UDINT;
  Attribute : DWORD;
END_VAR
```

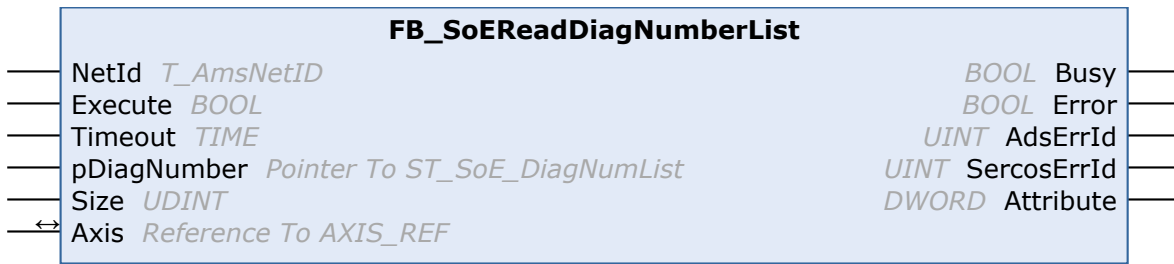
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
DiagNumber	UDINT	Returns the current diagnostic number.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

### Sample

```
fbDiagNumber : FB_SoEReadDiagNumber;
bDiagNumber  : BOOL;
DiagNumber   : UDINT;
(* NoAxis *)
Axis         : AXIS_REF;

IF bDiagNumber THEN
  fbDiagNumber(
    Axis      := Axis,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
    DiagNumber => DiagNumber
  );
IF NOT fbDiagNumber.Busy THEN
  fbDiagNumber(Axis := Axis, Execute := FALSE);
  bDiagNumber := FALSE;
END_IF
END_IF
```

### 4.2.7.3 FB\_SoEReadDiagNumberList



With the FB\_SoEReadDiagNumberList function block a history of the diagnosis numbers can be read out as a list (S-0-0375).

#### Inputs

```
VAR_INPUT
  NetId      : T_AmsNetID := '';
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
  pDiagNumber : POINTER TO ST_SoE_DiagNumList;
  Size       : UDINT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.
pDiagNumber	POINTER TO ST_SoE_DiagNumList	Pointer to the list of the last max. 30 error numbers. The list consists of the current and maximum number of bytes in the list as well as the 30 list items.
Size	UDINT	Size of the list in bytes (as Sizeof())

#### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```
VAR_OUTPUT
  busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
  Attribute : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.

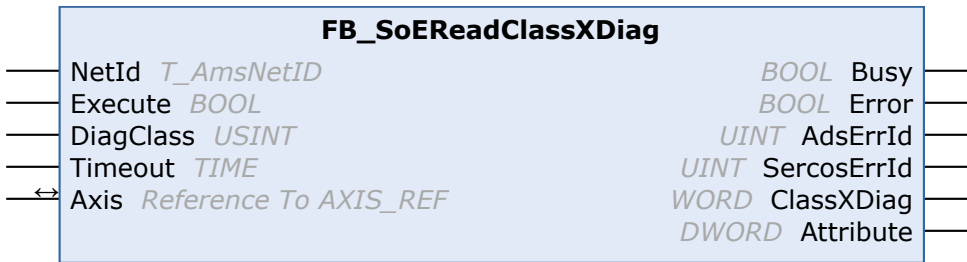
Name	Type	Description
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

```
fbDiagNumberList : FB_SoEReadDiagNumberList;
DiagNumberList   : BOOL;
stDiagNumberList : ST_SoE_DiagNumList;
(* NcAxis *)
Axis              : AXIS_REF;

IF DiagNumberList THEN
  fbDiagNumberList(
    Axis       := Axis,
    Execute    := TRUE,
    Timeout    := DEFAULT_ADS_TIMEOUT,
    pDiagNumber := ADR(stDiagNumberList),
    Size       := SIZEOF(stDiagNumberList),
  );
  IF NOT fbDiagNumberList.Busy THEN
    fbDiagNumberList(Axis := Axis, Execute := FALSE);
    DiagNumberList := FALSE;
  END_IF
END_IF
```

**4.2.7.4 FB\_SoEReadClassXDiag**



With the function block FB\_SoEReadClassXDiag, the current Class 1 diagnosis (S-0-0011) ... Class 3 diagnosis (S-0-0013) can be read out as WORD. There is the conversion function F\_ConvWordToSTAX5000C1D for the evaluation of the Class 1 diagnosis as a structure ST\_AX5000\_C1D, (see TwinCAT 3 PLC Lib Tc2\_Drive documentation).

**Inputs**

```
VAR_INPUT
  NetId      : T_AmsNetId := '';
  Execute    : BOOL;
  DiagClass  : USINT:= 1; (* 1: C1D (S-0-0011) is default, 2: C2D (S-0-0012), 3: C3D (S-0-0013) *)
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetId	String, which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
DiagClass	USINT	Specifies which diagnosis should be read. The diagnostics parameters may vary from vendor to vendor. All diagnostics parameters (C1D ... C3D) or all bits are not always implemented in them. 1: Error: Class 1 Diag (S-0-0011) 2: Warnings: Class 2 Diag (S-0-0012)

Name	Type	Description
		3: Information: Class 3 Diag (S-0-0013)
Timeout	TIME	Maximum time allowed for the execution of the function block.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
  ClassXDiag : WORD;
  Attribute : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
ClassXDiag	WORD	Returns the current Class X diagnosis.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

```
fbClassXDiag : FB_SoEReadClassXDiag;
bClassXDiag  : BOOL;
DiagClass    : USINT := 1;
Class1Diag   : WORD;
stAX5000C1D  : ST_AX5000_C1D;
Class2Diag   : WORD;
(* NcAxis *)
Axis         : AXIS_REF;

IF bClassXDiag THEN
  fbClassXDiag(
    Axis      := Axis,
    Execute   := TRUE,
    DiagClass := DiagClass,
    Timeout   := DEFAULT_ADS_TIMEOUT
  );
IF NOT fbClassXDiag.Busy THEN
  fbClassXDiag(Axis := Axis, Execute := FALSE);
  bClassXDiag := FALSE;
  CASE fbClassXDiag.DiagClass OF
  1:
    Class1Diag := fbClassXDiag.ClassXDiag;
    stAX5000C1D := F_ConvWordToSTAX5000C1D(Class1Diag);
  2:
    Class2Diag := fbClassXDiag.ClassXDiag;
```

```
END_CASE
END_IF
END_IF
```

## 4.2.8 Function blocks for determining current values

### 4.2.8.1 FB\_SoEReadAmplifierTemperature



With the FB\_SoEReadAmplifierTemperature function block the temperature of the drive (S-0-0384) can be read.

#### Inputs

```
VAR_INPUT
    NetId    : T_AmsNetID := '';
    Execute  : BOOL;
    Timeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```
VAR_OUTPUT
    Busy           : BOOL;
    Error          : BOOL;
    AdsErrId       : UINT;
    SercosErrId    : UINT;
    AmplifierTemperature : REAL;
    Attribute      : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.

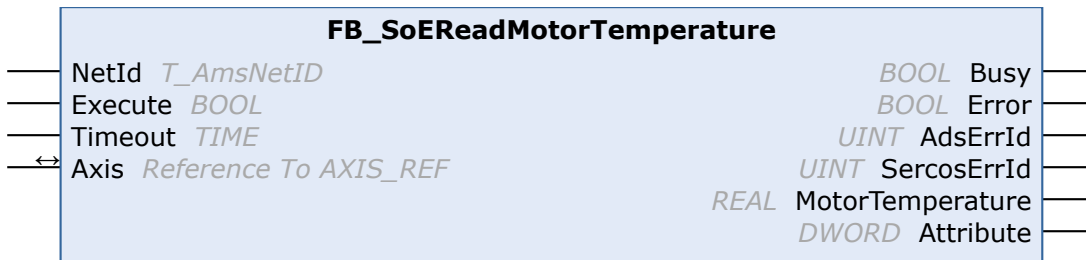
Name	Type	Description
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
AmplifierTemperature	REAL	Returns the drive temperature (e.g. 26.2 corresponds to 26.2 °C).
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

```
fbReadAmplifierTemp : FB_SoEReadAmplifierTemperature;
ReadAmplifierTemp : BOOL;
AmplifierTemperature : REAL;
(* NcAxis *)
Axis : AXIS_REF;

IF ReadAmplifierTemp THEN
  fbReadAmplifierTemp(
    Axis := Axis,
    Execute := TRUE,
    Timeout := DEFAULT_ADS_TIMEOUT,
    AmplifierTemperature => AmplifierTemperature
  );
  IF NOT fbReadAmplifierTemp.Busy THEN
    fbReadAmplifierTemp(Axis := Axis, Execute := FALSE);
    ReadAmplifierTemp := FALSE;
  END_IF
END_IF
```

**4.2.8.2 FB\_SoEReadMotorTemperature**



With the function block FB\_SoEReadMotorTemperature the temperature of the motor (S-0-0383) can be read. If the motor does not contain a temperature sensor, this is 0.0, i.e. 0.0 °C.

**Inputs**

```
VAR_INPUT
  NetId : T_AmsNetID := '';
  Execute : BOOL;
  Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

**Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**🔌 Outputs**

```
VAR_OUTPUT
    Busy          : BOOL;
    Error         : BOOL;
    AdsErrId      : UINT;
    SercosErrId   : UINT;
    MotorTemperature : REAL;
    Attribute     : DWORD;
END_VAR
```

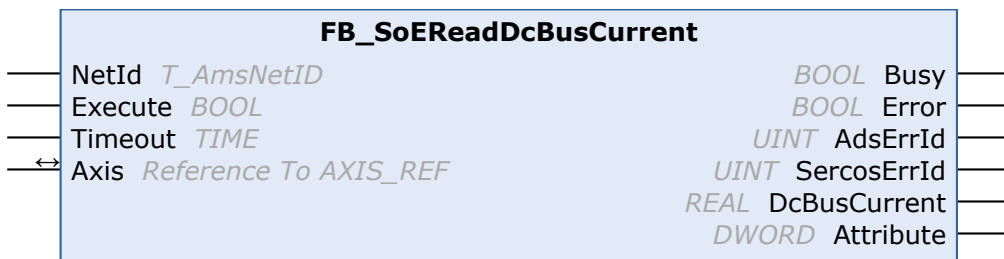
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
MotorTemperature	REAL	Returns the motor temperature (e.g. 30.5 corresponds to 30.5 °C). If the motor does not contain a temperature sensor, this is 0.0, i.e. 0.0 °C.
Attributes	DWORD	Returns the attributes of the Sercos parameter.

**Sample**

```
fbReadMotorTemp : FB_SoEReadMotorTemperature;
ReadMotorTemp   : BOOL;
MotorTemperature : REAL;
(* NcAxis *)
Axis            : AXIS_REF;

IF ReadMotorTemp THEN
    fbReadMotorTemp(
        Axis           := Axis,
        Execute        := TRUE,
        Timeout        := DEFAULT_ADS_TIMEOUT,
        MotorTemperature => MotorTemperature
    );
    IF NOT fbReadMotorTemp.Busy THEN
        fbReadMotorTemp(Axis := Axis, Execute := FALSE);
        ReadMotorTemp := FALSE;
    END_IF
END_IF
```

**4.2.8.3 FB\_SoEReadDcBusCurrent**



With the FB\_SoEAX5000ReadDcBusCurrent function block the DC-Bus current (S-0-0381) can be read.



 **Inputs**

```
VAR_INPUT
  NetId      : T_AmsNetID := '';
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy       : BOOL;
  Error      : BOOL;
  AdsErrId   : UINT;
  SercosErrId : UINT;
  DcBusCurrent : REAL;
  Attribute  : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
DcBusCurrent	REAL	Returns the DC bus current (e.g. 2,040 equals 2,040 A).
Attributes	DWORD	Returns the attributes of the Sercos parameter.

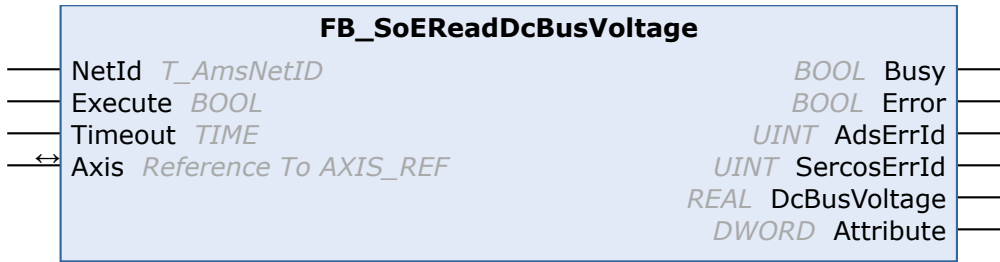
**Sample**

```
fbReadDcBusCurrent : FB_SoEReadDcBusCurrent;
ReadDcBusCurrent   : BOOL;
DcBusCurrent       : REAL;
(* NcAxis *)
Axis               : AXIS_REF;

IF ReadDcBusCurrent THEN
  fbReadDcBusCurrent (
    Axis       := Axis,
    Execute    := TRUE,
    Timeout    := DEFAULT_ADS_TIMEOUT,
    DcBusCurrent => DcBusCurrent
  );
IF NOT fbReadDcBusCurrent.Busy THEN
```

```
fbReadDcBusCurrent (Axis := Axis, Execute := FALSE);
ReadDcBusCurrent := FALSE;
END_IF
END_IF
```

### 4.2.8.4 FB\_SoEReadDcBusVoltage



With the FB\_SoEReadDcBusVoltage function block the Dc-Bus voltage of the drive (S-0-0380) can be read.

#### Inputs

```
VAR_INPUT
    NetId : T_AmsNetID := '';
    Execute : BOOL;
    Timeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```
VAR_OUTPUT
    Busy : BOOL;
    Error : BOOL;
    AdsErrId : UINT;
    SercosErrId : UINT;
    DcBusVoltage : REAL;
    Attribute : DWORD;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.

Name	Type	Description
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
DcBusVoltage	REAL	Returns the DC-Bus voltage (e.g. 294.0 corresponds to 294.0 V).
Attributes	DWORD	Returns the attributes of the Sercos parameter.

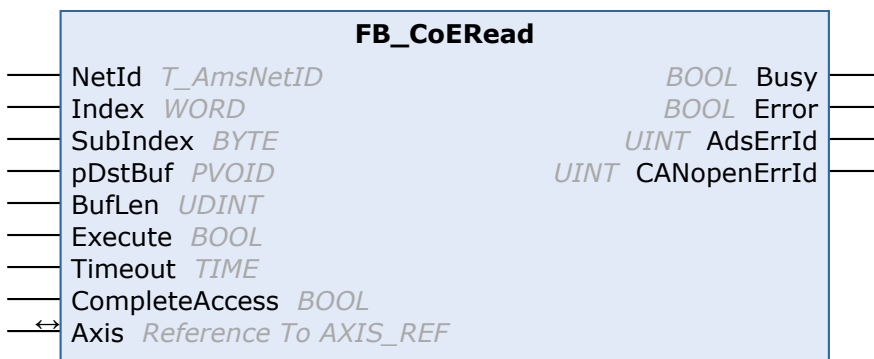
**Sample**

```
fbReadDcBusVoltage : FB_SoEReadDcBusVoltage;
ReadDcBusVoltage   : BOOL;
DcBusVoltage       : REAL;
(* NcAxis *)
Axis               : AXIS_REF;

IF ReadDcBusVoltage THEN
  fbReadDcBusVoltage(
    Axis       := Axis,
    Execute    := TRUE,
    Timeout    := DEFAULT_ADS_TIMEOUT,
    DcBusVoltage => DcBusVoltage
  );
IF NOT fbReadDcBusVoltage.Busy THEN
  fbReadDcBusVoltage(Axis := Axis, Execute := FALSE);
  ReadDcBusVoltage := FALSE;
END_IF
END_IF
```

## 4.3 General CoE

### 4.3.1 FB\_CoERead



The function block FB\_CoERead allows data to be read from an object directory of an EtherCAT slave through an SDO (Service Data Object) access. This requires the slave to have a mailbox and to support the CoE (CANopen over EtherCAT) protocol. With the help of the SubIndex and Index parameters a selection is made as to which object should be read out. Via CompleteAccess := TRUE the parameter can be read with sub-elements.

 **Inputs**

```
VAR_INPUT
  NetId       : T_AmsNetID; (*netID of PC with NC*)
  Index       : WORD; (*CoE object index*)
  SubIndex    : BYTE; (*CoE sub index*)
  pDstBuf     : PVOID; (*Contains the address of the buffer for the received data*)
  BufLen      : UDINT; (*Contains the max. number of bytes to be received*)
  Execute     : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
  Timeout     : TIME := DEFAULT_ADS_TIMEOUT; (*States the time before the function is
  cancelled.*)
  CompleteAccess : BOOL; (*Function block reads the complete object with all sub index*)
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Index	WORD	Index of the object that is to be read.
SubIndex	BYTE	Subindex of the object that is to be read.
pDstBuf	PVOID	Address (pointer) to the receive buffer
BufLen	UDINT	Maximum available buffer size (in bytes) for the data to be read
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.
CompleteAccess	BOOL	Via Complete Access the complete object can be accessed at once.



Index and SubIndex can be taken from the corresponding drive documentation. For the AX8000 from Beckhoff see [AX8000 object description](#).

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  CANopenErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the event of a set error output returns the ADS error code.
CANopenErrId	UINT	In the event of a set error output returns the CANopen error code.

### Example for implementation in ST:

```
PROGRAM MAIN
VAR
  fbCoERead      : FB_CoERead;
  NetId          : T_AmsNetID := '';
  Index          : WORD := 16#1018;
  SubIndex       : BYTE := 1;
  Execute        : BOOL := TRUE;
  Timeout        : TIME := T#5S;
  CompleteAccess : BOOL := TRUE;
  Axis           : AXIS_REF;
  vendorId      : UDINT := 2;
  Error         : BOOL;
```

```

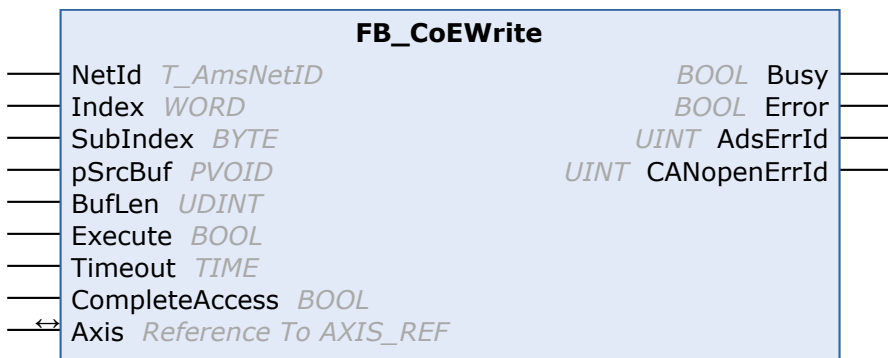
    AdsErrId      : UDINT;
    CANopenErrId  : UDINT;
END_VAR

fbCoERead(
    NetId      := NetId,
    Index      := Index,
    SubIndex   := SubIndex,
    pDstBuf    := ADR(vendorId),
    BufLen     := SIZEOF(vendorId),
    Execute    := Execute,
    Timeout    := Timeout,
    CompleteAccess := CompleteAccess,
    Axis      := Axis
);

IF NOT fbCoERead.Busy THEN
    Error      :=fbCoERead.Error;
    AdsErrId   :=fbCoERead.AdsErrId;
    CANopenErrId :=fbCoERead.CANopenErrId;
    Execute    := FALSE;
    fbCoERead(Execute := Execute, Axis := Axis);
END_IF

```

### 4.3.2 FB\_CoEWrite



With the function block FB\_CoEWrite, an object from the object directory of an EtherCAT slave can be written via an SDO (Service Data Object) download. This requires the slave to have a mailbox and to support the CoE (CANopen over EtherCAT) protocol. With the help of the SubIndex and Index parameters a selection is made as to which object should be written. Via CompleteAccess := TRUE the parameter can be written with sub-elements.

#### Inputs

```

VAR_INPUT
    NetId      : T_AmsNetID; (*netID of PC with NC*)
    Index      : WORD; (*CoE object index*)
    SubIndex   : BYTE; (*CoE sub index*)
    pSrcBuf    : PVOID; (*Contains the address of the buffer containing the data to be send*)
    BufLen     : UDINT; (*Contains the max. number of bytes to be received*)
    Execute    : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
    Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
    (*States the time before the function is cancelled.*)
    CompleteAccess : BOOL; (*Function block reads the complete object with all sub index*)
END_VAR

```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Index	WORD	Index of the object that is supposed to be written.
SubIndex	BYTE	Subindex of the object that is supposed to be written.
pDstBuf	PVOID	Address (pointer) to the transmit buffer
BufLen	UDINT	Amount of data to be sent in bytes
Execute	BOOL	The function block is enabled via a positive edge at this input.

Name	Type	Description
Timeout	TIME	Maximum time allowed for the execution of the function block.
CompleteAccess	BOOL	The complete object can be accessed all at once via Complete Access.



Index and SubIndex can be taken from the corresponding drive documentation. For the AX8000 from Beckhoff see [AX8000 object description](#).

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  CANopenErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the event of a set error output returns the ADS error code.
CANopenErrId	UINT	In the event of a set error output returns the CANopen error code.

### Example for implementation in ST:

```
PROGRAM MAIN
VAR
  fbCoEWrite      : FB_CoEWrite;
  NetId           : T_AmsNetID := '';
  Index          : WORD := 16#1018;
  SubIndex       : BYTE := 1;
  Execute        : BOOL := TRUE;
  Timeout        : TIME := T#5S;
  CompleteAccess : BOOL := TRUE;
  Axis           : AXIS_REF;
  vendorId       : UDINT := 2;
  Error          : BOOL;
  AdsErrId       : UDINT;
  CANopenErrId   : UDINT;END_VAR

fbCoEWrite(
  NetId      := NetId,
  Index      := Index,
  SubIndex   := SubIndex,
  pSrcBuf    := ADR(vendorId),
  BufLen     := SIZEOF(vendorId),
  Execute    := Execute,
  Timeout    := Timeout,
  CompleteAccess := CompleteAccess,
  Axis := Axis
```

```
);
IF NOT fbCoEWrite.Busy THEN
  Error      := fbCoEWrite.Error;
  AdsErrId   := fbCoEWrite.AdsErrId;
  CANopenErrId :=fbCoEWrite.CANopenErrId;
  Execute    := FALSE;
  fbCoEWrite(Execute := Execute, Axis := Axis);
END_IF
```

### 4.3.3 FB\_CoEExecuteCommand



With the function block FB\_CoEExecuteCommand a command can be executed.

#### Inputs

```
VAR_INPUT
  NetId      : T_AmsNetID := '';
  Index      : WORD;
  pSrcBuf    : PVOID;
  SrcBufLen  : UDINT;
  pDstBuf    : PVOID;
  DstBufLen  : UDINT;
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Index	WORD	CoE object index that FB_CoEExecuteCommand refers to.
pSrcBuf	PVOID	Address of the structure of the data to be sent
SrcBufLen	UDINT	Size in bytes of the structure of the data to be sent
pDstBuf	PVOID	Address of the structure of the data to be received
DstBufLen	UDINT	Size in bytes of the structure of the data to be received
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

#### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

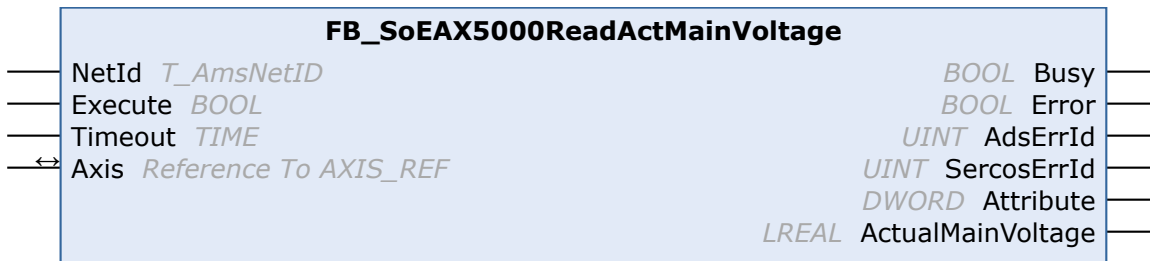
 **Outputs**

```
VAR_OUTPUT
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    Status    : _E_CoECommandStatus;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorId	UDINT	In the event of a set error output returns the ADS error code.
Status	_E_CoECommandStatus	Status of the command execution

## 4.4 AX5000 SoE

### 4.4.1 FB\_SoEAX5000ReadActMainVoltage



With the FB\_SoEAX5000ReadActMainVoltage function block the current peak value of the mains voltage of the AX5000 (P-0-0200) can be read.

 **Inputs**

```
VAR_INPUT
    NetId    : T_AmsNetID := '';
    Execute  : BOOL;
    Timeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String which contains the AMS Network ID of the PC (type: T_AmsNetId).
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.

 **Inputs/outputs**

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.



**🔌 Outputs**

```
VAR_OUTPUT
  Busy          : BOOL;
  Error         : BOOL;
  AdsErrId     : UINT;
  SercosErrId  : UINT;
  Attribute    : DWORD;
  ActualMainVoltage : LREAL;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.
Attributes	DWORD	Returns the attributes of the Sercos parameter.
ActualMainVoltage	LREAL	Returns the peak value of the current mains voltage of the AX5000 (e.g. 303.0 corresponds to 303.0 V).

**Sample**

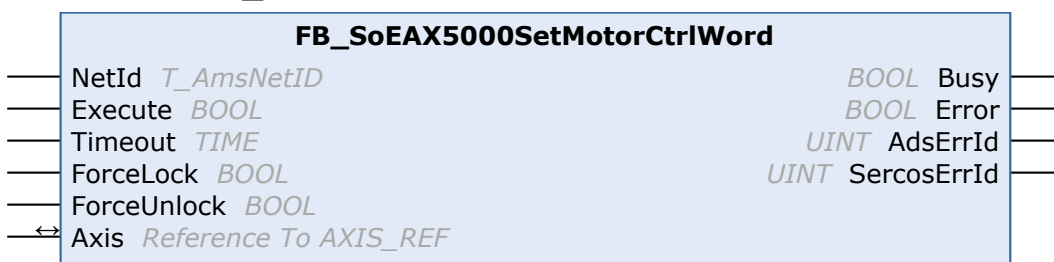
```
fbReadActMainVoltage : FB_SoEAX5000ReadActMainVoltage;
ReadActMainVoltage   : BOOL;
ActualMainVoltage    : REAL;
(* NoAxis *)
Axis                 : AXIS_REF;

IF ReadActMainVoltage THEN
  fbReadActMainVoltage(
    Axis       := Axis,
    Execute    := TRUE,
    Timeout    := DEFAULT_ADS_TIMEOUT,
    ActualMainVoltage => ActualMainVoltage
  );
IF NOT fbReadActMainVoltage.Busy THEN
  fbReadActMainVoltage(Axis := Axis, Execute := FALSE);
  ReadActMainVoltage := FALSE;
END_IF
END_IF
```

**Requirements**

Development environment	Target platform type	PLC library	AX5000 Firmware
TwinCAT 3	PC or CX (x86 or x64)	Tc2_MC2_Drive	≥v1.07 b0001

**4.4.2 FB\_SoEAX5000SetMotorCtrlWord**



With the function block **FB\_SoEAX5000SetMotorCtrlWord** the ForceLock bit (Bit 0) or the ForceUnlock bit can be set in the Motor Control Word (P-0-0096) to activate or release the brake. Normally the brake is automatically controlled via the Enable of the drive.

With the ForceLock bit, the brake can be activated independently from the Enable, with the ForceUnlock bit, the brake can be released independently from the Enable. In the case of simultaneously set ForceLock and ForceUnlock, ForceLock (Brake activated) has the higher priority.

### Inputs

```
VAR_INPUT
  NetId      : T_AmsNetId := '';
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
  ForceLock  : BOOL;
  ForceUnlock : BOOL;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	String containing the AMS network ID of the PC.
Execute	BOOL	The function block is enabled via a positive edge at this input.
Timeout	TIME	Maximum time allowed for the execution of the function block.
ForceLock	BOOL	Activates the brake independently of the enable.
ForceUnlock	BOOL	Releases the brake independently of the enable.

### Inputs/outputs

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

### Outputs

```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  AdsErrId  : UINT;
  SercosErrId : UINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
AdsErrId	UINT	In the case of a set Error output returns the ADS error code of the last executed command.
SercosErrId	UINT	In the case of a set Error output returns the Sercos error of the last executed command.

### Sample

```
fbSetMotorCtrlWord : FB_SoEAX5000SetMotorCtrlWord;
SetMotorCtrlWord   : BOOL;
ForceLock          : BOOL;
ForceUnlock        : BOOL;
(* NcAxis *)
Axis               : AXIS_REF;

IF SetMotorCtrlWord THEN
  fbSetMotorCtrlWord(
    Axis := Axis,
```

```

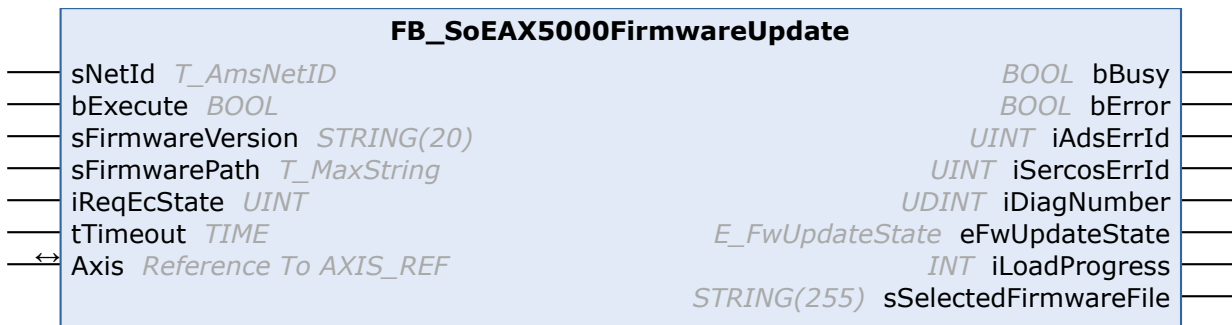
Execute      := TRUE,
Timeout     := DEFAULT_ADS_TIMEOUT,
ForceLock   := ForceLock,
ForceUnlock := ForceUnlock
);
IF NOT fbSetMotorCtrlWord.Busy THEN
  fbSetMotorCtrlWord(Axis := Axis, Execute := FALSE);
  SetMotorCtrlWord := FALSE;
END_IF
END_IF

```

**Requirements**

Development environment	Target platform type	PLC library	AX5000 Firmware
TwinCAT 3	PC or CX (x86 or x64)	Tc2_MC2_Drive	≥v1.07 b0001

**4.4.3 FB\_SoEAX5000FirmwareUpdate**



With the FB\_SoEAX5000FirmwareUpdate function block the Firmware of the AX5000 can be checked and automatically changed to a given version (Revision and Build) or to the current Build of the configured revision.

For the update:

- The configured slave type is determined, e.g. AX5103-0000-0010.
- The current slave is determined with the predefined slave address, e.g. AX5103-0000-0009.
- The current slave firmware is determined, e.g. v1.05\_b0009.
- A comparison of the configuration and the found slave regarding number of channels, current, revision and firmware is made.
- The name of the required firmware file is determined and a search for the file performed.
- The firmware update is executed (if necessary).
- The current slave with the predefined slave address is determined again.
- The slave is switched to the predefined EtherCAT state.

A successful update ends with eFwUpdateState = eFwU\_FwUpdateDone.

If the update is not required, this is signaled via eFwUpdateState = eFwU\_NoFwUpdateRequired.

The firmware update takes place via the specified channel (A=0 or B=1) from stDriveRef. In the case of two-channel devices only one of the two channels can be used. The other channel signals eFwUpdateState = eFwU\_UpdateViaOtherChannelActive or eFwUpdateState = eFwU\_UpdateViaOtherChannel.

During the firmware update (eFwUpdateState = eFwU\_FwUpdateInProgress), iLoadProgress signals the progress in percent.

**NOTICE**

**Faulty update due to interruptions**

Interruptions during the update may result in it not being executed or executed incorrectly. Afterwards, the terminal may no longer be usable without the appropriate firmware.

The rules during the update are:

- The PLC and TwinCAT must not be stopped.
- The EtherCAT connection must not be interrupted.
- The AX5000 must not be switched off.

**Inputs**

```
VAR_INPUT
  sNetId          : T_AmsNetID;
  bExecute        : BOOL;
  sFirmwareVersion : STRING(20); (* version string vx.yy_bnnnn, e.g. "v1.05_b0009" for v1.05 Build 0009*)
  sFirmwarePath   : T_MaxString; (* drive:\path, e.g. "C:\TwinCAT\Io\TcDriveManager\FirmwarePool" *)
  iReqEcState     : UINT := EC_DEVICE_STATE_OP;
  tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Type	Description
NetId	T_AmsNetID	AMS-NetID of the controller (IPC)
bExecute	BOOL	The function block is activated by a positive edge at this input.
sFirmwareVersion	STRING(20)	<p>Specifies the desired firmware version in the form of vx.yy_bnnnn, e.g. "v1.05_b0009" for Version v1.05 Build 0009.</p> <p>Release builds:</p> <ul style="list-style-type: none"> <li>• "v1.05_b0009" for a specific build, e.g. v1.05 Build 0009</li> <li>• "v1.05_b00??" latest build of a specified version, e.g. v1.05</li> <li>• "v1.??_b00??" latest build of a specified main version, e.g. v1</li> <li>• "v?.??_b00??" latest build of the latest version</li> <li>• " " latest build of the latest version</li> </ul> <p>Custom firmware builds:</p> <ul style="list-style-type: none"> <li>• "v1.05_b1009" for a specific build, e.g. v1.05 Build 0009</li> <li>• "v1.05_b10??" latest build of a specified version, e.g. v1.05</li> <li>• "v1.??_b10??" latest build of a specified main version, e.g. v1</li> <li>• "v?.??_b10??" latest build of the latest version</li> </ul> <p>...</p> <ul style="list-style-type: none"> <li>• "v1.05_b8909" for a specific build, e.g. v1.05 Build 8909</li> <li>• "v1.05_b89??" latest build of a specified version, e.g. v1.05</li> <li>• "v1.??_b89??" latest build of a specified main version, e.g. v1</li> <li>• "v?.??_b89??" latest build of the latest version</li> </ul> <p>Debug builds:</p>

Name	Type	Description
		<ul style="list-style-type: none"> <li>"v1.05_b9009" for a specific build, e.g. v1.05 Build 9009</li> <li>"v1.05_b90??" latest build of a specified version, e.g. v1.05</li> <li>"v1.??_b90??" latest build of a specified main version, e.g. v1</li> <li>"v?._?_b90??" latest build of the latest version</li> </ul>
sFirmwarePath	T_MaxString	Specifies the path for the firmware pool in which the firmware files are located, e.g. C:\TwinCAT\Io\TcDriveManager\FirmwarePool.
iReqEcState	UINT	Desired EtherCAT state after the update (only if an update is actually being executed). The statuses are defined as global constants in PLC Lib Tc2_EtherCAT.
tTimeout	TIME	Since the firmware update for large EtherCAT networks can take longer, only the timeout for individual internal ADS instances is specified here.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iAdsErrId      : UINT;
  iSercosErrId   : UINT;
  iDiagNumber    : UDINT;
  eFwUpdateState : E_FwUpdateState;
  iLoadProgress  : INT;
  sSelectedFirmwareFile : STRING(MAX_STRING_LENGTH); (* found firmware file, e.g. "AX5yxx_xxxx_0010_v1_05_b0009.efw" *)
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
bError	BOOL	This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId	UINT	Returns the ADS error code of the last executed command when the bError output is set.
iSercosErrId	UINT	In the case of a set bError output returns the Sercos error of the last executed command.
iDiagNumber	UDINT	In the case of a set bError output returns the drive error of the last executed firmware update.
eFwUpdateState	E_FwUpdateState	Returns the status of the firmware update.
iLoadProgress	INT	Returns the progress of the actual firmware update as a percentage.
sSelectedFirmwareFile	STRING(MAX_STRING_LENGTH)	Displays the name of the firmware file being searched for.

## Sample

```

VAR CONSTANT
  iNumOfDrives : INT := 2;
END_VAR

VAR
  fbFirmwareUpdate      : ARRAY [1..iNumOfDrives] OF FB_SoEAX5000FirmwareUpdate;
  Axes                  : ARRAY [1..iNumOfDrives] OF AXIS_REF;
  sFirmwareVersion      : ARRAY [1..iNumOfDrives] OF STRING(20) (* := 2('v1.04_b0002')*);
  eFwUpdateState        : ARRAY [1..iNumOfDrives] OF E_FwUpdateState;
  sSelectedFirmwareFile : ARRAY [1..iNumOfDrives] OF STRING(MAX_STRING_LENGTH);
  iUpdateState          : INT;
  bExecute              : BOOL;
  sNetIdIPC             : T_AmsNetId := '';
  sFirmwarePath         : T_MaxString := 'C:\TwinCAT\Io\TcDriveManager\FirmwarePool';
  I                     : INT;
  bAnyBusy              : BOOL;
  bAnyError             : BOOL;
END_VAR

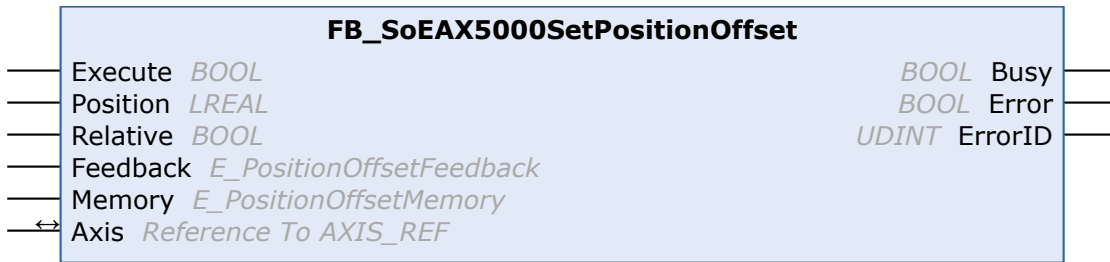
CASE iUpdateState OF
  0:
    IF bExecute THEN
      iUpdateState := 1;
    END_IF
  1:
    FOR I := 1 TO iNumOfDrives DO
      fbFirmwareUpdate[I](
        Axis      := Axes[I],
        bExecute   := TRUE,
        tTimeout   := T#15s,
        sFirmwareVersion := sFirmwareVersion[I],
        sFirmwarePath := sFirmwarePath,
        sNetId     := sNetIdIPC,
        iReqEcState := EC_DEVICE_STATE_OP,
        eFwUpdateState => eFwUpdateState[I],
      );
    END_FOR
    iUpdateState := 2;
  2:
    bAnyBusy      := FALSE;
    bAnyError     := FALSE;
    FOR I := 1 TO iNumOfDrives DO
      fbFirmwareUpdate[I](
        Axis      := Axes[I],
        eFwUpdateState => eFwUpdateState[I],
        sSelectedFirmwareFile => sSelectedFirmwareFile[I],
      );
      IF NOT fbFirmwareUpdate[I].bBusy THEN
        fbFirmwareUpdate[I](bExecute := FALSE, Axis := Axes[I]);
        IF fbFirmwareUpdate[I].bError THEN
          bAnyError := TRUE;
        END_IF
      ELSE
        bAnyBusy := TRUE;
      END_IF
    END_FOR
    IF NOT bAnyBusy THEN
      bExecute := FALSE;
      IF NOT bAnyError THEN
        iUpdateState := 0; (* OK *)
      ELSE
        iUpdateState := 0; (* Error *)
      END_IF
    END_IF
  END_CASE

```

## Requirements

Development environment	Target platform type	PLC libraries	AX5000 Firmware
TwinCAT 3	PC or CX (x86 or x64)	Tc2_MC2_Drive	≥v1.00 b0001

### 4.4.4 FB\_SoEAX5000SetPositionOffset



With the function block FB\_SoEAX5000SetPositionOffset a position offset can be saved in the memory of the AX5000 or of the digital nameplate of the connected motor encoder. To do this, the offset must first be configured with the DriveManager and then the same memory location (encoder / drive) must be used in the function block.

Alternatively, the function block FB\_SetPositionOffset [▶ 14] can also be used.

#### Inputs

```

VAR_INPUT
  Execute : BOOL;
  Position : LREAL;
  Relative : BOOL;
  Feedback : E_PositionOffsetFeedback;
  Memory : E_PositionOffsetMemory;
END_VAR
  
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New actual position of the NC axis
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	<u>E_PositionOffsetFeedback</u> [▶ 70]	Specifies which feedback is considered.
Memory	<u>E_PositionOffsetMemory</u> [▶ 69]	Specifies which memory the newly calculated position offset should be stored in.

#### Inputs/outputs

```

VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
  
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

#### Outputs

```

VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
  
```

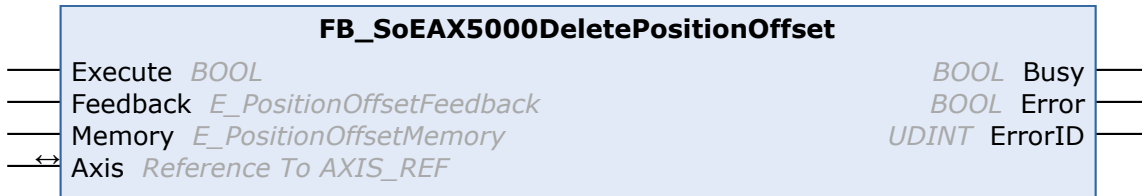
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.

Name	Type	Description
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	AX5000 Firmware
TwinCAT ≥4020.57 TwinCAT ≥4022.30	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.16.0	≥v2.11 b0001

**4.4.5 FB\_SoEAX5000DeletePositionOffset**



With the function block FB\_SoEAX5000DeletePositionOffset, a position offset can be deleted from the memory of the AX5000 or from the digital nameplate of the connected motor encoder.

Alternatively, the function block [FB\\_DeletePositionOffset \[► 12\]](#) can also be used.

**Inputs**

```
VAR_INPUT
    Execute : BOOL;
    Feedback : E_PositionOffsetFeedback;
    Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Feedback	<a href="#">E_PositionOffsetFeedback [► 70]</a>	Specifies which feedback is considered.
Memory	<a href="#">E_PositionOffsetMemory [► 69]</a>	Specifies the memory from which the position offset is to be deleted.

**Inputs/outputs**

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<a href="#">AXIS_REF</a>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**Outputs**

```
VAR_OUTPUT
    Busy : BOOL;
    Error : BOOL;
    ErrorID : UDINT;
END_VAR
```

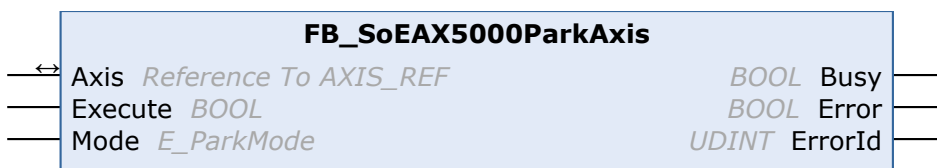


Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	AX5000 Firmware
TwinCAT ≥4020.57	PC or CX (x86 or x64)	Tc2_MC2_Drive	≥v2.11 b0001
TwinCAT ≥4022.30		≥V3.3.16.0	

**4.4.6 FB\_SoEAX5000ParkAxis**



This function block enables the parking functionality of an AX5000. A parked channel of an AX5000 is temporarily disabled.

In the case of a modular machine concept, certain motors may not be available. So that this does not lead directly to errors (e.g. feedback errors), the channels can be disabled via this.

Alternatively, the function block [FB\\_ParkAxis \[▶ 16\]](#) can also be used.

**Inputs**

```
VAR_INPUT
Execute : BOOL;
Mode : E_ParkMode
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	<a href="#">E_ParkMode [▶ 71]</a>	Mode for selecting whether the FB parks or releases the axis.

**Inputs/outputs**

```
VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<a href="#">AXIS_REF</a>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**Outputs**

```
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
```

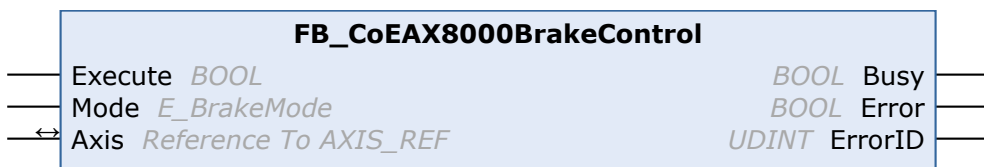
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC libraries	AX5000 Firmware
TwinCAT ≥ 4022.32 TwinCAT ≥ 4024.6	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.21.0	≥v2.12 b0001

## 4.5 AX8000 CoE

### 4.5.1 FB\_CoEAX8000BrakeControl



This function block can be used to manually open or close the brake of a motor that is operated on an AX8000.

It should be noted that the brake is permanently closed or opened via "Lock" or "Unlock".

It is recommended to open or close the brake manually only for the necessary duration. The brake should then be set back to "Automatic" mode using the function block.

Alternatively, the function block [FB\\_BrakeControl \[► 13\]](#) can also be used.

**Inputs**

```
VAR_INPUT
Execute : BOOL;
Mode : E_BrakeMode;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	E_BrakeMode [► 70]	The mode with which the brake is controlled.

**Inputs/outputs**

```
VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**🔌 Outputs**

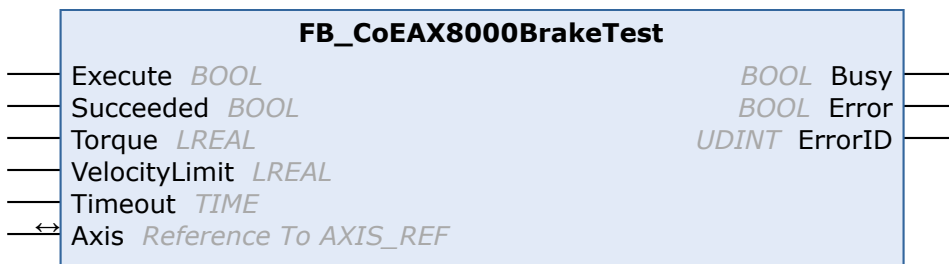
```
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	Firmware version
TwinCAT ≥ 4022.36 TwinCAT ≥ 4024.15	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.23.0	v1.04 b0001

**4.5.2 FB\_CoEAX8000BrakeTest**



The function block FB\_CoEAX8000BrakeTest can be used to implement a functional brake test.

This function block is used to switch the AX8000 to torque mode (Cyclic Synchronous Torque Mode (CST)). The torque parameterized at the input is transmitted to the drive as a setpoint. The torque is held until the "Timeout" has elapsed or a feedback ("Succeeded") is sent to the function block. This feedback is typically sent from the safety controller to the PLC. The AX8000 is then set to the original operation mode.

If the brake was not closed before the FB\_CoEAX8000BrakeTest was called, or if the brake does not hold the required test torque, then the speed limitation integrated in the AX8000 prevents uncontrolled acceleration of the axis. The velocity limit is parameterized via the "VelocityLimit" variable.

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

When using the function block, the axis is switched to CST mode. After using the function block (especially after error situations), the axis may still be in CST mode. This can lead to sudden and unplanned movements (especially with lifting axes) when the axis is released.

- Ensure that there is no hazard as defined by the risk assessment.
- Check the current operation mode via the function block MC\_ReadDriveOperationMode.
- If the axis is not in a position-related operation mode (CSV/CSP), transfer it before an enable:
  - *directly* with MC\_WriteDriveOperationMode into the desired position-related operation mode (CSV/CSP) or
  - *indirectly* with MC\_Halt / MC\_Stop into the desired position-related operation mode (CSV/CSP) (from TwinCAT 3.1.4024.40)

Other function blocks that switch the axis indirectly into a position-related operation mode can only do this to a limited extent and are therefore not to be used for a deliberate operation mode change.

⇒ Subsequently, it is necessary to check again whether the axis is really in a position-related operation mode (CSV/CSP), if not, an abort with error handling is required.

Information on the necessary changes in the configuration of the AX8000 can be found in the [AX8000 | Multi-axis servo system Functional description](#).

**🔌 Inputs**

```
VAR_INPUT
  Execute      : BOOL;
  Succeeded    : BOOL;
  Torque       : LREAL;
  VelocityLimit : LREAL;
  Timeout      : TIME;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Succeeded	BOOL	Feedback that the brake has held the specified torque. This feedback usually comes from the safety controller.
Torque	LREAL	Torque that should be effective [Nm].
VelocityLimit	LREAL	Velocity limit to prevent uncontrolled acceleration in torque mode.
Timeout	TIME	Maximum time for which the specified torque is effective.

**🔌/🔌 Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

**🔌 Outputs**

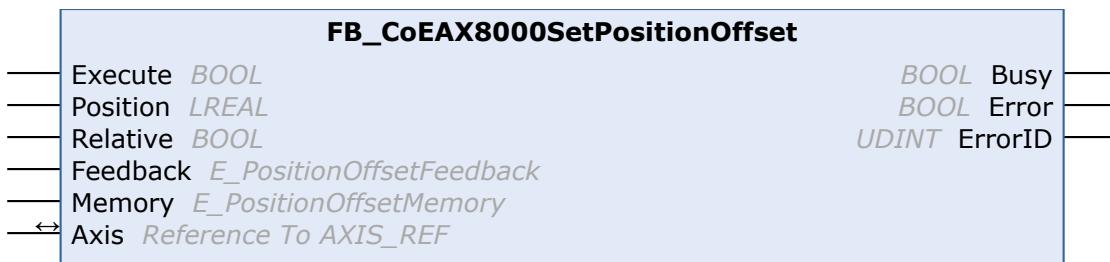
```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	Firmware version
TwinCAT ≥ 4022.36 TwinCAT ≥ 4024.15	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.23.0	v1.04 b0001

**4.5.3 FB\_CoEAX8000SetPositionOffset**



With the FB\_CoEAX8000SetPositionOffset function block, a position offset can be saved in the memory of the AX8000 or of the digital nameplate of the connected motor encoder. To do this, the offset must first be configured with the DriveManager and then the same memory location (encoder / drive) must be used in the function block.

Alternatively, the function block FB\_SetPositionOffset [▶ 14] can also be used.

**Inputs**

```
VAR_INPUT
Execute : BOOL;
Position : LREAL;
Relative : BOOL;
Feedback : E_PositionOffsetFeedback;
Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New actual position of the NC axis
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	<u>E_PositionOffsetFeedback</u> [▶ 70]	Specifies which feedback is considered.
Memory	<u>E_PositionOffsetMemory</u> [▶ 69]	Specifies which memory the newly calculated position offset should be stored in.

**Inputs/outputs**

```
VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	<u>AXIS_REF</u>	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

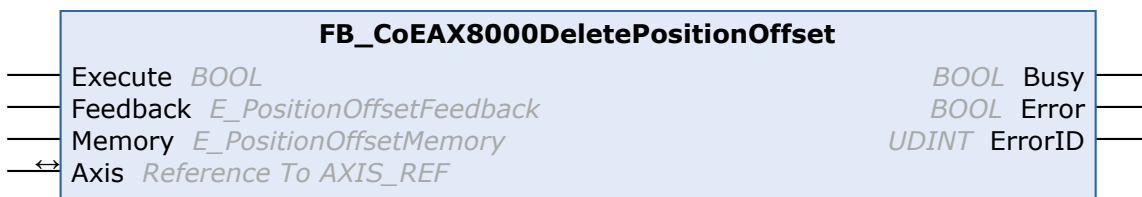
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC libraries	Firmware version
TwinCAT ≥4020.57 TwinCAT ≥4022.30	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.16.0	v1.04 b0001

### 4.5.4 FB\_CoEAX8000DeletePositionOffset



With the function block FB\_CoEAX8000DeletePositionOffset, a position offset can be deleted from the memory of the AX8000 or from the digital nameplate of the connected motor encoder.

Alternatively, the function block [FB\\_DeletePositionOffset](#) [► 12] can also be used.

 **Inputs**

```
VAR_INPUT
  Execute   : BOOL;
  Position  : LREAL;
  Relative  : BOOL;
  Feedback  : E_PositionOffsetFeedback;
  Memory    : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New position of the NC axis is displayed.
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	<u>E_PositionOffsetFeedback</u> [► 70]	Specifies which feedback is considered.
Memory	<u>E_PositionOffsetMemory</u> [► 69]	Specifies the memory from which the position offset is to be deleted.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

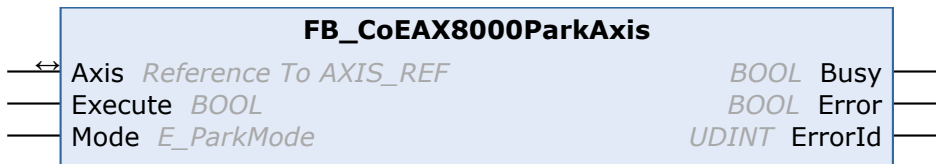
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC libraries	Firmware version
TwinCAT ≥4020.57 TwinCAT ≥4022.30	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.16.0	v1.04 b0001

**4.5.5 FB\_CoEAX8000ParkAxis**



This function block enables the parking functionality of an AX8000. A parked channel of an AX8000 is temporarily disabled.

In the case of a modular machine concept, certain motors may not be available. So that this does not lead directly to errors (e.g. feedback errors), the channels can be disabled via this.

Alternatively, the function block [FB\\_ParkAxis](#) [▶ 16] can also be used.

 **Inputs**

```
VAR_INPUT
  Execute : BOOL;
  Mode    : E_ParkMode;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Mode	E_ParkMode [▶ 71] [▶ 72]	Mode for selecting whether the FB parks or releases the axis.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

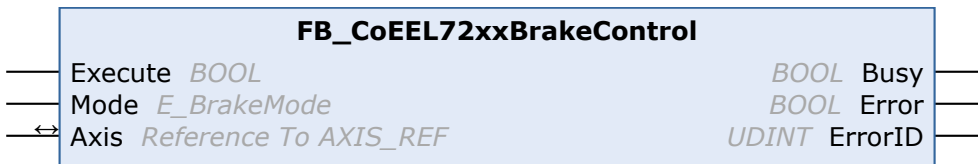
Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC libraries to include	Firmware version
TwinCAT ≥4024.60 TwinCAT ≥4026.8	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.41.0	v1.06 b0003

## 4.6 EL72xx CoE

### 4.6.1 FB\_CoEEL72xxBrakeControl



This function block can be used to manually open or close the brake of a motor that is operated on an EL72xx.

It should be noted that the brake is permanently closed or opened via "Lock" or "Unlock".

It is recommended to open or close the brake manually only for the necessary duration. The brake should then be set back to "Automatic" mode using the function block.

Alternatively, the function block [FB\\_BrakeControl](#) [▶ 13] can also be used.

 **Inputs**

```
VAR_INPUT
  Execute : BOOL;
  Mode : E_BrakeMode;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.



Name	Type	Description
Mode	E_BrakeMode [▶ 70]	The mode with which the brake is controlled.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

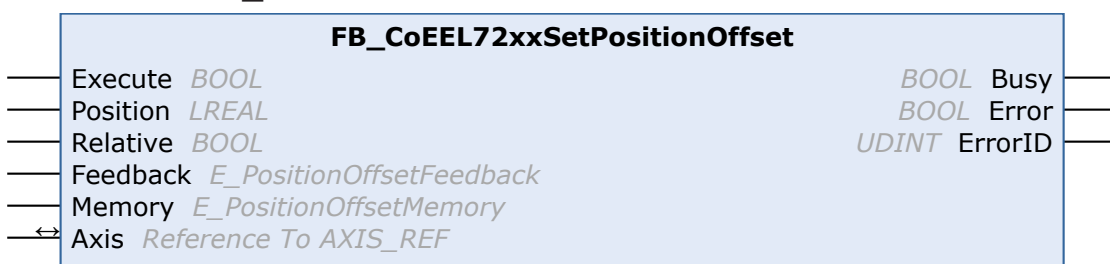
```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	Firmware version
TwinCAT ≥4024.48	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.34.0	From FW v01 (EL72xx only with OCT and from FW v16)

**4.6.2 FB\_CoEEL72xxSetPositionOffset**



With the function block FB\_CoEEL72xxSetPositionOffset, a position offset can be saved in the memory of the EL72xx. Saving in the digital nameplate of the connected motor encoder is currently not planned.

Alternatively, the function block [FB\\_SetPositionOffset \[▶ 14\]](#) can also be used.

 **Inputs**

```
VAR_INPUT
  Execute : BOOL;
  Position : LREAL;
  Relative : BOOL;
  Feedback : E_PositionOffsetFeedback;
  Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New actual position of the NC axis
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	E_PositionOffsetFeedback [▶ 70]	Specifies which feedback is considered.
Memory	E_PositionOffsetMemory [▶ 69]	Specifies which memory the newly calculated position offset should be stored in.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

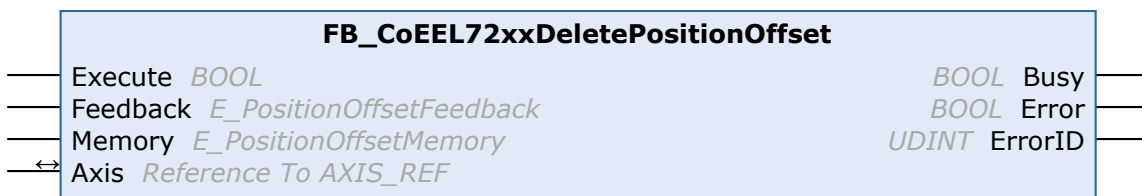
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	Firmware version
TwinCAT >4020.57 TwinCAT >4022.30 TwinCAT ≥4024.4	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.17.0	From FW v01 (AMI8xxx only Drive)

**4.6.3 FB\_CoEEL72xxDeletePositionOffset**



With the function block FB\_CoEEL72xxDeletePositionOffset, a position offset can be deleted from the memory of the EL72xx.

Alternatively, the function block [FB\\_DeletePositionOffset](#) [▶ 12] can also be used.

 **Inputs**

```
VAR_INPUT
  Execute : BOOL;
  Position : LREAL;
  Relative : BOOL;
  Feedback : E_PositionOffsetFeedback;
  Memory : E_PositionOffsetMemory;
END_VAR
```

Name	Type	Description
Execute	BOOL	The function block is enabled via a positive edge at this input.
Position	LREAL	New position of the NC axis is displayed.
Relative	BOOL	When this flag is set, the position is interpreted relatively.
Feedback	E_PositionOffsetFeedback [▶_70]	Specifies which feedback is considered.
Memory	E_PositionOffsetMemory [▶_69]	Specifies the memory from which the position offset is to be deleted.

 **Inputs/outputs**

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Type	Description
Axis	AXIS_REF	Axis data structure that unambiguously addresses an axis in the system. Among other parameters it contains the current axis status, including position, velocity or error state.

 **Outputs**

```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Type	Description
Busy	BOOL	This output is set when the function block is activated, and remains set until a feedback is received.
Error	BOOL	This output is set after the Busy output has been reset when an error occurs in the transmission of the command.
ErrorID	UDINT	In the event of a set error output returns the ADS error code.

**Requirements**

Development environment	Target platform type	PLC library	Firmware version
TwinCAT >4020.57 TwinCAT >4022.30 TwinCAT ≥4024.4	PC or CX (x86 or x64)	Tc2_MC2_Drive ≥V3.3.17.0	From FW v01 (AMI8xxx only Drive)

## 5 Data types

### 5.1 E\_FwUpdateState

The enumeration E\_FwUpdateState describes the state of a firmware update.

```

TYPE E_SoE_CmdState : (
  (* update states *)
  eFwU_NoError := 0,
  eFwU_CheckCfgIdentity,
  eFwU_CheckSlaveCount,
  eFwU_CheckFindSlavePos,
  eFwU_WaitForScan,
  eFwU_ScanningSlaves,
  eFwU_CheckScannedIdentity,
  eFwU_CheckScannedFirmware,
  eFwU_FindFirmwareFile,
  eFwU_WaitForUpdate,
  eFwU_WaitForSlaveState,
  eFwU_StartFwUpdate,
  eFwU_FwUpdateInProgress,
  eFwU_FwUpdateDone,
  eFwU_NoFwUpdateRequired,

  (* not updating via this channel *)
  eFwU_UpdateViaOtherChannelActive,
  eFwU_UpdatedViaOtherChannel,

  (* error states *)
  eFwU_GetSlaveIdentityError := -1,
  eFwU_GetSlaveCountError := -2,
  eFwU_GetSlaveAddrError := -3,
  eFwU_StartScanError := -4,
  eFwU_ScanStateError := -5,
  eFwU_ScanIdentityError := -6,
  eFwU_GetSlaveStateError := -7,
  eFwU_ScanFirmwareError := -8,
  eFwU_FindFileError := -9,
  eFwU_CfgTypeInNoAX5xxx := -10,
  eFwU_ScannedTypeInNoAX5xxx := -11,
  eFwU_ChannelMismatch := -12,
  eFwU_ChannelMismatch_1Cfg_2Scanned := -13,
  eFwU_ChannelMismatch_2Cfg_1Scanned := -14,
  eFwU_CurrentMismatch := -15,
  eFwU_FwUpdateError := -16,
  eFwU_ReqSlaveStateError := -17
);
END_TYPE

```

#### Update Status

Name	Description
eFwU_NoError	Initial state
eFwU_CheckCfgIdentity	Read the configured slave types (number of channels, current, revision).
eFwU_CheckSlaveCount	Determine the configured number of slaves.
eFwU_CheckFindSlavePos	Search for the slave address in the master object directory.
eFwU_WaitForScan	Wait for online scan.
eFwU_ScanningSlaves	Online scan of the slaves.
eFwU_CheckScannedIdentity	Read the scanned slave types (number of channels, current, revision).
eFwU_CheckScannedFirmware	Read the firmware version.
eFwU_FindFirmwareFile	Search for the selected firmware file.
eFwU_WaitForUpdate	Wait for status of the update.
eFwU_WaitForSlaveState	Determine the EtherCAT slave status.
eFwU_StartFwUpdate	Start the firmware update.
eFwU_FwUpdateInProgress	Firmware update active.

Name	Description
eFwU_FwUpdateDone	Firmware update successfully completed.
eFwU_NoFwUpdateRequired	No firmware update required.
eFwU_UpdateViaOtherChannelActive	Update takes place via the other axis channel.
eFwU_UpdatedViaOtherChannel	Update took place via the other axis channel.

**Update error**

Name	Description
eFwU_GetSlaveIdentityError	Reading of the configured slave type failed (see iAdsErrId).
eFwU_GetSlaveCountError	Determination of the configured number of slaves failed (see iAdsErrId).
eFwU_GetSlaveAddrError	Search for the slave address in the master object directory failed (see iAdsErrId).
eFwU_StartScanError	Start of the online scan failed (see iAdsErrId).
eFwU_ScanStateError	Online scan failed (see iAdsErrId).
eFwU_ScanIdentityError	Reading of the scanned slave types (number of channels, current, revision) failed (see iAdsErrId).
eFwU_GetSlaveStateError	Determination of the EtherCAT slave status failed (see iAdsErrId).
eFwU_ScanFirmwareError	Reading of the firmware version failed (see iAdsErrId + iSercosErrId).
eFwU_FindFileError	Search for the selected firmware file failed (see iAdsErrId).
eFwU_CfgTypeInNoAX5xxx	The configured slave is not an AX5000.
eFwU_ScannedTypeInNoAX5xxx	The scanned slave is not an AX5000.
eFwU_ChannelMismatch	The number of configured and found channels of the AX5000 do not match.
eFwU_ChannelMismatch_1Cfg_2Scanned	Single-channel device configured, but two-channel device found.
eFwU_ChannelMismatch_2Cfg_1Scanned	Two-channel device configured, but single-channel device found.
eFwU_CurrentMismatch	AX5000 type does not match in terms of current, e.g. AX5103 (3 A) configured, but AX5106 (6 A) found.
eFwU_FwUpdateError	General update error (see iAdsErrId)
eFwU_ReqSlaveStateError	Switching to the desired EtherCAT status failed.

## 5.2 E\_DriveErrorCodes

```

TYPE E_DriveErrorCodes : (
  MC_Error_MaxNcPositionOverrun           := 16#4BB0,
  MC_Error_MaxPositionOffsetOverrun       := 16#4BB1,
  MC_Error_MinNcPositionUnderrun         := 16#4BB2,
  MC_Error_MinPositionOffsetUnderrun      := 16#4BB3,
  MC_Error_WrongTargetForFeedbackOrMemory := 16#4BB4,
  MC_Error_PositionReinitializationFailed := 16#4BB5,
  MC_Error_CommandRejectedNoResponse     := 16#4BB6,
  MC_Error_CommandRejectedWithResponse   := 16#4BB7
) UDINT;
END_TYPE

```

## 5.3 E\_PositionOffsetMemory

```

TYPE E_PositionOffsetMemory : (
  ePositionOffsetMemory_Encoder := 0,
  ePositionOffsetMemory_Drive   := 1
) BYTE;
END_TYPE

```

Name	Description
ePositionOffsetMemory_Encoder	As memory for the new position offset the motor name plate of the encoder is selected.
ePositionOffsetMemory_Drive	As memory for the new positions offset the internal memory of the drive is selected.

## 5.4 E\_PositionOffsetFeedback

```

TYPE E_PositionOffsetFeedback : (
    ePositionOffsetFeedback1 := 0,
    ePositionOffsetFeedback2 := 1
) BYTE;
END_TYPE

```

Name	Description
ePositionOffsetFeedback1	Reference for the recalculation of the position offset is Feedback System 1.
ePositionOffsetFeedback2	Reference for the recalculation of the position offset is Feedback System 2.

## 5.5 E\_BrakeMode

Sets the mode for manual brake control.

```

TYPE E_BrakeMode : (
    eBrakeMode_Automatic,
    eBrakeMode_Lock,
    eBrakeMode_Unlock
) INT;
END_TYPE

```

Name	Description
eBrakeMode_Automatic	The brake of the connected motor is automatically opened or closed.
eBrakeMode_Lock	The brake of the connected motor remains permanently closed. Enabling the axis via the MC_Power does not release the brake.
eBrakeMode_Unlock	The brake of the connected motor is opened permanently. Disabling the axis via the MC_Power does not close the brake. See warning.

### WARNING

#### Risk of fatal or serious injury

Note that vertical axes can fall down and prevent this from happening!

## 5.6 E\_DeviceType

Enumeration for defining the Beckhoff drive device.

```

TYPE E_DeviceType : (
    DEVICETYPE_UNKNOWN           := 0,
    DEVICETYPE_SOE_DEFAULT       := 1,
    DEVICETYPE_AX2000           := 21,
    DEVICETYPE_EL72x1           := 22,
    DEVICETYPE_EL72x1_OCT       := 23,
    DEVICETYPE_EL72x1_OCT_SAFETY := 24,
    DEVICETYPE_AX8000           := 31,
    DEVICETYPE_AMI8xxx          := 41,
    DEVICETYPE_AMP8xxx          := 51,
    DEVICETYPE_EL74x1           := 61,
    DEVICETYPE_EL74x1_SAFETY    := 62,
    DEVICETYPE_ELM72xx_OCT      := 71,

```

```

DEVICETYPE_ELM72xx_OCT_SAFETY := 72
) UDINT;
END_TYPE

```

Name	Description
DEVICETYPE_UNKNOWN	Unknown drive unit
DEVICETYPE_SOE_DEFAULT	Standard SoE-drive
DEVICETYPE_AX2000	AX2000
DEVICETYPE_EL72x1	EL72x1
DEVICETYPE_EL72x1_OCT	EL72x1 with OCT
DEVICETYPE_EL72x1_OCT_SAFETY	EL72x1 with OCT and Safety
DEVICETYPE_AX8000	AX8000
DEVICETYPE_AMI8xxx	AMI8xxx
DEVICETYPE_AMP8xxx	AMP8xxx
DEVICETYPE_EL74x1	EL74x1
DEVICETYPE_EL74x1_SAFETY	EL74x1 with Safety
DEVICETYPE_ELM72xx_OCT	ELM72xx with OCT
DEVICETYPE_ELM72xx_OCT_SAFETY	ELM72xx with OCT and Safety

## 5.7 E\_ParkMode

This ENUM determines whether the function block [FB\\_ParkAxis](#) [▶ 16] parks or releases the corresponding channel of the drive.

```

TYPE E_ParkMode : (
    eParkMode_Park,
    eParkMode_Release
) INT;
END_TYPE

```

Name	Description
eParkMode_Park	Channel is to be parked.
eParkMode_Release	Channel is to be released.

## 5.8 ST\_DriveInfo

Structure with basic information on the Beckhoff servo hardware used, which is returned via the [FB\\_ReadDriveInfo](#) [▶ 15] function block.

```

TYPE ST_DriveInfo :
STRUCT
    NetId          : T_AmsNetId;
    SlaveAddress   : T_AmsPort;
    Channel        : USINT;
    MDPProfile     : BOOL;
    DeviceName     : STRING;
    DeviceType     : E_DeviceType;
END_STRUCT
END_TYPE

```

Name	Type	Description
NetId	T_AmsNetId	String containing the AMS network ID of the EtherCAT master device.
SlaveAddress	T_AmsPort	Fixed address of the EtherCAT slave
Channel	USINT	Channel of the EtherCAT slave
MDPProfile	BOOL	TRUE if an MDP protocol is used.
DeviceName	STRING	Contains the name of the device (not the free name of the device) in the I/O tree.
DeviceType	<a href="#">E_DeviceType</a> [▶ 70]	Enumeration that specifies which Beckhoff drive device is involved.

## 5.9 E\_SoEDataAccessMode

```
TYPE E_SoEDataAccessMode : (  
  eSoEDataAccessMode _Parallel := 0,  
  eSoEDataAccessMode _Sequential := 1  
) INT;  
END_TYPE
```

## 5.10 E\_AX5000ParkMode

This ENUM determines whether the [FB SoEAX5000ParkAxis](#) [► 57] parks or releases the corresponding channel of the AX5000.

```
TYPE E_AX5000ParkMode : (  
  eAX5000ParkMode_Park,  
  eAX5000ParkMode_Release  
) INT;  
END_TYPE
```

Name	Description
eAX5000ParkMode_Park	AX5000 channel is to be parked.
eAX5000ParkMode_Release	AX5000 channel is to be released.



## 6 Global constants

### 6.1 SoE Parameter Access

The parameters of the individual parameter sets can be accessed with the help of these constants.

#### Global\_Variables

```
VAR_GLOBAL CONSTANT
  S_0_IDNs : WORD := 16#0000;
  S_1_IDNs : WORD := 16#1000;
  S_2_IDNs : WORD := 16#2000;
  S_3_IDNs : WORD := 16#3000;
  S_4_IDNs : WORD := 16#4000;
  S_5_IDNs : WORD := 16#5000;
  S_6_IDNs : WORD := 16#6000;
  S_7_IDNs : WORD := 16#7000;

  P_0_IDNs : WORD := 16#8000;
  P_1_IDNs : WORD := 16#9000;
  P_2_IDNs : WORD := 16#A000;
  P_3_IDNs : WORD := 16#B000;
  P_4_IDNs : WORD := 16#C000;
  P_5_IDNs : WORD := 16#D000;
  P_6_IDNs : WORD := 16#E000;
  P_7_IDNs : WORD := 16#F000;
END_VAR
```

## 7 Support and Service

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

### Download finder

Our [download finder](#) contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

### Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for [local support and service](#) on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: [www.beckhoff.com](http://www.beckhoff.com)

You will also find further documentation for Beckhoff components there.

### Beckhoff Support

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

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

Hotline: +49 5246 963-157  
e-mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

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

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

Hotline: +49 5246 963-460  
e-mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20  
33415 Verl  
Germany

Phone: +49 5246 963-0  
e-mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
web: [www.beckhoff.com](http://www.beckhoff.com)



More Information:  
**[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

