Table of contents

1 Foreword ................................................................................................................................................. 5
  1.1 Notes on the documentation ............................................................................................................ 5
  1.2 Safety instructions ............................................................................................................................ 6

2 Overview .................................................................................................................................................. 7

3 Function blocks ........................................................................................................................................ 9
  3.1 General SoE ....................................................................................................................................... 9
    3.1.1 FB_SoEReset ................................................................................................................................ 9
    3.1.2 FB_SoEWritePassword .............................................................................................................. 10
    3.1.3 Function blocks for commands ................................................................................................. 11
    3.1.4 Function blocks for diagnostics ............................................................................................... 15
    3.1.5 Function blocks for determining current values ......................................................................... 20
  3.2 General CoE ..................................................................................................................................... 28
    3.2.1 Function blocks for determining current values ......................................................................... 28
  3.3 AX5000 SoE ....................................................................................................................................... 31
    3.3.1 FB_SoEAX5000ReadActMainVoltage ....................................................................................... 31
    3.3.2 FB_SoEAX5000SetMotorCtrlWord .......................................................................................... 32
    3.3.3 FB_SoEAX5000FirmwareUpdate ............................................................................................... 33
  3.4 F_GetVersionTcNcDrive ..................................................................................................................... 36
1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

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

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

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

Trademarks

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

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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


with corresponding applications or registrations in various other countries.

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.
1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

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

Personnel qualification

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

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨 DANGER 🚨</td>
<td>Serious risk of injury! Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</td>
</tr>
<tr>
<td>🚨 WARNING 🚨</td>
<td>Risk of injury! Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</td>
</tr>
<tr>
<td>🚨 CAUTION 🚨</td>
<td>Personal injuries! Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</td>
</tr>
<tr>
<td>📜 NOTE 📜</td>
<td>Damage to the environment or devices Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</td>
</tr>
</tbody>
</table>

Tip or pointer

This symbol indicates information that contributes to better understanding.
2 Overview

The Tc2_NcDrive library should no longer be used in newer projects. Please use the Tc2_MC2_Drive library instead (see documentation TwinCAT 3 PLC Lib Tc2_MC2_Drive).

The Tc2_NcDrive 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 Tc_MC2_Drive form wrapper function blocks around the function blocks of the Tc2_Drive library.

<table>
<thead>
<tr>
<th>Drive library</th>
<th>Use</th>
<th>Access to the drive</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tc2_Drive</strong></td>
<td>Use the Tc2_Drive library if you use the drive entirely from the PLC (i.e. without NC).</td>
<td>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)</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Tc2_NcDrive</strong></td>
<td>Use the Tc2_NcDrive library if you are using the drive via the NC with the Tc2_Nc libraries.</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Tc2_MC2_Drive</strong></td>
<td>Use the Tc2_MC2_Drive library if you are using the drive via the NC with the Tc2_MC2 library.</td>
<td>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.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_GetVersionTcNcDrive</td>
<td>Reads version information from the PLC library. The function has been replaced by the global structure stLibVersion_Tc2_NcDrive.</td>
</tr>
</tbody>
</table>
### Function blocks

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_ConvWordToSTAX5000C1D</td>
<td>Converts the C1D word (S-0-0011) of the AX5000 to an ST_AX5000_C1D structure. See: TwinCAT 3 PLC Lib documentation: Tc2_Drive.</td>
</tr>
</tbody>
</table>

#### Requests

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwinCAT on the development computer</td>
<td>3.1 Build 4016 or higher</td>
</tr>
<tr>
<td>TwinCAT on the Windows CE-Image</td>
<td>3.1 Build 4016 or higher</td>
</tr>
<tr>
<td>TwinCAT on the Windows XP-Image</td>
<td>3.1 Build 4016 or higher</td>
</tr>
</tbody>
</table>
3 Function blocks

3.1 General SoE

3.1.1 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.

VAR_INPUT

VAR_INPUT

sNetId   := T_AmsNetId := '';
bExecute : BOOL;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

sNetId: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
bExecute: The function block is enabled via a positive flank at this input.
tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT

VAR_IN_OUT

Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR
**Axis**: Axis data structure of the type NCTOPLC_AXIS_REF

### VAR_OUTPUT

```plaintext
VAR_OUTPUT

<table>
<thead>
<tr>
<th>Output</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
</tr>
<tr>
<td>iAdsErrId</td>
<td>UINT</td>
</tr>
<tr>
<td>iSercosErrId</td>
<td>UINT</td>
</tr>
</tbody>
</table>

END_VAR
```

**bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.

**bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

**iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.

**iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.

### Sample

```plaintext
Sample

```

```plaintext
FB_SoEReset : FB_SoEReset_ByDriveRef;
```

```plaintext
VAR_INPUT

<table>
<thead>
<tr>
<th>Input</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>sNetId</td>
<td>T_AmsNetId</td>
</tr>
<tr>
<td>bExecute</td>
<td>BOOL</td>
</tr>
<tr>
<td>tTimeout</td>
<td>TIME</td>
</tr>
<tr>
<td>sPassword</td>
<td>ST_SoE_String</td>
</tr>
</tbody>
</table>

END_VAR
```

```plaintext
sNetId : String containing the AMS network ID of the PC. (type: T_AmsNetId)

bExecute: The function block is enabled via a positive edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

sPassword: Password as a Sercos string

With the FB_SoEWritePassword function block (S-0-0267) the drive password can be set.
VAR_IN_OUT

VAR_IN

  Axis : NCTOPLC_AXIS_REF; (* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF.

VAR_OUTPUT

VAR_OUTPUT

  bBusy : BOOL;
  bError : BOOL;
  iAdsErrId : UINT;
  iSercosErrId : UINT;
END_VAR

bBusy: This output is set when the function block is activated, and remains set until an acknowledgement is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

Sample

fbWritePassword : FB_SoEWritePassword;
bWritePassword : BOOL;
sPassword : ST_SoE_String;
(* NcAxis *)

stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bWritePassword THEN
  fbWritePassword(  
    Axis := stNcToPlc,  
    bExecute := TRUE,  
    tTimeout := DEFAULT_ADS_TIMEOUT,  
    sPassword := sPassword  
  );
  IF NOT fbWritePassword.bBusy THEN
    fbWritePassword(Axis := stNcToPlc, bExecute := FALSE);
    bWritePassword := FALSE;
  END_IF
END_IF

3.1.3 Function blocks for commands

3.1.3.1 FB_SoEExecuteCommand

With the FB_SoEExecuteCommand function block a command can be executed.

VAR_INPUT

VAR_INPUT

  sNetId : T_AmsNetId := '';  
  nIdn : WORD;
Function blocks

bExecute : BOOL;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

sNetId: String, which contains the AMS-Network ID of the PC (Type: T_AmsNetId).
nIdn: Parameter number to which FB_SoEExecuteCommand refers, e.g. "P_0_IDN + 160" for P-0-0160.
bExecute: The function block is enabled via a positive edge at this input.
tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT
VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF

VAR_OUTPUT
VAR_OUTPUT
bBusy: This output is set when the function block is activated, and remains set until a feedback is received.
bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.
iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

Sample

{\texttt{fbExecuteCommand : FB_SoEExecuteCommand;}}
{\texttt{bExecuteCommand : BOOL;}}
{\texttt{nIdn : WORD;}}
{\texttt{(* NcAxis *)}}
{\texttt{stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;}}

{\texttt{IF bExecuteCommand THEN}}
{\texttt{nIdn := P_0_IDN + 160;}}
{\texttt{fbExecuteCommand(Axis := stNcToPlc, bExecute := TRUE,}}
{\texttt{tTimeout := DEFAULT_ADS_TIMEOUT,}}
{\texttt{nIdn := nIdn,}}
{\texttt{);}}
{\texttt{IF NOT fbExecuteCommand.bBusy THEN}}
{\texttt{fbExecuteCommand(Axis := stNcToPlc, bExecute := FALSE);}}
{\texttt{bExecuteCommand := FALSE;}}
{\texttt{END_IF}}
{\texttt{END_IF}}

3.1.3.2 FB_SoEWriteCommandControl

<table>
<thead>
<tr>
<th>sNetId</th>
<th>T_AmsNetId</th>
<th>BOOL</th>
<th>bBusy</th>
</tr>
</thead>
<tbody>
<tr>
<td>nIdn</td>
<td>WORD</td>
<td></td>
<td>bError</td>
</tr>
<tr>
<td>iParamControl</td>
<td>_SoE_CmdControl</td>
<td>UINT</td>
<td>iAdsErrId</td>
</tr>
<tr>
<td>bExecute</td>
<td>BOOL</td>
<td></td>
<td>iSercosErrId</td>
</tr>
<tr>
<td>tTimeout</td>
<td>TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Axis</td>
<td>NCTOPLC_AXIS_REF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Version: 1.2
TE1000
With the FB_SoEWriteCommandControl function block a command can be prepared, started or canceled.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sNetId</td>
<td>T_AmsNetId</td>
<td>String, which contains the AMS-Network ID of the PC</td>
</tr>
<tr>
<td>nIdn</td>
<td>WORD</td>
<td>Parameter number to which FB_SoEWriteCommandControl refers, e.g. &quot;P_0_IDN + 160&quot; for P-0-0160.</td>
</tr>
<tr>
<td>eCmdControl</td>
<td>E_SoE_CmdControl</td>
<td>Indicates, if a command should be prepared (eSoE_CmdControl_Set := 1), executed (eSoE_CmdControl_SetAndEnable := 3) or aborted (eSoE_CmdControl_Cancel := 0).</td>
</tr>
<tr>
<td>bExecute</td>
<td>BOOL</td>
<td>The function block is enabled via a positive edge at this input.</td>
</tr>
<tr>
<td>tTimeout</td>
<td>TIME</td>
<td>Maximum time allowed for the execution of the function block.</td>
</tr>
</tbody>
</table>

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis</td>
<td>NCTOPLC_AXIS_REF</td>
<td>Axis data structure of the type NCTOPLC_AXIS_REF</td>
</tr>
</tbody>
</table>

**VAR_OUTPUT**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>BOOL</td>
<td>This output is set when the function block is activated, and remains set until a feedback is received.</td>
</tr>
<tr>
<td>bError</td>
<td>BOOL</td>
<td>This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.</td>
</tr>
<tr>
<td>iAdsErrId</td>
<td>UINT</td>
<td>In the case of a set bError output returns the ADS error code of the last executed command.</td>
</tr>
<tr>
<td>iSercosErrId</td>
<td>UINT</td>
<td>In the case of a set bError output returns the Sercos error of the last executed command.</td>
</tr>
</tbody>
</table>

**Sample**

```plaintext
FB_WriteCommandControl := FB_SoEWriteCommandControl;
FB_SoEWriteCommandControl(nIdn := P_0_IDN + 160, bExecute := TRUE, tTimeout := DEFAULT_ADS_TIMEOUT, eCmdControl := E_SoE_CmdControl_SetAndEnable);
```

**Sample**

```plaintext
IF bWriteCommandControl THEN
  nIdn := P_0_IDN + 160;
  fbWriteCommandControl(Axis := stNcToPlc, bExecute := TRUE, tTimeout := DEFAULT_ADS_TIMEOUT, nIdn := nIdn, eCmdControl := eCmdControl);
ELSE IF NOT fbWriteCommandControl.bBusy THEN
  fbWriteCommandControl(Axis := stNcToPlc, bExecute := FALSE); fbWriteCommandControl := FALSE;
END_IF
```
Function blocks

3.1.3.3 FB_SoEReadCommandState

With the FB_SoEReadCommandState function block the command execution can be checked.

VAR_INPUT

| sNetId   | T_AmsNetId := ''; |
| nIdn     | WORD; |
| bExecute | BOOL; |
| tTimeout | TIME := DEFAULT_ADS_TIMEOUT; |

sNetId: String, which contains the AMS-Network ID of the PC (Type: T_AmsNetId).

nIdn: Parameter number to which FB_SoEReadCommandState refers, e.g. "P_0_IDN + 160" for P-0-0160.

bExecute: The function block is enabled via a positive edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT

| Axis | NCTOPLIC_AXIS_REF;(* reference to NC axis *) |

Axis: Axis data structure of the type NCTOPLIC_AXIS_REF

VAR_OUTPUT

| bBusy   | BOOL; |
| bError  | BOOL; |
| eCmdState | E_SoE_CmdState; |
| iAdsErrId | UINT; |
| iSercosErrId | UINT; |

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command:

eCmdState: 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
Function blocks

3.1.4 Function blocks for diagnostics

3.1.4.1 FB_SoEReadDiagMessage

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

VAR_INPUT

VAR_INPUT

sNetId : T_AmsNetId := '';
bExecute : BOOL;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

sNetId: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
bExecute: The function block is enabled via a positive flank at this input.
tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT

VAR_IN_OUT

Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF
**VAR_OUTPUT**

VAR_OUTPUT
- bBusy : BOOL;
- bError : BOOL;
- iAdsErrId : UINT;
- iSercosErrId : UINT;
- sDiagMessage : ST_SoE_String;
- dwAttribute : DWORD;
END_VAR

**bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.

**bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

**iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.

**iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.

**dwAttribute**: Returns the attributes of the Sercos parameter.

**sDiagMessage**: Returns the diagnosis message.

**Sample**

```plaintext
fbDiagMessage : FB_SoEReadDiagMessage;
bDiagMessage : BOOL;
sDiagMessage : ST_SoE_String;
("NcAxis ")
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bDiagMessage THEN
  fbDiagMessage(
    Axis := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    sDiagMessage=> sDiagMessage
  );
  IF NOT fbDiagMessage.bBusy THEN
    fbDiagMessage(Axis := stNcToPlc, bExecute := FALSE);
    bDiagMessage := FALSE;
  END_IF
END_IF
```

### 3.1.4.2 FB_SoEReadDiagNumber

With the FB_SoEReadDiagNumber function block the current diagnosis number can be read out as UDINT (S-0-0390).

**VAR_INPUT**

VAR_INPUT
- sNetId : T_AmsNetId := '';
- bExecute : BOOL;
- tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

**sNetId**: String containing the AMS network ID of the PC. (Type: T_AmsNetId)

**bExecute**: The function block is enabled via a positive flank at this input.
tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT

VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF

VAR_OUTPUT

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
iAdsErrId : UINT;
iSercosErrId : UINT;
iDiagNumber : UDINT;
dwAttribute : DWORD;
END_VAR

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

dwAttribute: Returns the attributes of the Sercos parameter.

iDiagNumber: Returns the current diagnostic number.

Sample

fbDiagNumber : FB_SoEReadDiagNumber;
bDiagNumber : BOOL;
iDiagNumber : UDINT;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bDiagNumber THEN
  fbDiagNumber(
    Axis := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    iDiagNumber => iDiagNumber
  );
  IF NOT fbDiagNumber.bBusy THEN
    fbDiagNumber(Axis := stNcToPlc, bExecute := FALSE);
    bDiagNumber := FALSE;
  END_IF
END_IF

3.1.4.3 FB_SoEReadDiagNumberList

With the FB_SoEReadDiagNumberList function block a history of the diagnosis numbers can be read out as a list (S-0-0375).
**VAR_INPUT**

- **sNetId:** String containing the AMS network ID of the PC. (type: T_AmsNetId)
- **bExecute:** The function block is enabled via a positive edge at this input.
- **tTimeout:** Maximum time allowed for the execution of the function block.
- **piDiagNumber:** 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 entries.
- **iSize:** Size of the list in bytes (as Sizeof())

**VAR_IN_OUT**

- **Axis:** Axis data structure of the type NCTOPLC_AXIS_REF

**VAR_OUTPUT**

- **bBusy:** This output is set when the function block is activated, and remains set until a feedback is received.
- **bError:** This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
- **iAdsErrId:** Supplies the ADS error code associated with the most recently executed command if the bError output is set.
- **iSercosErrId:** In the case of a set bError output returns the Sercos error of the last executed command.
- **dwAttribute:** Returns the attributes of the Sercos parameter.

**Sample**

```plaintext
fbDiagNumberList : FB_SoEReadDiagNumberList;
bDiagNumberList : BOOL;
stDiagNumberList : ST_SoE_DiagNumList;(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bDiagNumberList THEN
  fbDiagNumberList(
    Axis := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    piDiagNumber := ADR(stDiagNumberList),
    iSize := SIZEOF(stDiagNumberList),
  )
  IF NOT fbDiagNumberList.bBusy THEN
    fbDiagNumberList(Axis := stNcToPlc, bExecute := FALSE);
    bDiagNumberList := FALSE;
  END_IF
ENDIF
```
3.1.4.4 FB_SoEReadClassXDiag

With the FB_SoEReadClassXDiag function block the current class 1 diagnosis (S-0-0011) ... class 3 diagnosis (S-0-0013) can be read 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).

**VAR_INPUT**

```plaintext
VAR_INPUT
sNetId : T_AmsNetId := '';  
bExecute : BOOL;  
iDiagClass : USINT:= 1; (* 1: C1D (S-0-0011) is default, 2: C2D (S-0-0012), 3: C3D (S-0-0013) *)  
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;  
END_VAR
```

- **sNetId**: String containing the AMS network ID of the PC. (Type T_AmsNetId)
- **bExecute**: The function block is enabled via a positive edge at this input.
- **iDiagClass**: 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)
  
  3: Information messages: Class 3 Diag (S-0-0013)
- **tTimeout**: Maximum time allowed for the execution of the function block.

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)  
END_VAR
```

- **Axis**: Axis data structure of the type NCTOPLC_AXIS_REF

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT
bBusy : BOOL;  
bError : BOOL;  
iAdsErrId : UINT;  
iSercosErrId : UINT;  
wClassXDiag : WORD;  
dwAttribute : DWORD;  
END_VAR
```

- **bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.
- **bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
- **iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.
- **iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.
- **wClassXDiag**: Returns the current Class X diagnosis.
**dwAttribute**: Returns the attributes of the Sercos parameter.

**Sample**

```plaintext
fbClassXDiag : FB_SoEReadClassXDiag;
bClassXDiag : BOOL;
idClassXDiag : USINT := 1;
wClass1Diag : WORD;
stAX5000C1D : ST_AX5000_C1D;
wClass2Diag : WORD;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bClassXDiag THEN
    fbClassXDiag(    
        Axis := stNcToPlc,  
        iDiagClass := iDiagClass,  
        tTimeout := DEFAULT_ADS_TIMEOUT
    );
IF NOT fbClassXDiag.bBusy THEN
    fbClassXDiag(Axis := stNcToPlc, bExecute := FALSE);
    CASE fbClassXDiag.iDiagClass OF
        1:
            wClass1Diag := fbClassXDiag.wClassXDiag;
            stAX5000C1D := F_ConvWordToSTAX5000C1D(wClass1Diag);
        2:
            wClass2Diag := fbClassXDiag.wClassXDiag;
    END_CASE
END_IF
END_IF
```

### 3.1.5 Function blocks for determining current values

#### 3.1.5.1 FB_SoERead

With the FB_SoERead function block a parameter can be read.

**VAR_INPUT**

```plaintext
VAR_INPUT
    sNetId : T_AmsNetId := '';
    nIdn : WORD;
    nElement : BYTE;
    pDstBuf : PVOID;
    cbBufLen : UDINT;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

- **sNetId**: String, which contains the AMS-Network ID of the PC (Type: T_AmsNetId).

- **nIdn**: Parameter number to which FB_SoERead refers, e.g. "S_0_IDN + 33" for S-0-0033

- **nElement**: Specifies which part of the parameter should be accessed, e.g. 16#40 is the value (Value) of the parameter.
Function blocks

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: ADR() of the variables to which the value should be read.

cbBufLen: SIZEOF() of the variables to which the value should be read.

bExecute: The function block is enabled via a positive edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT

VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF; (* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF.

VAR_OUTPUT

VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  iAdsErrId : UINT;
  iSercosErrId : UINT;
  dwAttribute : DWORD;
END_VAR

bBusy: This output is set when the function block is activated, and remains set until an acknowledgement is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

dwAttribute: Returns the attributes of the Sercos parameter.

Sample

fbRead : FB_SoERead;
bRead : BOOL;
iReadValue : UINT;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bRead THEN
  nIdn := S_0_IDN + 33;
  fbRead(
    Axis := stNcToPlc,
    nIdn := nIdn,
    nElement := 16#40,
    pDstBuf := ADR(iReadValue),
    cbBufLen := SIZEOF(iReadValue),
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbRead.bBusy THEN
    fbRead(Axis := stNcToPlc, bExecute := FALSE);
With the FB_SoEWrite function block a parameter can be written.

**VAR_INPUT**

```plaintext
VAR_INPUT
  sNetId : T_AmsNetId := '';
  nIdn : WORD;
  nElement : BYTE;
  pSrcBuf : PVOID;
  cbBufLen : UDINT;
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
  sPassword : ST_SoE_String;
END_VAR
```

*sNetId*: String, which contains the AMS-Network ID of the PC (Type: T_AmsNetId).

*nIdn*: Parameter number to which FB_SoERead refers, e.g. "S_0_IDN + 47" for S-0-0047.

*nElement*: 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.

```plaintext
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;
```

*pSrcBuf*: ADR() of the variable containing the value to be written.

*cbBufLen*: SIZEOF() of the variable containing the value to be written.

*bExecute*: The function block is enabled via a positive edge at this input.

*tTimeout*: Maximum time allowed for the execution of the function block.

*sPassword*: Password as a Sercos string. Not currently used. The password must be written with FB_SoEWritePassword [10].
VAR_IN_OUT

Axis: Axis data structure of the type NCTOPLC_AXIS_REF

VAR_OUTPUT

bBusy: This output is set when the function block is activated, and remains set until an acknowledgement is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

Sample

fbWrite : FB_SoEWrite;
bWrite : BOOL;
iWriteValue : UINT;
sPassword : ST_SoE_String;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bWrite THEN
  nIdn := S_0_IDN + 33;
  fbWrite(
    Axis := stNcToPlc,
    nIdn := nIdn,
    nElement := 16#40,
    pSrcBuf := ADR(iWriteValue),
    cbBufLen := SIZEOF(iWriteValue),
    sPassword := sPassword,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbWrite.bBusy THEN
    fbWrite(Axis := stNcToPlc, bExecute := FALSE);
    bWrite := FALSE;
  END_IF
END_IF

3.1.5.3 FB_SoEReadAmplifierTemperature

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

sNetId T_AmsNetId
bExecute BOOL bBusy
Timeout TIME bError
Axis NCTOPLC_AXIS_REF iAdsErrId
BOOL bError
REAL fAmplifierTemperature
DWORD dwAttribute
VAR_INPUT
VAR_INPUT
sNetId : T_AmsNetId := '';
bExecute : BOOL;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

sNetId: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
bExecute: The function block is enabled via a positive flank at this input.
tTimeout: Maximum time allowed for the execution of the function block.

VAR_IN_OUT
VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF

VAR_OUTPUT
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
iAdsErrId : UINT;
iSercosErrId : UINT;
fAmplifierTemperature : REAL;
dwAttribute : DWORD;
END_VAR

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.
bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.
iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.
dwAttribute: Returns the attributes of the Sercos parameter.
fAmplifierTemperature: Returns the drive temperature (e.g. 26.2 corresponds to 26.2 °C).

Sample

fbReadAmplifierTemp :
FB_SoEReadAmplifierTemperature;
bReadAmplifierTemp : BOOL;
fAmplifierTemperature : REAL;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bReadAmplifierTemp THEN
 fbReadAmplifierTemp(
 Axis := stNcToPlc,
 bExecute := TRUE,
 tTimeout := DEFAULT_ADS_TIMEOUT,
 fAmplifierTemperature=>fAmplifierTemperature
 );
IF NOT fbReadAmplifierTemp.bBusy THEN
 fbReadAmplifierTemp(Axis := stNcToPlc, bExecute := FALSE);
bReadAmplifierTemp := FALSE;
END_IF
END_IF
### 3.1.5.4 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.

#### VAR_INPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sNetId</td>
<td>String containing the AMS network ID of the PC.</td>
</tr>
<tr>
<td>bExecute</td>
<td>The function block is enabled via a positive flank at this input.</td>
</tr>
<tr>
<td>tTimeout</td>
<td>Maximum time allowed for the execution of the function block.</td>
</tr>
</tbody>
</table>

- **sNetId**: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
- **bExecute**: The function block is enabled via a positive flank at this input.
- **tTimeout**: Maximum time allowed for the execution of the function block.

#### VAR_IN_OUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis</td>
<td>Axis data structure of the type NCTOPLC_AXIS_REF</td>
</tr>
</tbody>
</table>

- **Axis**: Axis data structure of the type NCTOPLC_AXIS_REF

#### VAR_OUTPUT

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bBusy</td>
<td>This output is set when the function block is activated, and remains set until a feedback is received.</td>
</tr>
<tr>
<td>bError</td>
<td>This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.</td>
</tr>
<tr>
<td>iAdsErrId</td>
<td>In the case of a set bError output returns the ADS error code of the last executed command.</td>
</tr>
<tr>
<td>iSercosErrId</td>
<td>In the case of a set bError output returns the Sercos error of the last executed command.</td>
</tr>
<tr>
<td>fMotorTemperature</td>
<td>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.</td>
</tr>
<tr>
<td>dwAttribute</td>
<td>Returns the attributes of the Sercos parameter.</td>
</tr>
</tbody>
</table>

- **bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.
- **bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
- **iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.
- **iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.
- **fMotorTemperature**: 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.
- **dwAttribute**: Returns the attributes of the Sercos parameter.

#### Sample

```plaintext
fbReadMotorTemp := FB_SoEReadMotorTemperature;
bReadMotorTemp := BOOL;
fMotorTemperature := REAL;
(* NcAxis *)
stNcToPlc AT %I* := NCTOPLC_AXLESTRUCT;
IF bReadMotorTemp AND NOT bInit THEN
  fbReadMotorTemp;
```
Function blocks

```
Axis := stNcToPlc,
bExecute := TRUE,
tTimeout := DEFAULT_ADS_TIMEOUT,
fMotorTemperature=>fMotorTemperature
);
IF NOT fbReadMotorTemp.bBusy THEN
    fbReadMotorTemp(Axis := stNcToPlc, bExecute := FALSE);
bReadMotorTemp := FALSE;
END_IF
END_IF
```

3.1.5.5 FB_SoEReadDcBusCurrent

With the function block FB_SoEAX5000ReadDcBusCurrent the DC-Bus current (S-0-0381) can be read.

**VAR_INPUT**

```plaintext
VAR_INPUT
  sNetId : T_AmsNetId := '';
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

- **sNetId**: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
- **bExecute**: The function block is enabled via a positive flank at this input.
- **tTimeout**: Maximum time allowed for the execution of the function block.

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF; /* reference to NC axis */
END_VAR
```

- **Axis**: Axis data structure of the type NCTOPLC_AXIS_REF

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  iAdsErrId : UINT;
  iSercosErrId : UINT;
  fDcBusCurrent : REAL;
  dwAttribute : DWORD;
END_VAR
```

- **bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.
- **bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
- **iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.
- **iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.
- **dwAttribute**: Returns the attributes of the Sercos parameter.
- **fDcBusCurrent**: Returns the DC bus current (e.g. 2,040 equals 2,040 A).
Sample

```plaintext
Sample

```fbReadDCBusCurrent : FB_SoEReadDCBusCurrent_ByDriveRef;
bReadDCBusCurrent : BOOL;
fDCBusCurrent : REAL;
{" NcAxis "}
stNcToPlc AT %I* : NCTOPLC_AXISSTRUCT;

IF bReadDCBusCurrent THEN
  fbReadDCBusCurrent(
    Axis := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    fDCBusCurrent=>fDCBusCurrent
  );
  IF NOT fbReadDCBusCurrent.bBusy THEN
    fbReadDCBusCurrent(Axis := stNcToPlc, bExecute := FALSE);
    bReadDCBusCurrent := FALSE;
  END_IF
ENDIF
```

### 3.1.5.6 FB_SoEReadDCBusVoltage

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

**VAR_INPUT**

```plaintext
VAR_INPUT
  sNetId : T_AmsNetId := '';
bExecute : BOOL;
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

- `sNetId`: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
- `bExecute`: The function block is enabled via a positive flank at this input.
- `tTimeout`: Maximum time allowed for the execution of the function block.

**VAR_IN_OUT**

```plaintext
VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR
```

- `Axis`: Axis data structure of the type NCTOPLC_AXIS_REF

**VAR_OUTPUT**

```plaintext
VAR_OUTPUT
  bBusy : BOOL;
bError : BOOL;
iAdsErrId : UINT;
iSercosErrId : UINT;
fDCBusVoltage : REAL;
dwAttribute : DWORD;
END_VAR
```

- `bBusy`: This output is set when the function block is activated, and remains set until a feedback is received.
- `bError`: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.

iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.

dwAttribute: Returns the attributes of the Sercos parameter.

fDcBusVoltage: Returns the DC-Bus voltage (e.g. 294.0 corresponds to 294.0 V).

Sample

```plaintext
fbReadDcBusVoltage : FB_SoEReadDcBusVoltage;
bReadDcBusVoltage : BOOL;
fDcBusVoltage : REAL;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bReadDcBusVoltage THEN
  fbReadDcBusVoltage( 
    Axis := stNcToPlc, 
    bExecute := TRUE, 
    tTimeout := DEFAULT_ADS_TIMEOUT, 
    fDcBusVoltage=>fDcBusVoltage 
  );
  IF NOT fbReadDcBusVoltage.bBusy THEN
    fbReadDcBusVoltage(Axis := stNcToPlc, bExecute := FALSE);
    bReadDcBusVoltage := FALSE;
  END_IF
END_IF
```

3.2 General CoE

3.2.1 Function blocks for determining current values

3.2.1.1 FB_CoERead

The function block FB_CoERead allows data to be read from an object dictionary 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.

**VAR_INPUT**

```plaintext
VAR_INPUT
  sNetId    : T_AmsNetId;(*netID of PC with NC*)
  iIndex    : WORD;(*CoE object index*)
  iSubIndex : BYTE;(*CoE sub index*)
  pDstBuf   : PVOID;(*Contains the address of the buffer for the received data*)
  iBufLen   : UDINT;(*Contains the max. number of bytes to be received*)
  bExecute  : BOOL;(*Function block execution is triggered by a rising edge at this input.*)
  tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
```
sNetId: String containing the AMS network ID of the PC. (type: T_AmsNetId)

iIndex: Index of the object that is to be read.

iSubIndex: Subindex of the object that is to be read.

pDstBuf: Address (pointer) to the receive buffer

iBufLen: Maximum available buffer size (in bytes) for the data to be read

bExecute: The function block is enabled via a positive edge at this input.

tTimeout: Maximum time allowed for the execution of the function block.

bCompleteAccess: Via Complete Access the complete object can be accessed at once.

VAR_OUTPUT

bBusy : BOOL;
bError : BOOL;
iAdsErrId : UINT;
iCANopenErrId : UINT;
END_VAR

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.

bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

iAdsErrId: In the event of a set bError output returns the ADS error code.

iCANopenErrId: In the event of a set bError output returns the CANopen error code.

VAR_IN_OUT

Axis : NCTOPLC_AXIS_REF;
END_VAR

Axis: Axis data structure of type NCTOPLC_AXIS_REF, which addresses an axis uniquely within the system. Among other things it contains the current status of the axis such as the position, the velocity and the error state.

3.2.1.2 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.
VAR_INPUT

VAR_INPUT
sNetId : T_AmsNetId; (*netID of PC with NC*)
iIndex : WORD; (*CoE object index*)
iSubIndex : BYTE; (*CoE sub index*)
pDstBuf : PVOID; (*Contains the address of the buffer for the received data*)
iBufLen : UDINT; (*Contains the max. number of bytes to be received*)
bExecute : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
(*States the time before the function is cancelled.*)
bCompleteAccess : BOOL; (*Function block reads the complete object with all sub index*)

sNetId: String containing the AMS network ID of the PC. (type: T_AmsNetId)
iIndex: Index of the object that is supposed to be written.
iSubIndex: Sub-index of the object that is supposed to be written.
pDstBuf: Address (pointer) to the send buffer
iBufLen: Maximum available buffer size (in bytes) for the data to be read
bExecute: The function block is enabled via a positive edge at this input.
tTimeout: Maximum time allowed for the execution of the function block.
bCompleteAccess: Via Complete Access the complete object can be accessed at once.

VAR_OUTPUT

VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
iAdsErrId : UINT;
iCANopenErrId : UINT;

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.
bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId: In the event of a set bError output returns the ADS error code.
iCANopenErrId: In the event of a set bError output returns the CANopen error code.

VAR_IN_OUT

VAR_IN_OUT
Axis : NCTOPLC_AXIS_REF;

Axis: Axis data structure of type NCTOPLC_AXIS_REF, which addresses an axis uniquely within the system. Among other things it contains the current status of the axis such as the position, the velocity and the error state.
With the FB_SoEAX5000ReadActMainVoltage function block the current peak value of the mains voltage of the AX5000 (P-0-0200) can be read.

**VAR_INPUT**

```
VAR_INPUT
  sNetId : T_AmsNetId := '';
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

- **sNetId**: String containing the AMS network ID of the PC. (Type: T_AmsNetId)
- **bExecute**: The function block is enabled via a positive flank at this input.
- **tTimeout**: Maximum time allowed for the execution of the function block.

**VAR_IN_OUT**

```
VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF;(* reference to NC axis *)
END_VAR
```

- **Axis**: Axis data structure of the type NCTOPLC_AXIS_REF

**VAR_OUTPUT**

```
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  iAdsErrId : UINT;
  iSercosErrId : UINT;
  dwAttribute : DWORD;
  fActualMainVoltage : LREAL;
END_VAR
```

- **bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.
- **bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
- **iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.
- **iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.
- **dwAttribute**: Returns the attributes of the Sercos parameter.
- **fActualMainVoltage**: Returns the peak value of the current mains voltage of the AX5000 (e.g. 303.0 corresponds to 303.0 V).
### 3.3.2 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.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>sNetId</th>
<th>T_AmsNetId</th>
</tr>
</thead>
<tbody>
<tr>
<td>bExecute</td>
<td>BOOL</td>
</tr>
<tr>
<td>tTimeout</td>
<td>TIME := DEFAULT_ADS_TIMEOUT</td>
</tr>
<tr>
<td>bForceLock</td>
<td>BOOL</td>
</tr>
<tr>
<td>bForceUnlock</td>
<td>BOOL</td>
</tr>
<tr>
<td>Axis</td>
<td>NCTOPLC_AXIS_REF</td>
</tr>
</tbody>
</table>

**VAR_INPUT**

- **sNetId**: String containing the AMS network ID of the PC. (type: T_AmsNetId)
- **bExecute**: The function block is enabled via a positive edge at this input.
- **tTimeout**: Maximum time allowed for the execution of the function block.
- **bForceLock**: Activates the brake independently of the enable.
- **bForceUnlock**: Releases the brake independently of the enable.

**VAR_IN_OUT**

<table>
<thead>
<tr>
<th>Axis</th>
<th>NCTOPLC_AXIS_REF</th>
</tr>
</thead>
</table>

**Axis**: Axis data structure of the type NCTOPLC_AXIS_REF
VAR_OUTPUT

VAR_OUTPUT
- bBusy : BOOL;
- bError : BOOL;
- iAdsErrId : UINT;
- iSercosErrId : UINT;
END_VAR

**bBusy**: This output is set when the function block is activated, and remains set until a feedback is received.

**bError**: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.

**iAdsErrId**: In the case of a set bError output returns the ADS error code of the last executed command.

**iSercosErrId**: In the case of a set bError output returns the Sercos error of the last executed command.

**Sample**

```plaintext
Sample

fbSetMotorCtrlWord : FB_SoEAX5000SetMotorCtrlWord;
bSetMotorCtrlWord : BOOL;
bForceLock : BOOL;
bForceUnlock : BOOL;
("NcAxis *")
stNcToPlc AT %I*: NCTOPLC_AXLESTRUCT;
IF bSetMotorCtrlWord THEN
  fbSetMotorCtrlWord(
    Axis := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    bForceLock := bForceLock,
    bForceUnlock:= bForceUnlock
  );
  IF NOT fbSetMotorCtrlWord.bBusy THEN
    fbSetMotorCtrlWord(Axis := stNcToPlc, bExecute := FALSE);
    bSetMotorCtrlWord := FALSE;
  END_IF
END_IF
```

### 3.3.3 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.

**NOTE**

**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.

**VAR_INPUT**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sNetId</td>
<td>AMS-NetID of the controller (IPC).</td>
</tr>
<tr>
<td>bExecute</td>
<td>The function block is activated by a positive edge at this input.</td>
</tr>
<tr>
<td>sFirmwareVersion</td>
<td>Specifies the desired firmware version in the form of vx.yy_bnnnn, e.g. &quot;v1.05_b0009&quot; for Version v1.05 Build 0009.</td>
</tr>
<tr>
<td>sFirmwarePath</td>
<td>*drive:\path, e.g. &quot;C:\TwinCAT\Io\TcDriveManagerFirmwarePool&quot;</td>
</tr>
<tr>
<td>iReqEcState</td>
<td>UINT := EC_DEVICE_STATE_OP;</td>
</tr>
<tr>
<td>tTimeout</td>
<td>TIME := DEFAULT_ADS_TIMEOUT;</td>
</tr>
</tbody>
</table>

sNetId: AMS-NetID of the controller (IPC).

bExecute: The function block is activated by a positive edge at this input.

sFirmwareVersion: Specifies the desired firmware version in the form of vx.yy_bnnnn, e.g. "v1.05_b0009" for Version v1.05 Build 0009.

Release-Builds:

• "v1.05_b0009" for a specific build, e.g. v1.05 Build 0009
• "v1.05_b00??" latest build of a specified version, e.g. v1.05
• "v1.??_b00??" latest build of a specified main version, e.g. v1
• "v.??_b00??" latest build of the latest version
• "*" latest build of the latest version

Custom Firmware-Builds:

• "v1.05_b1009" for a specific build, e.g. v1.05 Build 0009
• "v1.05_b10??" latest build of a specified version, e.g. v1.05
• "v1.??_b10??" latest build of a specified main version, e.g. v1
• "v.??_b10??" latest build of the latest version

...
Function blocks

• "v?.??_b9??" latest build of the latest version

Debug-Builds:
• "v1.05_b9009" for a specific build, e.g. v1.05 Build 9009
• "v1.05_b90??" latest build of a specified version, e.g. v1.05
• "v1.??.b90??" latest build of a specified main version, e.g. v1
• "v?.??_b90??" latest build of the latest version

sFirmwarePath: Specifies the path for the firmware pool in which the firmware files are located, e.g. C:\TwinCAT\Io\TcDriveManager\FirmwarePool.

iReqEcState: Desired EtherCAT state after the update (only if an update is actually being executed). The statuses are defined in PLC Lib Tc2_EtherCAT as global constants.

tTimeout: Since the firmware update for large EtherCAT networks can take longer, only the timeout for individual internal ADS instances is specified here.

VAR IN_OUT

VAR_IN_OUT
  Axis : NCTOPLC_AXLESTRUCT;(* reference to NC axis *)
END_VAR

Axis: Axis data structure of the type NCTOPLC_AXIS_REF

VAR OUTPUT

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

bBusy: This output is set when the function block is activated, and remains set until a feedback is received.
bError: This output is set after the bBusy output has been reset when an error occurs in the transmission of the command.
iAdsErrId: In the case of a set bError output returns the ADS error code of the last executed command.
iSercosErrId: In the case of a set bError output returns the Sercos error of the last executed command.
iDiagNumber: In the case of a set bError output returns the drive error of the last executed firmware update.
eFwUpdateState: Returns the status of the firmware update.
iLoadProgress: Returns the progress of the actual firmware update as a percentage.
sSelectedFirmwareFile: 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;
stNcToPlc AT i*: ARRAY [1..iNumOfDrives] OF NCTOPLC_AXLESTRUCT;
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 := stNcToPlc[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 := stNcToPlc[I],
        eFwUpdateState => eFwUpdateState[I],
        sSelectedFirmwareFile => sSelectedFirmwareFile[I],
      );
      IF NOT fbFirmwareUpdate[I].bBusy THEN
        fbFirmwareUpdate[I](bExecute := FALSE, Axis := stNcToPlc[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

3.4  F_GetVersionTcNcDrive

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

FUNCTION F_GetVersionTcNcDrive: UINT
VAR_INPUT
  nVersionElement : INT;
END_VAR

nVersionElement: Version element to be read. Possible parameters:

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