

Handbuch | DE

# PLC-Bibliothek: TcNCDrive

TwinCAT 2 | TX1200, PlcNc, TcNcUtilities



TwinCAT Motion





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Sicherheitshinweise .....	6
<b>2</b>	<b>POUs der TcNcDrive.lib</b> .....	<b>7</b>
<b>3</b>	<b>Allgemeine SoE FBs</b> .....	<b>9</b>
3.1	FB_SoEReset .....	9
3.2	FB_SoEWritePassword .....	10
3.3	Kommando FBs .....	11
3.3.1	FB_SoEExecuteCommand .....	11
3.3.2	FB_SoEWriteCommandControl .....	13
3.3.3	FB_SoEReadCommandState .....	14
3.4	Diagnose FBs .....	16
3.4.1	FB_SoEReadDiagMessage .....	16
3.4.2	FB_SoEReadDiagNumber .....	17
3.4.3	FB_SoEReadDiagNumberList .....	18
3.4.4	FB_SoEReadClassXDiag .....	20
3.5	FBs für aktuelle Werte .....	21
3.5.1	FB_SoERead .....	21
3.5.2	FB_SoEWrite .....	23
3.5.3	FB_SoEReadAmplifierTemperature .....	24
3.5.4	FB_SoEReadMotorTemperature .....	26
3.5.5	FB_SoEReadDcBusCurrent .....	27
3.5.6	FB_SoEReadDcBusVoltage .....	28
<b>4</b>	<b>AX5000 spezifische FBs</b> .....	<b>30</b>
4.1	FB_SoEAX5000ReadActMainVoltage .....	30
4.2	FB_SoEAX5000SetMotorCtrlWord .....	31
4.3	FB_SoEAX5000FirmwareUpdate .....	32
<b>5</b>	<b>F_GetVersionTcNcDrive</b> .....	<b>36</b>
<b>6</b>	<b>E_FwUpdateState</b> .....	<b>37</b>



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

**EtherCAT** 

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Sicherheitshinweise

### Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!  
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

#### **GEFAHR**

##### **Akute Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### **WARNUNG**

##### **Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

#### **VORSICHT**

##### **Schädigung von Personen!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

#### **HINWEIS**

##### **Schädigung von Umwelt oder Geräten**

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.



##### **Tipp oder Fingerzeig**

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

## 2 POU's der TcNcDrive.lib

In dieser Bibliothek sind Funktionen und Funktionsbausteine für SoE-Antriebe enthalten, die per NC-Achsstruktur auf den Antrieb zugreifen.

Die TcNcDrive.lib enthält Wrapper-Bausteine um die Bausteine der TcDrive.lib.

Unterschiede bei der Verwendung der Drive-Libs in Verbindung mit AX5000 und mit Bosch-Rexroth IndraDrive CS sind zu berücksichtigen. Beispiel siehe unten.

Die TcNcDrive.lib sollte dann verwendet werden, wenn der Antrieb über die NC mit den Bibliotheken TcNc.lib bzw. TcMc.lib verwendet wird. Hierzu wird auf den Antrieb über die NC-Achsstruktur (NC\_TO\_PLC) zugegriffen, die auch in den Bausteinen der TcNc.lib bzw. TcMc.lib verwendet werden. Die Bausteine in der TcNcDrive.lib ermitteln eigenständig über die NC-AchsID aus der NC\_TO\_PLC-Struktur die Zugriffsdaten auf den Antrieb (NetID, Adresse und Kanalnummer). Siehe Beispiel bei den jeweiligen Funktionsbausteinen in der Dokumentation der TcNcDrive.lib.

Hinweis: Um auf Parameter im Antrieb zuzugreifen, für die kein spezieller Baustein implementiert wurde, können die Bausteine FB\_SoERead und FB\_SoEWrite verwendet werden.

### Funktionen

Name	Beschreibung
<a href="#">F_GetVersionTcNcDrive [► 36]</a>	Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.
<a href="#">F_ConvWordToSTAX5000C1D</a>	Siehe Dokumentation TcDrive.lib.  Konvertiert das C1D-Wort (S-0-0011) des AX5000 in eine Struktur ST_AX5000_C1D.

### Funktionsbausteine

Name	Beschreibung
<a href="#">FB_SoEReset [► 9]</a>	Antriebsreset ausführen (S-0-0099)
<a href="#">FB_SoEWritePassword [► 10]</a>	Setzen des Antriebspassworts (S-0-0267)
<a href="#">FB_SoEReadDiagMessage [► 16]</a>	Lesen der Diagnosenachricht (S-0-0095)
<a href="#">FB_SoEReadDiagNumber [► 17]</a>	Lesen der Diagnosenummer (S-0-0390)
<a href="#">FB_SoEReadDiagNumberList [► 18]</a>	Lesen der Diagnosenummernliste (bis zu 30 Einträge) (S-0-0375)
<a href="#">FB_SoEExecuteCommand [► 11]</a>	Ausführen eines Kommandos
<a href="#">FB_SoEWriteCommandControl [► 13]</a>	Setzen des Command Control
<a href="#">FB_SoEReadCommandState [► 14]</a>	Prüfen des Kommandostatus
<a href="#">FB_SoEReadClassXDiag [► 20]</a>	Lesen der Class 1 Diagnose (S-0-0011) ... Class 3 Diagnose (S-0-0013)
<a href="#">FB_SoERead [► 21]</a>	Lesen eines Parameters
<a href="#">FB_SoEWrite [► 23]</a>	Schreiben eines Parameters
<a href="#">FB_SoEReadAmplifierTemperature [► 24]</a>	Lesen der Antriebstemperatur (S-0-0384)

Name	Beschreibung
FB_SoEReadMotorTemperature [► 26]	Lesen der Motortemperatur (S-0-0383)
FB_SoEReadDcBusCurrent [► 27]	Lesen des Dc-Bus-Stroms (S-0-0381)
FB_SoEReadDcBusVoltage [► 28]	Lesen der Dc-Bus-Spannung (S-0-0380)
FB_SoEAX5000ReadActMainVoltage [► 30]	Lesen der Netzspannung (P-0-0200)
FB_SoEAX5000SetMotorCtrlWord [► 31]	Setzen des Motor Control Words (P-0-0096)
FB_SoEAX5000FirmwareUpdate [► 32]	Automatischer Firmware-Update des AX5000

### Beispielprojekt und Beispielkonfiguration für AX5000-Diagnose

Siehe <https://infosys.beckhoff.com/content/1031/tcplclibncdrive/Resources/pro/10954810635.pro>, <https://infosys.beckhoff.com/content/1031/tcplclibncdrive/Resources/tsm/10954813579.tsm>

### Beispielprojekt und Beispielkonfiguration für IndraDriveCS-Diagnose

Siehe <https://infosys.beckhoff.com/content/1031/tcplclibncdrive/Resources/pro/10954814987.pro>, <https://infosys.beckhoff.com/content/1031/tcplclibncdrive/Resources/tsm/10954816395.tsm> (ab TcNcDrive.lib v0.0.25)

### Anforderungen

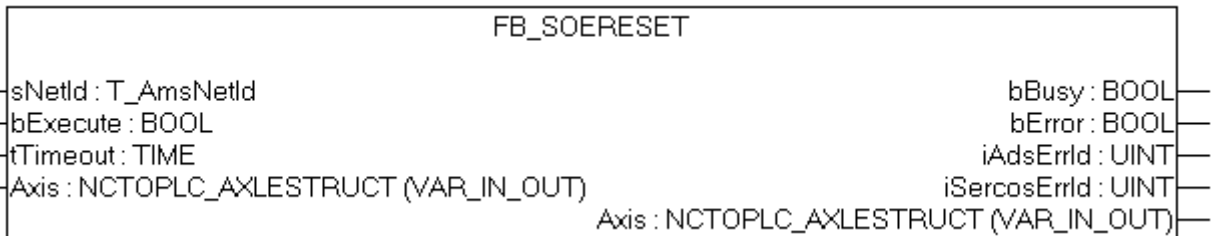
### Voraussetzungen

Komponente	Version
TwinCAT auf dem Entwicklungsrechner	2.10 Build 1335 oder höher
TwinCAT auf dem Windows CE-Image	2.10 Build 1333 oder höher
TwinCAT auf dem Windows XP-Image	2.10 Build 1333 oder höher



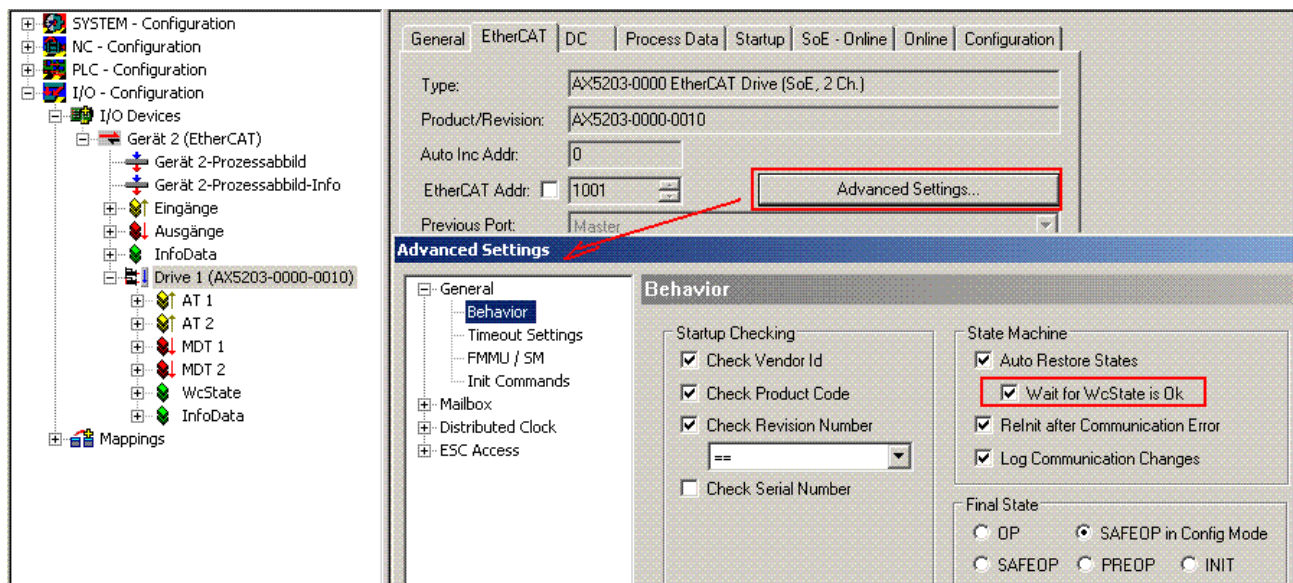
### 3 Allgemeine SoE FBs

#### 3.1 FB\_SoEReset



Mit dem Funktionsbaustein FB\_SoEReset kann ein Antriebsreset (S-0-0099) ausgeführt werden. Bei mehrkanaligen Geräten müssen ggf. beide Kanäle einen Reset ausführen. Die Timeoutzeit muss 10s betragen, da der Reset je nach Fehler bis zu 10s dauern kann.

Für den AX5000 muss das Flag "Wait For WcState is OK" in den Advanced EtherCAT Settings aktiviert sein.



Ein NC-Reset wird nicht ausgeführt. Falls ein NC-Reset nötig ist, kann er über den MC\_Reset-Baustein aus der TcMc.lib ausgeführt werden.

#### VAR\_INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

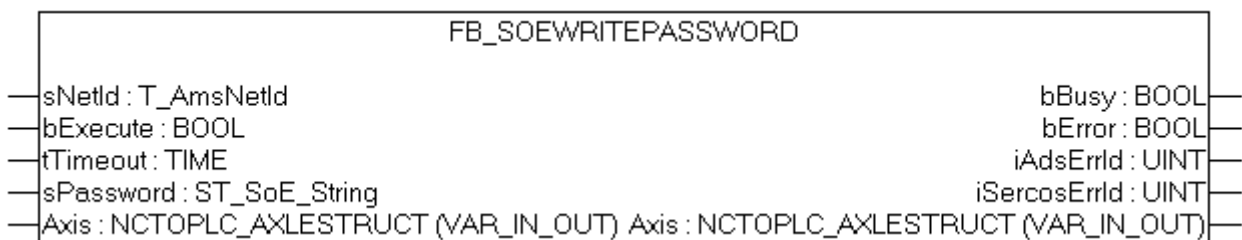
**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**Beispiel**

```
fbSoEReset : FB_SoEReset_ByDriveRef;
bSoEReset : BOOL;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bSoEReset THEN
  fbSoEReset(
    Axis      := stNcToPlc,
    bExecute  := TRUE,
    tTimeout  := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbSoEReset.bBusy THEN
    fbSoEReset(Axis := stNcToPlc, bExecute := FALSE);
    bSoEReset := FALSE;
  END_IF
END_IF
```

**3.2 FB\_SoEWritePassword**

Mit dem Funktionsbaustein FB\_SoEWritePassword kann das Antriebspasswort (S-0-0267) gesetzt werden.

**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**sPassword:** enthält das Passwort als Sercos-String

**VAR\_IN\_OUT**

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

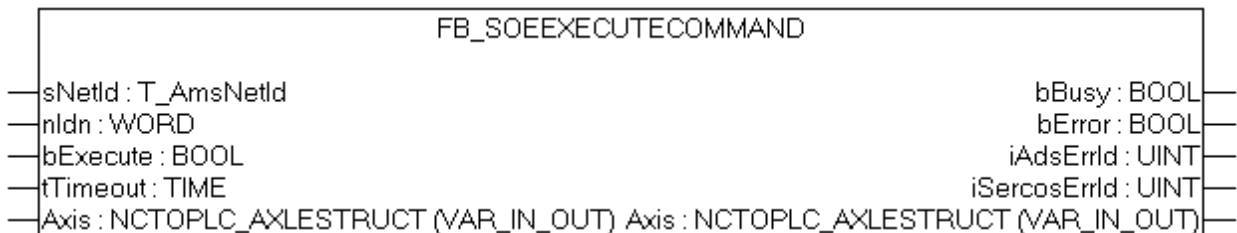
**Beispiel**

```
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
```

**3.3 Kommando FBs**

**3.3.1 FB\_SoEExecuteCommand**



Mit dem Funktionsbaustein FB\_SoEExecuteCommand kann ein Kommando ausgeführt werden.

**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**nIdn:** Parameternummer, auf das sich das FB\_SoEExecuteCommand\_ByDriveRef bezieht, "P\_0\_IDN + 160" für P-0-0160

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**VAR\_IN\_OUT**

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

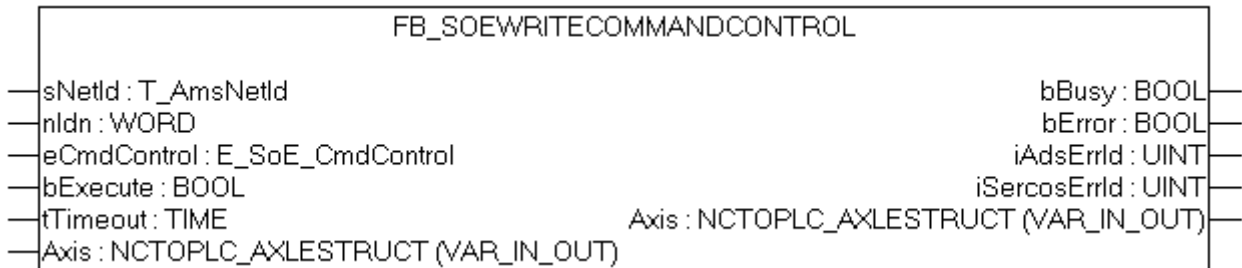
**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**Beispiel**

```
fbExecuteCommand : FB_SoEExecuteCommand;
bExecuteCommand  : BOOL;
nIdn             : WORD;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bExecuteCommand THEN
  nIdn := P_0_IDN + 160;
  fbExecuteCommand(
    Axis       := stNcToPlc,
    bExecute  := TRUE,
    tTimeout  := DEFAULT_ADS_TIMEOUT,
    nIdn     := nIdn,
  );
  IF NOT fbExecuteCommand.bBusy THEN
    fbExecuteCommand(Axis := stNcToPlc, bExecute := FALSE);
    bExecuteCommand := FALSE;
  END_IF
END_IF
```

### 3.3.2 FB\_SoEWriteCommandControl



Mit dem Funktionsbaustein FB\_SoEWriteCommandControl kann ein Kommando vorbereitet, gestartet oder abgebrochen werden.

#### VAR\_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId := '';
  nIdn        : WORD;
  eCmdControl : E_SoE_CmdControl;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**nIdn:** Parameternummer, auf das sich das FB\_SoEWriteCommandControl\_ByDriveRef bezieht, z.B. "P\_0\_IDN + 160" für P-0-0160

**eCmdControl:** Gibt an, ob das vorbereitet (eSoE\_CmdControl\_Set := 1), ausgeführt (eSoE\_CmdControl\_SetAndEnable := 3) oder abgebrochen (eSoE\_CmdControl\_Cancel := 0) werden soll

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

```
VAR_OUTPUT
  bbBusy      : BOOL;
  bError      : BOOL;
  iAdsErrId   : UINT;
  iSercosErrId : UINT;
END_VAR
```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

#### Beispiel

```
fbWriteCommandControl : FB_SoEWriteCommandControl;
bWriteCommandControl  : BOOL;
nIdn                  : WORD;
```

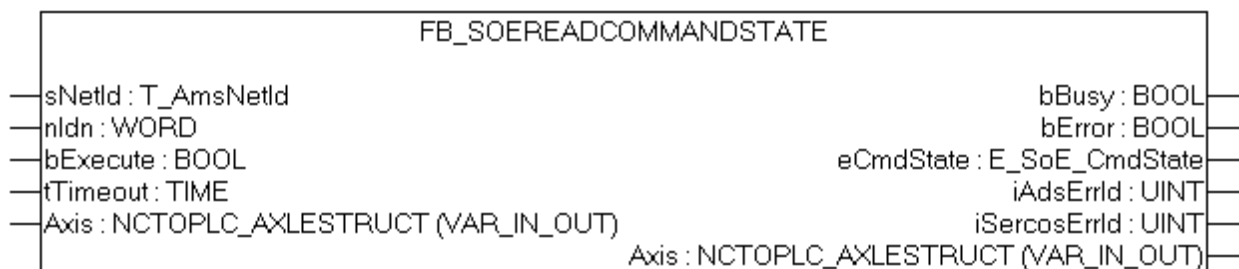
```

eCmdControl      : E_SoE_CmdControl;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bWriteCommandControl THEN
  nIdn := P_0_IDN + 160;
  fbWriteCommandControl(
    Axis      := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    nIdn := nIdn,
    eCmdControl := eCmdControl
  );
IF NOT fbWriteCommandControl.bBusy THEN
  fbWriteCommandControl(Axis := stNcToPlc, bExecute := FALSE);
  bWriteCommandControl := FALSE;
END_IF
END_IF

```

### 3.3.3 FB\_SoEReadCommandState



Mit dem Funktionsbaustein FB\_SoEReadCommandState kann die Kommandoausführung überprüft werden.

#### VAR\_INPUT

```

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

```

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**Idn:** Parameternummer, auf das sich das FB\_SoEReadCommandState\_ByDriveRef bezieht, z.B. "P\_0\_IDN + 160" für P-0-0160

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

```

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

```

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

```

VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  eCmdState : E_SoE_CmdState;
  iAdsErrId : UINT;
  iSercosErrId : UINT;
END_VAR

```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

**eCmdState:** Liefert den Kommandostatus

```
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
```

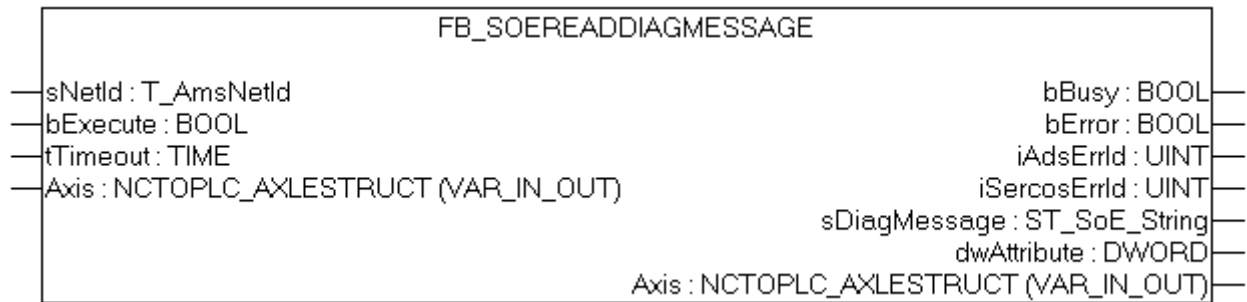
## Beispiel

```
fbReadCommandState : FB_SoEReadCommandState;
bReadCommandState : BOOL;
nIdn                : WORD;
eCmdState           :E_SoE_CmdState;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bReadCommandState THEN
  nIdn := P_0_IDN + 160;
  fbReadCommandState(
    Axis      := stNcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    nIdn     := nIdn,
    eCmdState => eCmdState
  );
  IF NOT fbReadCommandState.bBusy THEN
    fbReadCommandState(Axis := stNcToPlc, bExecute := FALSE);
    bReadCommandState := FALSE;
  END_IF
END_IF
```

## 3.4 Diagnose FBs

### 3.4.1 FB\_SoEReadDiagMessage



Mit dem Funktionsbaustein FB\_SoEReadDiagMessage kann die Diagnosenachricht als Sercos-String (S-0-0095) ausgelesen werden.

#### VAR\_INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

**sDiagMessage:** Liefert die Diagnosenachricht.

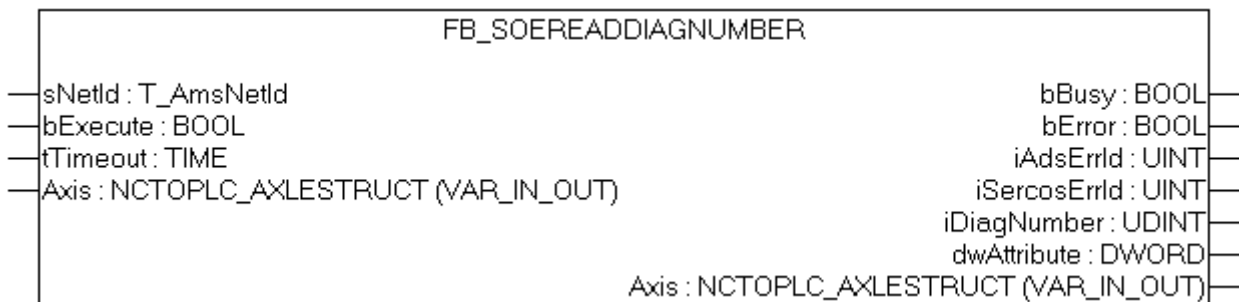


**Beispiel**

```
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
```

**3.4.2 FB\_SoEReadDiagNumber**



Mit dem Funktionsbaustein FB\_SoEReadDiagNumber kann die aktuelle Diagnosenummer als UDINT (S-0-0390) ausgelesen werden.

**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**VAR\_IN\_OUT**

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iAdsErrId  : UINT;
  iSercosErrId : UINT;
```

```

    iDiagNumber      : UDINT;
    dwAttribute      : DWORD;
END_VAR

```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

**iDiagNumber:** Liefert die aktuelle Diagnosenummer.

### Beispiel

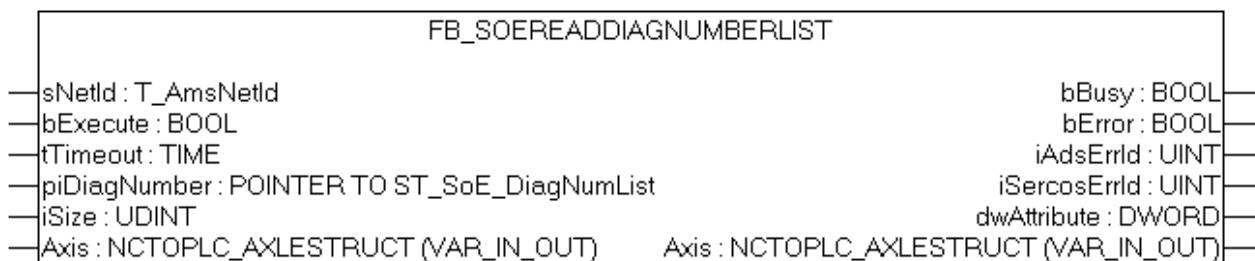
```

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

```

## 3.4.3 FB\_SoEReadDiagNumberList



Mit dem Funktionsbaustein FB\_SoEReadDiagNumberList kann eine Historie der Diagnosenummern als Liste (S-0-0375) ausgelesen werden.

### VAR\_INPUT

```

VAR_INPUT
    sNetId      : T_AmsNetId := '';
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
    piDiagNumber : POINTER TO ST_SoE_DiagNumList;
    iSize       : UDINT;
END_VAR

```

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**piDiagNumber:** Zeiger auf Liste der letzten max. 30 Fehlernummern. Die Liste besteht aus aktueller und maximaler Anzahl von Bytes in der Liste, sowie den 30 Listeneinträgen

**iSize:** Größe der Liste in Bytes (als Sizeof())

## VAR\_IN\_OUT

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

**Axis:** Achsstruktur

## VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei einem gesetzten bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei einem gesetzten bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

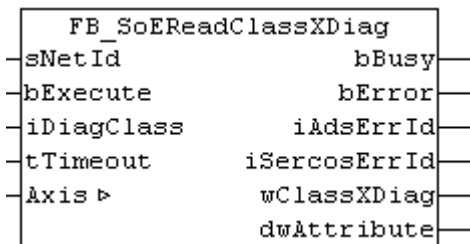
**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

## Beispiel

```
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
END_IF
```

### 3.4.4 FB\_SoEReadClassXDiag



Mit dem Funktionsbaustein FB\_SoEReadClassXDiag kann die aktuelle Class 1 Diagnose(S-0-0011) ... Class 3 Diagnose (S-0-0013) als WORD ausgelesen werden. Für die Auswertung der Class 1 Diagnose als Struktur ST\_AX5000\_C1D gibt es eine Konvertierungsfunktion F\_ConvWordToSTAX5000C1D. Siehe Dokumentation TcDrive.lib.

#### VAR\_INPUT

```
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:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**iDiagClass:** Gibt an, welche Diagnose gelesen werden soll. Die Diagnose Parameter können sich von Hersteller zu Hersteller unterscheiden. Nicht immer sind alle Diagnose Parameter (C1D ... C3D) oder alle Bits darin implementiert.

- 1: Fehler: Class 1 Diag (S-0-0011)
- 2: Warnungen: Class 2 Diag (S-0-0012)
- 3: Informationen: Class 3 Diag (S-0-0013)

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**wClassXDiag:** Liefert die aktuelle Class X Diagnose.

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```
fbClassXDiag : FB_SoEReadClassXDiag;
bClassXDiag  : BOOL;
iDiagClass   : USINT := 1;
wClass1Diag  : WORD;
stAX5000C1D  : ST_AX5000_C1D;
wClass2Diag  : WORD;

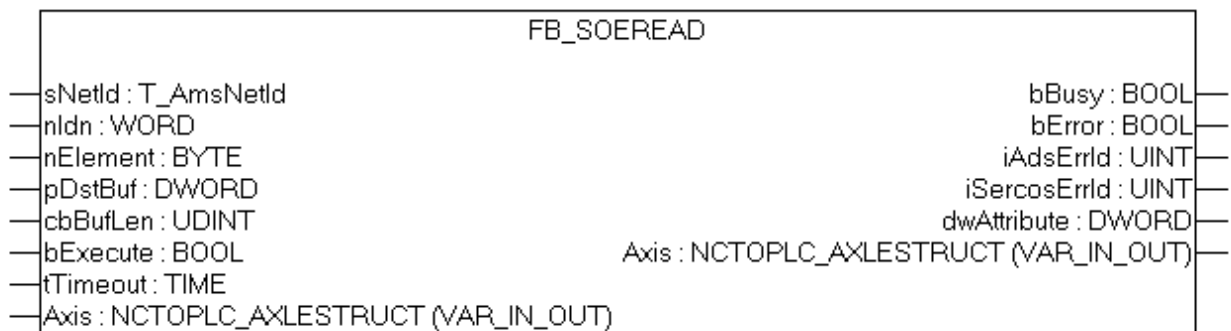
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bClassXDiag THEN
  fbClassXDiag(
    Axis      := stNcToPlc,
    bExecute  := TRUE,
    iDiagClass := iDiagClass,
    tTimeout  := DEFAULT_ADS_TIMEOUT
  );
  IF NOT fbClassXDiag.bBusy THEN
    fbClassXDiag(Axis := stNcToPlc, bExecute := FALSE);
    bClassXDiag := FALSE;

    CASE fbClassXDiag.iDiagClass OF
    1:
      wClass1Diag := fbClassXDiag.wClassXDiag;
      stAX5000C1D := F_ConvWordToSTAX5000C1D(wClass1Diag);

    2:
      wClass2Diag := fbClassXDiag.wClassXDiag;
    END_CASE
  END_IF
END_IF
```

### 3.5 FBs für aktuelle Werte

#### 3.5.1 FB\_SoERead



Mit dem Funktionsbaustein FB\_SoERead kann ein Parameter eingelesen werden.

**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**nIdn:** Parameternummer, auf das sich das FB\_SoERead bezieht, "S\_0\_IDN + 33" für S-0-0033

**nElement:** Gibt an, auf welchen Teil des Parameters zugegriffen werden soll, z.B. 16#40 ist der Wert (Value) des Parameters

```

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() der Variablen, in die der Wert gelesen werden soll.

**cbBufLen:** SIZEOF() der Variablen, in die der Wert gelesen werden soll.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

## VAR\_IN\_OUT

```

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

```

**Axis:** Achsstruktur.

## VAR\_OUTPUT

```

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

```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

## Beispiel

```

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);

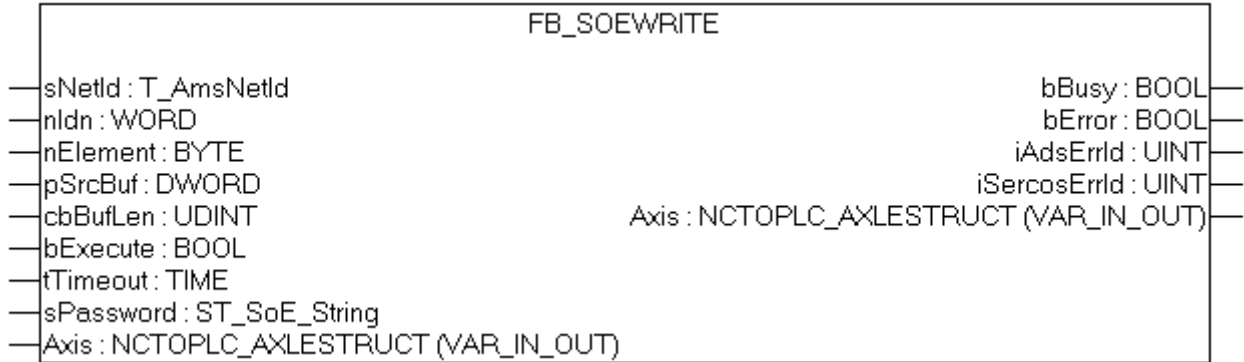
```

```

        bRead := FALSE;
    END_IF
END_IF

```

### 3.5.2 FB\_SoEWrite



Mit dem Funktionsbaustein FB\_SoEWrite kann ein Parameter geschrieben werden.

#### VAR\_INPUT

```

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

```

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**nIdn:** Parameternummer, auf das sich das FB\_SoERead bezieht, "S\_0\_IDN + 47" für S-0-0047

**nElement:** Gibt an, auf welchen Teil des Parameters zugegriffen werden soll, z.B. 16#40 ist der Wert (Value) des Parameters. Meist kann nur auf den Wert schreibend zugegriffen werden, andere Bestandteile des Parameters sind schreibgeschützt.

```

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() der Variablen, die den zu schreibenden Wert enthält.

**cbBufLen:** sizeof() der Variablen, die den zu schreibenden Wert enthält

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**sPassword:** enthält das Passwort als Sercos-String. Wird z.Z. noch nicht verwendet. Das Passwort muss mit [FB\\_SoEWritePassword](#) ([10](#)) geschrieben werden.

#### VAR\_IN\_OUT

```

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

```

**Axis:** Achsstruktur.

## VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

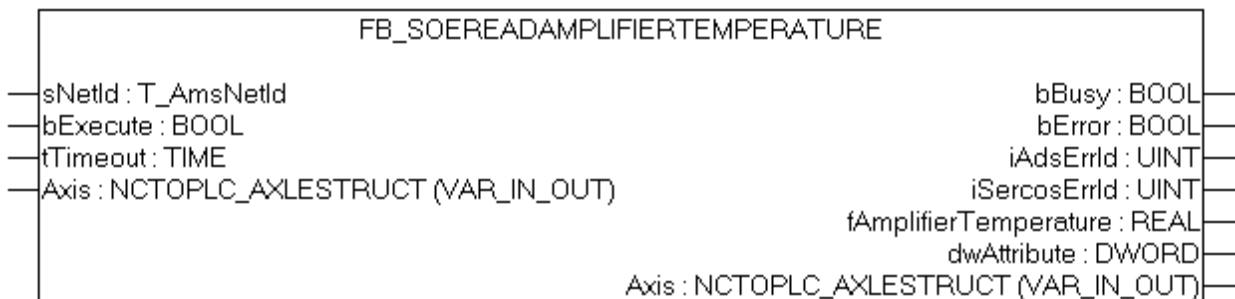
**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

## Beispiel

```
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
```

### 3.5.3 FB\_SoEReadAmplifierTemperature



Mit dem Funktionsbaustein FB\_SoEReadAmplifierTemperature kann die Temperatur des Antriebs (S-0-0384) eingelesen werden.



**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**VAR\_IN\_OUT**

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

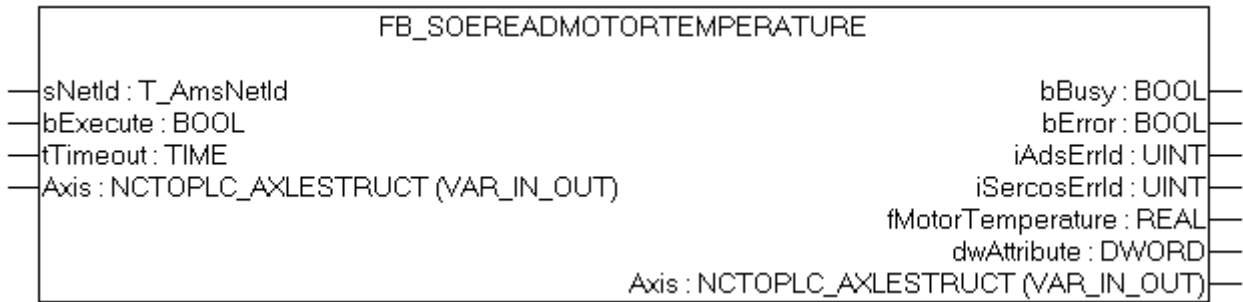
**fAmplifierTemperature:** Liefert die Antriebstemperatur (z.B. 26.2 entspricht 26.2°C).

**Beispiel**

```
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
```

### 3.5.4 FB\_SoEReadMotorTemperature



Mit dem Funktionsbaustein FB\_SoEReadMotorTemperature kann die Temperatur des Motor (S-0-0383) eingelesen werden. Falls der Motor keinen Temperatursensor enthält, steht hier 0.0, heißt 0.0°C.

#### VAR\_INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

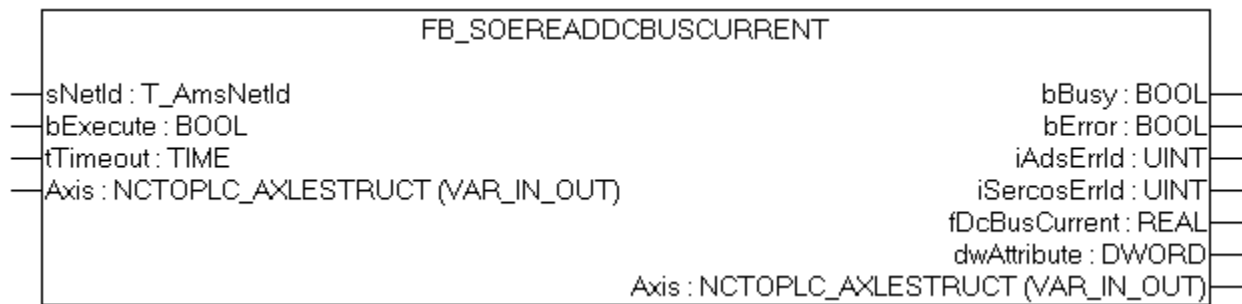
**fMotorTemperature:** Liefert die Motortemperatur (z.B. 30.5 entspricht 30.5°C). Falls der Motor keinen Temperatursensor enthält, steht hier 0.0, heißt 0.0°C.

#### Beispiel

```
fbReadMotorTemp : FB_SoEReadMotorTemperature;
bReadMotorTemp  : BOOL;
fMotorTemperature : REAL;
```

```
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bReadMotorTemp AND NOT bInit THEN
  fbReadMotorTemp(
    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
```

### 3.5.5 FB\_SoEReadDcBusCurrent



Mit dem Funktionsbaustein FB\_SoEAX5000ReadDcBusCurrent kann der DC-Bus-Strom (S-0-0381) eingelesen werden.

#### VAR INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

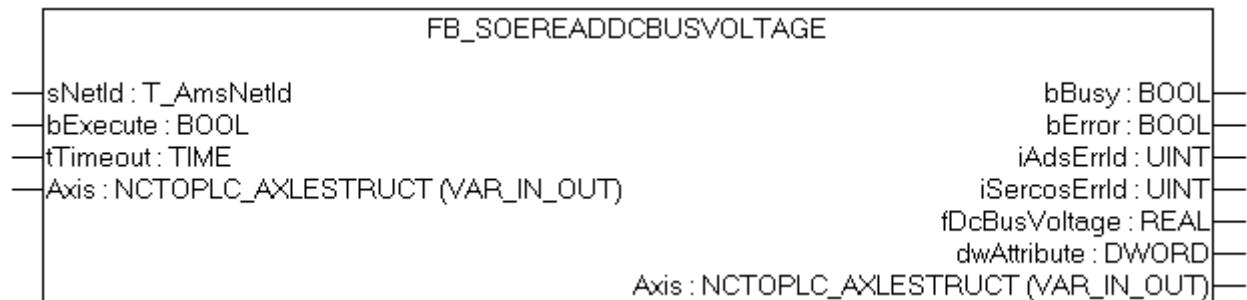
**fDcBusCurrent:** Liefert den DC-Bus-Strom (z.B. 2.040 entspricht 2.040A).

### Beispiel

```
fbReadDcBusCurrent : FB_SoEReadDcBusCurrent_ByDriveRef;
bReadDcBusCurrent  : BOOL;
fDcBusCurrent      : REAL;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
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
END_IF
```

## 3.5.6 FB\_SoEReadDcBusVoltage



Mit dem Funktionsbaustein FB\_SoEReadDcBusVoltage kann die DC-Bus-Spannung des Antriebs (S-0-0380) eingelesen werden.

### VAR\_INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

### VAR\_IN\_OUT

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

**Axis:** Achsstruktur

**VAR\_OUTPUT**

```

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

```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

**fDcBusVoltage:** Liefert die DC-Bus-Spannung (z.B. 294.0 entspricht 294.0V).

**Beispiel**

```

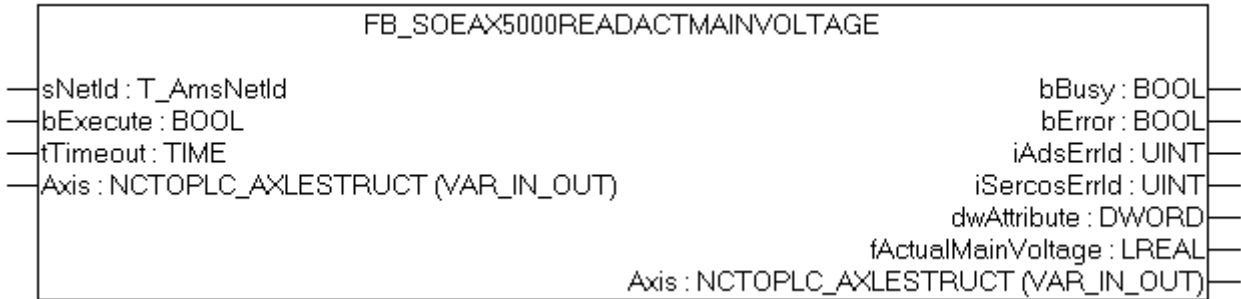
fbReadDcBusVoltage : FB_SoEReadDcBusVoltage;
bReadDcBusVoltage  : BOOL;
fDcBusVoltage      : REAL;

(* 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

```

## 4 AX5000 spezifische FBs

### 4.1 FB\_SoEAX5000ReadActMainVoltage



Mit dem Funktionsbaustein FB\_SoEAX5000ReadActMainVoltage kann der aktuelle Scheitelwert der Netzspannung des AX5000 (P-0-0200) eingelesen werden.

#### VAR\_INPUT

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

#### VAR\_OUTPUT

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

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**dwAttribute:** Liefert das Attribut des Sercos-Parameters.

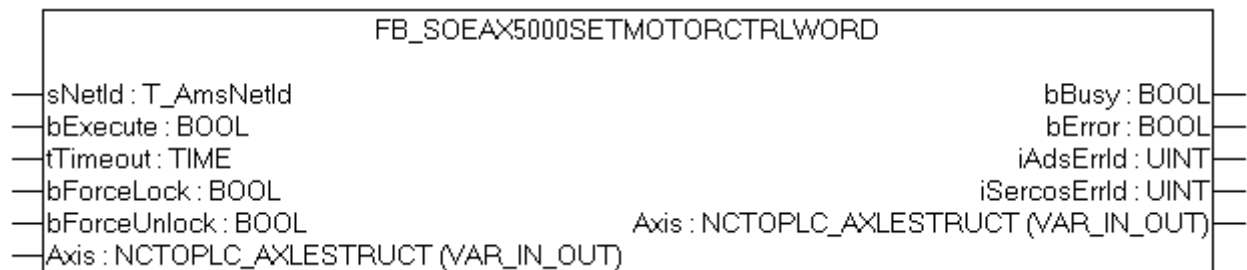
**fActualMainVoltage:** Liefert den Scheitelwert der aktuellen Netzspannung des AX5000 (z.B. 303.0 entspricht 303.0V ).

**Beispiel**

```
fbReadActMainVoltage : FB_SoEAX5000ReadActMainVoltage;
bReadActMainVoltage  : BOOL;
fActualMainVoltage   : REAL;

(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;
IF bReadActMainVoltage THEN
  fbReadActMainVoltage(
    Axis      := stNcToPlc,
    bExecute  := TRUE,
    tTimeout  := DEFAULT_ADS_TIMEOUT,
    fActualMainVoltage=>fActualMainVoltage
  );
IF NOT fbReadActMainVoltage.bBusy THEN
  fbReadActMainVoltage(Axis := stNcToPlc, bExecute := FALSE);
  bReadActMainVoltage := FALSE;
END_IF
END_IF
```

## 4.2 FB\_SoEAX5000SetMotorCtrlWord



Mit dem Funktionsbaustein FB\_SoEAX5000SetMotorCtrlWord kann das ForceLock-Bit (Bit 0) bzw. das ForceUnlock-Bit im Motor Control Word (P-0-0096) gesetzt werden, um die Bremse zu setzen oder zu lösen. Im Normalfall wird die Bremse automatisch über das Enable des Antriebs gehandhabt.

Mit dem ForceLock-Bit kann die Bremse unabhängig vom Enable eingeworfen werden, mit dem ForceUnlock-Bit kann die Bremse unabhängig vom Enable gelöst werden. Bei gleichzeitig gesetztem ForceLock und ForceUnlock hat das ForceLock (Bremse gesetzt) die höhere Priorität.

**VAR\_INPUT**

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

**sNetId:** Ist ein String, der die AMS-Netzwerkennung des PCs enthält.

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**tTimeout:** Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

**bForceLock:** Bremse unabhängig vom Enable aktivieren.

**bForceUnlock:** Bremse unabhängig vom Enable lösen.

**VAR\_IN\_OUT**

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

**Axis:** Achsstruktur.

**VAR\_OUTPUT**

```

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

```

**bBusy:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**Beispiel**

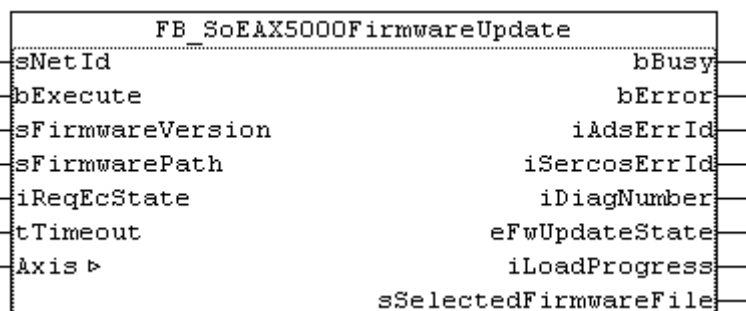
```

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

```

**4.3 FB\_SoEAX5000FirmwareUpdate**

Mit dem Funktionsbaustein FB\_SoEAX5000FirmwareUpdate kann die Firmware des AX5000 überprüft und automatisch auf eine bestimmte Version (Revision und Build) oder auf das aktuellste Build der konfigurierten Revision geändert werden.

Zum Updaten wird:

- der konfigurierte Slave-Typ ermittelt, z.B. AX5103-0000-0010
- der aktuelle Slave mit der vorgegebenen Slaveadresse ermittelt, z.B. AX5103-0000-0009
- die aktuelle Slavefirmware ermittelt, z.B. v1.05\_b0009
- ein Vergleich der Konfiguration und des gefundenen Slaves, auf Anzahl der Kanäle, Strom, Revision, Firmware ausgeführt
- der Name des erforderlichen Firmware-Files ermittelt und die Datei gesucht



- der Firmwareupdate (falls erforderlich) ausgeführt
- der aktuelle Slave mit der vorgegebenen Slaveadresse erneut ermittelt
- der Slave in den vorgegebenen EtherCAT-State geschaltet

Ein erfolgreicher Update endet mit **eFwUpdateState = eFwU\_FwUpdateDone**, ist der Update nicht erforderlich, wird diese über **eFwUpdateState = eFwU\_NoFwUpdateRequired** signalisiert. Der Firmwareupdate erfolgt über den angegebenen Kanal (A=0 oder B=1) aus der stDriveRef. Bei zweikanaligen Geräten kann nur einer der beiden Kanäle hierfür verwendet werden. Der andere Kanal signalisiert das über **eFwUpdateState = eFwU\_UpdateViaOtherChannelActive** bzw. **= eFwU\_UpdateViaOtherChannel**.

Während des Firmwareupdates (**eFwUpdateState = eFwU\_FwUpdateInProgress**) signalisiert **iLoadProgress** den Fortschritt in Prozent.

**Achtung: Während des Updates dürfen die SPS und TwinCAT nicht gestoppt, die EtherCAT-Verbindung nicht unterbrochen und der AX5000 nicht ausgeschaltet werden!**

**VAR\_INPUT**

```
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
```

**sNetId:** AMS-NetID der Steuerung (IPC).

**bExecute:** Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

**sFirmwareVersion:** Gibt die gewünschte Firmware-Version in Form von vx.yy\_bnnnn an, z.B. "v1.05\_b0009" für Version v1.05 Build 0009.

Release-Builds:

- "v1.05\_b0009" für ein spezifisches Build, zum Beispiel v1.05 Build 0009
- "v1.05\_b00???" aktuellstes Build einer vorgegebenen Version, zum Beispiel v1.05
- "v1.??\_b00???" aktuellstes Build einer vorgegebenen Hauptversion, zum Beispiel v1
- "v?.??\_b00???" aktuellstes Build der aktuellsten Version
- "" aktuellstes Build der aktuellsten Version

Kundenspezifische Firmware-Builds:

- "v1.05\_b1009" für ein spezifisches Build, zum Beispiel v1.05 Build 0009
- "v1.05\_b10???" aktuellstes Build einer vorgegebenen Version, zum Beispiel v1.05
- "v1.??\_b10???" aktuellstes Build einer vorgegebenen Hauptversion, zum Beispiel v1
- "v?.??\_b10???" aktuellstes Build der aktuellsten Version

...

- "v1.05\_b8909" für ein spezifisches Build, zum Beispiel v1.05 Build 8909
- "v1.05\_b89???" aktuellstes Build einer vorgegebenen Version, zum Beispiel v1.05
- "v1.??\_b89???" aktuellstes Build einer vorgegebenen Hauptversion, zum Beispiel v1
- "v?.??\_b89???" aktuellstes Build der aktuellsten Version

Debug-Builds:

- "v1.05\_b9009" für ein spezifisches Build, zum Beispiel v1.05 Build 9009
- "v1.05\_b90???" aktuellstes Build einer vorgegebenen Version, zum Beispiel v1.05
- "v1.??\_b90???" aktuellstes Build einer vorgegebenen Hauptversion, zum Beispiel v1
- "v?.??\_b90???" aktuellstes Build der aktuellsten Version

**sFirmwarePath:** Gibt den Pfad für den Firmwarepool an, in dem sich die Firmware-Dateien befinden, z.B. "C:\TwinCAT\Io\TcDriveManager\FirmwarePool".

**iReqEcState:** Gewünschter EtherCAT-State nach dem Update (nur wenn tatsächlich ein Update ausgeführt wird). Die States sind in der TcEtherCAT.lib als globale Konstanten definiert.

**tTimeout:** Da der Firmwareupdate bei großen EtherCAT-Netzwerken länger dauern kann, wird hier nur der Timeout für einzelne interne ADS-Instanzen vorgegeben.

## VAR\_IN\_OUT

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

**Axis:** Achsstruktur.

## 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:** Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

**bError:** Dieser Ausgang wird, nachdem der bBusy-Ausgang zurückgesetzt wurde, gesetzt, sollte ein Fehler bei der Übertragung des Kommandos erfolgen.

**iAdsErrId:** Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehles

**iSercosErrId:** Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehles

**iDiagNumber:** Liefert bei gesetztem bError-Ausgang den Antriebsfehler des letzten Firmware-Updates

**eFwUpdateState:** Liefert den Status der Firmware-Updates. Siehe [E\\_FwUpdateState](#) [▶ 37].

**iLoadProgress:** Liefert den Fortschritt des eigentlichen Firmware-Update in Prozent.

**sSelectedFirmwareFile:** Zeigt den Namen der gesuchten Firmware-Datei an.

## Beispiel

```
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

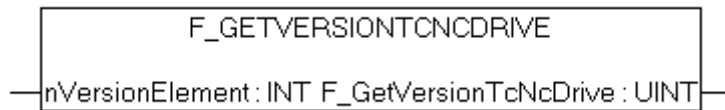
IF NOT bAnyBusy THEN
  bExecute := FALSE;

  IF NOT bAnyError THEN
    iUpdateState := 0; (* OK *)
  ELSE
    iUpdateState := 3; (* Error *)
  END_IF
END_IF

3:
(* Error handling *)
iUpdateState := 0;

END_CASE
```

## 5 F\_GetVersionTcNcDrive



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

### FUNCTION F\_GetVersionTcNcDrive : UINT

```

VAR_INPUT
    nVersionElement : INT;
END_VAR
  
```

**nVersionElement** : Versionselement, das gelesen werden soll. Mögliche Parameter:

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

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0 Build >= 1329	PC or CX (x86)	TcDrive.lib, TcEtherCAT.lib, TcUtilities.Lib, TcSystem.lib
TwinCAT v2.10.0 Build >= 1329	CX (ARM)	

## 6 E\_FwUpdateState

Der E\_FwUpdateState beschreibt den Zustand eines Firmware-Updates.

```

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

```

eFwU_NoError
: Initialzustand

eFwU_CheckCfgIdentity
: Einlesen des konfigurierten Slavetypen (Anzahl Kanäle, Strom,
Revision)

eFwU_CheckSlaveCount
: Ermitteln der konfigurierten Slaveanzahl

eFwU_CheckFindSlavePos
: Suchen der Slave-Adresse im Master-Objektverzeichnis

eFwU_WaitForScan
: Warten auf Online-Scan

eFwU_ScanningSlaves
: Online-Scan der Slaves

eFwU_CheckScannedIdentity
: Einlesen des gescannten Slavetypen (Anzahl Kanäle, Strom,
Revision)

```

```
eFwU_CheckScannedFirmware
: Einlesen der Firmware-Version

eFwU_FindFirmwareFile
: Suchen nach der gewählten Firmware-Datei

eFwU_WaitForUpdate
: Warten auf State des Updates

eFwU_WaitForSlaveState
: Ermitteln des EtherCAT Slave-States

eFwU_StartFwUpdate
: Starten des Firmware-Updates

eFwU_FwUpdateInProgress
: Firmwareupdate aktiv

eFwU_FwUpdateDone
: Firmwareupdate erfolgreich beendet

eFwU_NoFwUpdateRequired
: Kein Firmwareupdate erforderlich

    eFwU_UpdateViaOtherChannelActive    : Update
erfolgt über den anderen Achskanal

eFwU_UpdatedViaOtherChannel
: Update erfolgte über den anderen Achskanal

Update Fehler

eFwU_GetSlaveIdentityError
: Einlesen des konfigurierten Slavetypen schlug fehl, siehe
iAdsErrId

eFwU_GetSlaveCountError
: Ermitteln der konfigurierten Slaveanzahl schlug fehl, siehe
iAdsErrId

eFwU_GetSlaveAddrError
: Suchen der Slave-Adresse im Master-Objektverzeichnis schlug fehl,
siehe iAdsErrId

eFwU_StartScanError
: Starten des Online-Scan schlug fehl, siehe iAdsErrId

eFwU_ScanStateError
: Online-Scan schlug fehl, siehe iAdsErrId

eFwU_ScanIdentityError
: Einlesen des gescannten Slavetypen (Anzahl Kanäle, Strom,
Revision) schlug fehl, siehe iAdsErrId

eFwU_GetSlaveStateError
: Ermitteln des EtherCAT Slave-States schlug fehl, siehe
iAdsErrId

eFwU_ScanFirmwareError
: Einlesen der Firmware-Version schlug fehl, siehe iAdsErrId +
iSercosErrId
```

```
eFwU_FindFileError
: Suchen nach der gewählten Firmware-Datei schlug fehl, siehe
iAdsErrId

eFwU_CfgTypeInNoAX5xxx
: Der konfigurierte Slave ist kein AX5000

eFwU_ScannedTypeInNoAX5xxx
: Der gescannte Slave ist kein AX5000

eFwU_ChannelMismatch
: Anzahl der konfigurierten bzw. gefundenen Kanäle des AX5000
passen nicht zusammen

    eFwU_ChannelMismatch_1Cfg_2Scanned : Einkanaliges
Gerät konfiguriert aber zweikanaliges Gerät gefunden

    eFwU_ChannelMismatch_2Cfg_1Scanned : Zweikanaliges
Gerät konfiguriert aber einkanaliges Gerät gefunden

eFwU_CurrentMismatch
: AX5000-Type paßt vom Strom her nicht, z.B. AX5103 (3A)
konfiguriert aber AX5106 (6A) gefunden

eFwU_FwUpdateError
: Allgemeiner Updatefehler, siehe iAdsErrId

eFwU_ReqSlaveStateError
: Umschalten in den gewünschten EtherCAT-State schlug fehl
```





Mehr Informationen:  
**[www.beckhoff.de/tx1200](http://www.beckhoff.de/tx1200)**

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

