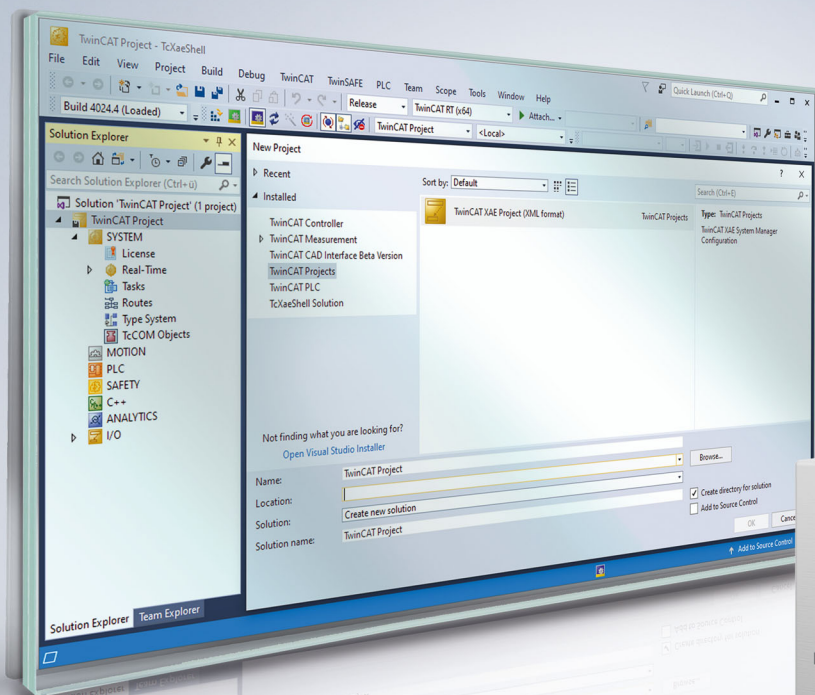


# BECKHOFF New Automation Technology

Handbuch | DE

## TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2\_NcDrive





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>5</b>
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit	6
1.3	Hinweise zur Informationssicherheit	7
<b>2</b>	<b>Übersicht</b>	<b>8</b>
<b>3</b>	<b>Funktionsbausteine</b>	<b>10</b>
3.1	Allgemein SoE	10
3.1.1	FB_SoEReset	10
3.1.2	FB_SoEWritePassword	11
3.1.3	Funktionsbausteine für Kommandos	13
3.1.4	Funktionsbausteine für die Diagnose	17
3.1.5	Funktionsbausteine für die Ermittlung aktueller Werte	23
3.2	Allgemein CoE	32
3.2.1	Funktionsbausteine für die Ermittlung aktueller Werte	32
3.3	AX5000 SoE	34
3.3.1	FB_SoEAX5000ReadActMainVoltage	34
3.3.2	FB_SoEAX5000SetMotorCtrlWord	36
3.3.3	FB_SoEAX5000FirmwareUpdate	37
3.4	F_GetVersionTcNcDrive	41
<b>4</b>	<b>Support und Service</b>	<b>42</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, stets die aktuell gültige 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 weiterentwickelt.

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® 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 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit. Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

### 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.

### Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

### Warnungen vor Personenschäden

#### **GEFAHR**

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

#### **WARNUNG**

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

#### **VORSICHT**

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

### Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:  
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Übersicht



Die Bibliothek Tc2\_NcDrive sollte in neueren Projekten nicht mehr verwendet werden. Verwenden Sie stattdessen die Bibliothek Tc2\_MC2\_Drive (siehe Dokumentation [TwinCAT 3 PLC Lib Tc2\\_MC2\\_Drive](#)).

Die Bibliothek Tc2\_NcDrive umfasst Funktionen und Funktionsbausteine für SoE-Antriebe, die per MC2-Achsstruktur (AXIS\_REF) auf den Antrieb zugreifen.

### Drive-Bibliotheken

Die drei Drive-Bibliotheken Tc2\_Drive, Tc2\_NcDrive und Tc2\_MC2\_Drive sind für unterschiedliche Funktionszwecke entwickelt worden, sind in ihrem Funktionsumfang aber nahezu identisch. Die Bausteine der Bibliotheken Tc2\_NcDrive und Tc2\_MC2\_Drive bilden Wrapper-Bausteine um die Bausteine der Bibliothek Tc2\_Drive.

Drive-Bibliothek	Verwendung	Zugriff auf den Antrieb	Bemerkungen
<b>Tc2_Drive</b> Siehe: Dokumentation <a href="#">TwinCAT 3 PLC Lib: Tc2_Drive</a>	Verwenden Sie die Bibliothek Tc2_Drive, wenn Sie den Antrieb komplett aus der SPS heraus verwenden (also ohne NC).	Auf den Antrieb wird über eine Drive-Referenz zugegriffen. Bibliotheksintern wird dafür die Struktur ST_DriveRef mit der NetID als String verwendet. Zu Verlinkungszwecken wird zusätzlich eine Struktur ST_PlcDriveRef mit der NetID als Bytearray zur Verfügung gestellt.  (Siehe Drive-Referenz ST_DriveRef)	Wenn Sie auf Parameter im Antrieb zugreifen wollen, für die kein spezieller Baustein implementiert wurde, verwenden Sie die Bausteine FB_SoERead_ByDriveRef und FB_SoEWrite_ByDriveRef. Diese Funktionsbausteine sind in der PLC Lib Tc2_EtherCAT im Ordner SoE Interface implementiert.
<b>Tc2_NcDrive</b> Siehe: Dokumentation <a href="#">TwinCAT 3 PLC Lib: Tc2_NcDrive</a>	Verwenden Sie die Bibliothek Tc2_NcDrive, wenn Sie den Antrieb über die NC mit der Bibliotheken Tc2_Nc verwenden.	Auf den Antrieb wird über die NC-Achsstruktur (NC_TO_PLC) zugegriffen. Die Bausteine der Bibliothek Tc2_NcDrive ermitteln eigenständig über die NC-AchsID aus der NC-Achsstruktur die Zugriffsdaten auf den Antrieb (NetID, Adresse und Kanalnummer).	Wenn Sie auf Parameter im Antrieb zugreifen wollen, für die kein spezieller Baustein implementiert wurde, verwenden Sie die Bausteine FB_SoERead und FB_SoEWrite.
<b>Tc2_MC2_Drive</b> Siehe: Dokumentation <a href="#">TwinCAT 3 PLC Lib Tc2_MC2_Drive</a>	Verwenden Sie die Bibliothek Tc2_MC2_Drive, wenn Sie den Antrieb über NC mit der Bibliothek Tc2_MC2 verwenden.	Auf den Antrieb wird über die MC2-Achsreferenz (AXIS_REF) zugegriffen. Die Bausteine der Bibliothek Tc2_MC2_Drive ermitteln eigenständig über die NC-AchsID aus der MC2-Achsreferenz die Zugriffsdaten auf den Antrieb (NetID, Adresse und Kanalnummer).	Wenn Sie auf Parameter im Antrieb zugreifen wollen, für die kein spezieller Baustein implementiert wurde, verwenden Sie die Bausteine FB_SoERead und FB_SoEWrite.



Beachten Sie die Unterschiede bei der Verwendung der Drive-Bibliotheken mit AX5000 und Bosch Rexroth IndraDrive CS (siehe Beispiele)

### Funktionen

Name	Beschreibung
F_GetVersionTcNcDrive [ <a href="#">▶ 41</a> ]	Liest Versionsinformationen der SPS-Bibliothek aus. Diese Funktion wurde durch die globale Struktur stLibVersion_Tc2_NcDrive ersetzt.



Name	Beschreibung
F_ConvWordToSTAX5000C1D	Konvertiert das C1D-Wort (S-0-0011) des AX5000 in eine Struktur ST_AX5000_C1D. Siehe: Dokumentation TwinCAT 3 PLC Lib: Tc2_Drive.

**Funktionsbausteine**

Name	Beschreibung
<a href="#">FB_SoEReset [► 10]</a>	Führt ein Reset des Antriebs aus (S-0-0099).
<a href="#">FB_SoEWritePassword [► 11]</a>	Setzt das Antriebsspasswort (S-0-0267).
<a href="#">FB_SoEReadDiagMessage [► 17]</a>	Liest die Diagnosenachricht (S-0-0095).
<a href="#">FB_SoEReadDiagNumber [► 18]</a>	Liest die Diagnosenummer (S-0-0390).
<a href="#">FB_SoEReadDiagNumberList [► 20]</a>	Liest die Diagnosenummernliste (bis zu 30 Einträge) (S-0-0375)
<a href="#">FB_SoEExecuteCommand [► 13]</a>	Führt ein Kommando aus.
<a href="#">FB_SoEWriteCommandControl [► 14]</a>	Setzt das Command Control.
<a href="#">FB_SoEReadCommandState [► 15]</a>	Prüft den Kommandostatus.
<a href="#">FB_SoEReadClassXDiag [► 21]</a>	Liest die Class-1-Diagnose (S-0-0011) ... Class-3-Diagnose (S-0-0013).
<a href="#">FB_SoERead [► 23]</a>	Liest einen Parameter.
<a href="#">FB_SoEWrite [► 25]</a>	Schreibt einen Parameter.
<a href="#">FB_SoEReadAmplifierTemperature [► 26]</a>	Liest die Antriebstemperatur (S-0-0384).
<a href="#">FB_SoEReadMotorTemperature [► 28]</a>	Liest die Motortemperatur (S-0-0383).
<a href="#">FB_SoEReadDcBusCurrent [► 29]</a>	Liest den Dc-Bus-Stroms (S-0-0381).
<a href="#">FB_SoEReadDcBusVoltage [► 30]</a>	Liest die Dc-Bus-Spannung (S-0-0380).
<a href="#">FB_SoEAX5000ReadActMainVoltage [► 34]</a>	Liest die Netzspannung (P-0-0200).
<a href="#">FB_SoEAX5000SetMotorCtrlWord [► 36]</a>	Setzt das Motor Control Word (P-0-0096).
<a href="#">FB_SoEAX5000FirmwareUpdate [► 37]</a>	Führt ein automatisches Firmware-Update für den AX5000 aus.
<a href="#">FB_CoERead [► 32]</a>	Liest einen Parameter.
<a href="#">FB_CoEWrite [► 33]</a>	Schreibt einen Parameter.

**Anforderungen**

Komponente	Version
TwinCAT auf dem Entwicklungsrechner	3.1 Build 4016 oder höher
TwinCAT auf dem Windows CE-Image	3.1 Build 4016 oder höher
TwinCAT auf dem Windows XP-Image	3.1 Build 4016 oder höher

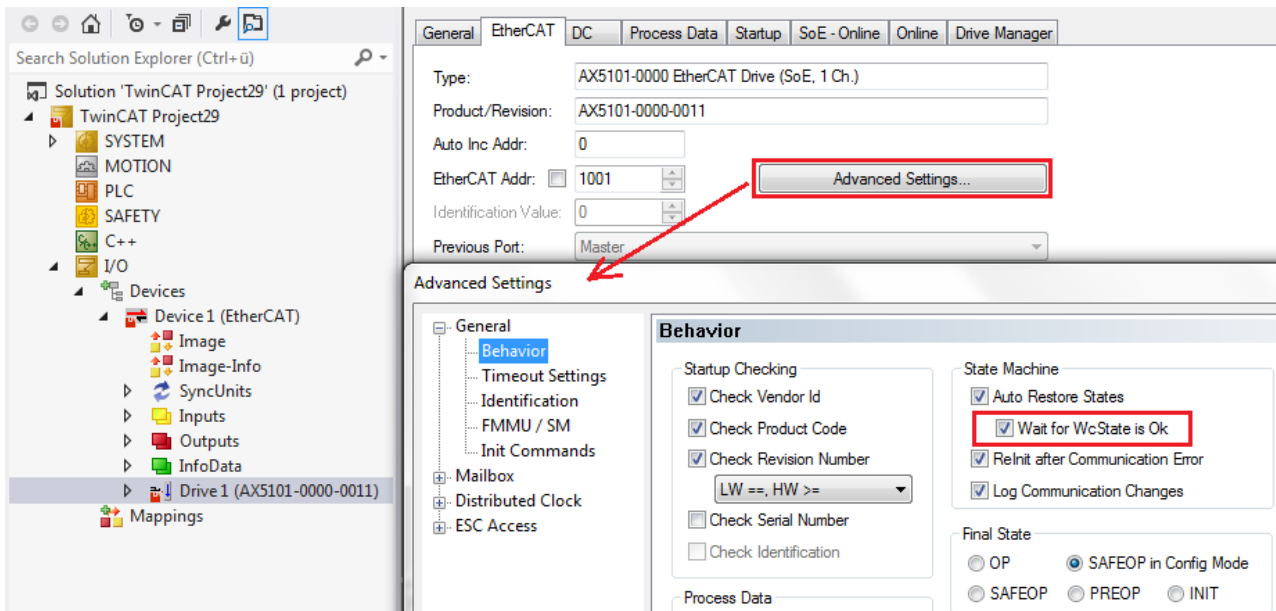
### 3 Funktionsbausteine

#### 3.1 Allgemein SoE

##### 3.1.1 FB\_SoEReset



Mit dem Funktionsbaustein FB\_SoEReset kann ein Reset des Antriebs (S-0-0099) ausgeführt werden. Bei mehrkanaligen Geräten müssen ggf. beide Kanäle einen Reset ausführen. Die Timeout-Zeit muss 10 s betragen, da der Reset je nach Fehler bis zu 10 s dauern kann. Für den AX5000 muss das Flag „Wait For WcState is OK“ in den erweiterten EtherCAT-Einstellungen aktiviert sein.



Ein NC-Reset wird nicht ausgeführt. Wenn ein NC-Reset nötig ist, kann es über den Baustein MC\_Reset aus der PLC Lib Tc2\_MC2 ausgeführt werden.

#### Eingänge

```

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

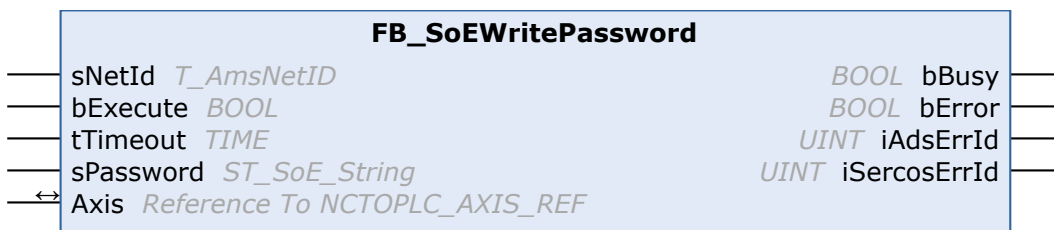
Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

**Beispiel**

```
fbSoEReset : FB_SoEReset_ByDriveRef;
bSoEReset : BOOL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_Axis_REF;

IF bSoEReset THEN
  fbSoEReset(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
  );
IF NOT fbSoEReset.bBusy THEN
  fbSoEReset(Axis := NcToPlc, bExecute := FALSE);
  bSoEReset := FALSE;
END_IF
END_IF
```

**3.1.2 FB\_SoEWritePassword**



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

 **Eingänge**

```
VAR_INPUT
  sNetId : T_AmsNetId := '';
  bExecute : BOOL;
```

```
tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
sPassword : ST_SoE_String;
END_VAR
```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
sPassword	ST_SoE_String	Passwort als Sercos-String

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

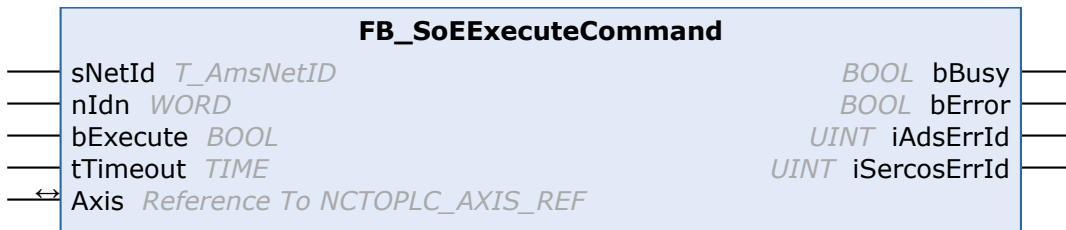
**Beispiel**

```
fbWritePassword : FB_SoEWritePassword;
bWritePassword : BOOL;
sPassword : ST_SoE_String;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

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

### 3.1.3 Funktionsbausteine für Kommandos

#### 3.1.3.1 FB\_SoEExecuteCommand



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

#### Eingänge

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
nIdn	WORD	Parameternummer, auf die sich FB_SoEExecuteCommand bezieht, z. B. „P_0_IDN + 160“ für P-0-0160.
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

#### Ausgänge

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.

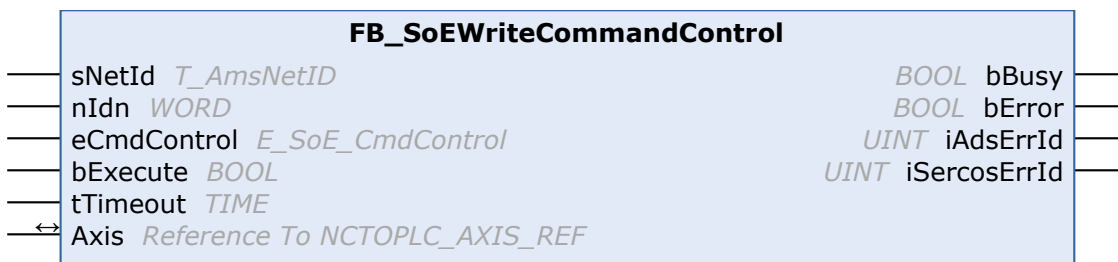
Name	Typ	Beschreibung
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

**Beispiel**

```
fbExecuteCommand : FB_SoEExecuteCommand;
bExecuteCommand : BOOL;
nIdn : WORD;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bExecuteCommand THEN
  nIdn := P_0_IDN + 160;
  fbExecuteCommand(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    nIdn := nIdn,
  );
IF NOT fbExecuteCommand.bBusy THEN
  fbExecuteCommand(Axis := NcToPlc, bExecute := FALSE);
  bExecuteCommand := FALSE;
END_IF
END_IF
```

**3.1.3.2 FB\_SoEWriteCommandControl**



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

**Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
nIdn	WORD	Parameternummer, auf die sich FB_SoEExecuteCommand bezieht, z. B. „P_0_IDN + 160“ für P-0-0160.
eCmdControl	E_SoE_CmdControl	Gibt an, ob das Kommando vorbereitet (eSoE_CmdControl_Set := 1), ausgeführt (eSoE_CmdControl_SetAndEnable := 3) oder abgebrochen (eSoE_CmdControl_Cancel := 0) werden soll.
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

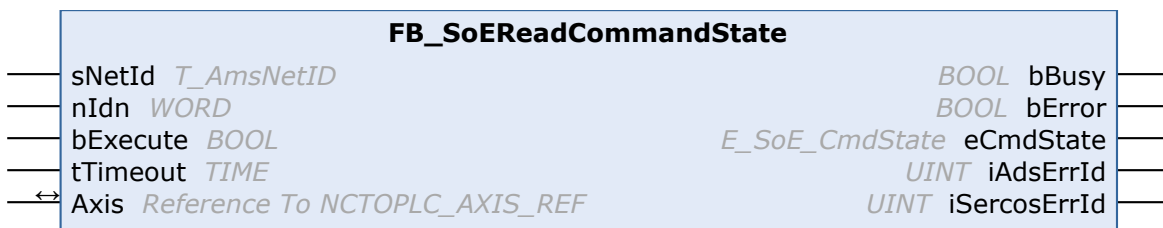
Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

**Beispiel**

```
fbWriteCommandControl : FB_SoEWriteCommandControl;
bWriteCommandControl : BOOL;
nIdn : WORD;
eCmdControl : E_SoE_CmdControl;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

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

**3.1.3.3 FB\_SoEReadCommandState**



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

 **Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
nIdn	WORD	Parameternummer, auf die sich FB_SoEExecuteCommand bezieht, z. B. „P_0_IDN + 160“ für P-0-0160.
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>NCTOPLC_AXIS_REF</u>	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
eCmdState	E_SoE_CmdState	Liefert den Kommandostatus (s.u.).
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

```
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
```

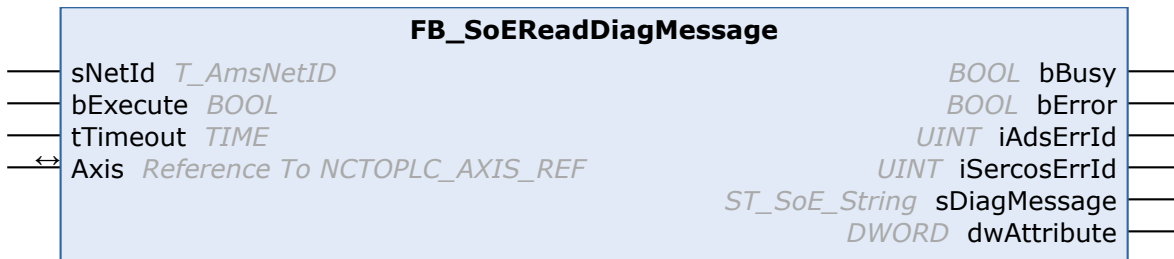
**Beispiel**

```
fbReadCommandState : FB_SoEReadCommandState;
bReadCommandState : BOOL;
nIdn : WORD;
eCmdState : E_SoE_CmdState;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bReadCommandState THEN
  nIdn := P_0_IDN + 160;
  fbReadCommandState(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    nIdn := nIdn,
    eCmdState => eCmdState
  );
IF NOT fbReadCommandState.bBusy THEN
  fbReadCommandState(Axis := NcToPlc, bExecute := FALSE);
  bReadCommandState := FALSE;
END_IF
END_IF
```

### 3.1.4 Funktionsbausteine für die Diagnose

#### 3.1.4.1 FB\_SoEReadDiagMessage



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

**Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

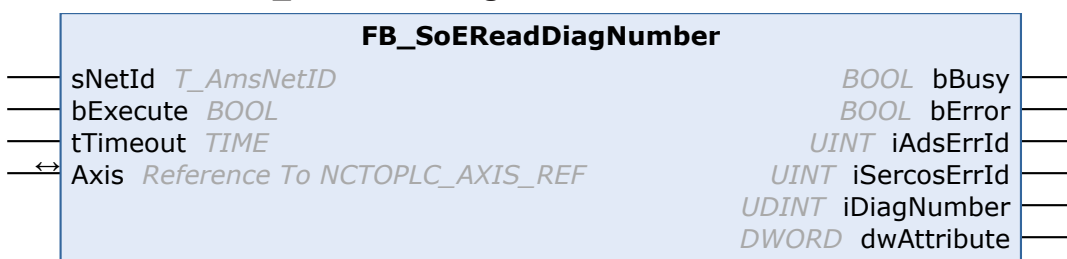
Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
sDiagMessage	ST_SoE_String	Liefert die Diagnosenachricht.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```
fbDiagMessage : FB_SoEReadDiagMessage;
bDiagMessage : BOOL;
sDiagMessage : ST_SoE_String;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

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

**3.1.4.2 FB\_SoEReadDiagNumber**



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

 **Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
iDiagNumber	UDINT	Liefert das Attribut des Sercos-Parameters.
dwAttribute	DWORD	Liefert die aktuelle Diagnosenummer.

**Beispiel**

```
fbDiagNumber : FB_SoEReadDiagNumber;
bDiagNumber : BOOL;
iDiagNumber : UDINT;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

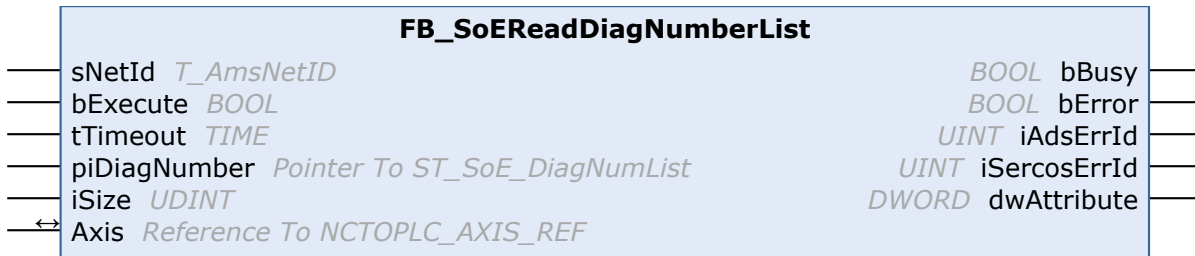
IF bDiagNumber THEN
    fbDiagNumber(
        Axis := NcToPlc,
```

```

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

```

### 3.1.4.3 FB\_SoEReadDiagNumberList



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

#### Eingänge

```

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

```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
piDiagNumber	POINTER TO ST_SoE_DiagNumList	Zeiger auf die Liste der letzten max. 30 Fehlernummern. Die Liste besteht aus der aktuellen und maximalen Anzahl von Bytes in der Liste sowie den 30 Listeneinträgen.
iSize	UDINT	Größe der Liste in Bytes (als sizeof())

#### Ein-/Ausgänge

```

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

```

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

#### Ausgänge

```

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

```

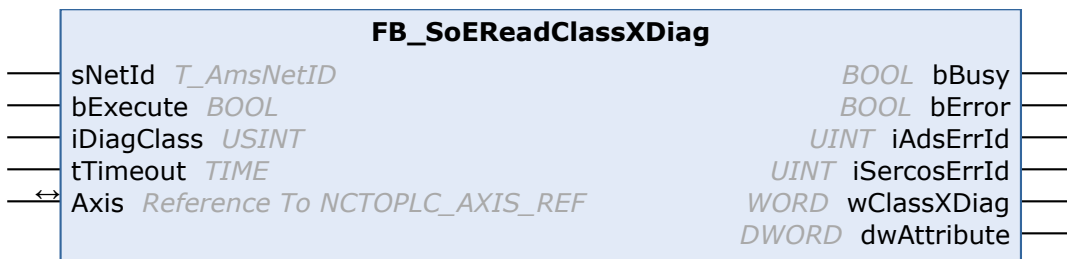
Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```
fbDiagNumberList : FB_SoEReadDiagNumberList;
bDiagNumberList : BOOL;
stDiagNumberList : ST_SoE_DiagNumList;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

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

**3.1.4.4 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 die Konvertierungsfunktion F\_ConvWordToSTAX5000C1D (siehe Dokumentation TwinCAT 3 PLC Lib Tc2\_Drive).

**Eingänge**

```
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
```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.

Name	Typ	Beschreibung
iDiagClass	USINT	Gibt an, welche Diagnose gelesen werden soll. Die Diagnoseparameter können sich von Hersteller zu Hersteller unterscheiden. Nicht immer sind alle Diagnoseparameter (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	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
wClassXDiag	WORD	Liefert die aktuelle Class-X-Diagnose.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```
fbClassXDiag : FB_SoEReadClassXDiag;
bClassXDiag : BOOL;
iDiagClass : USINT := 1;
wClass1Diag : WORD;
stAX5000C1D : ST_AX5000_C1D;
wClass2Diag : WORD;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bClassXDiag THEN
  fbClassXDiag(
    Axis := NcToPlc,
    bExecute := TRUE,
    iDiagClass := iDiagClass,
    tTimeout := DEFAULT_ADS_TIMEOUT
```

```

);
IF NOT fbClassXDiag.bBusy THEN
  fbClassXDiag(Axis := NcToPlc, 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

```

### 3.1.5 Funktionsbausteine für die Ermittlung aktueller Werte

#### 3.1.5.1 FB\_SoERead



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

#### Eingänge

```

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

```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
nIdn	WORD	Parameternummer, auf die sich FB_SoERead bezieht, z. B. „S_0_IDN + 33“ für S-0-0033.
nElement	BYTE	Gibt an, auf welchen Teil des Parameters zugegriffen werden soll, z. B. ist 16#40 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	PVOID	ADR() der Variablen, in die der Wert gelesen werden soll.
cbBufLen	UDINT	SIZEOF() der Variablen, in die der Wert gelesen werden soll.
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.

Name	Typ	Beschreibung
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

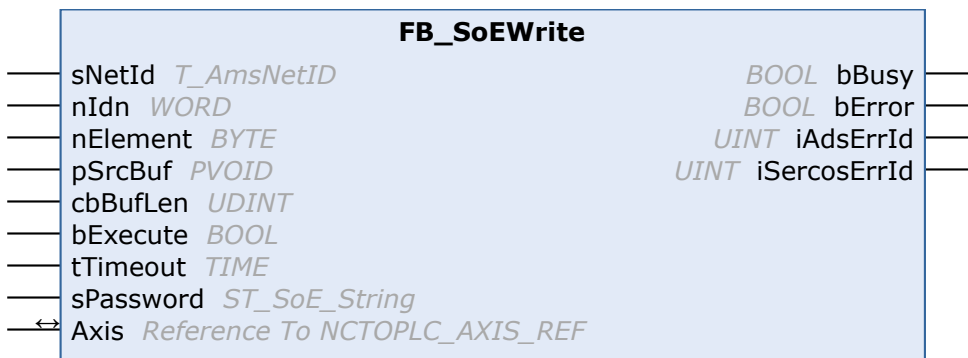
**Beispiel**

```
fbRead      : FB SoERead;
bRead       : BOOL;
iReadValue  : UINT;
nIdn        : WORD;
bReadValue  : UINT;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bRead THEN
  nIdn := S_0_IDN + 33;
  fbRead(
    Axis      := NcToPlc,
    nIdn      := nIdn,
    nElement  := 16#40,
    pDstBuf   := ADR(iReadValue),
    cbBufLen  := SIZEOF(iReadValue),
    bExecute  := TRUE,
    tTimeout  := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbRead.bBusy THEN
    fbRead(Axis := NcToPlc, bExecute := FALSE);
    bRead      := FALSE;
  END_IF
END_IF
```



### 3.1.5.2 FB\_SoEWrite



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

#### Eingänge

```
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
```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
nIdn	WORD	Parameternummer, auf die sich FB_SoERead bezieht, z. B. „S_0_IDN + 47“ für S-0-0047.
nElement	BYTE	Gibt an, auf welchen Teil des Parameters zugegriffen werden soll, z. B. 16#40 ist der Wert (Value) des Parameters. Meist kann nur schreibend auf den Wert 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	PVOID	ADR() der Variablen, die den zu schreibenden Wert enthält.
cbBufLen	UDINT	SIZEOF() der Variablen, die den zu schreibenden Wert enthält.
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
sPassword	ST_SoE_String	Passwort als Sercos-String. Wird z. Z. noch nicht verwendet. Das Passwort muss mit <u>FB_SoEWritePassword</u> [► 11] geschrieben werden.

#### Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

**Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

**Beispiel**

```
fbWrite      : FB_SoEWrite;
bWrite       : BOOL;
nIdn         : WORD;
iWriteValue  : UINT;
sPassword    : ST_SoE_String;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bWrite THEN
  nIdn := S_0_IDN + 33;
  fbWrite(
    Axis      := NcToPlc,
    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 := NcToPlc, bExecute := FALSE);
    bWrite := FALSE;
  END_IF
END_IF
```

**3.1.5.3 FB\_SoEReadAmplifierTemperature**



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

 **Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>NCTOPLC_AXIS_REF</u>	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
fAmplifierTemperature	REAL	Liefert die Antriebstemperatur (z. B. 26.2 entspricht 26.2 °C).
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

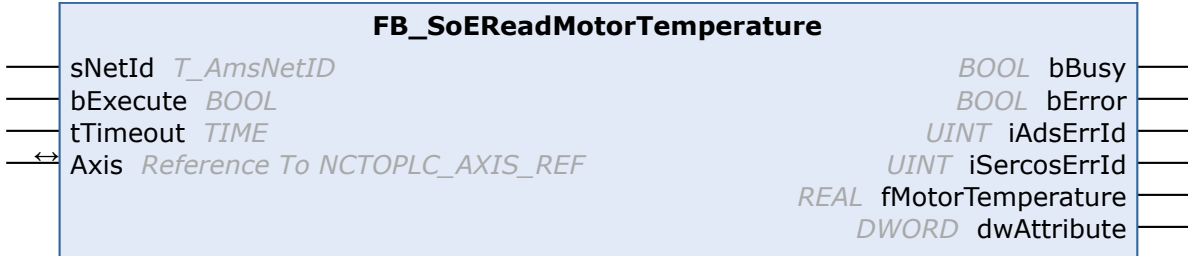
**Beispiel**

```
fbReadAmplifierTemp :
FB_SoEReadAmplifierTemperature;
bReadAmplifierTemp : BOOL;
fAmplifierTemperature : REAL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bReadAmplifierTemp THEN
  fbReadAmplifierTemp(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
```

```
fAmplifierTemperature=>fAmplifierTemperature
);
IF NOT fbReadAmplifierTemp.bBusy THEN
  fbReadAmplifierTemp(Axis := NcToPlc, bExecute := FALSE);
  bReadAmplifierTemp := FALSE;
END_IF
END_IF
```

### 3.1.5.4 FB\_SoEReadMotorTemperature



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

#### Eingänge

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

#### Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsenstruktur vom Typ NCTOPLC_AXIS_REF

#### Ausgänge

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.

Name	Typ	Beschreibung
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
fMotorTemperature	REAL	Liefert die Motortemperatur (z. B. 30.5 entspricht 30.5°C). Wenn der Motor keinen Temperatursensor enthält, steht hier 0.0, d. h. 0.0°C.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```

fbReadMotorTemp : FB_SoEReadMotorTemperature;
bReadMotorTemp : BOOL;
fMotorTemperature : REAL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

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

**3.1.5.5 FB\_SoEReadDcBusCurrent**



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

**Eingänge**

```

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
fDcBusCurrent	REAL	Liefert das Attribut des Sercos-Parameters.
dwAttribute	DWORD	Liefert den DC-Bus-Strom (z. B. 2.040 entspricht 2.040 A).

**Beispiel**

```
fbReadDcBusCurrent : FB_SoEReadDcBusCurrent_ByDriveRef;
bReadDcBusCurrent : BOOL;
fDcBusCurrent : REAL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bReadDcBusCurrent THEN
  fbReadDcBusCurrent(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    fDcBusCurrent=>fDcBusCurrent
  );
IF NOT fbReadDcBusCurrent.bBusy THEN
  fbReadDcBusCurrent(Axis := NcToPlc, bExecute := FALSE);
  bReadDcBusCurrent := FALSE;
END_IF
END_IF
```

**3.1.5.6 FB\_SoEReadDcBusVoltage**



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

 **Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
fDcBusVoltage	REAL	Liefert die DC-Bus-Spannung (z. B. 294.0 entspricht 294.0 V).
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.

**Beispiel**

```
fbReadDcBusVoltage : FB_SoEReadDcBusVoltage;
bReadDcBusVoltage : BOOL;
fDcBusVoltage : REAL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bReadDcBusVoltage THEN
  fbReadDcBusVoltage(
    Axis := NcToPlc,
```

```

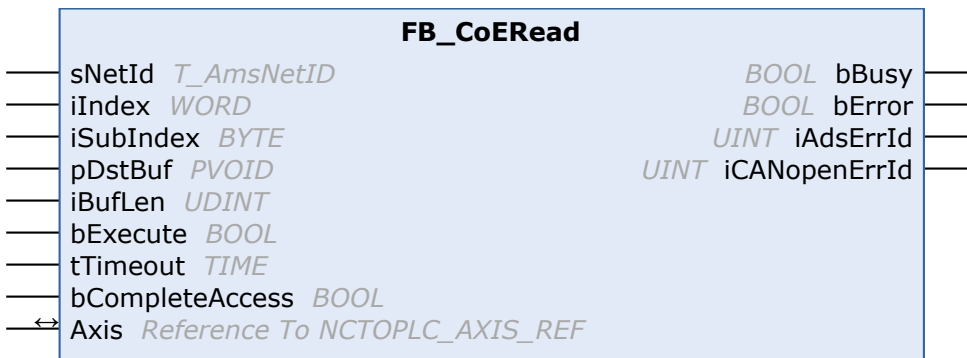
bExecute := TRUE,
tTimeout := DEFAULT_ADS_TIMEOUT,
fDcBusVoltage=>fDcBusVoltage
);
IF NOT fbReadDcBusVoltage.bBusy THEN
fbReadDcBusVoltage(Axis := NcToPlc, bExecute := FALSE);
bReadDcBusVoltage := FALSE;
END_IF
END_IF

```

## 3.2 Allgemein CoE

### 3.2.1 Funktionsbausteine für die Ermittlung aktueller Werte

#### 3.2.1.1 FB\_CoERead



Mit dem Funktionsbaustein FB\_CoERead können per SDO-(Service Daten Objekt)-Zugriff Daten aus dem Objektverzeichnis eines EtherCAT-Slaves ausgelesen werden. Dazu muss der Slave eine Mailbox besitzen und das CoE-(CANopen over EtherCAT)-Protokoll unterstützen. Mithilfe der Parameter SubIndex und Index wird ausgewählt, welches Objekt ausgelesen werden soll. Über CompleteAccess := TRUE kann der Parameter mit Unterelementen gelesen werden.

#### Eingänge

```

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*)
END_VAR

```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
iIndex	WORD	Index des Objektes, das gelesen werden soll.
iSubIndex	BYTE	Subindex des Objektes, das gelesen werden soll.
pDstBuf	PVOID	Adresse (Pointer) auf den Empfangspuffer
iBufLen	UDINT	Maximal verfügbare Puffergröße für die zu lesenden Daten in Bytes
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
bCompleteAccess	BOOL	Über Complete Access kann das komplette Objekt auf einmal zugegriffen werden.



 **Ein-/Ausgänge**

```
VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF, die eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse wie die Position, die Geschwindigkeit und den Fehlerzustand.

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei einem gesetzten bError-Ausgang den ADS-Fehlercode.
iCANopenErrId	UINT	Liefert bei einem gesetzten bError-Ausgang den CANopen-Fehlercode.

**3.2.1.2 FB\_CoEWrite**



Mit dem Funktionsbaustein FB\_CoEWrite kann per SDO-(Service Daten Objekt)-Download ein Objekt aus dem Objektverzeichnis eines EtherCAT-Slaves beschrieben werden. Dazu muss der Slave eine Mailbox besitzen und das CoE-(CANopen over EtherCAT)-Protokoll unterstützen. Mithilfe der Parameter SubIndex und Index wird ausgewählt, welches Objekt beschrieben werden soll. Über CompleteAccess := TRUE kann der Parameter mit Unterelementen geschrieben werden.

 **Eingänge**

```
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*)
END_VAR
```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
iIndex	WORD	Index des Objektes, das beschrieben werden soll.
iSubIndex	BYTE	Subindex des Objektes, das beschrieben werden soll.
pDstBuf	PVOID	Adresse (Pointer) auf den Sendepuffer
iBufLen	UDINT	Maximal verfügbare Puffergröße für die zu lesenden Daten in Bytes
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
bCompleteAccess	BOOL	Über Complete Access kann das komplette Objekt auf einmal zugegriffen werden.

 **Ein-/Ausgänge**

```
VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF, die eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse wie die Position, die Geschwindigkeit und den Fehlerzustand.

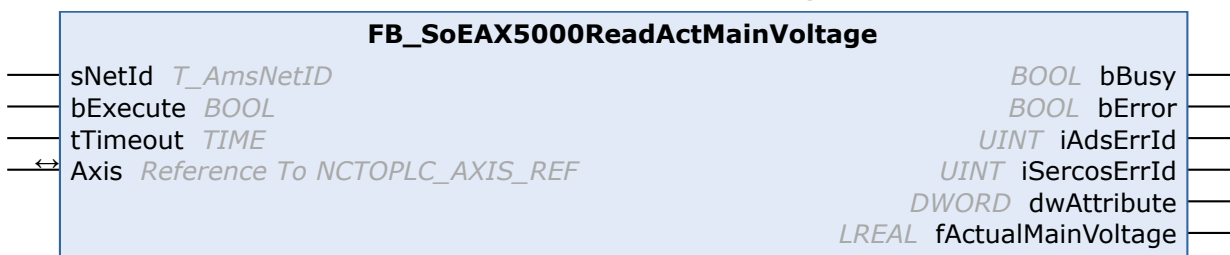
 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei einem gesetzten bError-Ausgang den ADS-Fehlercode.
iCANopenErrId	UINT	Liefert bei einem gesetzten bError-Ausgang den CANopen-Fehlercode.

### 3.3 AX5000 SoE

#### 3.3.1 FB\_SoEAX5000ReadActMainVoltage



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

 **Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
dwAttribute	DWORD	Liefert das Attribut des Sercos-Parameters.
fActualMainVoltage	LREAL	Liefert den Scheitelwert der aktuellen Netzspannung des AX5000 (z. B. 303.0 entspricht 303.0 V).

**Beispiel**

```
fbReadActMainVoltage : FB_SoEAX5000ReadActMainVoltage;
bReadActMainVoltage : BOOL;
fActualMainVoltage : REAL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

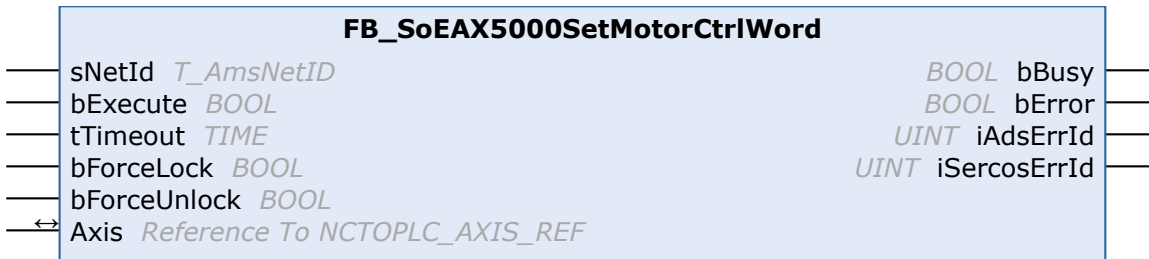
IF bReadActMainVoltage THEN
  fbReadActMainVoltage(
    Axis := NcToPlc,
```

```

bExecute := TRUE,
tTimeout := DEFAULT_ADS_TIMEOUT,
fActualMainVoltage=>fActualMainVoltage
);
IF NOT fbReadActMainVoltage.bBusy THEN
  fbReadActMainVoltage(Axis := NcToPlc, bExecute := FALSE);
  bReadActMainVoltage := FALSE;
END_IF
END_IF

```

### 3.3.2 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 aktivieren oder zu lösen. Im Normalfall wird die Bremse automatisch über das Enable des Antriebs gesteuert.

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

#### Eingänge

```

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

```

Name	Typ	Beschreibung
sNetId	T_AmsNetId	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
bExecute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
tTimeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
bForceLock	BOOL	Aktiviert die Bremse unabhängig vom Enable.
bForceUnlock	BOOL	Löst die Bremse unabhängig vom Enable.

#### Ein-/Ausgänge

```

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

```

Name	Typ	Beschreibung
Axis	NCTOPLC_AXIS_REF	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

#### Ausgänge

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;

```

```
iAdsErrId : UINT;
iSercosErrId : UINT;
END_VAR
```

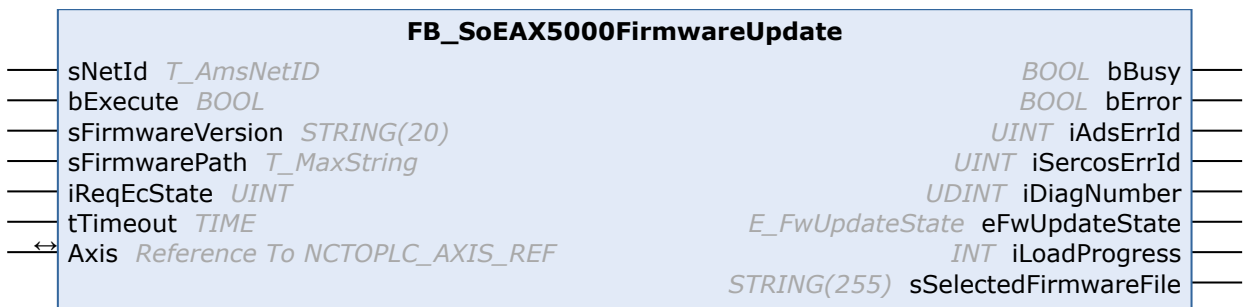
Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrId	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrId	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

**Beispiel**

```
fbSetMotorCtrlWord : FB_SoEAX5000SetMotorCtrlWord;
bSetMotorCtrlWord : BOOL;
bForceLock : BOOL;
bForceUnlock : BOOL;
(* NcAxis *)
NcToPlc AT %I* : NCTOPLC_AXIS_REF;

IF bSetMotorCtrlWord THEN
  fbSetMotorCtrlWord(
    Axis := NcToPlc,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
    bForceLock := bForceLock,
    bForceUnlock := bForceUnlock
  );
  IF NOT fbSetMotorCtrlWord.bBusy THEN
    fbSetMotorCtrlWord(Axis := NcToPlc, bExecute := FALSE);
    bSetMotorCtrlWord := FALSE;
  END_IF
END_IF
```

**3.3.3 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.

Für das Update wird

- der konfigurierte Slavetyp 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 und Firmware ausgeführt
- der Name des erforderlichen Firmware-Files ermittelt und die Datei gesucht
- der Firmware-Update (falls erforderlich) ausgeführt
- der aktuelle Slave mit der vorgegebenen Slaveadresse erneut ermittelt

- der Slave in den vorgegebenen EtherCAT-Status geschaltet

Ein erfolgreiches Update endet mit eFwUpdateState = eFwU\_FwUpdateDone.

Wenn das Update nicht erforderlich ist, wird dies über eFwUpdateState = eFwU\_NoFwUpdateRequired signalisiert.

Das Firmware-Update erfolgt über den angegebenen Kanal (A=0 oder B=1) aus stDriveRef. Bei zweikanaligen Geräten kann nur einer der beiden Kanäle verwendet werden. Der andere Kanal signalisiert eFwUpdateState = eFwU\_UpdateViaOtherChannelActive bzw. eFwUpdateState = eFwU\_UpdateViaOtherChannel.

Während des Firmware-Updates (eFwUpdateState = eFwU\_FwUpdateInProgress) signalisiert iLoadProgress den Fortschritt in Prozent.

**HINWEIS**

**Fehlerhaftes Update durch Unterbrechungen**

Unterbrechungen während des Updates können dazu führen, dass dieses nicht oder fehlerhaft ausgeführt wird. Die Klemme kann danach ohne die passende Firmware möglicherweise nicht mehr verwendet werden.

Während des Updates gilt:

- Die SPS und TwinCAT dürfen nicht gestoppt werden.
- Die EtherCAT-Verbindung darf nicht unterbrochen werden.
- Der AX5000 darf nicht ausgeschaltet werden.

**Eingänge**

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

Name	Typ	Beschreibung
sNetId	T_AmsNetId	AMS-NetID der Steuerung (IPC).
bExecute	BOOL	Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.
sFirmwareVersion	STRING(20)	<p>Gibt die gewünschte Firmware-Version in Form von vx.yy_bnnnn an, z. B. „v1.05_b0009“ für Version v1.05 Build 0009.</p> <p><b>Release-Builds:</b></p> <ul style="list-style-type: none"> <li>• "v1.05_b0009" für ein spezifisches Build, z. B. v1.05 Build 0009</li> <li>• "v1.05_b00??" aktuellstes Build einer vorgegebenen Version, z. B. v1.05</li> <li>• "v1.??_b00??" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1</li> <li>• "v?.??_b00??" aktuellstes Build der aktuellsten Version</li> <li>• " " aktuellstes Build der aktuellsten Version</li> </ul> <p><b>Kundenspezifische Firmware-Builds:</b></p> <ul style="list-style-type: none"> <li>• "v1.05_b1009" für ein spezifisches Build, z. B. v1.05 Build 0009</li> </ul>

Name	Typ	Beschreibung
		<ul style="list-style-type: none"> <li>"v1.05_b10???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05</li> <li>"v1.??_b10???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1</li> <li>"v??.??_b10???" aktuellstes Build der aktuellsten Version</li> </ul> <p>...</p> <ul style="list-style-type: none"> <li>"v1.05_b8909" für ein spezifisches Build, z. B. v1.05 Build 8909</li> <li>"v1.05_b89???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05</li> <li>"v1.??_b89???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1</li> <li>"v??.??_b89???" aktuellstes Build der aktuellsten Version</li> </ul> <p><b>Debug-Builds:</b></p> <ul style="list-style-type: none"> <li>"v1.05_b9009" für ein spezifisches Build, z. B. v1.05 Build 9009</li> <li>"v1.05_b90???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05</li> <li>"v1.??_b90???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1</li> <li>"v??.??_b90???" aktuellstes Build der aktuellsten Version</li> </ul>
sFirmwarePath	T_MaxString	Gibt den Pfad für den Firmware-Pool an, in dem sich die Firmware-Dateien befinden, z. B. C:\TwinCAT\Io\TcDriveManager\FirmwarePool.
iReqEcState	UINT	Gewünschter EtherCAT-Status nach dem Update (nur wenn tatsächlich ein Update ausgeführt wird). Die Status sind in der PLC Lib Tc2_EtherCAT als globale Konstanten definiert.
tTimeout	TIME	Da der Firmware-Update bei großen EtherCAT-Netzwerken länger dauern kann, wird hier nur der Timeout für einzelne interne ADS-Instanzen vorgegeben.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>NCTOPLC_AXIS_REF</u>	Achsdatenstruktur vom Typ NCTOPLC_AXIS_REF

 **Ausgänge**

```
VAR_OUTPUT
bBusy           : BOOL;
bError          : BOOL;
iAdsErrId      : UINT;
iSercosErrId   : UINT;
iDiagNumber    : UDINT;
eFwUpdateState : E_FwUpdateState;
iLoadProgress  : INT;
sSelectedFirmwareFile : STRING(MAX_STRING_LENGTH); (* found firmware file, e.g. "AX5yxx_xxxx_-0010_v1_05_b0009.efw" *)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
bError	BOOL	Dieser Ausgang wird gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
iAdsErrld	UINT	Liefert bei gesetztem bError-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
iSercosErrld	UINT	Liefert bei gesetztem bError-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.
iDiagNumber	UDINT	Liefert bei gesetztem bError-Ausgang den Antriebsfehler des letzten Firmware-Updates.
eFwUpdateState	E_FwUpdateState	Liefert den Status der Firmware-Updates.
iLoadProgress	INT	Liefert den Fortschritt des eigentlichen Firmware-Updates in Prozent.
sSelectedFirmwareFile	STRING(MAX_STRING_LENGTH)	Zeigt den Namen der gesuchten Firmware-Datei an.

### Beispiel

```

VAR CONSTANT
  iNumOfDrives : INT := 2;
END_VAR

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

CASE iUpdateState OF
0:
  IF bExecute THEN
    iUpdateState := 1;
  END_IF
1:
  FOR I := 1 TO iNumOfDrives DO
    fbFirmwareUpdate[I] (
      Axis := NcToPlc[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 := NcToPlc[I],
      eFwUpdateState => eFwUpdateState[I],
      sSelectedFirmwareFile => sSelectedFirmwareFile[I],
    );
    IF NOT fbFirmwareUpdate[I].bBusy THEN
      fbFirmwareUpdate[I] (bExecute := FALSE, Axis := NcToPlc[I]);
    END_IF
  END_FOR

```

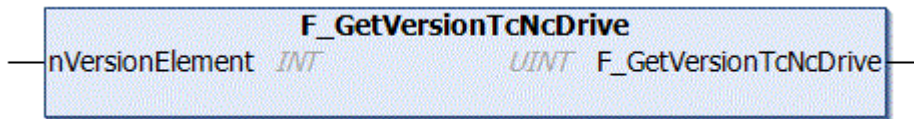


```

    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



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;

## 4 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: [www.beckhoff.com](http://www.beckhoff.com)

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

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

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

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

### Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49 5246 963-0  
E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
Internet: [www.beckhoff.com](http://www.beckhoff.com)



Mehr Informationen:  
**[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)**

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

