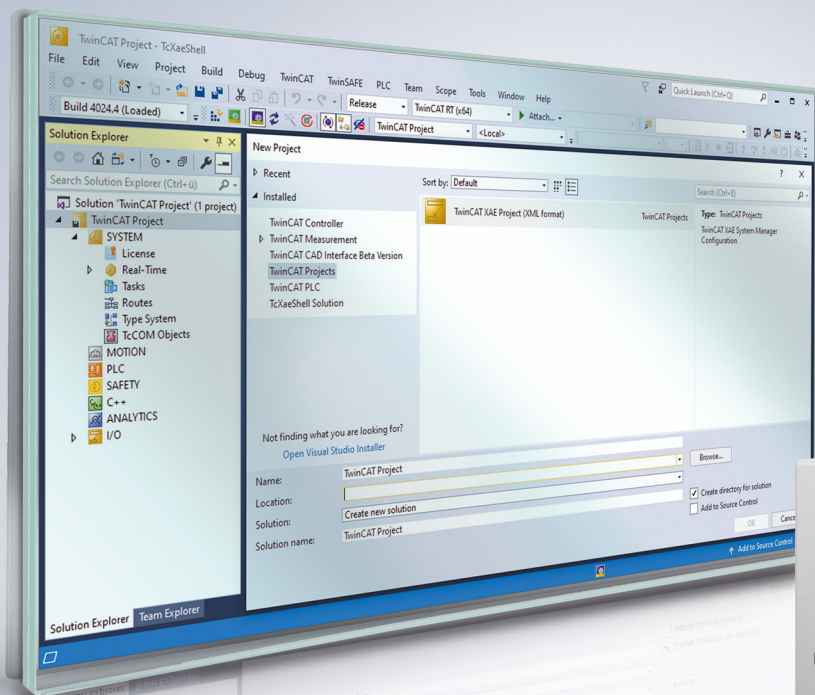


BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2_MC2_Drive



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Funktionsbausteine	10
3.1	Allgemein Beckhoff	10
3.1.1	FB_BrakeControl.....	10
3.1.2	FB_SetPositionOffset.....	11
3.1.3	FB_DeletePositionOffset.....	12
3.1.4	FB_ReadDriveInfo.....	13
3.2	Allgemein SoE.....	14
3.2.1	FB_SoERead	14
3.2.2	FB_SoEWrite	18
3.2.3	FB_SoEReset	21
3.2.4	FB_SoEWritePassword.....	22
3.2.5	FB_SoESetDataAccessMode	24
3.2.6	Funktionsbausteine für Kommando	24
3.2.7	Funktionsbausteine für die Diagnose.....	29
3.2.8	Funktionsbausteine für die Ermittlung aktueller Werte.....	35
3.3	Allgemein CoE	41
3.3.1	FB_CoERead	41
3.3.2	FB_CoEWrite	43
3.3.3	FB_CoEExecuteCommand (Kommando)	45
3.4	AX5000 SoE.....	46
3.4.1	FB_SoEAX5000ReadActMainVoltage	46
3.4.2	FB_SoEAX5000SetMotorCtrlWord	47
3.4.3	FB_SoEAX5000FirmwareUpdate	49
3.4.4	FB_SoEAX5000SetPositionOffset	52
3.4.5	FB_SoEAX5000DeletePositionOffset	54
3.4.6	FB_SoEAX5000ParkAxis.....	55
3.5	F_GetVersionTcMc2Drive	56
3.6	AX8000 CoE	56
3.6.1	FB_CoEAX8000BrakeControl.....	56
3.6.2	FB_CoEAX8000BrakeTest	57
3.6.3	FB_CoEAX8000SetPositionOffset.....	59
3.6.4	FB_CoEAX8000DeletePositionOffset	60
3.7	EL72xx CoE	61
3.7.1	FB_CoEEL72xxBrakeControl.....	61
3.7.2	FB_CoEEL72xxSetPositionOffset.....	62
3.7.3	FB_CoEEL72xxDeletePositionOffset.....	64
4	Datentypen.....	66
4.1	E_FwUpdateState	66

4.2	E_PositionOffsetMemory	67
4.3	E_PositionOffsetFeedback.....	67
4.4	E_DriveErrorCodes	68
4.5	E_AX5000ParkMode.....	68
4.6	E_BrakeMode	68
4.7	ST_DriveInfo	69
4.8	E_DeviceType.....	69
5	Globale Konstanten	70
5.1	SoE Parameter Access	70
6	Beispiele	71
7	Support und Service	72

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.

Zuwendungen 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_MC2_Drive 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
Tc2_Drive Siehe: Dokumentation TwinCAT 3 PLC Lib: Tc2_Drive	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.
Tc2_NcDrive Siehe: Dokumentation TwinCAT 3 PLC Lib: Tc2_NcDrive	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.
Tc2_MC2_Drive Siehe: Dokumentation TwinCAT 3 PLC Lib Tc2_MC2_Drive	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_GetVersionTcMc2Drive ► 56	Liest Versionsinformationen der SPS-Bibliothek aus. Die Funktion wurde durch die globale Struktur stLibVersion_Tc2_MC2_Drive ersetzt.
F_ConvWordToSTAX5000C1D	Konvertiert das C1D-Wort (S-0-0011) des AX5000 in eine Struktur ST_AX5000_C1D. Siehe: Dokumentation TwinCAT3 PLC Lib:Tc2_Drive.

Funktionsbausteine

Name	Beschreibung
FB SoEReset [▶ 21]	Führt ein Reset des Antriebs aus (S-0-0099).
FB SoEWritePassword [▶ 22]	Setzt das Antriebspasswort (S-0-0267).
FB SoEReadDiagMessage [▶ 29]	Liest die Diagnosenachricht (S-0-0095).
FB SoEReadDiagNumber [▶ 31]	Liest die Diagnosenummer (S-0-0390).
FB SoEReadDiagNumberList [▶ 32]	Liest die Diagnosenummernliste (bis zu 30 Einträge) (S-0-0375).
FB SoEReadClassXDiag [▶ 33]	Liest die Class-1-Diagnose (S-0-0011) und die Class-3-Diagnose (S-0-0013).
FB SoEExecuteCommand [▶ 24]	Führt ein Kommando aus.
FB SoEWriteCommandControl [▶ 26]	Setzt das Command Control.
FB SoEReadCommandState [▶ 27]	Prüft den Kommandostatus.
FB SoERead [▶ 14]	Liest einen Parameter.
FB SoEWrite [▶ 18]	Schreibt einen Parameter.
FB SoEReadAmplifierTemperature [▶ 35]	Liest die Antriebstemperatur (S-0-0384).
FB SoEReadMotorTemperature [▶ 36]	Liest die Motortemperatur (S-0-0383).
FB SoEReadDcBusCurrent [▶ 38]	Liest den Dc-Bus-Strom (S-0-0381).
FB SoEReadDcBusVoltage [▶ 39]	Liest die Dc-Bus-Spannung (S-0-0380).
FB SoEAX5000ReadActMainVoltage [▶ 46]	Liest die Netzspannung (P-0-0200).
FB SoEAX5000SetMotorCtrlWord [▶ 47]	Setzt das Motor Control Word (P-0-0096).
FB SoEAX5000FirmwareUpdate [▶ 49]	Führt ein automatisches Firmware-Update für den AX5000 aus.
FB SoEAX5000SetPositionOffset [▶ 52]	Speichert einen Positions-Offset.
FB SoEAX5000DeletePositionOffset [▶ 54]	Löscht einen Positions-Offset.
FB CoERead [▶ 41]	Liest einen Parameter.
FB CoEWrite [▶ 43]	Schreibt einen Parameter.
FB CoEAX80000SetPositionOffset [▶ 59]	Speichert einen Positions-Offset.
FB CoEAX80000DeletePositionOffset [▶ 60]	Löscht einen Positions-Offset.

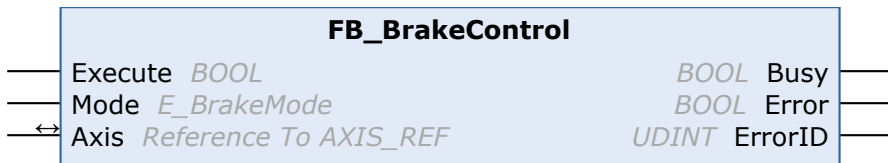
Anforderungen

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 Beckhoff

3.1.1 FB_BrakeControl



Mit dem Funktionsbaustein FB_BrakeControl kann die Bremse eines Motors, der an einer Beckhoff-Servo-Hardware betrieben wird, manuell geöffnet oder geschlossen werden.

Dabei ist zu beachten, dass die Bremse über „Lock“, beziehungsweise „Unlock“, dauerhaft geschlossen oder geöffnet wird.

Es wird empfohlen, die Bremse nur für die nötige Dauer manuell zu öffnen oder zu schließen. Anschließend sollte die Bremse mit dem Funktionsbaustein wieder in den Modus „Automatic“ gesetzt werden.

Eingänge

```

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Mode	E_BrakeMode [▶ 68]	Der Modus, mit dem die Bremse angesteuert wird.

Ein-/Ausgänge

```

VAR_IN_OUT
Axis : AXIS_REF;
END_VAR
  
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

```

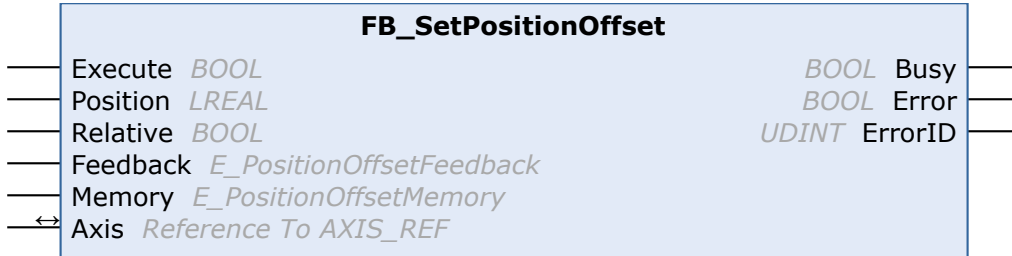
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
  
```

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥4024.48	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.34.0		≥0032

3.1.2 FB_SetPositionOffset



Mit dem Funktionsbaustein FB_SetPositionOffset kann ein Positions-Offset im Speicher einer Beckhoff-Servo-Hardware abgespeichert werden. Ein Speichern im digitalen Typenschild des angeschlossenen Motor-Encoders ist ebenfalls möglich. Dazu muss das Offset zuvor mit dem DriveManager konfiguriert und anschließend im Funktionsbaustein der gleiche Speicherort (Encoder / Drive) verwendet werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Ist-Position der NC-Achse
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ <u>E_PositionOffsetFeedback</u> [▶ 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ <u>E_PositionOffsetMemory</u> [▶ 67]. Gibt an, in welchem Speicher der neu berechnete Positions-Offset abgelegt werden soll.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥4024.48	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.34.0		≥0032

3.1.3 FB_DeletePositionOffset



Mit dem Funktionsbaustein FB_DeletePositionOffset kann ein Positions-Offset im Speicher einer Beckhoff-Servo-Hardware gelöscht werden.

 **Eingänge**

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ E_PositionOffsetFeedback [▶ 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ E_PositionOffsetMemory [▶ 67]. Gibt an, aus welchem Speicher der Positions-Offset gelöscht werden soll.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

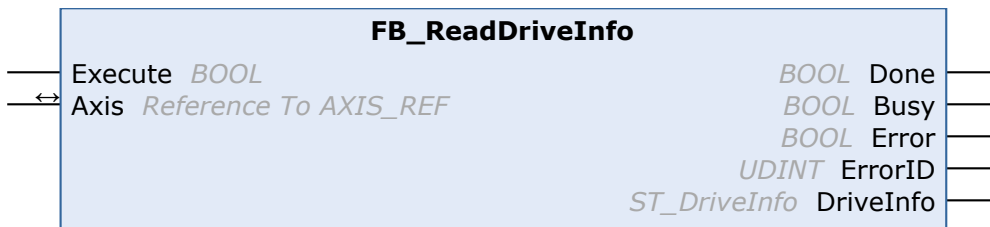
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Ziellplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥4024.48	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.34.0		≥0032

3.1.4 FB_ReadDriveInfo



Mit dem Funktionsbaustein FB_ReadDriveInfo können Basis-Informationen für die Kommunikation mit einer Beckhoff-Servo-Hardware gelesen werden.

 **Eingänge**

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

Name	Type	Beschreibung
Done	BOOL	TRUE, wenn der Befehl erfolgreich ausgeführt wurde.
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.
DriveInfo	ST_DriveInfo	Datenstruktur mit Basis-Informationen, um mit einer Beckhoff-Servo-Hardware zu kommunizieren.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥4024.48	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.34.0		≥0032

3.2 Allgemein SoE

3.2.1 FB_SoERead



Mit dem Funktionsbaustein FB_SoERead kann ein Parameter eingelesen werden.

Per Default werden Attribute und Value parallel gelesen.

Wenn der Aufruf bei einem Gerät eines Fremdherstellers einen ADS-Fehler generieren sollte, kann die fehlende Unterstützung dieser schnelleren Parameterzugriffsmethode die Ursache sein.

Mit dem Baustein FB_SoESetDataAccessMode [▶ 24](#) kann dann bei Bedarf auf die langsamere sequenzielle Zugriffsmethode umgestellt werden.

 **Eingänge**

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

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Idn	WORD	Parameternummer, auf die sich FB_SoERead bezieht, z. B. „S_0_IDNs + 33“ für S-0-0033
Element	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.
BufLen	UDINT	SIZEOF() der Variablen, in die der Wert gelesen werden soll.
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.



Die Idns können der entsprechenden Antriebsdokumentation entnommen werden. Für den AX5000 von Beckhoff siehe [AX5000 Idn-Beschreibung](#).

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel

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

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

Parameterzugriff

Beim Parameterzugriff ist zu beachten, dass bei einigen Parametern die zusätzlichen Informationen über „ActualLength“ & „MaxLength“ für den Zugriff nötig sind.

Um zu ermitteln, bei welchen Parametern dies notwendig ist, können in der Parameter-Liste des Drive-Managers über das Kontextmenu der Kopfzeile die Attribute eingeblendet werden.

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

Enthält die 5. Stelle von rechts einen Wert 4,5,6 oder 7, so werden die zusätzlichen Informationen mit übertragen.

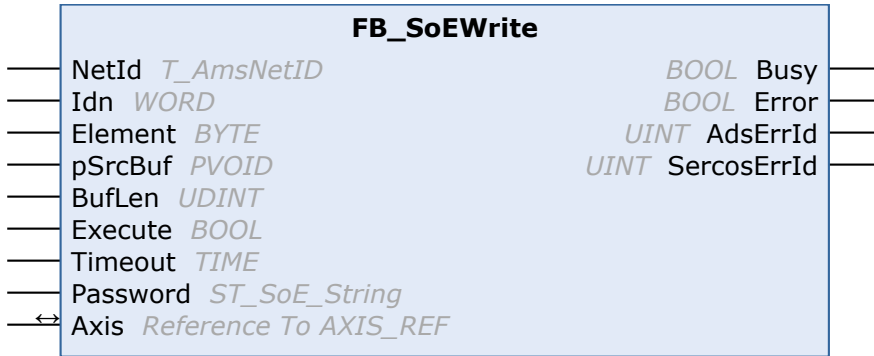
Index	Name	Actual value	Set value	Attribute
S-0-0007	Feedback acquisition capture point (#4)	us	0 us	0x60110001
S-0-0011	Class 1 diagnostic (C1D)		0b0000 0000 0000 0000	0x70010001
S-0-0012	Class 2 diagnostic (C2D)		0b0000 0000 0000 0000	0x70010001
S-0-0013	Class 3 diagnostic (C3D)		0b0000 0000 0000 0000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

Die Struktur, um die Seriennummer (S-0-0432) auszulesen, sieht dann entsprechend so aus:

```

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

3.2.2 FB_SoEWrite



Mit dem Funktionsbaustein FB_SoEWrite kann ein Parameter geschrieben werden.

Eingänge

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

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Idn	WORD	Parameternummer, auf die sich FB_SoERead bezieht, z. B. „S_0_IDNs + 47“ für S-0-0047, siehe AX5000 Idn-Beschreibung .
Element	BYTE	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;
SrcBuf	PVOID	ADR() der Variablen, die den zu schreibenden Wert enthält.
BufLen	UDINT	SIZEOF() der Variablen, in die der Wert gelesen werden soll.
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

Name	Typ	Beschreibung
Password	ST_SoE_String	Passwort als Sercos-String. Wird z. Z. noch nicht verwendet. Das Passwort muss mit FB_SoEWritePassword geschrieben werden.



Die Idns können der entsprechenden Antriebsdokumentation entnommen werden. Für den AX5000 von Beckhoff siehe [AX5000 Idn-Beschreibung](#).

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

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

Beispiel

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

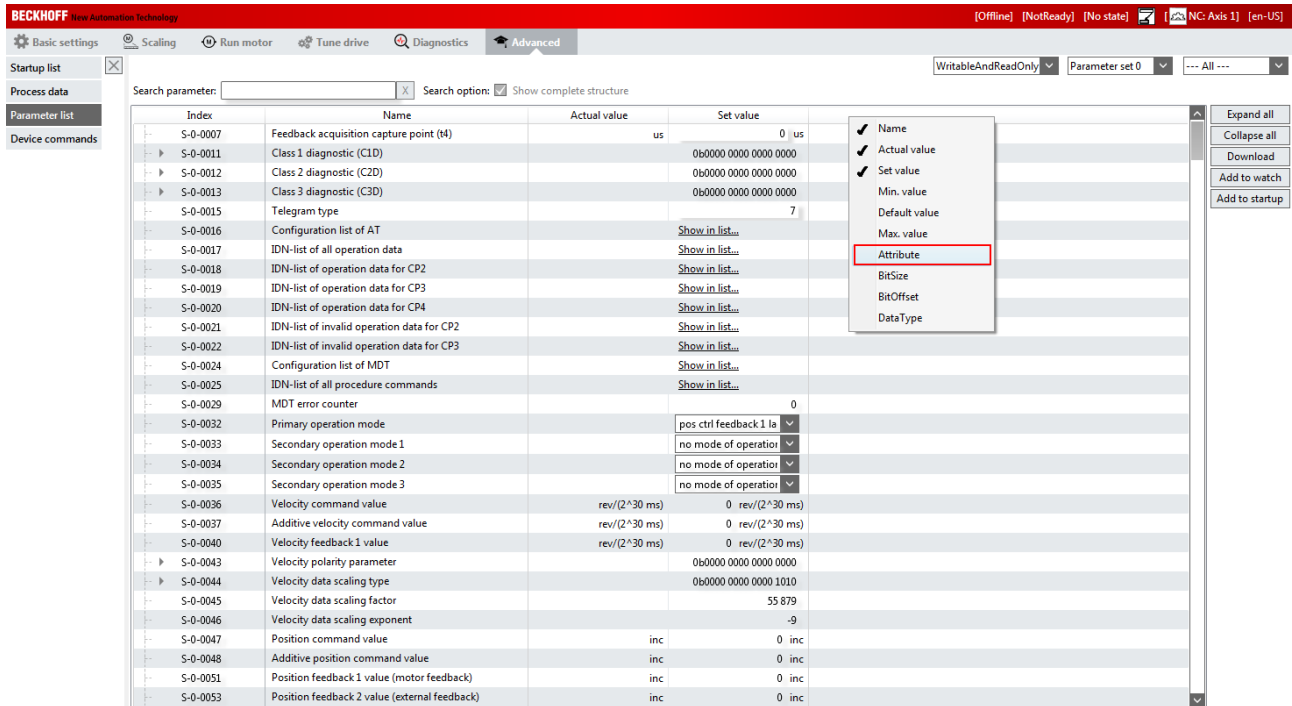
IF Write THEN
  Idn := S_0_IDNs + 33;
  fbWrite(
    Axis      := Axis,
    Idn       := Idn,
    Element   := 16#40,
    pSrcBuf   := ADR(WriteValue),
    BufLen    := SIZEOF(WriteValue),
    Password  := Password,
    Execute   := TRUE,
    Timeout   := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbWrite.Busy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

```
Write := FALSE;
END_IF
END_IF
```

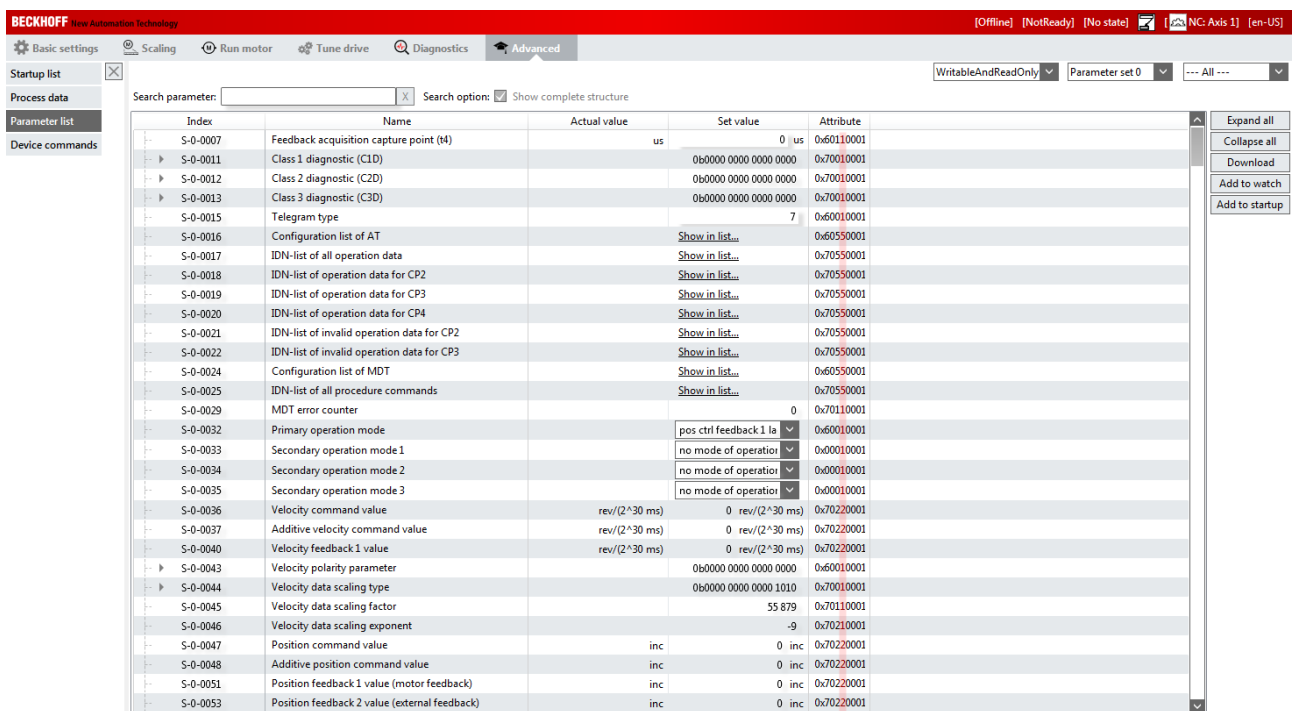
Parameterzugriff

Beim Parameterzugriff ist zu beachten, dass bei einigen Parametern die zusätzlichen Informationen über „ActualLength“ & „MaxLength“ für den Zugriff nötig sind.

Um zu ermitteln, bei welchen Parametern dies notwendig ist, können in der Parameter-Liste des Drive-Managers über das Kontextmenu der Kopfzeile die Attribute eingblendet werden.



Enthält die 5. Stelle von rechts einen Wert 4,5,6 oder 7, so werden die zusätzlichen Informationen mit übertragen.

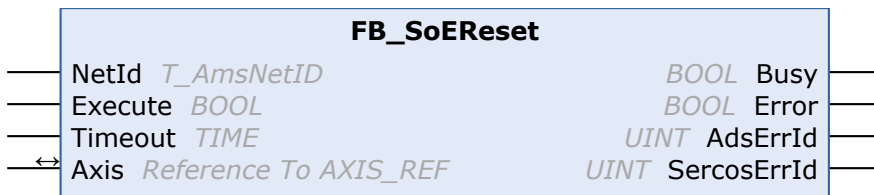


Die Struktur, um die Seriennummer (S-0-0432) auszulesen, sieht dann entsprechend so aus:

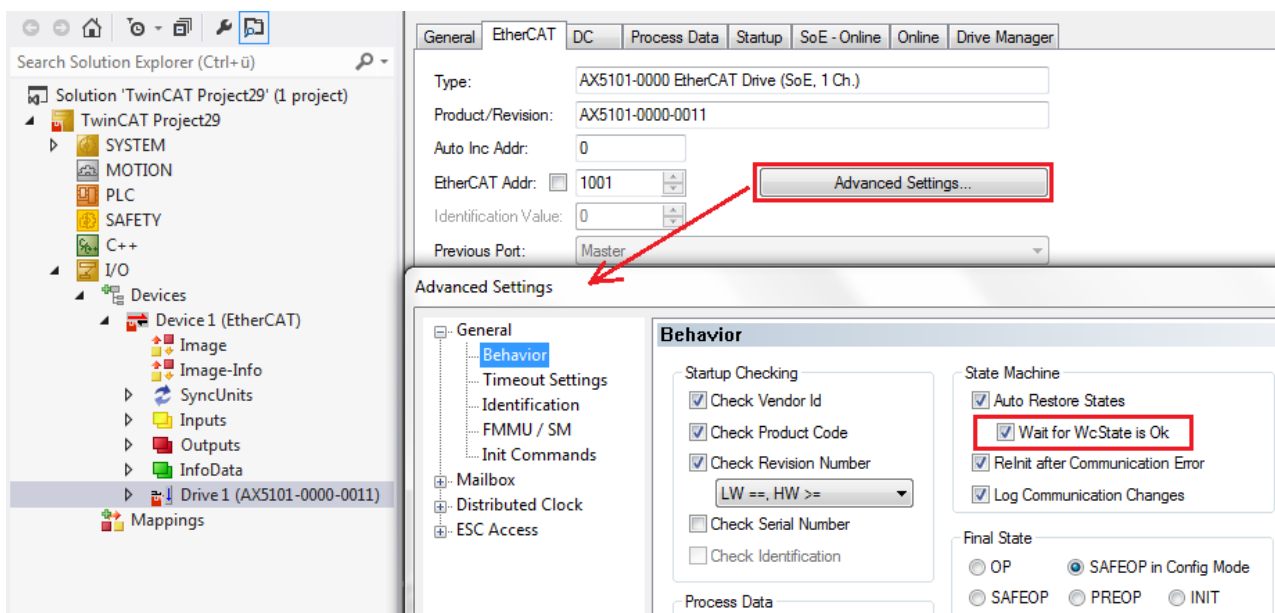
```

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

3.2.3 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
  NetId : T_AmsNetID := '';
  Execute : BOOL;
  Timeout : TIME := T#10s;
END_VAR
    
```

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME := T#10s	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

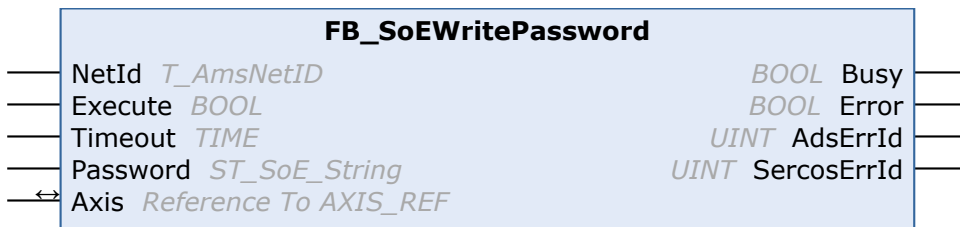
Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
AdsErrId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

Beispiel

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

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

3.2.4 FB_SoEWritePassword



Mit dem Funktionsbaustein FB_SoEWritePassword kann das Antriebspasswort (S-0-0267) gesetzt werden.

 **Eingänge**

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

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

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

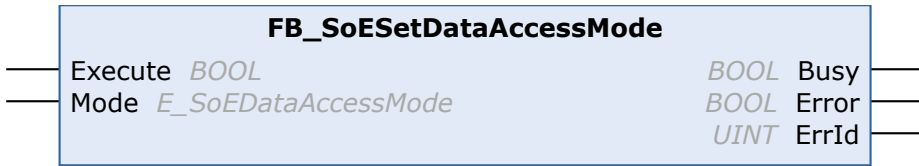
Beispiel

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

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

```
WritePassword := FALSE;
END_IF
END_IF
```

3.2.5 FB_SoESetDataAccessMode



Mit dem Funktionsbaustein FB_SoESetDataAccessMode wird der Zugriffsmodus auf SoE Parameter gesetzt. Per Default wird parallel auf Attribute & Value zugegriffen. Sollte dieser Zugriffsmodus vom Gerätehersteller nicht unterstützt werden, kann ein sequenzieller Zugriff erzwungen werden. Grundsätzlich ist aber der parallele Zugriff schneller.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Mode	E_SoEDataAccessMode	Legt den Zugriffsmodus fest.

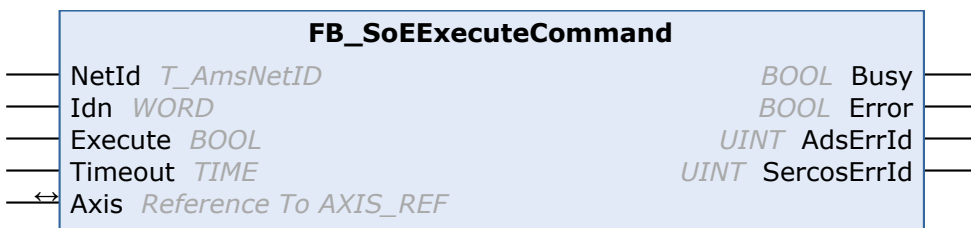
Ausgänge

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

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

3.2.6 Funktionsbausteine für Kommando

3.2.6.1 FB_SoEExecuteCommand



Mit dem Funktionsbaustein FB_SoEExecuteCommand kann ein Kommando ausgeführt werden.

 **Eingänge**

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

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Idn	WORD	Parameternummer, auf die sich FB_SoEExecuteCommand bezieht, z. B. „P_0_IDNs + 160“ für P-0-0160.
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel

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

IF ExecuteCommand THEN
  Idn := P_0_IDNs + 160;
  fbExecuteCommand(
    Axis := Axis,
    Execute := TRUE,
    Timeout := DEFAULT_ADS_TIMEOUT,
```

```

    Idn      := Idn,
  );
  IF NOT fbExecuteCommand.Busy THEN
    fbExecuteCommand(Axis := Axis, Execute := FALSE);
    ExecuteCommand := FALSE;
  END_IF
END_IF

```

3.2.6.2 FB_SoEWriteCommandControl



Mit dem Funktionsbaustein FB_SoEWriteCommandControl kann ein Kommando vorbereitet, gestartet oder abgebrochen werden.

Eingänge

```

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

```

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Idn	WORD	Parameternummer, auf die sich FB_SoEWriteCommandControl bezieht, z. B. „P_0_IDNs + 160“ für P-0-0160.
CmdControl	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.
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

Ein-/Ausgänge

```

VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR

```

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

```

VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;

```

```

    AdsErrId   : UINT;
    SercosErrId : UINT;
END_VAR

```

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

Beispiel

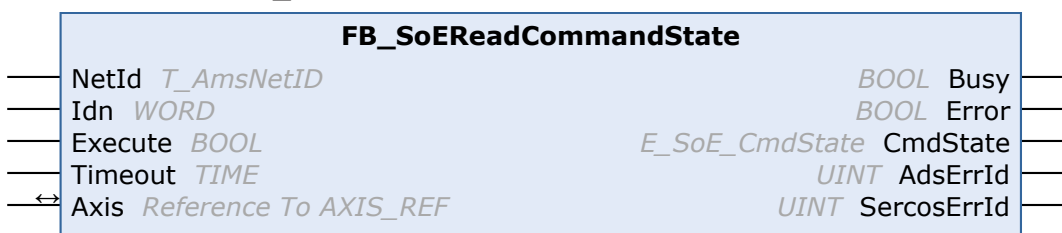
```

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

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

```

3.2.6.3 FB_SoEReadCommandState



Mit dem Funktionsbaustein FB_SoEReadCommandState kann die Kommandoausführung überprüft werden.

Eingänge

```

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

```

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Idn	WORD	Parameternummer, auf die sich FB_SoEReadCommandState_ByDriveRef, z. B. „P_0_IDNs + 160“ für P-0-0160

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Name	Typ	Beschreibung
		- Fehler bei der Kommandoausführung, es wurde in den Fehlerstate gewechselt
AdsErrId	UINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode des zuletzt ausgeführten Befehls.
SercosErrId	UINT	Liefert bei gesetztem Error-Ausgang den Sercos-Fehler des zuletzt ausgeführten Befehls.

Beispiel

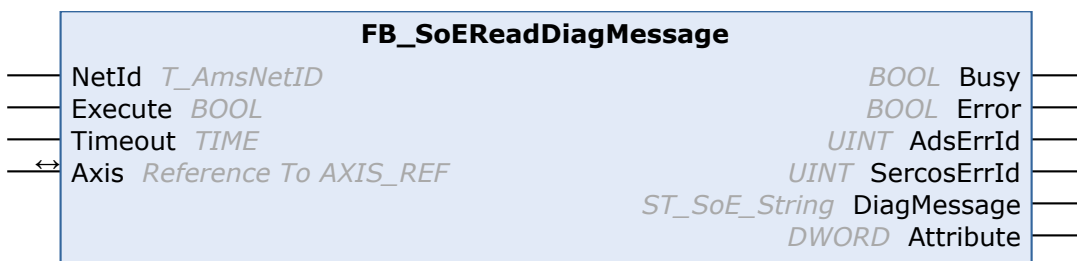
```

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

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

3.2.7 Funktionsbausteine für die Diagnose

3.2.7.1 FB_SoEReadDiagMessage



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

Eingänge

```

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

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

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

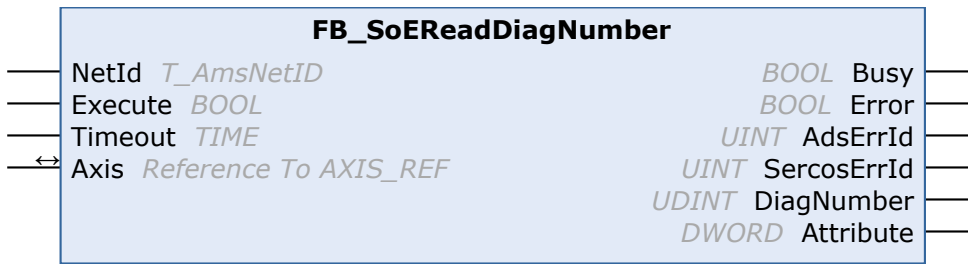
Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
AdsErrId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

Beispiel

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

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

3.2.7.2 FB_SoEReadDiagNumber



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

Eingänge

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

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

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

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

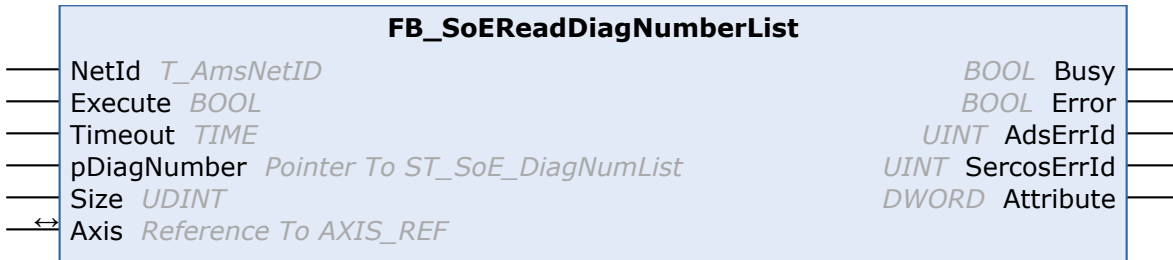
Name	Typ	Beschreibung
DiagNumber	UDINT	Liefert die aktuelle Diagnosenummer.
Attribute	DWORD	Liefert das Attribut des Sercos-Parameters.

Beispiel

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

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

3.2.7.3 FB_SoEReadDiagNumberList



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

Eingänge

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

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
pDiagNumber	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.
Size	UDINT	Größe der Liste in Bytes (als Sizeof())

Ein-/Ausgänge

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


Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

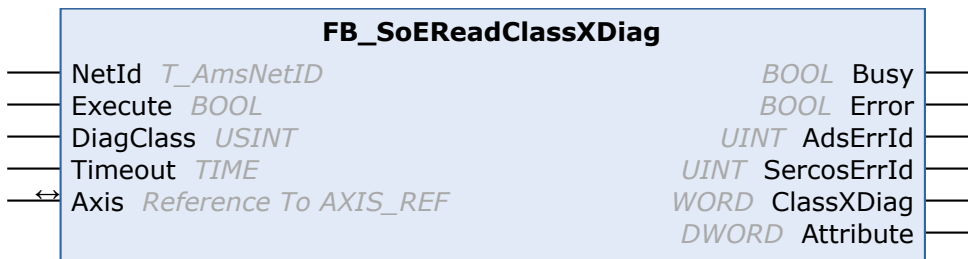
Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
AdsErrId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

Beispiel

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

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

3.2.7.4 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
  NetId      : T_AmsNetId := '';
  Execute    : BOOL;
  DiagClass  : USINT:= 1; (* 1: C1D (S-0-0011) is default, 2: C2D (S-0-0012), 3: C3D (S-0-0013) *)
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

Name	Typ	Beschreibung
NetId	T_AmsNetId := "	String, der die AMS-Netzwerkennung des PCs enthält (Typ T_AmsNetId).
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
DiagClass	USINT:= 1; (* 1: C1D (S-0-0011) is default, 2: C2D (S-0-0012), 3: C3D (S-0-0013) *)	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)
Timeout	TIME := DEFAULT_ADS_TIMEOUT	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel

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

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

3.2.8 Funktionsbausteine für die Ermittlung aktueller Werte

3.2.8.1 FB_SoEReadAmplifierTemperature



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

Eingänge

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

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

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

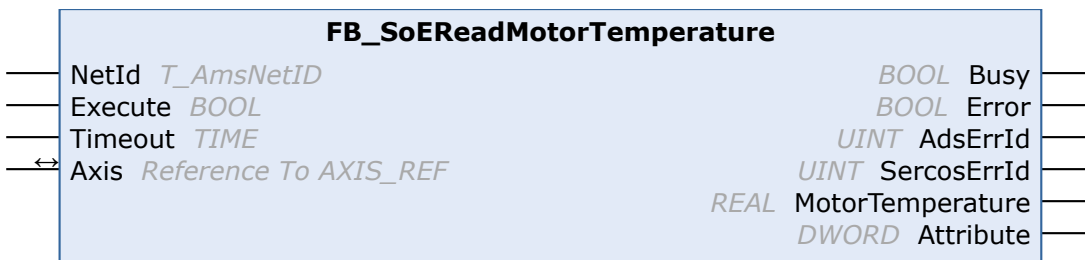
Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
AdsErrId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

Beispiel

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

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

3.2.8.2 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
  NetId      : T_AmsNetID := '';
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

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

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel

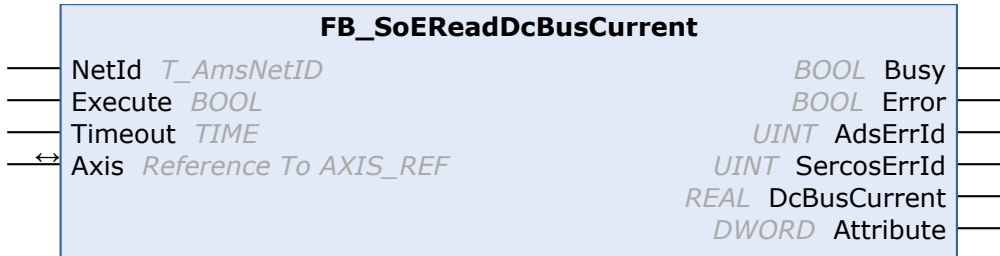
```
fbReadMotorTemp : FB_SoEReadMotorTemperature;
ReadMotorTemp   : BOOL;
MotorTemperature : REAL;
(* NcAxis *)
Axis            : AXIS_REF;

IF ReadMotorTemp THEN
  fbReadMotorTemp(
    Axis := Axis,
```

```

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

3.2.8.3 FB_SoEReadDcBusCurrent



Mit dem Funktionsbaustein FB_SoEAX5000ReadDcBusCurrent kann der DC-Bus-Strom (S-0-0381) eingelesen werden.

Eingänge

```

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

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

Ein-/Ausgänge

```

VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
    
```

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

```

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

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

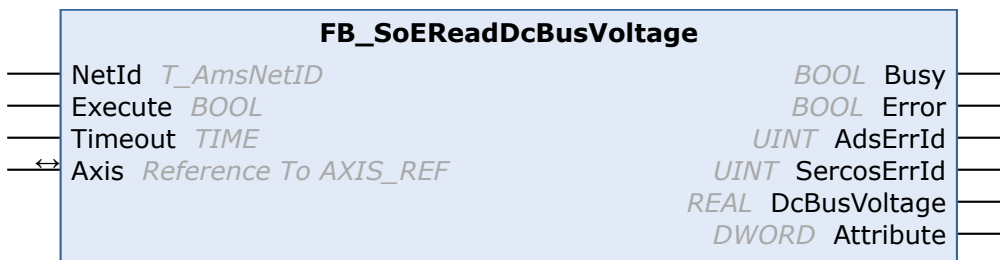
Beispiel

```

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

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

3.2.8.4 FB_SoEReadDcBusVoltage



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

Eingänge

```

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

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

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

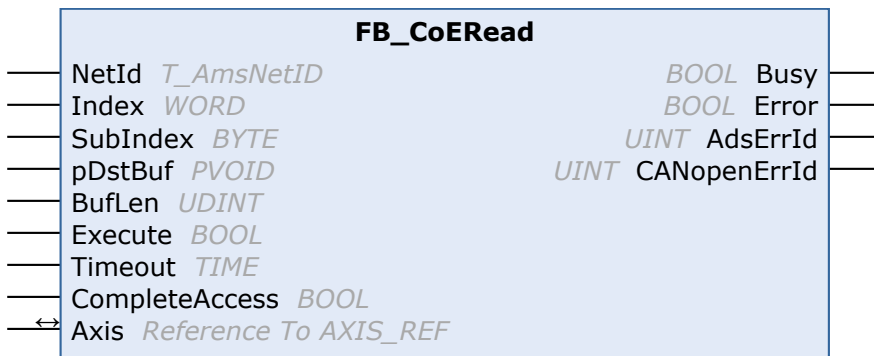
Beispiel

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

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


3.3 Allgemein CoE

3.3.1 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
  NetId      : T_AmsNetID; (*netID of PC with NC*)
  Index      : WORD; (*CoE object index*)
  SubIndex   : BYTE; (*CoE sub index*)
  pDstBuf    : PVOID; (*Contains the address of the buffer for the received data*)
  BufLen     : UDINT; (*Contains the max. number of bytes to be received*)
  Execute    : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT; (*States the time before the function is
cancelled.*)
  CompleteAccess : BOOL; (*Function block reads the complete object with all sub index*)
END_VAR
```

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



Index und SubIndex können der entsprechenden Antriebsdokumentation entnommen werden. Für den AX8000 von Beckhoff siehe [AX8000 Objektbeschreibung](#).

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

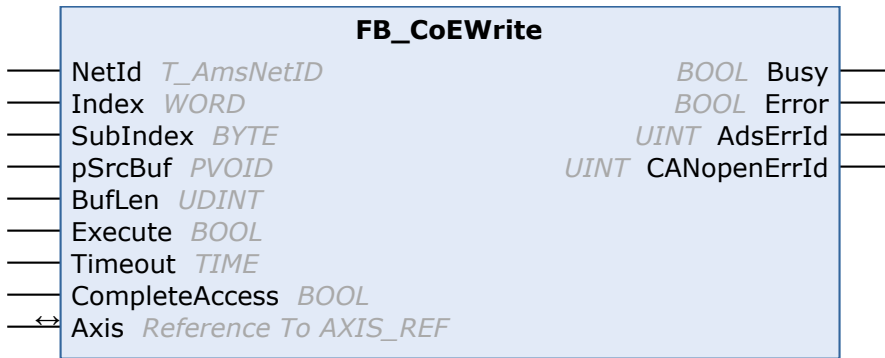
Beispiel für eine Implementierung in ST:

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

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

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

3.3.2 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
  NetId      : T_AmsNetID; (*netID of PC with NC*)
  Index      : WORD; (*CoE object index*)
  SubIndex   : BYTE; (*CoE sub index*)
  pSrcBuf    : PVOID; (*Contains the address of the buffer containing the data to be send*)
  BufLen     : UDINT; (*Contains the max. number of bytes to be received*)
  Execute    : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
  (*States the time before the function is cancelled.*)
  CompleteAccess : BOOL; (*Function block reads the complete object with all sub index*)
END_VAR
    
```

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Index	WORD	Index des Objektes, das beschrieben werden soll.
SubIndex	BYTE	Subindex des Objektes, das beschrieben werden soll.
pDstBuf	PVOID	Adresse (Pointer) auf den Sendepuffer
BufLen	UDINT	Anzahl der zu sendenden Daten in Bytes
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.
CompleteAccess	BOOL	Über Complete Access kann auf das komplette Objekt auf einmal zugegriffen werden.



Index und SubIndex können der entsprechenden Antriebsdokumentation entnommen werden. Für den AX8000 von Beckhoff siehe [AX8000 Objektbeschreibung](#).

Ein-/Ausgänge

```

VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
    
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel für eine Implementierung in ST:

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

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

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

3.3.3 FB_CoEExecuteCommand (Kommando)



Mit dem Funktionsbaustein FB_CoEExecuteCommand kann ein Kommando ausgeführt werden.

Eingänge

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

Name	Typ	Beschreibung
NetId	T_AmsNetID	String, der die AMS-Netzwerkennung des PCs enthält (Typ: T_AmsNetId).
Index	WORD	CoE Objekt-Index, auf den sich FB_CoEExecuteCommand bezieht.
pSrcBuf	PVOID	Adresse der Struktur der zu sendenden Daten
SrcBufLen	UDINT	Größe der Struktur der zu sendenden Daten in Byte
pDstBuf	PVOID	Adresse der Struktur der zu empfangenden Daten
DstBufLen	UDINT	Größe der Struktur der zu empfangenen Daten in Byte
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Timeout	TIME	Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

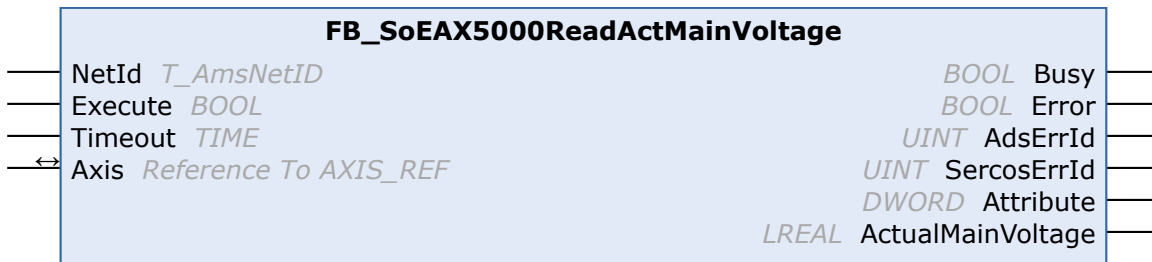
Ausgänge

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

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorId	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.
Status	<u>_E_CoECommandStatus</u>	Status der Kommandoausführung

3.4 AX5000 SoE

3.4.1 FB_SoEAX5000ReadActMainVoltage



Mit dem Funktionsbaustein FB_SoEAX5000ReadActMainVoltage kann der aktuelle Scheitelwert der Netzspannung des AX5000 (P-0-0200) eingelesen werden.

Eingänge

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

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

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

```
SercosErrId      : UINT;
Attribute        : DWORD;
ActualMainVoltage : LREAL;
END_VAR
```

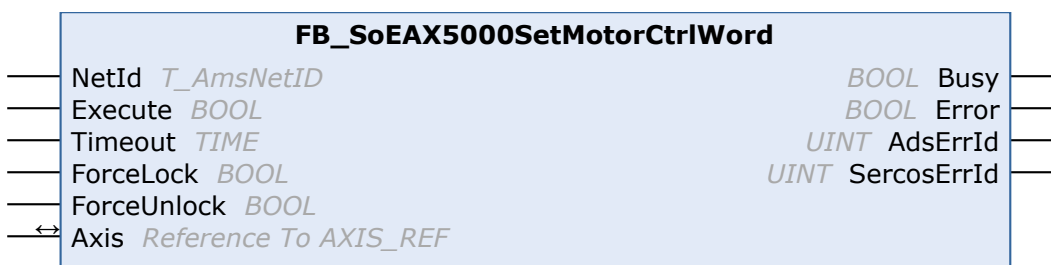
Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
AdsErrId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

Beispiel

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

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

3.4.2 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 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
  NetId      : T_AmsNetId := '';
  Execute    : BOOL;
  Timeout    : TIME := DEFAULT_ADS_TIMEOUT;
```

```
ForceLock : BOOL;
ForceUnlock : BOOL;
END_VAR
```

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

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

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

Beispiel

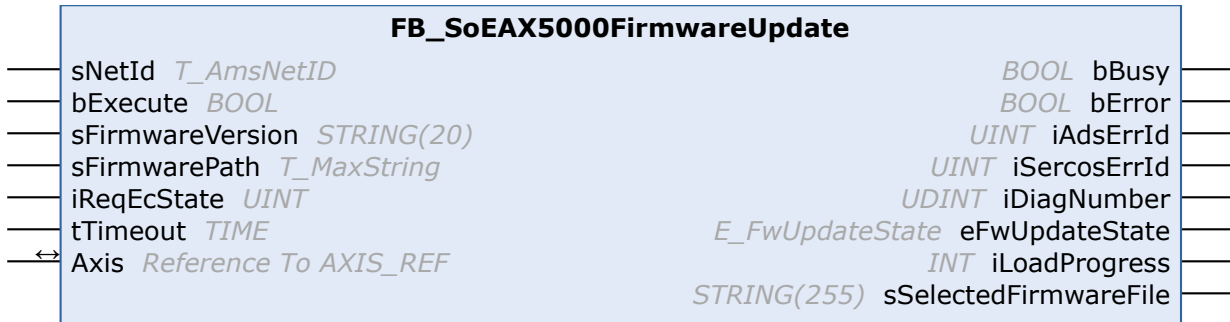
```
fbSetMotorCtrlWord : FB_SoEAX5000SetMotorCtrlWord;
SetMotorCtrlWord : BOOL;
ForceLock : BOOL;
ForceUnlock : BOOL;
(* NcAxis *)
Axis : AXIS_REF;

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



```
SetMotorCtrlWord := FALSE;
END_IF
END_IF
```

3.4.3 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.

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
NetId	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>Release-Builds:</p> <ul style="list-style-type: none"> • "v1.05_b0009" für ein spezifisches Build, z. B. v1.05 Build 0009 • "v1.05_b00???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05 • "v1.??_b00???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1 • "v?.??_b00???" aktuellstes Build der aktuellsten Version • " " aktuellstes Build der aktuellsten Version <p>Kundenspezifische Firmware-Builds:</p> <ul style="list-style-type: none"> • "v1.05_b1009" für ein spezifisches Build, z. B. v1.05 Build 0009 • "v1.05_b10???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05 • "v1.??_b10???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1 • "v?.??_b10???" aktuellstes Build der aktuellsten Version <p>...</p> <ul style="list-style-type: none"> • "v1.05_b8909" für ein spezifisches Build, z. B. v1.05 Build 8909 • "v1.05_b89???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05 • "v1.??_b89???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1 • "v?.??_b89???" aktuellstes Build der aktuellsten Version <p>Debug-Builds:</p> <ul style="list-style-type: none"> • "v1.05_b9009" für ein spezifisches Build, z. B. v1.05 Build 9009 • "v1.05_b90???" aktuellstes Build einer vorgegebenen Version, z. B. v1.05 • "v1.??_b90???" aktuellstes Build einer vorgegebenen Hauptversion, z. B. v1 • "v?.??_b90???" aktuellstes Build der aktuellsten Version

Name	Typ	Beschreibung
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 das 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 : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **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
Busy	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 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
```

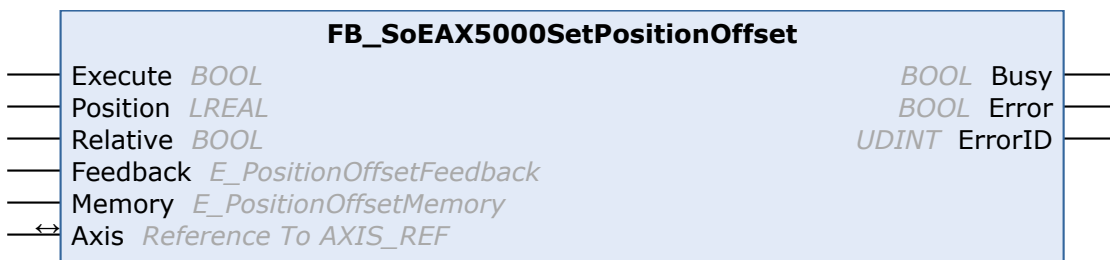
```

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

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

```

3.4.4 FB_SoEAX5000SetPositionOffset



Mit dem Funktionsbaustein FB_SoEAX5000SetPositionOffset kann ein Positions-Offset im Speicher des AX5000 oder im digitalen Typenschild des angeschlossenen Motor-Encoders abgespeichert werden. Dazu muss das Offset zuvor mit dem DriveManager konfiguriert und anschließend im Funktionsbaustein der gleiche Speicherort (Encoder / Drive) verwendet werden.

 **Eingänge**

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Ist-Position der NC-Achse
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ E_PositionOffsetFeedback [► 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ E_PositionOffsetMemory [► 67]. Gibt an, in welchem Speicher der neu berechnete Positions-Offset abgelegt werden soll.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

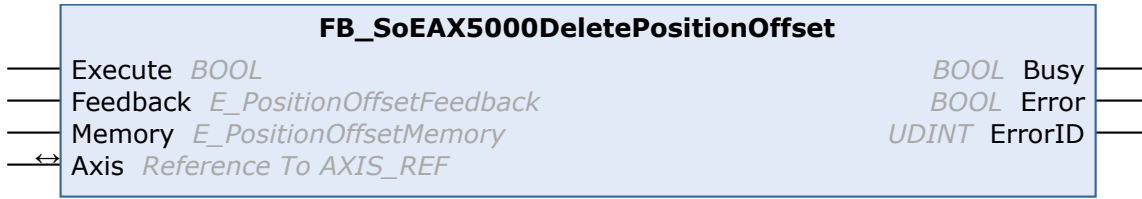
```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken
TwinCAT ≥4020.57	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.16.0
TwinCAT ≥4022.30		

3.4.5 FB_SoEAX5000DeletePositionOffset



Mit dem Funktionsbaustein FB_SoEAX5000DeletePositionOffset kann ein Positions-Offset im Speicher des AX5000 oder im digitalen Typenschild des angeschlossenen Motor-Encoders gelöscht werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ <u>E_PositionOffsetFeedback</u> [▶ 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ <u>E_PositionOffsetMemory</u> [▶ 67]. Gibt an, aus welchem Speicher der Positions-Offset gelöscht werden soll.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken
TwinCAT ≥4020.57 TwinCAT ≥4022.30	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.16.0

3.4.6 FB_SoEAX5000ParkAxis



Mit dem Funktionsbaustein wird die Parkfunktionalität eines AX5000 aktiviert. Ein geparkter Kanal eines AX5000 ist temporär deaktiviert.

Im Falle eines modularen Maschinenkonzepts sind unter Umständen bestimmte Motore nicht vorhanden. Damit dies nicht direkt zu Fehlern führt (z. B. Feedbackfehlern), können die Kanäle hierüber deaktiviert werden.

Eingänge

```
VAR_INPUT
Execute : BOOL;
Mode : E_AX5000ParkMode
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Mode	E_AX5000ParkMode [▶_68]	Modus zur Auswahl, ob der FB die Achse parkt oder freigibt.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥ 4022.32 TwinCAT ≥ 4024.6	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.21.0		

3.5 F_GetVersionTcMc2Drive



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

FUNCTION F_GetVersionTcMc2Drive: UINT

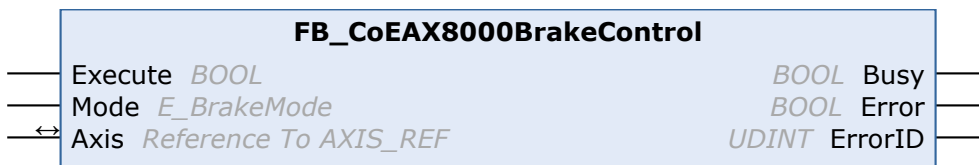
```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

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

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

3.6 AX8000 CoE

3.6.1 FB_CoEAX8000BrakeControl



Mit diesem Funktionsbaustein kann die Bremse eines Motors, der an einem AX8000 betrieben wird, manuell geöffnet oder geschlossen werden.

Dabei ist zu beachten, dass die Bremse über „Lock“, beziehungsweise „Unlock“, dauerhaft geschlossen oder geöffnet wird.

Es wird empfohlen, die Bremse nur für die nötige Dauer manuell zu öffnen oder zu schließen. Anschließend sollte die Bremse mit dem Funktionsbaustein wieder in den Modus „Automatic“ gesetzt werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Mode	E_BrakeMode [▶ 68]	Der Modus, mit dem die Bremse angesteuert wird.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

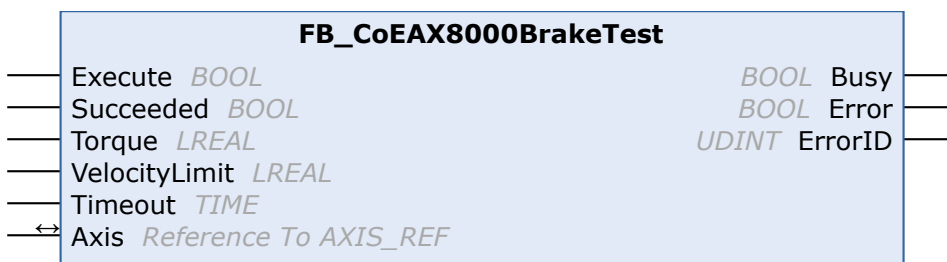
```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Ziellplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥ 4022.36 TwinCAT ≥ 4024.15	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.23.0		

3.6.2 FB_CoEAX8000BrakeTest



Der Funktionsbaustein FB_CoEAX8000BrakeTest kann für die Umsetzung eines funktionalen Bremsentests verwendet werden.

Mit Hilfe dieses Funktionsbausteins wird der AX8000 in die Drehmomentbetriebsart („Cyclic Synchronous Torque Mode“ (CST)) geschaltet. Das am Eingang parametrisierte Drehmoment „Torque“ wird als Sollwert an den Antrieb übermittelt. Das Drehmoment wird gehalten bis die Zeit „Timeout“ abgelaufen ist, oder eine Rückmeldung („Succeeded“) an den Baustein gesendet wird. Diese Rückmeldung wird typischerweise aus der Sicherheitssteuerung an die PLC gesendet. Anschließend wird der AX8000 in die ursprüngliche Betriebsart gesetzt.

Wurde die Bremse vor dem Aufruf des FB_CoEAX8000BrakeTest nicht geschlossen, oder die Bremse hält das geforderte Prüfmoment nicht, dann verhindert die im AX8000 integrierte Drehzahlklammerung ein unkontrolliertes Beschleunigen der Achse. Das Geschwindigkeitslimit wird über die Variable „VelocityLimit“ parametrisiert.

⚠ GEFAHR

Lebensgefahr oder Gefahr von schweren Verletzungen oder Sachschäden durch unbeabsichtigte Bewegungen der Achse

Bei Verwendung des Funktionsbausteins wird die Achse in den CST-Mode geschaltet. Nach Verwendung des Funktionsbausteins (insbesondere nach Fehlersituationen), kann es dazu kommen, dass sich die Achse weiterhin im CST-Mode befindet. Dieses kann beim Freigeben der Achse zu plötzlichen und ungeplanten Bewegungen (insbesondere bei Hubachsen) führen.

- Stellen Sie sicher, dass keine Gefahr im Sinne der Risikobewertung entsteht.
- Prüfen Sie die aktuelle Betriebsart über den Funktionsbaustein MC_ReadDriveOperationMode.
- Wenn sich die Achse nicht in einer positionsbezogenen Betriebsart (CSV/CSP) befindet, überführen Sie diese vor einer Freigabe:
 - *direkt* mit MC_WriteDriveOperationMode in die gewünschte positionsbezogene Betriebsart (CSV/CSP) oder
 - *indirekt* mit MC_Halt / MC_Stop in die gewünschte positionsbezogene Betriebsart (CSV/CSP) (ab TwinCAT 3.1.4024.40)

Weitere Funktionsbausteine, die die Achse indirekt in eine positionsbezogene Betriebsart schalten, können dies nur eingeschränkt und sind daher nicht für einen bewussten Betriebsartwechsel zu verwenden.

⇒ Anschließend ist ein erneutes Prüfen nötig, ob sich die Achse auch wirklich in einer positionsbezogenen Betriebsart (CSV/CSP) befindet, falls nicht, ist ein Abbruch mit Fehlerbehandlung erforderlich.

Informationen über nötige Änderungen in der Konfiguration des AX8000 können der [AX8000 | Multiachs-Servosystem Funktionsbeschreibung](#) entnommen werden.

🔌 Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Succeeded	BOOL	Rückmeldung, dass die Bremse das vorgegebene Drehmoment gehalten hat. Diese Rückmeldung kommt üblicherweise aus der Sicherheitssteuerung.
Torque	LREAL	Drehmoment, das wirken soll [Nm].
VelocityLimit	LREAL	Geschwindigkeitslimit zur Vermeidung von unkontrolliertem Beschleunigen in der Drehmomentbetriebsart.
Timeout	TIME	Maximale Zeit, die das vorgegebene Drehmoment wirkt.

🔌/🔌 Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

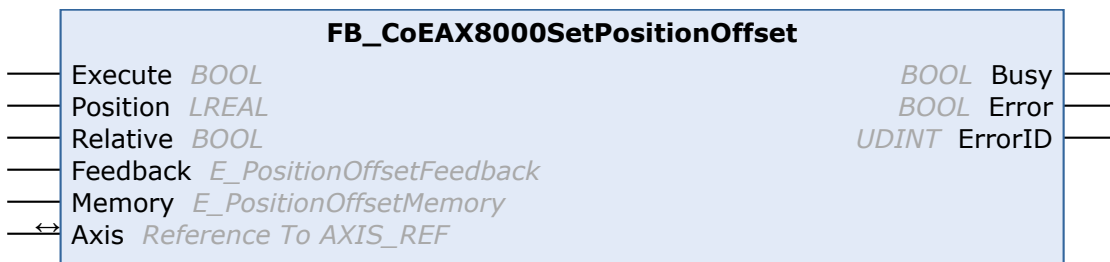
```
VAR_OUTPUT
Busy : BOOL;
Error : BOOL;
ErrorID : UDINT;
END_VAR
```

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥ 4022.36 TwinCAT ≥ 4024.15	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.23.0		

3.6.3 FB_CoEAX8000SetPositionOffset



Mit dem Funktionsbaustein FB_CoEAX8000SetPositionOffset kann ein Positions-Offset im Speicher des AX8000 oder im digitalen Typenschild des angeschlossenen Motor-Encoders abgespeichert werden. Dazu muss das Offset zuvor mit dem DriveManager konfiguriert und anschließend im Funktionsbaustein der gleiche Speicherort (Encoder / Drive) verwendet werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Ist-Position der NC-Achse
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ <i>E_PositionOffsetFeedback</i> [► 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ <i>E_PositionOffsetMemory</i> [► 67]. Gibt an, in welchem Speicher der neu berechnete Positions-Offset abgelegt werden soll.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

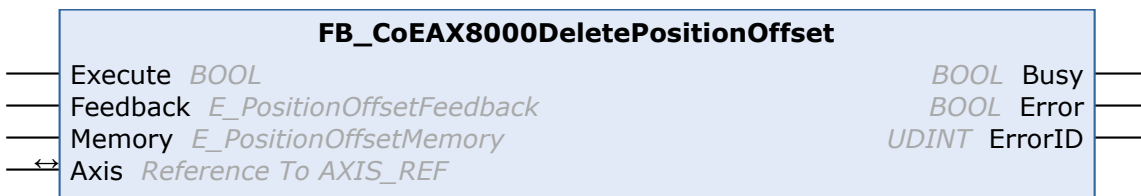
```
VAR_OUTPUT
  Busy      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken
TwinCAT ≥4020.57	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.16.0
TwinCAT ≥4022.30		

3.6.4 FB_CoEAX8000DeletePositionOffset



Mit dem Funktionsbaustein FB_CoEAX8000DeletePositionOffset kann ein Positions-Offset im Speicher des AX8000 oder im digitalen Typenschild des angeschlossenen Motor-Encoders gelöscht werden.

 **Eingänge**

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Position der NC-Achse wird angezeigt.

Name	Typ	Beschreibung
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ E_PositionOffsetFeedback [► 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ E_PositionOffsetMemory [► 67]. Gibt an, aus welchem Speicher der Positions-Offset gelöscht werden soll.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

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

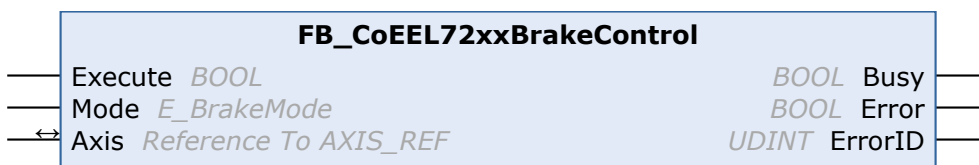
Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken
TwinCAT ≥4020.57	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.16.0
TwinCAT ≥4022.30		

3.7 EL72xx CoE

3.7.1 FB_CoEEL72xxBrakeControl



Mit diesem Funktionsbaustein kann die Bremse eines Motors, der an einer EL72xx betrieben wird, manuell geöffnet oder geschlossen werden.

Dabei ist zu beachten, dass die Bremse über „Lock“, beziehungsweise „Unlock“, dauerhaft geschlossen oder geöffnet wird.

Es wird empfohlen, die Bremse nur für die nötige Dauer manuell zu öffnen oder zu schließen. Anschließend sollte die Bremse mit dem Funktionsbaustein wieder in den Modus „Automatic“ gesetzt werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Mode	<u>E_BrakeMode</u> [► 68]	Der Modus, mit dem die Bremse angesteuert wird.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT ≥4024.48	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.34.0		≥0032

3.7.2 FB_CoEEL72xxSetPositionOffset



Mit dem Funktionsbaustein FB_CoEEL72xxSetPositionOffset kann ein Positions-Offset im Speicher der EL72xx abgespeichert werden. Ein Speichern im digitalen Typenschild des angeschlossenen Motor-Encoders ist momentan nicht vorgesehen.

 **Eingänge**

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Ist-Position der NC-Achse
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ E_PositionOffsetFeedback [► 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ E_PositionOffsetMemory [► 67]. Gibt an, in welchem Speicher der neu berechnete Positions-Offset abgelegt werden soll.

 **Ein-/Ausgänge**

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

Name	Typ	Beschreibung
Axis	AXIS_REF	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

 **Ausgänge**

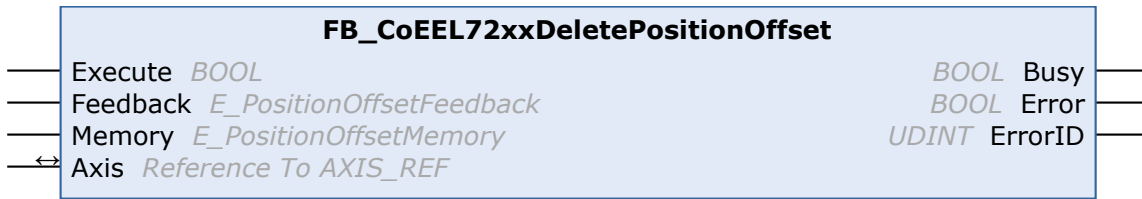
```
VAR_OUTPUT
  Busy : BOOL;
  Error : BOOL;
  ErrorID : UDINT;
END_VAR
```

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielpattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT >4020.57 TwinCAT >4022.30 TwinCAT ≥4024.4	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.17.0	≥ 19	≥0030

3.7.3 FB_CoEEL72xxDeletePositionOffset



Mit dem Funktionsbaustein FB_CoEEL72xxDeletePositionOffset kann ein Positions-Offset im Speicher der EL72xx gelöscht werden.

Eingänge

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

Name	Typ	Beschreibung
Execute	BOOL	Der Baustein wird über eine positive Flanke an diesem Eingang aktiviert.
Position	LREAL	Neue Position der NC-Achse wird angezeigt.
Relative	BOOL	Wenn dieses Flag gesetzt wird, wird die Position relativ interpretiert.
Feedback	E_PositionOffsetFeedback	Enumeration vom Typ <u>E_PositionOffsetFeedback</u> [▶ 67]. Gibt an, welches Feedback betrachtet wird.
Memory	E_PositionOffsetMemory	Enumeration vom Typ <u>E_PositionOffsetMemory</u> [▶ 67]. Gibt an, aus welchem Speicher der Positions-Offset gelöscht werden soll.

Ein-/Ausgänge

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

Name	Typ	Beschreibung
Axis	<u>AXIS_REF</u>	Achsdatenstruktur, welche eine Achse eindeutig im System adressiert. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

Ausgänge

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

Name	Type	Beschreibung
Busy	BOOL	Dieser Ausgang wird bei der Aktivierung des Funktionsbausteins gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
Error	BOOL	Dieser Ausgang wird gesetzt, nachdem der Busy-Ausgang zurückgesetzt wurde, wenn ein Fehler bei der Übertragung des Kommandos auftritt.
ErrorID	UDINT	Liefert bei gesetztem Error-Ausgang den ADS-Fehlercode.

Voraussetzungen

Entwicklungsumgebung	Zielplattformtyp	Einzubindende SPS Bibliotheken	Firmwareversion	Revision
TwinCAT >4020.57 TwinCAT >4022.30 TwinCAT ≥4024.4	PC oder CX (x86 oder x64)	Tc2_MC2_Drive ≥V3.3.17.0	≥ 19	≥0030

4 Datentypen

4.1 E_FwUpdateState

Die Enumeration 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

Name	Beschreibung
eFwU_NoError	Initialzustand
eFwU_CheckCfgIdentity	Einlesen der konfigurierten Slavetypen (Anzahl Kanäle, Strom, Revision).
eFwU_CheckSlaveCount	Ermitteln der konfigurierten Slaveanzahl.
eFwU_CheckFindSlavePos	Suchen der Slaveadresse im Master-Objektverzeichnis.
eFwU_WaitForScan	Warten auf Online-Scan.
eFwU_ScanningSlaves	Online-Scan der Slaves.
eFwU_CheckScannedIdentity	Einlesen der gescannten Slavetypen (Anzahl Kanäle, Strom, Revision).
eFwU_CheckScannedFirmware	Einlesen der Firmware-Version.
eFwU_FindFirmwareFile	Suchen nach der gewählten Firmware-Datei.
eFwU_WaitForUpdate	Warten auf Status des Updates.
eFwU_WaitForSlaveState	Ermitteln des EtherCAT-Slave-Status.
eFwU_StartFwUpdate	Starten des Firmware-Updates.
eFwU_FwUpdateInProgress	Firmware-Update aktiv.

Name	Beschreibung
eFwU_FwUpdateDone	Firmware-Update erfolgreich beendet.
eFwU_NoFwUpdateRequired	Kein Firmware-Update erforderlich.
eFwU_UpdateViaOtherChannelActive	Update erfolgt über den anderen Achskanal.
eFwU_UpdatedViaOtherChannel	Update erfolgte über den anderen Achskanal.

Update-Fehler

Name	Beschreibung
eFwU_GetSlaveIdentityError	Einlesen des konfigurierten Slavetypen schlug fehl (siehe iAdsErrId).
eFwU_GetSlaveCountError	Ermitteln der konfigurierten Slaveanzahl schlug fehl (siehe iAdsErrId).
eFwU_GetSlaveAddrError	Suchen der Slaveadresse 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 der gescannten Slavetypen (Anzahl Kanäle, Strom, Revision) schlug fehl (siehe iAdsErrId).
eFwU_GetSlaveStateError	Ermitteln des EtherCAT-Slave-Status 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	Die Anzahl der konfigurierten und 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-Typ 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-Status schlug fehl.

4.2 E_PositionOffsetMemory

```

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

Name	Beschreibung
ePositionOffsetMemory_Encoder	Als Speicher für den neuen Positions-Offset wird das Motortypenschild des Encoders gewählt.
ePositionOffsetMemory_Drive	Als Speicher für den neuen Positions-Offset wird der interne Speicher des Drives gewählt.

4.3 E_PositionOffsetFeedback

```

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

Name	Beschreibung
ePositionOffsetFeedback1	Bezug für die Neuberechnung des Positions-Offsets ist Feedback-System 1.
ePositionOffsetFeedback2	Bezug für die Neuberechnung des Positions-Offsets ist Feedback-System 2.

4.4 E_DriveErrorCodes

```

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

```

4.5 E_AX5000ParkMode

Mit diesem Enum wird festgelegt, ob der `FB_SoEAX5000ParkAxis` [► 55] den korrespondierenden Kanal des AX5000 parkt oder freigibt.

```

TYPE E_AX5000ParkMode : (
  eAX5000ParkMode_Park,
  eAX5000ParkMode_Release
) INT;
END_TYPE

```

Name	Beschreibung
eAX5000ParkMode_Park	AX5000 Kanal soll geparkt werden.
eAX5000ParkMode_Release	AX5000 Kanal soll freigegeben werden.

4.6 E_BrakeMode

Legt den Modus für die manuelle Bremsenansteuerung fest.

```

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

```

Name	Beschreibung
eBrakeMode_Automatic	Die Bremse des angeschlossenen Motors wird automatisch geöffnet oder geschlossen.
eBrakeMode_Lock	Die Bremse des angeschlossenen Motors bleibt dauerhaft geschlossen. Das Enablen der Achse über den MC_Power öffnet die Bremse nicht.
eBrakeMode_Unlock	Die Bremse des angeschlossenen Motors wird dauerhaft geöffnet. Das Disablen der Achse über den MC_Power schließt die Bremse nicht. Siehe Warnung.

WARNUNG

Gefahr von tödlichen oder schweren Verletzungen

Beachten und vermeiden Sie, dass vertikale Achsen herunterfallen können!

4.7 ST_DriveInfo

Struktur mit Basis-Informationen zur verwendeten Beckhoff-Servo-Hardware, welche über den Funktionsbaustein [FB_ReadDriveInfo](#) [[▶ 13](#)] zurückgegeben wird.

```

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

Name	Typ	Beschreibung
NetId	T_AmsNetId	String, der die AMS-Netzwerkennung des EtherCAT-Master-Gerätes enthält.
SlaveAddress	T_AmsPort	Feste Adresse des EtherCAT-Slaves
Channel	USINT	Kanal des EtherCAT-Slaves
MDPProfile	BOOL	TRUE, falls ein MDP-Protokoll verwendet wird.
DeviceName	STRING	Enthält den Namen des Gerätes (nicht den freien Namen des Gerätes) im IO-Baum.
DeviceType	E_DeviceType ▶ 69	Enumeration, die angibt, um welches Beckhoff-Antriebsgerät es sich handelt.

4.8 E_DeviceType

Enumeration zur Definition des Beckhoff-Antriebsgeräts.

```

TYPE E_DeviceType : (
  DEVICETYPE_UNKNOWN          := 0,
  DEVICETYPE_SOE_DEFAULT     := 1,
  DEVICETYPE_AX2000          := 21,
  DEVICETYPE_EL72x1         := 22,
  DEVICETYPE_EL72x1_OCT      := 23,
  DEVICETYPE_EL72x1_OCT_SAFETY := 24,
  DEVICETYPE_AX8000          := 31,
  DEVICETYPE_AMI8xxx         := 41,
  DEVICETYPE_AMP8xxx         := 51,
  DEVICETYPE_EL74x1          := 61,
  DEVICETYPE_EL74x1_SAFETY   := 62,
  DEVICETYPE_ELM72xx_OCT     := 71,
  DEVICETYPE_ELM72xx_OCT_SAFETY := 72
) UDINT;
END_TYPE
    
```

Name	Beschreibung
DEVICETYPE_UNKNOWN	Unbekanntes Antriebsgerät
DEVICETYPE_SOE_DEFAULT	Standard SoE-Antrieb
DEVICETYPE_AX2000	AX2000
DEVICETYPE_EL72x1	EL72x1
DEVICETYPE_EL72x1_OCT	EL72x1 mit OCT
DEVICETYPE_EL72x1_OCT_SAFETY	EL72x1 mit OCT und Safety
DEVICETYPE_AX8000	AX8000
DEVICETYPE_AMI8xxx	AMI8xxx
DEVICETYPE_AMP8xxx	AMP8xxx
DEVICETYPE_EL74x1	EL74x1
DEVICETYPE_EL74x1_SAFETY	EL74x1 mit Safety
DEVICETYPE_ELM72xx_OCT	ELM72xx mit OCT
DEVICETYPE_ELM72xx_OCT_SAFETY	ELM72xx mit OCT und Safety

5 Globale Konstanten

5.1 SoE Parameter Access

Mit Hilfe dieser Konstanten kann auf die Parameter der einzelnen Parametersätze zugegriffen werden.

Global_Variables

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

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

6 Beispiele

Beispielprojekt und -konfiguration für die Diagnose von AX5000

Download: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_MC2_Drive/Resources/2327326731.zip

Beispielprojekt und -konfiguration für die Diagnose von IndraDrive Cs

Download: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_MC2_Drive/Resources/2327323403.zip

7 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

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

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

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

