

取扱説明書

PLCライブラリ: Tc2\_NcDrive

TwinCAT 3

バージョン: 1.1  
日付: 2020-06-17

**BECKHOFF**



# 目次

1	序文	5
1.1	取扱説明書に関する注記	5
1.2	安全に関する指示事項	5
2	概要	7
3	ファンクションブロック	10
3.1	汎用SoE	10
3.1.1	FB_SoEReset	10
3.1.2	FB_SoEWritePassword	11
3.1.3	コマンド用ファンクションブロック	13
3.1.4	診断用ファンクションブロック	17
3.1.5	現在値を判定するファンクションブロック	23
3.2	汎用CoE	32
3.2.1	現在値を判定するファンクションブロック	32
3.3	AX5000 SoE	35
3.3.1	FB_SoEAX5000ReadActMainVoltage	35
3.3.2	FB_SoEAX5000SetMotorCtrlWord	36
3.3.3	FB_SoEAX5000FirmwareUpdate	37
3.4	F_GetVersionTcNcDrive	41



# 1 序文

## 1.1 取扱説明書に関する注記

この説明は対応する国内規格を熟知した、トレーニングを受けた制御、オートメーションエンジニアリングの専門技術者のみの使用を対象としています。

コンポーネントのインストールとコミッショニングの際には、取扱説明書および以下の注意事項と説明に従うことが重要です。

技術者には各設置およびコミッショニングのそれぞれの時点で、発行された取扱説明書を使用する義務があります。

本製品を使用するうえでの責任者は、本製品の用途および使用方法が、関連するすべての法律、法規、ガイドラインおよび規格を含む、安全に関するすべての要件を満たしていることを確認してください。

### 免責事項

この取扱説明書の記載内容は、一般的な製品説明および性能を記載したものであり、場合により記載通りに動作しないことがあります。

製品の情報・仕様は予告なく変更されます。

この説明書に記載されているデータ、図および説明に基づいて、すでに納品されている製品の変更を要求することはできません。

### 商標

Beckhoff®、TwinCAT®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS®およびXPlanar®は、Beckhoff Automation GmbHの登録商標です。

この取扱説明書で使用されているその他の名称は商標である可能性があり、第三者が独自の目的のために使用すると所有者の権利を侵害する可能性があります。

### 特許出願

EtherCAT Technologyについては、欧州特許 EP1590927、EP1789857、EP1456722およびEP2137893、ドイツ特許DE102015105702に記載されていますが、これらに限定されるものではありません。

## EtherCAT®

EtherCAT®は、Beckhoff Automation GmbH (ドイツ)によりライセンスを受けた登録商標および特許技術です。

### 著作権

© Beckhoff Automation GmbH & Co. KG, Germany.

明示的な許可なく、本書の複製、配布、使用、および他への内容の転載は禁止されています。

これに違反した者は損害賠償の責任を負います。すべての権利は、特許、実用新案、意匠の付与の際に留保されます。

## 1.2 安全に関する指示事項

### 安全に関する注意事項

この取扱説明書に記載された安全に関する指示や注意事項はよくお読みになり、必ず指示に従ってください。

## 納入仕様

すべての製品は、用途に適した特定のハードウェア構成およびソフトウェア構成を有する状態で供給されます。ハードウェアまたはソフトウェアに取扱説明書に記載されている以外の変更を加えることは許可されていません。許可されていない変更を加えると、Beckhoff Automation GmbH & Co. KGの保証の対象外となります。

## 使用者の資格

この説明書は関連する国内法規を熟知した、制御およびオートメーションエンジニアリングの専門家の使用を目的としています。

## 安全記号の説明

この取扱説明書では、安全に関する指示や注意事項とともに以下の安全記号を使用します。安全に関する指示事項はよくお読みになり、必ず指示に従ってください。

### ⚠ 危険

#### 重大な人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に直ちに危害を及ぼします。

### ⚠ 警告

#### 人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼします。

### ⚠ 注意

#### 人的傷害の恐れ

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼす恐れがあります。

### 注記

#### 物的損害と環境汚染

この記号が付いた安全に関する注意事項に従わないと、物的損害と環境汚染をもたらす恐れがあります。

#### ● ヒントまたはアドバイス

**i** この記号が示す情報により、さらに理解が深まります。

## 2 概要



Tc2\_NcDriverライブラリは新しいプロジェクトでは使用せず、代わりにTc2\_MC2\_Driveライブラリを使用してください（マニュアル『[TwinCAT 3 PLC Lib Tc2\\_MC2\\_Drive](#)』を参照）。

Tc2\_NcDriveライブラリには、MC2軸構造（AXIS\_REF）によってドライブにアクセスするSoEドライブ用のファンクションおよびファンクションブロックが含まれています。

### ドライブライブラリ

異なる機能用途向けに3つのドライブライブラリTc2\_Drive、Tc2\_NcDrive、およびTc2\_MC2\_Driveが開発されていますが、各ライブラリの機能はほぼ同一です。ライブラリTc2\_NcDriveおよびTc2\_MC2\_Driveのファンクションブロックは、Tc2\_Driveライブラリのファンクションブロックを内包するラッパーファンクションブロックを形成します。

ドライブライブラリ	使用	ドライブへのアクセス	注
Tc2_Drive マニュアル 『 <a href="#">TwinCAT 3 PLC Lib: Tc2_Drive</a> 』を参照。	ドライブをすべてPLCから（NCを介さずに）使用する場合は、Tc2_Driveライブラリを使用します。	ドライブには、ドライブリファレンスによってアクセスされません。ドライブへアクセスするために、このライブラリではST_DriveRef構造体、および文字列のNetIDが使用されます。リンクを行うために、ST_PlcDriveRefと呼ばれる構造体、およびバイト列のNetIDも提供されています。 （「」を参照）	特殊なファンクションブロックが実装されていないドライブのパラメータにアクセスする場合は、ファンクションブロックFB_SoERead_ByDriveRefおよびFB_SoEWrite_ByDriveRefを使用します。これらのファンクションブロックは、SoE InterfaceフォルダのPLC Lib Tc2_EtherCATに実装されています。
Tc2_NcDrive マニュアル 『 <a href="#">TwinCAT 3 PLC Lib: Tc2_NcDrive</a> 』を参照。	Tc2_NcDriveライブラリによってNCを介してドライブを使用している場合は、Tc2_NcDriveライブラリを使用します。	ドライブには、NC軸構造体（NC_TO_PLC）によってアクセスされます。Tc2_NcDriveライブラリのファンクションブロックは、NC軸構造体のNC軸IDによって、ドライブへのアクセスデータ（NetID、アドレス、およびチャンネル番号）を独自に判別します。	特殊なファンクションブロックが実装されていないドライブのパラメータにアクセスする場合は、ファンクションブロックFB_SoEReadおよびFB_SoEWriteを使用します。
Tc2_MC2_Drive マニュアル 『 <a href="#">TwinCAT 3 PLC Lib Tc2_MC2_Drive</a> 』を参照。	Tc2_MC2DriveライブラリによってNCを介してドライブを使用している場合は、Tc2_MC2_Driveライブラリを使用します。	ドライブには、MC2軸リファレンス（AXIS_REF）によってアクセスされます。Tc2_MC2_Driveライブラリのファンクションブロックは、MC2軸リファレンスのNC軸IDによって、ドライブへのアクセスデータ（NetID、アドレス、およびチャンネル番号）を独自に判別します。	特殊なファンクションブロックが実装されていないドライブのパラメータにアクセスする場合は、ファンクションブロックFB_SoEReadおよびFB_SoEWriteを使用します。



AX5000のドライブライブラリ、およびBosch Rexroth IndraDrive CSのドライブライブラリ使用時の違いについて注意してください（を参照）。

## ファンクション

名称	説明
<a href="#">F_GetVersionTcNcDrive [▶_41]</a>	PLCライブラリからバージョン情報を読み取ります。このファンクションは、グローバル構造体 <code>stLibVersion_Tc2_NcDrive</code> で置換されています。
<a href="#">F_ConvWordToSTAX5000C1D</a>	AX5000のC1Dワード (S-0-0011) を <code>ST_AX5000_C1D</code> 構造体に変換します。 TwinCAT 3 PLC Libマニュアル: <code>Tc2_Drive</code> をご参照ください。

## ファンクションブロック

名称	説明
<a href="#">FB_SoEReset [▶_10]</a>	ドライブ (S-0-0099) をリセットします。
<a href="#">FB_SoEWritePassword [▶_11]</a>	ドライブパスワード (S-0-0267) を設定します。
<a href="#">FB_SoEReadDiagMessage [▶_17]</a>	診断メッセージ (S-0-0095) を読み取ります。
<a href="#">FB_SoEReadDiagNumber [▶_18]</a>	診断番号 (S-0-0390) を読み取ります。
<a href="#">FB_SoEReadDiagNumberList [▶_20]</a>	診断番号リスト (最大30エントリ) (S-0-0375) を読み取ります。
<a href="#">FB_SoEExecuteCommand [▶_13]</a>	コマンドを実行します。
<a href="#">FB_SoEWriteCommandControl [▶_14]</a>	コマンドコントロールを設定します。
<a href="#">FB_SoEReadCommandState [▶_15]</a>	コマンドのステータスをチェックします。
<a href="#">FB_SoEReadClassXDiag [▶_21]</a>	クラス1診断 (S-0-0011) ~ クラス3診断 (S-0-0013) を読み取ります。
<a href="#">FB_SoERead [▶_23]</a>	パラメータを読み取ります。
<a href="#">FB_SoEWrite [▶_25]</a>	パラメータを書き込みます。
<a href="#">FB_SoEReadAmplifierTemperature [▶_27]</a>	ドライブ温度 (S-0-0384) を読み取ります。
<a href="#">FB_SoEReadMotorTemperature [▶_28]</a>	モータ温度 (S-0-0383) を読み取ります。
<a href="#">FB_SoEReadDcBusCurrent [▶_29]</a>	DCバス電流 (S-0-0381) を読み取ります。
<a href="#">FB_SoEReadDcBusVoltage [▶_31]</a>	DCバス電圧 (S-0-0380) を読み取ります。
<a href="#">FB_SoEAX5000ReadActMainVoltage [▶_35]</a>	主電圧 (P-0-0200) を読み取ります。
<a href="#">FB_SoEAX5000SetMotorCtrlWord [▶_36]</a>	モータコントロールワード (P-0-0096) を設定します。
<a href="#">FB_SoEAX5000FirmwareUpdate [▶_37]</a>	AX5000に対して自動ファームウェア更新を実行します。
<a href="#">FB_CoERead [▶_32]</a>	パラメータを読み取ります。
<a href="#">FB_CoEWrite [▶_33]</a>	パラメータを書き込みます。



## 要件

コンポーネント	バージョン
開発用コンピュータのTwinCAT	3.1 Build 4016以降
Windows CE-ImageのTwinCAT	3.1 Build 4016以降
Windows XP-ImageのTwinCAT	3.1 Build 4016以降

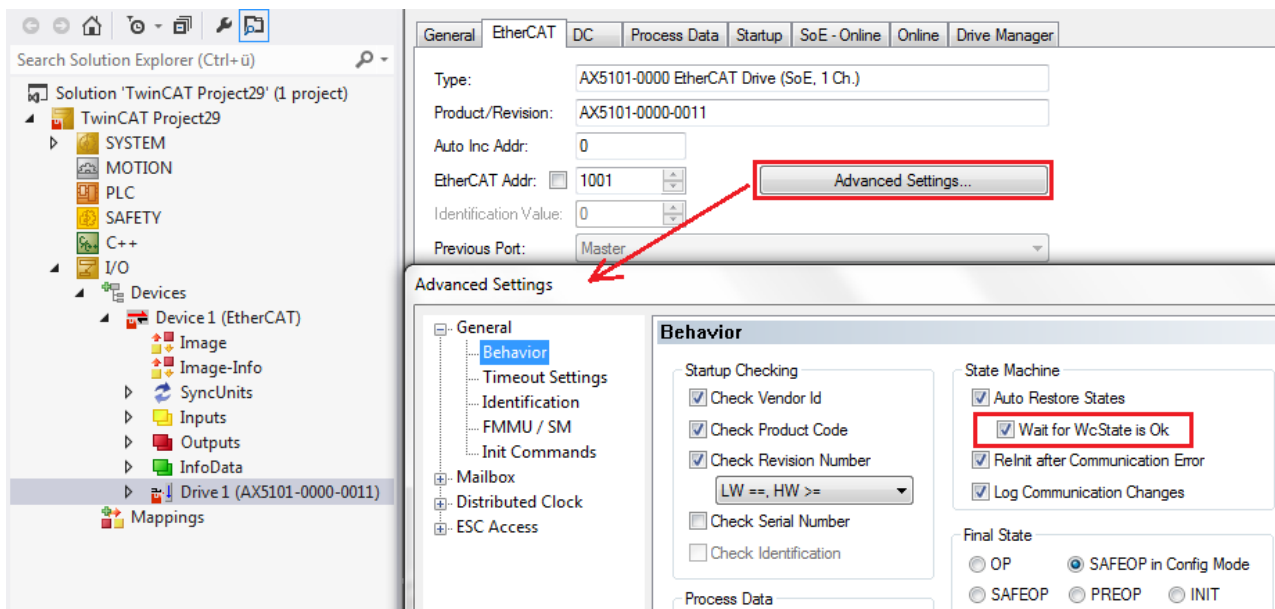
## 3 ファンクションブロック

### 3.1 汎用SoE

#### 3.1.1 FB\_SoEReset



ファンクションブロックFB\_SoEResetを使用して、ドライブ（S-0-0099）をリセットできます。マルチチャンネルデバイスの場合は、必要に応じて両方のチャンネルでリセットを実行する必要があります。エラーによっては、リセットに最大10秒かかるため、タイムアウト時間は10秒にする必要があります。AX5000の場合、EtherCATの詳細設定でフラグ[Wait For WcState is OK]を有効にする必要があります。



NCリセットは実行されません。NCリセットが必要な場合は、Tc2\_MC2 PLCライブラリからファンクションブロックMC\_Resetを使用して実行できます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

**VAR\_IN\_OUT**

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

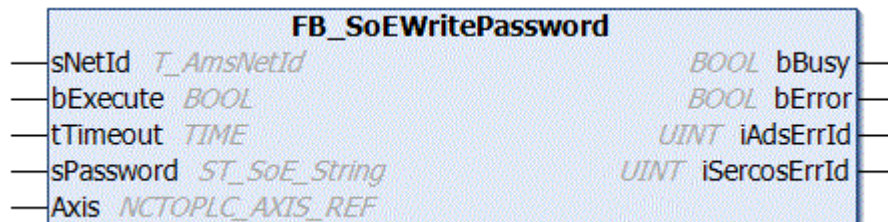
**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**サンプル**

```
fbSoEReset : FB_SoEReset_ByDriveRef;
bSoEReset : BOOL;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

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

**3.1.2 FB\_SoEWritePassword**

ファンクションブロックFB\_SoEWritePassword (S-0-0267) を使用して、ドライブパスワードを設定できません。

**VAR\_INPUT**

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

**sNetId**: PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**sPassword**: Sercos文字列のパスワードです。

**VAR\_IN\_OUT**

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

**Axis**: タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、確認レスポンスを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**サンプル**

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

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

### 3.1.3 コマンド用ファンクションブロック

#### 3.1.3.1 FB\_SoEExecuteCommand



FB\_SoEExecuteCommandファンクションブロックを使用して、コマンドを実行できます。

#### VAR\_INPUT

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

**sNetId**: PCのAMSネットワークIDを含む文字列です（タイプ: T\_AmsNetId）。

**nIdn**: FB\_SoEExecuteCommandが参照するパラメータ番号です。例えば、P-0-0160の場合は「P\_0\_IDN + 160」となります。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

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

**Axis**: タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

### サンプル

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

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

### 3.1.3.2 FB\_SoEWriteCommandControl



FB\_SoEWriteCommandControlファンクションブロックを使用して、コマンドを準備、開始、またはキャンセルできます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です（タイプ: T\_AmsNetId）。

**nIdn:** FB\_SoEWriteCommandControlが参照するパラメータ番号です。例えば、P-0-0160の場合は「P\_0\_IDN + 160」となります。

**eCmdControl:** コマンドを準備（eSoE\_CmdControl\_Set := 1）、実行（eSoE\_CmdControl\_SetAndEnable := 3）、または中止（eSoE\_CmdControl\_Cancel := 0）するかどうかを示します。

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

```

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

```

**bbBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bbBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

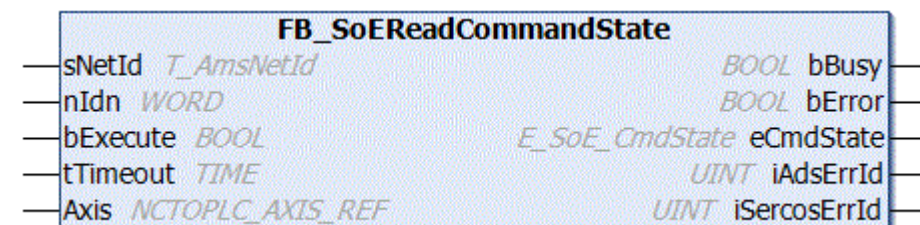
**サンプル**

```

fbWriteCommandControl : FB_SoEWriteCommandControl;
bWriteCommandControl : BOOL;
nIdn : WORD;
eCmdControl : E_SoE_CmdControl;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

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

```

**3.1.3.3 FB\_SoEReadCommandState**

FB\_SoEReadCommandStateファンクションブロックを使用して、コマンド実行をチェックできます。

**VAR\_INPUT**

```

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

```

**sNetId**: PCのAMSネットワークIDを含む文字列です（タイプ: T\_AmsNetId）。

**Idn**: FB\_SoEReadCommandStateが参照するパラメータ番号です。例えば、P-0-0160の場合は「P\_0\_IDN + 160」となります。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

## VAR\_IN\_OUT

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

**Axis**: タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

## VAR\_OUTPUT

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返しません。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返しません。

**eCmdState**: コマンドのステータスを返します。

```
eSoE_CmdState_NotSet = 0
- kein Kommando aktiv

eSoE_CmdState_Set = 1
- Kommando gesetzt (vorbereitet) aber (noch) nicht ausgeführt

eSoE_CmdState_Executed = 2
- Kommando wurde ausgeführt

eSoE_CmdState_SetEnabledExecuted = 3
- Kommando gesetzt (vorbereitet) und ausgeführt

eSoE_CmdState_SetAndInterrupted = 5
- Kommando wurde gesetzt aber unterbrochen

eSoE_CmdState_SetEnabledNotExecuted = 7
- Kommandoausführung ist noch aktiv

eSoE_CmdState_Error = 15
- Fehler bei der Kommandoausführung, es wurde in den Fehlerstate gewechselt
```

## サンプル

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



```

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

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

```

### 3.1.4 診断用ファンクションブロック

#### 3.1.4.1 FB\_SoEReadDiagMessage



FB\_SoEReadDiagMessageファンクションブロックを使用して、診断メッセージをSercos文字列（S-0-0095）として読み取れます。

#### VAR\_INPUT

```

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

```

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

```

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

```

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

```

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

```

```
sDiagMessage : ST_SoE_String;
dwAttribute : DWORD;
END_VAR
```

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute**: Sercosパラメータの属性を返します。

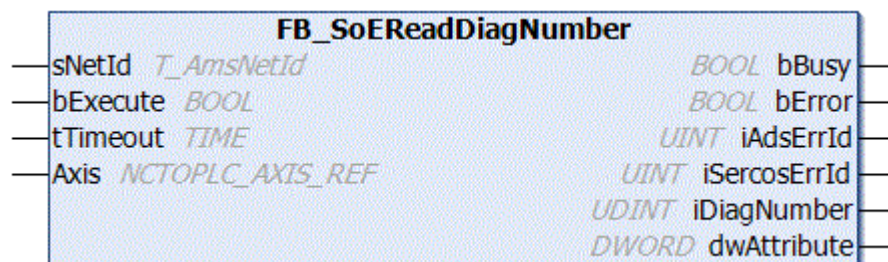
**sDiagMessage**: 診断メッセージを返します。

### サンプル

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

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

### 3.1.4.2 FB\_SoEReadDiagNumber



FB\_SoEReadDiagNumber ファンクションブロックを使用して、現在の診断番号をUDINT (S-0-0390) として読み取れます。

#### VAR\_INPUT

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

**sNetId**: PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

## VAR\_IN\_OUT

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

**Axis**: タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

## VAR\_OUTPUT

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute**: Sercosパラメータの属性を返します。

**iDiagNumber**: 現在の診断番号を返します。

## サンプル

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

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

### 3.1.4.3 FB\_SoEReadDiagNumberList



FB\_SoEReadDiagNumberListファンクションブロックを使用して、診断番号の履歴をリスト（S-0-0375）として読み取れます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

**piDiagNumber:** 最後の最大30件のエラー番号のリストへのポインタです。このリストは、リストの現在のバイト数および最大バイト数と、30件のリストエントリで構成されます。

**iSize:** バイト単位でのリストのサイズ (Sizeof()) です。

#### VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** bError出力が設定されている場合、最後に実行されたコマンドに関連するADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute:** Sercosパラメータの属性を返します。

## サンプル

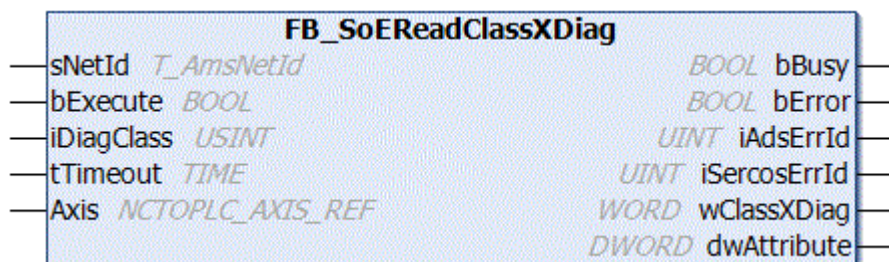
```

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

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

```

## 3.1.4.4 FB\_SoEReadClassXDiag



FB\_SoEReadClassXDiagファンクションブロックを使用して、現在のクラス1診断（S-0-0011）～クラス3診断（S-0-0013）をWORDとして読み取れます。構造体ST\_AX5000\_C1Dとしてクラス1 診断を評価する変換関数F\_ConvWordToSTAX5000C1Dが用意されています（TwinCAT 3 PLC Lib Tc2\_Driveマニュアルを参照）。

## VAR\_INPUT

```

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

```

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**iDiagClass:** どの診断を読み取るかを指定します。診断パラメータは、ベンダーによって異なる場合があります。すべての診断パラメータ（C1D～C3D）、またはすべてのビットが診断パラメータに含まれているとは限りません。

1: エラー: クラス1診断（S-0-0011）

2: 警告: クラス2診断（S-0-0012）

3: 情報メッセージ: クラス3診断（S-0-0013）

**tTimeout:** ファンクションブロックの最大許容実行時間です。

**VAR\_IN\_OUT**

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**wClassXDiag:** 現在のクラスX診断を返します。

**dwAttribute:** Sercosパラメータの属性を返します。

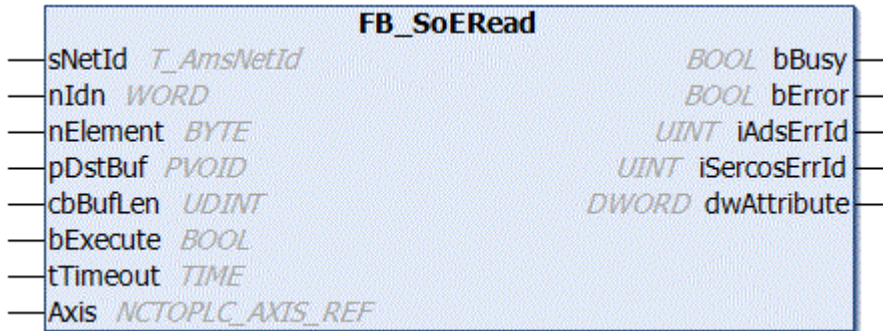
**サンプル**

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

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

### 3.1.5 現在値を判定するファンクションブロック

#### 3.1.5.1 FB\_SoERead



FB\_SoEReadファンクションブロックを使用して、パラメータを読み取れます。

#### VAR\_INPUT

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

**sNetId**: PCのAMSネットワークIDを含む文字列です（タイプ: T\_AmsNetId）。

**nIdn**: FB\_SoEReadが参照するパラメータ番号です。例えば、P-0-0033の場合は「S\_0\_IDN + 33」となります。

**nElement**: パラメータのどの部分にアクセスするかを指定します。例えば、16#40がパラメータの値 (Value) です。

```
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**: 値が読み取られる変数のADR () です。

**cbBufLen**: 値が読み取られる変数のsizeof () です。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**VAR\_IN\_OUT**

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、確認レスポンスを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute:** Sercosパラメータの属性を返します。

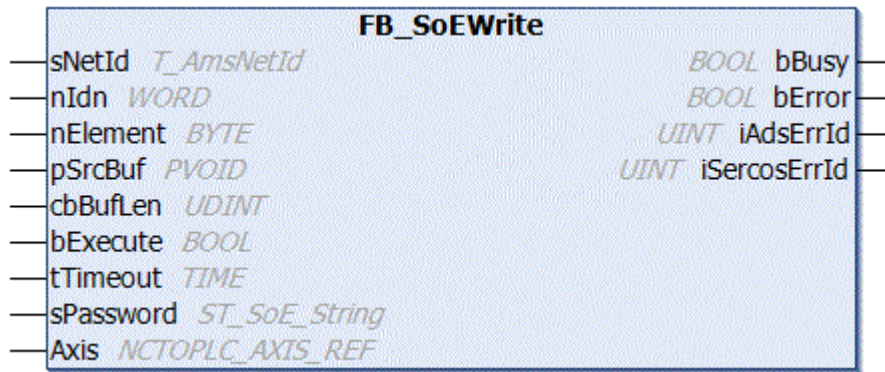
**サンプル**

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

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



## 3.1.5.2 FB\_SoEWrite



FB\_SoEWriteファンクションブロックを使用して、パラメータを書き込みます。

## VAR\_INPUT

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

**sNetId**: PCのAMSネットワークIDを含む文字列です（タイプ: T\_AmsNetId）。

**nIdn**: FB\_SoEReadが参照するパラメータ番号です。例えば、P-0-0047の場合は「S\_0\_IDN + 47」となります。

**nElement**: パラメータのどの部分にアクセスするかを指定します。例えば、16#40がパラメータの値 (Value) です。通常、この値には書き込みアクセスのみ可能であり、このパラメータの他の部分は読み取り専用です。

```
EC_SOE_ELEMENT_DATASTATE :BYTE :=16#01;
EC_SOE_ELEMENT_NAME      :BYTE :=16#02;
EC_SOE_ELEMENT_ATTRIBUTE :BYTE :=16#04;
EC_SOE_ELEMENT_UNIT      :BYTE :=16#08;
EC_SOE_ELEMENT_MIN       :BYTE :=16#10;
EC_SOE_ELEMENT_MAX       :BYTE :=16#20;
EC_SOE_ELEMENT_VALUE     :BYTE :=16#40;
EC_SOE_ELEMENT_DEFAULT   :BYTE :=16#80;
```

**pSrcBuf**: 書き込まれる値を含む変数のADR()です。

**cbBufLen**: 書き込まれる値を含む変数のsizeof()です。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**sPassword**: Sercos文字列のパスワードです。現在は使用されていません。パスワードは FB\_SoEWritePassword [▶ 11]で書き込む必要があります。

**VAR\_IN\_OUT**

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、確認レスポンスを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返しません。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返しません。

**サンプル**

```
fbWrite : FB_SoEWrite;
bWrite : BOOL;
iWriteValue : UINT;
sPassword : ST_SoE_String;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

IF bWrite THEN
  nIdn := S_0_IDN + 33;
  fbWrite(
    Axis := stNcToPlc,
    nIdn := nIdn,
    nElement := 16#40,
    pSrcBuf := ADR(iWriteValue),
    cbBufLen := SIZEOF(iWriteValue),
    sPassword := sPassword,
    bExecute := TRUE,
    tTimeout := DEFAULT_ADS_TIMEOUT,
  );
  IF NOT fbWrite.bBusy THEN
    fbWrite(Axis := stNcToPlc, bExecute := FALSE);
    bWrite := FALSE;
  END_IF
END_IF
```

### 3.1.5.3 FB\_SoEReadAmplifierTemperature



ファンクションブロックFB\_SoEReadAmplifierTemperatureを使用して、ドライブの温度（S-0-0384）を読み取れます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute:** Sercosパラメータの属性を返します。

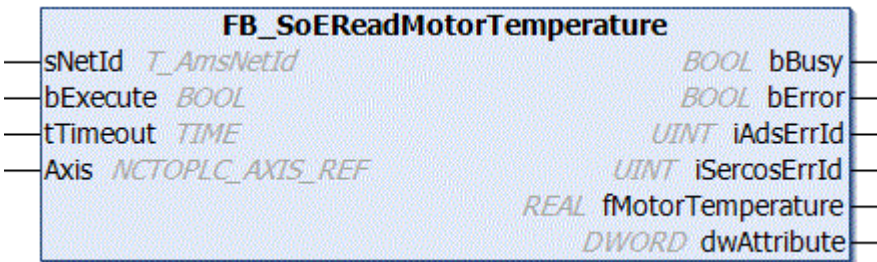
**fAmplifierTemperature:** ドライブ温度を返します（例えば、26.2は26.2°Cを意味します）。

サンプル

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

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

3.1.5.4 FB\_SoEReadMotorTemperature



ファンクションブロックFB\_SoEReadMotorTemperature を使用して、モータの温度 (S-0-0383) を読み取れます。モータに温度センサが搭載されていない場合、0.0 (0.0°C) となります。

VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

VAR\_OUTPUT

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

```
fMotorTemperature : REAL;
dwAttribute       : DWORD;
END_VAR
```

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute**: Sercosパラメータの属性を返します。

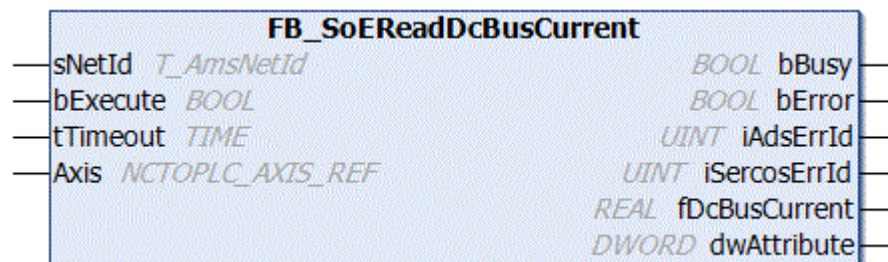
**fMotorTemperature**: モータ温度を返します（例えば、30.5は30.5°Cを意味します）。モータに温度センサが搭載されていない場合、0.0（0.0°C）となります。

**サンプル**

```
fbReadMotorTemp : FB_SoEReadMotorTemperature;
bReadMotorTemp : BOOL;
fMotorTemperature : REAL;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

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

**3.1.5.5 FB\_SoEReadDcBusCurrent**



ファンクションブロックFB\_SoEAX5000ReadDcBusCurrentを使用して、DCバス電流（S-0-0381）を読み取れます。

**VAR\_INPUT**

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

**sNetId**: PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

### VAR\_IN\_OUT

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

**Axis**: タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

### VAR\_OUTPUT

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute**: Sercosパラメータの属性を返します。

**fDcBusCurrent**: DCバス電流を返します(例えば、2,040は2,040 Aを意味します)。

### サンプル

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

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

### 3.1.5.6 FB\_SoEReadDcBusVoltage



FB\_SoEReadDcBusVoltageファンクションブロックを使用して、ドライブのDCバス電圧（S-0-0380）を読み取れます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute:** Sercosパラメータの属性を返します。

**fDcBusVoltage:** DCバス電圧を返します（例えば、294.0は294.0 Vを意味します）。

## サンプル

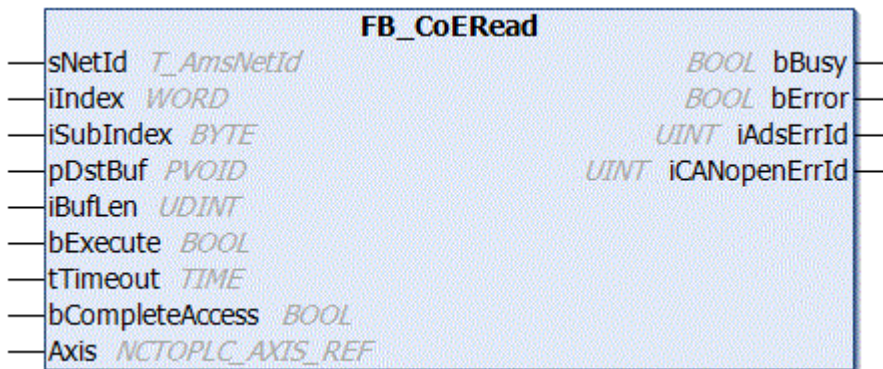
```
fbReadDcBusVoltage : FB_SoEReadDcBusVoltage;
bReadDcBusVoltage : BOOL;
fDcBusVoltage : REAL;
(* NcAxis *)
stNcToPlc AT %I* : NCTOPLC_AXLESTRUCT;

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

## 3.2 汎用CoE

### 3.2.1 現在値を判定するファンクションブロック

#### 3.2.1.1 FB\_CoERead



ファンクションブロックFB\_CoEReadを使用して、SOD (Service Data Object) アクセス経由でEtherCATスレーブのオブジェクト辞書からデータを読み取れます。これを行うには、スレーブにメールボックスが存在している、およびスレーブがCoE (CANopen over EtherCAT) プロトコルをサポートしている必要があります。SubIndexおよびIndexパラメータを使用して、どのオブジェクトを読み出すかを選択します。CompleteAccess := TRUEの場合、サブエレメントを使用してパラメータを読み取れます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)



**iIndex**: 読み取られるオブジェクトのインデックスです。

**iSubIndex**: 読み取られるオブジェクトのサブインデックスです。

**pDstBuf**: 受信バッファのアドレス（ポインタ）です。

**iBufLen**: 読み取るデータの最大許容バッファサイズ（バイト）です。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**bCompleteAccess**: Complete Accessにより、オブジェクト全体に一度にアクセスできます。

## VAR\_OUTPUT

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

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力がADSエラーコードを返します。

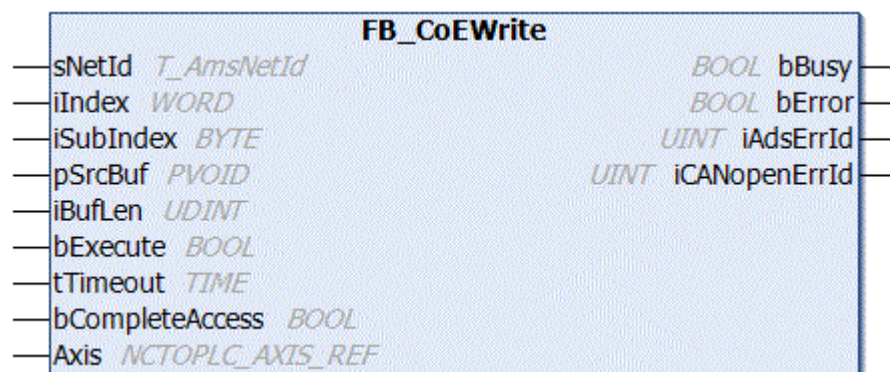
**iCANopenErrId**: 設定されている場合、bError出力がCANopenエラーコードを返します。

## VAR\_IN\_OUT

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

**Axis**: システムで軸を一意にアドレス指定する、タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。この構造体には、位置、速度、およびエラー状態といった軸の現在の状態が含まれます。

### 3.2.1.2 FB\_CoEWrite



ファンクションブロックFB\_CoEWriteを使用して、EtherCATスレーブのオブジェクトディレクトリのオブジェクトをSDO (Service Data Object) ダウンロードによって書き込みます。これを行うには、スレーブにメールボックスが存在している、およびスレーブがCoE (CANopen over EtherCAT) プロトコルをサポートしている必要があります。SubIndexおよびIndexパラメータを使用して、どのオブジェクトを書き込むかを選択します。CompleteAccess := TRUEの場合、サブエレメントを使用してパラメータを書き込みます。

**VAR\_INPUT**

```

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

```

**sNetId**: PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**iIndex**: 書き込まれるオブジェクトのインデックスです。

**iSubIndex**: 書き込まれるオブジェクトのサブインデックスです。

**pDstBuf**: 送信バッファのアドレス (ポインタ) です。

**iBufLen**: 読み取るデータの最大許容バッファサイズ (バイト) です。

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**bCompleteAccess**: Complete Accessにより、オブジェクト全体に一度にアクセスできます。

**VAR\_OUTPUT**

```

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

```

**bBusy**: ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError**: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力がADSエラーコードを返します。

**iCANopenErrId**: 設定されている場合、bError出力がCANopenエラーコードを返します。

**VAR\_IN\_OUT**

```

VAR_IN_OUT
  Axis : NCTOPLC_AXIS_REF;
END_VAR

```

**Axis**: システムで軸を一意にアドレス指定する、タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。この構造体には、位置、速度、およびエラー状態といった軸の現在の状態が含まれます。

## 3.3 AX5000 SoE

### 3.3.1 FB\_SoEAX5000ReadActMainVoltage



FB\_SoEAX5000ReadActMainVoltageファンクションブロックを使用して、AX5000の主電圧の現在のピーク値 (P-0-0200) を読み取れます。

#### VAR\_INPUT

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

**sNetId:** PCのAMSネットワークIDを含む文字列です。(タイプ: T\_AmsNetId)

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout:** ファンクションブロックの最大許容実行時間です。

#### VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

#### VAR\_OUTPUT

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**dwAttribute:** Sercosパラメータの属性を返します。

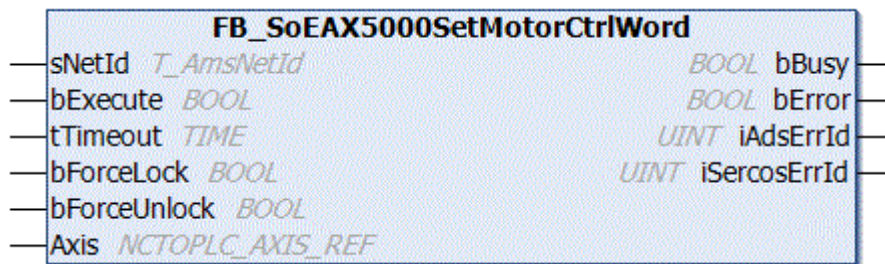
**fActualMainVoltage**: AX5000の現在の主電圧のピーク値を返します（例えば、303.0は303.0 Vを意味します）。

### サンプル

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

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

### 3.3.2 FB\_SoEAX5000SetMotorCtrlWord



ファンクションブロックFB\_SoEAX5000SetMotorCtrlWordを使用して、Motor Control Word (P-0-0096) で ForceLockビット（ビット0）またはForceUnlockビットをセットし、ブレーキを作動または解放できます。通常、ブレーキはドライブのイネーブルによって自動的に制御されます。

ForceLockビットによって、ブレーキをイネーブルとは無関係に作動できます。また、ForceUnlockビットによって、ブレーキをイネーブルとは無関係に解放できます。ForceLockとForceUnlockが同時に設定されている場合は、ForceLock（ブレーキ作動）が優先されます。

#### VAR\_INPUT

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

**sNetId**: PCのAMSネットワークIDを含む文字列です。（タイプ: T\_AmsNetId）

**bExecute**: ファンクションブロックは、この入力の立ち上がりで有効になります。

**tTimeout**: ファンクションブロックの最大許容実行時間です。

**bForceLock**: イネーブルとは無関係にブレーキを作動します。

**bForceUnlock**: イネーブルとは無関係にブレーキを解放します。

**VAR\_IN\_OUT**

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

**VAR\_OUTPUT**

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返します。

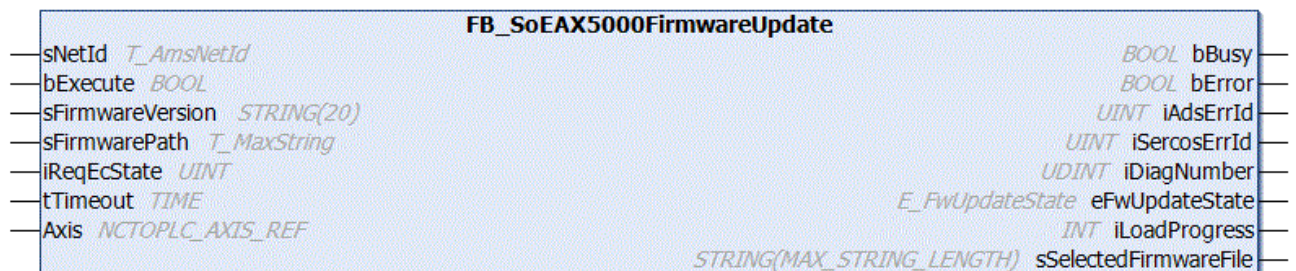
**iSercosErrId:** 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返します。

**サンプル**

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

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

**3.3.3 FB\_SoEAX5000FirmwareUpdate**



FB\_SoEAX5000FirmwareUpdateファンクションブロックを使用して、AX5000のファームウェアをチェックし、指定されたバージョン（リビジョンおよびビルド）、または設定されているリビジョンの現在のビルドを自動的に変更できます。

更新時には:

- ・ 設定されているスレーブタイプ（AX5103-0000-0010など）が判定されます。
- ・ 事前定義されているスレーブアドレス（AX5103-0000-0009など）の現在のスレーブが判定されます。
- ・ 現在のスレーブのファームウェア（v1.05\_b0009など）が判定されます。
- ・ 設定と、検出されたスレーブのチャンネル数、電流、リビジョン、およびファームウェアの比較が行われます。
- ・ 必要なファームウェアファイル名が判定され、このファイルの検索が実行されます。
- ・ 必要に応じて、ファームウェア更新が実行されます。
- ・ 事前定義されているスレーブアドレスの現在のスレーブが再度判定されます。
- ・ スレーブが、事前定義されているEtherCAT状態に切り替えられます。

更新が正常に終了すると、eFwUpdateState = eFwU\_FwUpdateDoneとなります。

更新が不要な場合はeFwUpdateState = eFwU\_NoFwUpdateRequiredとなり、更新不要である信号が出力されません。

ファームウェア更新は、stDriveRefで指定されたチャンネル（A=0またはB=1）を経由して行われます。2チャンネルデバイスの場合は、2つのチャンネルの一方のみを使用できます。他のチャンネルは、eFwUpdateState = eFwU\_UpdateViaOtherChannelActiveまたはeFwUpdateState = eFwU\_UpdateViaOtherChannelの信号を出力します。

ファームウェア更新中（eFwUpdateState = eFwU\_FwUpdateInProgress）は、iLoadProgressが進捗の信号をパーセンテージで出力します。

### 注記

#### 中断による更新の失敗

更新中に中断が発生すると、更新が実行されない、または更新が正しく行われられない可能性があります。その場合、適切なファームウェアが存在せず、ターミナルが使用できなくなるおそれがあります。

更新中は、以下のルールを順守する必要があります。

- ・ PLCおよびTwinCATを停止してはいけません。
- ・ EtherCAT接続を中断してはいけません。
- ・ AX5000の電源をオフしてはいけません。

#### VAR\_INPUT

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

**sNetId:** コントローラ（IPC）のAMS-NetIDです。

**bExecute:** ファンクションブロックは、この入力の立ち上がりで有効になります。

**sFirmwareVersion:** 目的のファームウェアバージョンを「vx.yy\_bnnnn」の形式（例えば、バージョンv1.05ビルド0009の場合は「v1.05\_b0009」）で指定します。

リリースビルド:

- ・ 「v1.05\_b0009」 特定のビルド（この場合はv1.05ビルド0009）

- ・ 「v1.05\_b00??」 指定されたバージョン（この場合はv1.05）の最新ビルド
- ・ 「v1.??\_b00??」 指定されたメインバージョン（この場合はv1）の最新ビルド
- ・ 「v?.??\_b00??」 最新バージョンの最新ビルド
- ・ 「 」 最新バージョンの最新ビルド

カスタムファームウェアビルド:

- ・ 「v1.05\_b1009」 特定のビルド（この場合はv1.05ビルド0009）
- ・ 「v1.05\_b10??」 指定されたバージョン（この場合はv1.05）の最新ビルド
- ・ 「v1.??\_b10??」 指定されたメインバージョン（この場合はv1）の最新ビルド
- ・ 「v?.??\_b10??」 最新バージョンの最新ビルド

...

- ・ 「v1.05\_b8909」 特定のビルド（この場合はv1.05ビルド8909）
- ・ 「v1.05\_b89??」 指定されたバージョン（この場合はv1.05）の最新ビルド
- ・ 「v1.??\_b89??」 指定されたメインバージョン（この場合はv1）の最新ビルド
- ・ 「v?.??\_b89??」 最新バージョンの最新ビルド

デバッグビルド:

- ・ 「v1.05\_b9009」 特定のビルド（この場合はv1.05ビルド9009）
- ・ 「v1.05\_b90??」 指定されたバージョン（この場合はv1.05）の最新ビルド
- ・ 「v1.??\_b90??」 指定されたメインバージョン（この場合はv1）の最新ビルド
- ・ 「v?.??\_b90??」 最新バージョンの最新ビルド

**sFirmwarePath:** ファームウェアファイルが存在するファームウェアプールのパス（例えばC:¥TwinCAT¥Io¥TcDriveManager¥FirmwarePool）を指定します。

**iReqEcState:** 更新後のEtherCATの状態です（更新が実際に実行される場合のみ）。状態は、PLC Lib Tc2\_EtherCATでグローバル定数として定義されます。

**tTimeout:** 大規模なEtherCATネットワークでのファームウェア更新にはより長く時間がかかるため、ここでは個々の内部ADSインスタンスのタイムアウトのみを指定します。

## VAR\_IN\_OUT

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

**Axis:** タイプNCTOPLC\_AXIS\_REFの軸データ構造体です。

## VAR\_OUTPUT

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

**bBusy:** ファンクションブロックが有効な場合、この出力がセットされ、フィードバックを受信するまでセットされたままの状態となります。

**bError:** コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後で、この出力がセットされます。

**iAdsErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのADSエラーコードを返しません。

**iSercosErrId**: 設定されている場合、bError出力が、最後に実行されたコマンドのSercosエラーを返しません。

**iDiagNumber**: 設定されている場合、bError出力が、最後に実行されたファームウェア更新のドライブエラーを返します。

**eFwUpdateState**: ファーストクラスのステータスを返します。

**iLoadProgress**: 実際のファームウェア更新の進捗をパーセンテージで返します。

**sSelectedFirmwareFile**: 検索するファームウェアファイル名を表示します。

## サンプル

```

VAR CONSTANT
  iNumOfDrives : INT := 2;
END_VAR

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

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

```

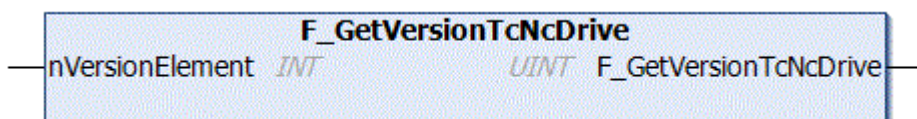


```

        bAnyError := TRUE;
    END_IF
    ELSE
        bAnyBusy := TRUE;
    END_IF
END_FOR
IF NOT bAnyBusy THEN
    bExecute := FALSE;
    IF NOT bAnyError THEN
        iUpdateState := 0; (* OK *)
    ELSE
        iUpdateState := 0; (* Error *)
    END_IF
END_IF
END_CASE

```

### 3.4 F\_GetVersionTcNcDrive



このファンクションを使用して、PLCライブラリのバージョン情報を読み込みできます。

#### ファンクションF\_GetVersionTcNcDrive: UINT

```

VAR_INPUT
    nVersionElement : INT;
END_VAR

```

**nVersionElement**: 読み込みするバージョン要素です。使用可能なパラメータ:

- ・ 1 : メジャー番号
- ・ 2 : マイナー番号
- ・ 3 : リビジョン番号