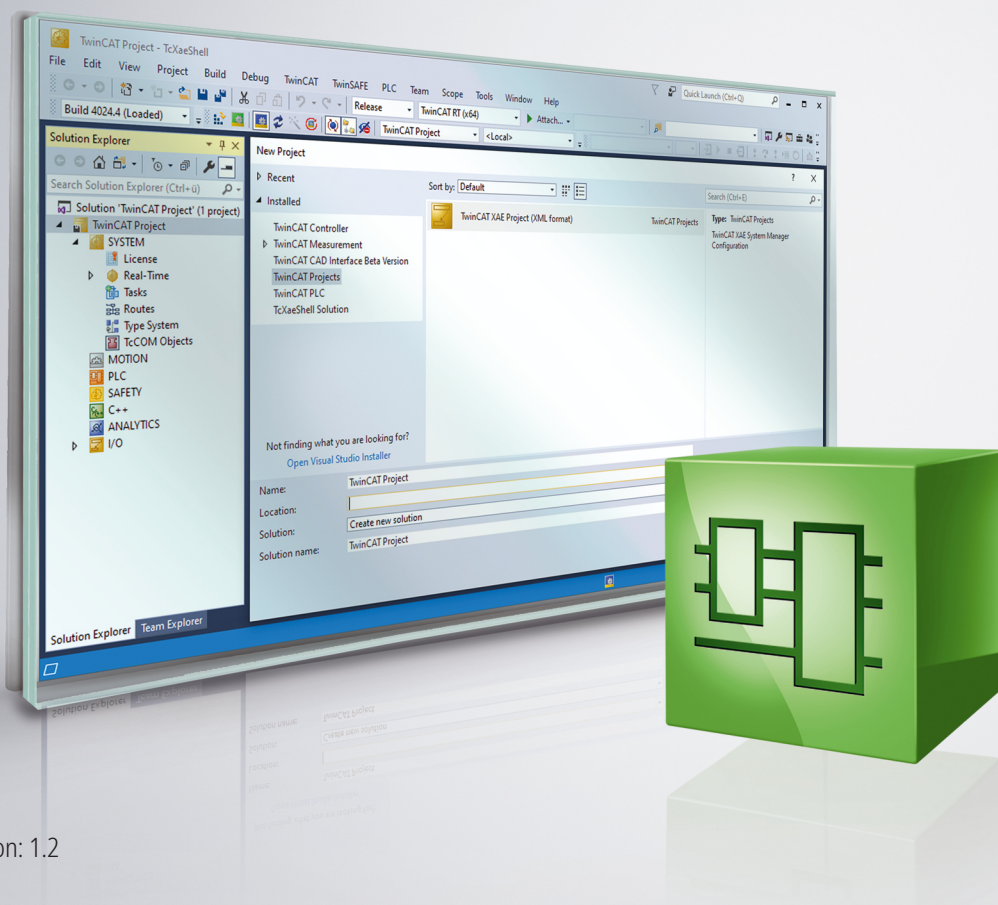


Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2_DataExchange



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
2	Übersicht	7
3	Ereignisgesteuerte Übertragungsbausteine	8
3.1	FB_ReadAdsSymByName	8
3.2	FB_WriteAdsSymByName.....	9
3.3	FB_WriteBoolOnDelta	10
3.4	FB_WriteByteOnDelta	12
3.5	FB_WriteWordOnDelta	13
3.6	FB_WriteDWordOnDelta	14
3.7	FB_WriteRealOnDelta	16
3.8	FB_WriteLRealOnDelta	17
4	Überwachungsbausteine	19
4.1	FB_CheckWatchdog.....	19
4.2	FB_WriteWatchdog	20
5	Datentypen	22
5.1	E_AdsComMode	22
6	Globale Konstanten	23
6.1	Bibliotheksversion.....	23

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Symbole

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

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

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

Tipp oder Fingerzeig



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

2 Übersicht

Die vorliegenden Funktionsbausteine vereinfachen den ereignisgesteuerten Datenaustausch zwischen TwinCAT-SPS-Laufzeitsystemen und/oder anderen ADS-Geräten (TwinCAT NC, Busklemmen Controller, ...).

Der Funktionsbaustein FB_WriteXXXOnDelta() führt einen Schreibvorgang aus, wenn das Eingangssignal einen bestimmten Grenzwert über-/ bzw. unterschritten hat. Wie oft das Eingangssignal überprüft werden soll, kann eingestellt werden. Durch das ereignisgesteuerte Schreiben der Daten wird die Belastung der Feldbusses minimiert. Tritt bei der Übertragung ein Fehler auf, so wird der Vorgang so lange wiederholt, bis die Verbindung wieder vorhanden ist. Als Quell- und Zielvariablen sind alle in der TwinCAT PLC unterstützten Datentypen zugelassen. Symbolnamen werden ebenfalls unterstützt.

Zur Überwachung einzelner Kommunikationspartner stehen Watchdog-Bausteine zur Verfügung. Der zu überwachende Teilnehmer versendet zyklisch einen inkrementierenden Zähler. Bei dem Empfänger wird überprüft, ob sich der Zählerstand innerhalb einer bestimmten Zeit verändert.

Schreib-/Lesebausteine

Name	Beschreibung
FB_ReadAdsSymByName [▶ 8]	Liest eine Variable eines beliebigen Datentyps per Variablenname
FB_WriteAdsSymByName [▶ 9]	Schreibt eine Variable eines beliebigen Datentyps per Variablenname
FB_WriteBoolOnDelta [▶ 10]	Schreibt ereignisgesteuert eine Variable vom Typ BOOLEAN
FB_WriteByteOnDelta [▶ 12]	Schreibt ereignisgesteuert eine Variable vom Typ BYTE
FB_WriteWordOnDelta [▶ 13]	Schreibt ereignisgesteuert eine Variable vom Typ WORD
FB_WriteDWordOnDelta [▶ 14]	Schreibt ereignisgesteuert eine Variable vom Typ DWORD
FB_WriteRealOnDelta [▶ 16]	Schreibt ereignisgesteuert eine Variable vom Typ REAL
FB_WriteLRealOnDelta [▶ 17]	Schreibt ereignisgesteuert eine Variable vom Typ LREAL

Überwachungsbausteine

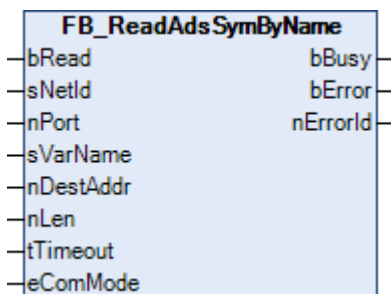
Name	Beschreibung
FB_WriteWatchdog [▶ 20]	Schreibt zyklisch ein Watchdogsignal (inkrementierender Zähler)
FB_CheckWatchdog [▶ 19]	Überwacht das empfangende Watchdogsignal

3 Ereignisgesteuerte Übertragungsbausteine

Schreib-/Lesebausteine

Name	Beschreibung
FB_ReadAdsSymByName [▶ 8]	Liest eine Variable eines beliebigen Datentyps per Variablenname
FB_WriteAdsSymByName [▶ 9]	Schreibt eine Variable eines beliebigen Datentyps per Variablenname
FB_WriteBoolOnDelta [▶ 10]	Schreibt ereignisgesteuert eine Variable vom Typ BOOLEAN
FB_WriteByteOnDelta [▶ 12]	Schreibt ereignisgesteuert eine Variable vom Typ BYTE
FB_WriteWordOnDelta [▶ 13]	Schreibt ereignisgesteuert eine Variable vom Typ WORD
FB_WriteDWordOnDelta [▶ 14]	Schreibt ereignisgesteuert eine Variable vom Typ DWORD
FB_WriteRealOnDelta [▶ 16]	Schreibt ereignisgesteuert eine Variable vom Typ REAL
FB_WriteLRealOnDelta [▶ 17]	Schreibt ereignisgesteuert eine Variable vom Typ LREAL

3.1 FB_ReadAdsSymByName



Der Funktionsbaustein ermöglicht das Lesen eines beliebigen Wertes von einer anderen Steuerung mithilfe des Symbolnamens.

Bei einer positiven Flanke am Eingang *bRead* liest der Baustein den Wert der Variablen *sVarName* von dem ausgewählten ADS-Gerät (z. B. SPS). Das ADS-Gerät wird durch die AMS-NetId (*sNetId*) und die AMS-Portnummer (*nPort*) angegeben. Der Wert wird in die Variable geschrieben, auf die *nDestAddr* zeigt.

Mithilfe des Eingangs *eComMode* kann die interne Arbeitsweise des Bausteins verändert werden:

- *eComMode := eAdsComModeSecureCom*: Nach jedem Lesevorgang wird der Handle der SPS-Variablen wieder freigegeben. Dieser Modus sollte verwendet werden, wenn sehr langsam Werte ausgetauscht werden.
- *eComMode := eAdsComModeFastCom*: Solange sich die Eingänge *sVarName*, *sNetId* und *nPort* nicht verändern, wird der Handle der SPS-Variablen nicht nach jedem Lesevorgang freigegeben. Dieser Modus sollte verwendet werden, wenn sehr häufig Werte ausgetauscht werden.

VAR_INPUT

```
VAR_INPUT
  bRead          : BOOL;
  sNetId         : T_AmsNetId;
  nPort          : T_AmsPort := 851;
  sVarName       : STRING(255);
  nDestAddr      : PVOID;
  nLen           : UDINT;
  tTimeout       : TIME := DEFAULT_ADS_TIMEOUT;
  eComMode       : E_AdsComMode := eAdsComModeSecureCom;
END_VAR
```

bRead: Der Baustein liest den Inhalt der Variablen *sVarName* des ausgewählten ADS-Gerätes und schreibt diesen in die Variable, auf die der Zeiger *nDestAddr* zeigt.

sNetId: AMS-NetId des ADS-Gerätes, von dem der Wert gelesen werden soll. (Typ *T_AmsNetId*)

nPort: AMS-Portnummer des ADS-Gerätes, von dem der Wert gelesen werden soll. (Typ T_AmsPort)

sVarName: Symbolname der zu lesenden Variablen auf dem ausgewählten ADS-Gerät (max. 255 Zeichen).

nDestAddr: Adresse der Variablen, in der der gelesene Wert geschrieben wird.

nLen: Länge der zu lesenden Variable in Byte.

tTimeout: Zeit bis zum Abbruch der Abarbeitung.

eComMode: Enum, mit dem bestimmt wird, ob der Handle der SPS-Variablen nach jedem Lesen wieder freigegeben wird.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: Die Übertragung ist aktiv.

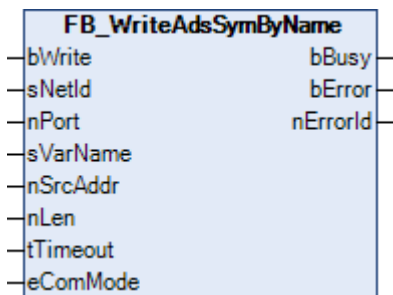
bError: Bei der Übertragung ist ein Fehler aufgetreten.

nErrorId: ADS-Fehlernummer, falls ein Fehler aufgetreten ist.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.1.0	Tc2_DataExchange

3.2 FB_WriteAdsSymByName



Der Funktionsbaustein ermöglicht das Schreiben eines beliebigen Wertes auf eine andere Steuerung mithilfe des Symbolnamens.

Bei einer positiven Flanke am Eingang *bWrite* schreibt der Baustein den Wert, auf den der Zeiger *nSrcAddr* zeigt, in die Variable *sVarName* des ausgewählten ADS-Gerätes (z. B. SPS). Das ADS-Gerät wird durch die AMS-NetId (*sNetId*) und die AMS-Portnummer (*nPort*) angegeben.

Mithilfe des Eingangs *eComMode* kann die interne Arbeitsweise des Bausteins verändert werden:

- *eComMode := eAdsComModeSecureCom:* Nach jedem Schreibvorgang wird der Handle der SPS-Variablen wieder freigegeben. Dieser Modus sollte verwendet werden, wenn sehr langsam Werte ausgetauscht werden.
- *eComMode := eAdsComModeFastCom:* Solange sich die Eingänge *sVarName*, *sNetID* und *nPort* nicht ändern, wird der Handle der SPS-Variablen nicht nach jedem Schreibvorgang freigegeben. Dieser Modus sollte verwendet werden, wenn sehr häufig Werte ausgetauscht werden.

VAR_INPUT

```
VAR_INPUT
  bWrite      : BOOL;
  sNetId     : T_AmsNetId;
  nPort      : T_AmsPort := 851;
```

```

sVarName      : STRING(255);
nSrcAddr      : PVOID;
nLen          : UDINT;
tTimeout      : TIME := DEFAULT_ADS_TIMEOUT;
eComMode      : E_AdsComMode := eAdsComModeSecureCom;
END_VAR

```

bWrite: Der Baustein schreibt den Inhalt der Variablen, auf die der Zeiger *nSrcAddr* zeigt, in die Variable *sVarName* des ausgewählten ADS-Gerätes.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ *T_AmsNetId*)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ *T_AmsPort*)

sVarName: Symbolname der zu beschreibenden Variablen auf dem ausgewählten ADS-Gerät (max. 255 Zeichen).

nSrcAddr: Adresse der Variablen, in der der zu schreibende Wert steht.

nLen: Länge der zu schreibenden Variablen in Byte.

tTimeout: Zeit bis zum Abbruch der Abarbeitung.

eComMode: Enum, mit dem bestimmt wird, ob der Handle der SPS-Variablen nach jedem Schreiben wieder freigegeben wird.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR

```

bBusy: Die Übertragung ist aktiv.

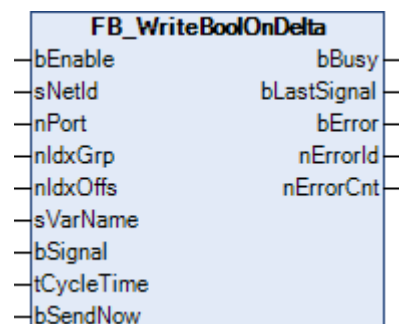
bError: Bei der Übertragung ist ein Fehler aufgetreten.

nErrorId: ADS-Fehlernummer, falls ein Fehler aufgetreten ist.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.1.0	Tc2_DataExchange

3.3 FB_WriteBoolOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ BOOLEAN.

Der Funktionsbaustein *FB_WriteBoolOnDelta* prüft zyklisch, ob sich der Wert am Eingang *bSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird eine Änderung erkannt, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-

NetId und durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signals.

VAR_INPUT

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  bSignal      : BOOL;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

bSignal: Variable, deren Wert übertragen werden soll.

tCycleTime: Zykluszeit, in der überprüft wird, ob sich das Eingangssignal verändert hat.

bSendNow: Durch eine positive Flanke wird der Wert unmittelbar übertragen.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy        : BOOL := FALSE;
  bLastSignal  : BOOL;
  bError       : BOOL := FALSE;
  nErrorId     : UDINT := 0;
  nErrorCnt    : UDINT := 0;
END_VAR
```

bBusy: Die Übertragung ist aktiv.

bLastSignal: Zuletzt übertragender Wert.

bError: Bei der Übertragung ist ein Fehler aufgetreten.

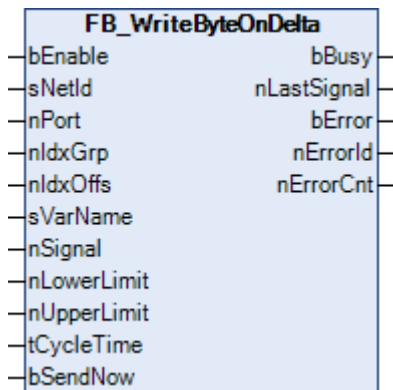
nErrorId: Fehlernummer, falls ein Fehler aufgetreten ist.

nErrorCnt: Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

3.4 FB_WriteByteOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ BYTE.

Der Funktionsbaustein FB_WriteByteOnDelta prüft zyklisch, ob sich der Wert am Eingang *nSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird bei dem Vergleich festgestellt, dass der aktuelle Wert um den Wert *nUpperLimit* größer oder um den Wert *nLowerLimit* kleiner ist, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-NetId und durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signals.

VAR_INPUT

```

VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  nSignal      : BYTE;
  nLowerLimit  : BYTE;
  nUpperLimit  : BYTE;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR

```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

nSignal: Variable, dessen Wert übertragen werden soll.

nLowerLimit: Unterer Grenzwert.

nUpperLimit: Oberer Grenzwert.

tCycleTime: Zykluszeit, in der überprüft wird, ob das Eingangssignal die Grenzwerte überschritten hat.

bSendNow: Durch eine positive Flanke wird der Wert unmittelbar übertragen.

VAR_OUTPUT

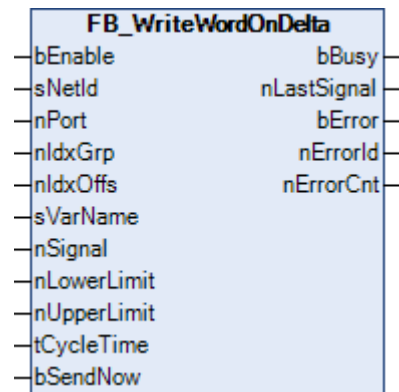
```
VAR_OUTPUT
  bBusy          : BOOL := FALSE;
  nLastSignal    : BYTE;
  bError        : BOOL := FALSE;
  nErrorId      : UDINT := 0;
  nErrorCnt     : UDINT := 0;
END_VAR
```

- bBusy:** Die Übertragung ist aktiv.
- nLastSignal:** Zuletzt übertragender Wert.
- bError:** Bei der Übertragung ist ein Fehler aufgetreten.
- nErrorId:** Fehlernummer, falls ein Fehler aufgetreten ist.
- nErrorCnt:** Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

3.5 FB_WriteWordOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ WORD.

Der Funktionsbaustein FB_WriteWordOnDelta prüft zyklisch, ob sich der Wert am Eingang *nSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird bei dem Vergleich festgestellt, dass der aktuelle Wert um den Wert *nUpperLimit* größer oder um den Wert *nLowerLimit* kleiner ist, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-NetId und durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signal.

VAR_INPUT

```
VAR_INPUT
  bEnable          : BOOL := FALSE;
  sNetId           : T_AmsNetId;
  nPort            : T_AmsPort;
  nIdxGrp          : UDINT;
  nIdxOffs         : UDINT;
  sVarName         : STRING;
  nSignal          : WORD;
  nLowerLimit      : WORD;
  nUpperLimit      : WORD;
```

```

tCycleTime      : TIME := t#0s;
bSendNow        : BOOL;
END_VAR

```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

nSignal: Variable, dessen Wert übertragen werden soll.

nLowerLimit: Unterer Grenzwert.

nUpperLimit: Oberer Grenzwert.

tCycleTime: Zykluszeit, in der überprüft wird, ob das Eingangssignal die Grenzwerte überschritten hat.

bSendNow: Durch eine positive Flanke wird der Wert unmittelbar übertragen.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL := FALSE;
  nLastSignal : WORD;
  bError     : BOOL := FALSE;
  nErrorId   : UDINT := 0;
  nErrorCnt  : UDINT := 0;
END_VAR

```

bBusy: Die Übertragung ist aktiv.

nLastSignal: Zuletzt übertragender Wert.

bError: Bei der Übertragung ist ein Fehler aufgetreten.

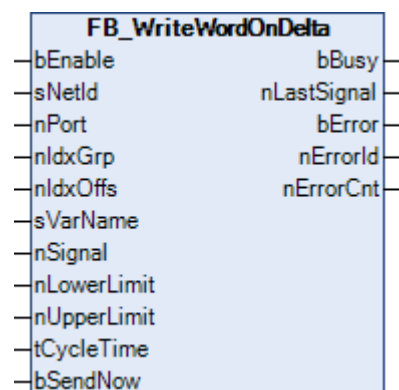
nErrorId: Fehlernummer, falls ein Fehler aufgetreten ist.

nErrorCnt: Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

3.6 FB_WriteDWordOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ DWORD.

Der Funktionsbaustein FB_WriteDWordOnDelta überprüft zyklisch, ob sich der Wert am Eingang *nSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird bei dem Vergleich festgestellt, dass der aktuelle Wert um den Wert *nUpperLimit* größer oder um den Wert *nLowerLimit* kleiner ist, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-NetId und durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signals.

VAR_INPUT

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  nSignal      : BYTE;
  nLowerLimit  : BYTE;
  nUpperLimit  : BYTE;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

nSignal: Variable, dessen Wert übertragen werden soll.

nLowerLimit: Unterer Grenzwert.

nUpperLimit: Oberer Grenzwert.

tCycleTime: Zykluszeit, in der überprüft wird, ob das Eingangssignal die Grenzwerte überschritten hat.

bSendNow: Durch eine positive Flanke wird der Wert unmittelbar übertragen.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy        : BOOL := FALSE;
  nLastSignal  : DWORD;
  bError       : BOOL := FALSE;
  nErrorId     : UDINT := 0;
  nErrorCnt    : UDINT := 0;
END_VAR
```

bBusy: Die Übertragung ist aktiv.

nLastSignal: Zuletzt übertragender Wert.

bError: Bei der Übertragung ist ein Fehler aufgetreten.

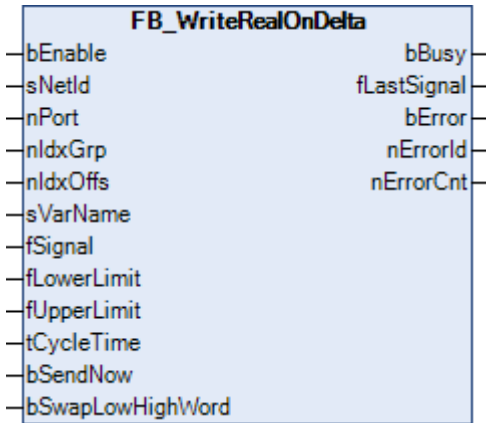
nErrorId: Fehlernummer, falls ein Fehler aufgetreten ist.

nErrorCnt: Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

3.7 FB_WriteRealOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ REAL.

Der Funktionsbaustein FB_WriteRealOnDelta prüft zyklisch, ob sich der Wert am Eingang *fSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird bei dem Vergleich festgestellt, dass der aktuelle Wert um den Wert *fUpperLimit* größer oder um den Wert *fLowerLimit* kleiner ist, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-NetId und durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Je nach verwendeter Hardware ist die interne Darstellung von Fließkommazahlen unterschiedlich. Während Intel das „little endian“-Format benutzt, wird bei Motorola das „big-endian“-Format angewendet. Um einen Austausch von Fließkommazahlen zu ermöglichen, kann durch die Eingangsvariable *bSwapLowHighWord* eine Anpassung durchgeführt werden. Dieses ist z. B. notwendig, wenn zwischen der TwinCAT SPS auf einem PC und einem BC9000 Fließkommazahlen ausgetauscht werden sollen.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signals.

VAR_INPUT

```

VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  fSignal      : REAL;
  fLowerLimit  : REAL;
  fUpperLimit  : REAL;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
  bSwapLowHighWord : BOOL := FALSE;
END_VAR

```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

- nIdxOffs:** Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.
- sVarName:** Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.
- fSignal:** Variable, dessen Wert übertragen werden soll.
- fLowerLimit:** Unterer Grenzwert.
- fUpperLimit:** Oberer Grenzwert.
- tCycleTime:** Zykluszeit, in der überprüft wird, ob das Eingangssignal die Grenzwerte überschritten hat.
- bSendNow:** Durch eine positive Flanke wird der Wert unmittelbar übertragen.
- bSwapLowHighWord:** Das niederwertige und das höherwertige WORD werden gedreht.

VAR_OUTPUT

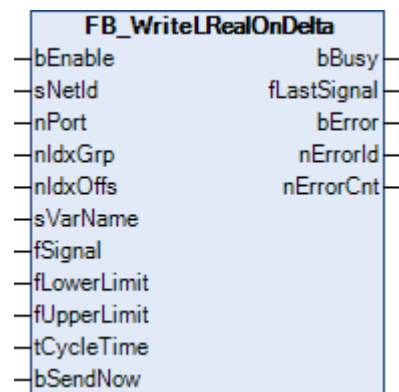
```
VAR_OUTPUT
  bBusy          : BOOL := FALSE;
  fLastSignal    : REAL;
  bError         : BOOL := FALSE;
  nErrorId       : UDINT := 0;
  nErrorCnt      : UDINT := 0;
END_VAR
```

- bBusy:** Die Übertragung ist aktiv.
- fLastSignal:** Zuletzt übertragender Wert.
- bError:** Bei der Übertragung ist ein Fehler aufgetreten.
- nErrorId:** Fehlernummer, falls ein Fehler aufgetreten ist.
- nErrorCnt:** Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

3.8 FB_WriteLRealOnDelta



Der Funktionsbaustein ermöglicht das ereignisgesteuerte Schreiben einer Variablen vom Typ LREAL.

Der Funktionsbaustein **FB_WriteLRealOnDelta** prüft zyklisch, ob sich der Wert am Eingang *fSignal* geändert hat. Die Zykluszeit für die Prüfung wird durch den Parameter *tCycleTime* bestimmt. Wird für *tCycleTime* 0 s angegeben, so wird das Eingangssignal in jedem SPS-Zyklus überprüft. Wird bei dem Vergleich festgestellt, dass der aktuelle Wert um den Wert *fUpperLimit* größer oder um den Wert *fLowerLimit* kleiner ist, so wird der Wert des Signals an das angegebene ADS-Gerät geschickt. Der Empfänger wird durch die AMS-NetId und

durch die Portnummer adressiert (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Signals.

VAR_INPUT

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  fSignal      : LREAL;
  fLowerLimit  : LREAL;
  fUpperLimit  : LREAL;
  tCycleTime   : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das der Wert übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das der Wert übertragen werden soll. (T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in der der Wert übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in dem der Wert übertragen werden soll.

fSignal: Variable, dessen Wert übertragen werden soll.

fLowerLimit: Unterer Grenzwert.

fUpperLimit: Oberer Grenzwert.

tCycleTime: Zykluszeit, in der überprüft wird, ob das Eingangssignal die Grenzwerte überschritten hat.

bSendNow: Durch eine positive Flanke wird der Wert unmittelbar übertragen.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy        : BOOL := FALSE;
  fLastSignal  : LREAL;
  bError       : BOOL := FALSE;
  nErrorId     : UDINT := 0;
  nErrorCnt    : UDINT := 0;
END_VAR
```

bBusy: Die Übertragung ist aktiv.

fLastSignal: Zuletzt übertragender Wert.

bError: Bei der Übertragung ist ein Fehler aufgetreten.

nErrorId: Fehlernummer, falls ein Fehler aufgetreten ist.

nErrorCnt: Anzahl der fehlerhaften Übertragungsversuche.

Voraussetzungen

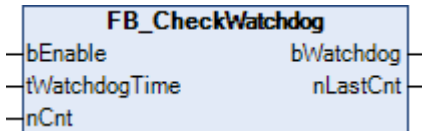
Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

4 Überwachungsbausteine

Überwachungsbausteine

Name	Beschreibung
FB_WriteWatchdog [▶ 20]	Schreibt zyklisch ein Watchdogsignal (inkrementierender Zähler)
FB_CheckWatchdog [▶ 19]	Überwacht das empfangende Watchdogsignal

4.1 FB_CheckWatchdog



Überwachen eines Watchdog-Signals, das mit dem Baustein [FB_WriteWatchdog \[▶ 20\]](#) übertragen wird.

Das zu überwachende Gerät verschickt regelmäßig einen sich verändernden Zählerstand an das Gerät, das die Übertragung kontrollieren soll. Dort wird mithilfe des Funktionsbausteins FB_CheckWatchdog der Zählerstand überwacht. Verändert sich dieser innerhalb einer bestimmten Zeit nicht, so wird der Ausgang *bWatchdog* auf TRUE gesetzt. Wird für *tWatchdogTime* ein Wert von 0 s angegeben, so wird das Signal *bWatchdog* auf FALSE gesetzt. Die Zeit *tWatchdogTime* sollte ein Vielfaches (5-10faches) der Zeit sein, in der das Überwachungssignal zugeschickt wird.

VAR_INPUT

```

VAR_INPUT
    bEnable      : BOOL := FALSE;
    tWatchdogTime : TIME := t#0s;
    nCnt         : UDINT;
END_VAR
  
```

bEnable: Baustein freischalten.

tWatchdogTime: Zeitdauer, in der sich *nCnt* verändern muss.

nCnt: Aktueller Zählerstand des Watchdog-Signals.

VAR_OUTPUT

```

VAR_OUTPUT
    bWatchdog : BOOL := FALSE;
    nLastCnt  : UDINT;
END_VAR
  
```

bWatchdog: FALSE signalisiert ein gültiges Überwachungssignal. Wird für die Zeitdauer *tWatchdogTime* keine Veränderung von *nCnt* erkannt, so geht der Ausgang auf TRUE.

nLastCnt: Letzter, erfolgreich übertragender Zählerstand des Überwachungssignals.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

4.2 FB_WriteWatchdog



Schreiben eines Watchdog-Signals in ein anderes ADS-Gerät (TwinCAT PLC, Busklemmen Controller, ...).

Der Funktionsbaustein FB_WriteWatchdog schreibt zyklisch einen 32-Bit-Zähler in ein anderes ADS-Gerät. Bei jeder erfolgreichen Übertragung wird der Zähler inkrementiert. Beim Empfänger kann mithilfe des Bausteins FB_CheckWatchdog dieses Signal ausgewertet werden. Adressiert wird der Empfänger durch die AMS-NetId und durch die Portnummer (siehe auch Identifikation ADS-Gerät). Die Position innerhalb des Empfängers wird durch die Index-Group/Index-Offset oder den Symbolnamen angegeben. Üblicherweise ist dieses das Eingangsabbild oder der Merkerbereich.

Damit der Zählerstand nicht zu häufig übertragen wird, sollte die Zeit *tWatchdogTime* nicht kleiner als 1 s sein. Wird für *tWatchdogTime* 0 s angegeben, so wird die Übertragung des Signals gesperrt. Beachten sie auch die Beschreibung des Funktionsbausteins [FB_CheckWatchdog\(\)](#) [► 19].

Wird der Eingang *bEnable* auf FALSE gesetzt, so erfolgt keine weitere Übertragung des Watchdog-Signals.

VAR_INPUT

```
VAR_INPUT
  bEnable      : BOOL := FALSE;
  sNetId       : T_AmsNetId;
  nPort        : T_AmsPort;
  nIdxGrp      : UDINT;
  nIdxOffs     : UDINT;
  sVarName     : STRING;
  tWatchdogTime : TIME := t#0s;
  bSendNow     : BOOL;
END_VAR
```

bEnable: Baustein freischalten.

sNetId: AMS-NetId des ADS-Gerätes, in das das Watchdog-Signal übertragen werden soll. (Typ T_AmsNetId)

nPort: AMS-Portnummer des ADS-Gerätes, in das das Watchdog-Signal übertragen werden soll. (Typ T_AmsPort)

nIdxGrp: Index-Group innerhalb des ADS-Gerätes, in das das Watchdog-Signal übertragen werden soll.

nIdxOffs: Index-Offset innerhalb des ADS-Gerätes, in das das Watchdog-Signal übertragen werden soll.

sVarName: Symbolname innerhalb des ADS-Gerätes, in das das Watchdog-Signal übertragen werden soll.

tWatchdogTime: Zykluszeit, in der das Watchdog-Signal übertragen wird.

bSendNow: Durch eine positive Flanke wird das Watchdog-Signal unmittelbar übertragen.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL := FALSE;
  nLastCnt    : UDINT := 0;
  bError      : BOOL := FALSE;
  nErrorId    : UDINT := 0;
END_VAR
```

bBusy: Übertragung ist aktiv.

nLastCnt: Zuletzt übertragender Zählerstand.

bError: Bei der Übertragung ist ein Fehler aufgetreten.

nErrorId: Fehlernummer, falls ein Fehler aufgetreten ist.

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.0.0	Tc2_DataExchange

5 Datentypen

Name	Beschreibung
E_ADSComMode [► 22]	Einstellung der Kommunikationsart sicher oder schnell.

5.1 E_AdsComMode

```
TYPE E_AdsComMode :  
(  
  eAdsComModeSecureCom := 0,  
  eAdsComModeFastCom   := 1  
);  
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	erforderliche TC3 SPS-Bibliothek
TwinCAT v3.1.0	Tc2_DataExchange

6 Globale Konstanten

6.1 Bibliotheksversion

Alle Bibliotheken haben eine bestimmte Version. Diese Version ist u. a. im SPS-Bibliotheks-Repository zu sehen. Eine globale Konstante enthält die Information über die Bibliotheksversion:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_DataExchange : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_DataExchange: Versionsinformation der Tc2_DataExchange (Typ: ST_LibVersion)

Um zu sehen, ob die Version, die Sie haben auch die Version ist, die Sie brauchen, benutzen Sie die Funktion F_CmpLibVersion (definiert in der Tc2_System-Bibliothek).



Alle anderen Möglichkeiten Bibliotheksversionen zu vergleichen, die Sie von TwinCAT 2 kennen, sind veraltet.

Mehr Informationen:
www.beckhoff.de/te1000

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

