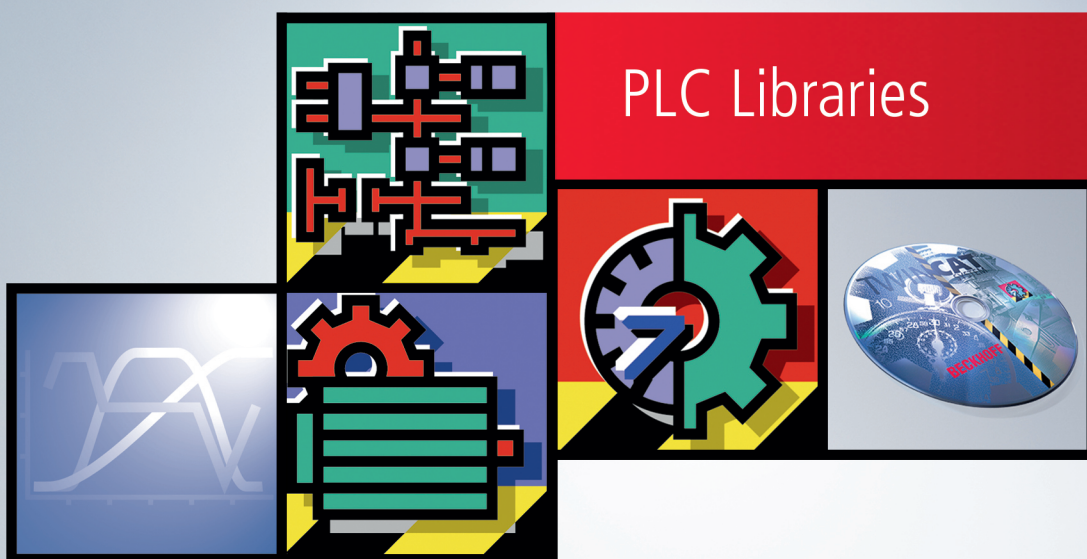


Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: TcPlcCoupler



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Funktionsbausteine	9
3.1	ReadWriteTerminalReg	9
3.2	CouplerReset	11
3.3	FB_ReadCouplerDiag	13
3.4	FB_ReadCouplerRegs	14
3.5	FB_WriteCouplerRegs	15
4	Funktionen	17
4.1	F_GetVersionTcPlcCoupler	17
5	Datenstrukturen	18
5.1	PLCINTFSTRUCT	18
5.2	E_CouplerErrType	18
5.3	ST_CouplerDiag	18
5.4	ST_CouplerReg	19
5.5	ST_CouplerTable	19
5.6	ST_FlashCode	20
6	Anhang	21
6.1	Fehlercodes: 2 Byte PLC Interface	21

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Erklärung der Symbole

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

GEFAHR

Akute Verletzungsgefahr!

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

WARNUNG

Verletzungsgefahr!

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

VORSICHT

Schädigung von Personen!

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

HINWEIS

Schädigung von Umwelt oder Geräten

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

Tipp oder Fingerzeig



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

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 beinhaltet Funktionsbausteine für einen komfortablen Zugriff auf Register der Klemmen über das Kontroll/Status-Byte der Klemme (Registerkommunikation) und für die Kommunikation mit den Beckhoff-Kopplern über das 2 Byte SPS-Interface. Die Funktionsbausteine können z.B. für die Parametrierung der Klemmen über den Feldbus benutzt werden.

Nur die intelligenten Klemmen besitzen eine Registerstruktur. Zu den intelligenten Klemmen zählen z.B. alle analogen Ein- und Ausgangsklemmen. Das Status/Kontroll-Byte der Klemme ist in dem Prozessabbild nur dann sichtbar, wenn die Klemme als komplexe Klemme gemappt wurde. Jedes Klemmenkanal besitzt eine eigene Registerstruktur mit maximal 64 Registern. Bei einem kompakten Mapping, sind die Kontroll/Status-Bytes im Prozessabbild nicht sichtbar.

Für den Registerzugriff über das 2 Byte SPS-Interface müssen die Status- und Kontrollwortvariablen des SPS-Interfaces ebenfalls in das Prozessabbild gemappt werden. Bei einigen Feldbussen kann dies im TwinCAT-Systemmanager für den jeweiligen Koppler konfiguriert werden (Lightbus, Profibus), bei anderen (z.B. Interbus S) wird dafür spezielle Konfigurationssoftware benötigt (z.B. KS2000). Die Status- und Kontrollvariablen werden mit den entsprechenden Ein- und Ausgangsvariablen der Funktionsbausteine verknüpft.

Um eventuelle Änderungen in den Registern permanent zu speichern, muss die Spannungsversorgung des Kopplers unterbrochen werden.

HINWEIS

EEPROM-Schädigung beim zyklischen Schreiben

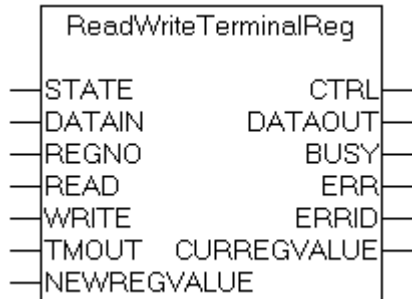
Die Funktionsbausteine der Bibliothek führen beim Aufruf Schreib-Lese-Zugriffe auf die Register der Klemmen oder Koppler. Wenn diese zyklisch aufgerufen werden, kann der EEPROM Speicher zerstört werden. Die Funktionsbausteine sind im Wesentlichen dafür entwickelt worden, um aus dem SPS-Programm relativ einfach eine Konfiguration der Klemmen/Koppler oder eine Diagnose im Fehlerfall durchführen zu können.

Inhalt der Bibliothek

Name	Beschreibung
ReadWriteTerminalReg [► 9]	Auf Register der Klemme über das Kontroll/Status-Byte der Klemme zugreifen (Registerkommunikation)
CouplerReset [► 11]	Reset des Kopplers über das 2 Byte SPS-Interface
FB_ReadCouplerDiag [► 13]	Den Blinkcode des Kopplers über das 2 Byte SPS-Interface auslesen
FB_ReadCouplerRegs [► 14]	Tabellenregister des Kopplers lesen
FB_WriteCouplerRegs [► 15]	Tabellenregister des Kopplers beschreiben
F_GetVersionTcPlcCoupler [► 17]	Versionsinformationen der Bibliothek lesen

3 Funktionsbausteine

3.1 ReadWriteTerminalReg



Der Funktionsbaustein "ReadWriteTerminalReg" ermöglicht einen komfortableren Zugriff auf die Register der Klemme über das Status-/Controlbyte des Klemmenkanals (Registerkommunikation). In der Standardbetriebsart werden die Daten-Ein/Ausgänge der intelligenten Klemmen (z.B. einer analogen Ausgangsklemme) für den Austausch der analogen Ausgangsdaten benutzt. Ein Handshake über das Status-/Controlbyte ermöglicht einen Registerzugriff, dabei werden die Daten-Ein/Ausgangsvariablen zum Übertragen der Registerwerte benutzt. Über eine positive Flanke an dem READ- oder WRITE-Eingang wird das Register mit der Nummer REGNO gelesen bzw. in dem Register geschrieben. Der Registerschreibschutz wird von dem Funktionsbaustein bei einem Schreibzugriff aufgehoben und anschließend neu gesetzt. Bei einem Schreibzugriff auf einem Register wird der neue Registerwert gelesen und steht an dem Ausgang CURREGVALUE zur Verfügung. Um Änderungen der Registerwerte permanent zu speichern, muss die Spannungsversorgung des Kopplers unterbrochen werden. Die Variablen STATE, DATAIN, CTRL, DATAOUT müssen mit den entsprechenden IO-Variablen des Klemmenkanals im TwinCAT-System Manager verknüpft werden.

VAR_INPUT

```
VAR_INPUT
  STATE      : BYTE;
  DATAIN    : WORD;
  REGNO      : BYTE;
  READ       : BOOL;
  WRITE      : BOOL;
  TMOUT      : TIME;
  NEWREGVALUE : WORD;
END_VAR
```

STATE: Statusbyte des Klemmenkanals.

DATAIN: Dateneingangswort des Klemmenkanals.

REGNO: Nummer des Registers auf den ein Schreib- bzw. Lesezugriff erfolgen soll.

READ: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert und der aktuelle Registerwert gelesen. Bei Erfolg steht der Registerwert in der Ausgangsvariablen CURREGVALUE zur Verfügung.

WRITE: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert und in den Register REGNO der Wert der Eingangsvariablen NEWREGVALUE geschrieben. Anschließend wird der aktuelle Wert des Registers gelesen und steht bei Erfolg in der Ausgangsvariablen CURREGVALUE zur Verfügung.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung der Funktion nicht überschritten werden darf.

NEWREGVALUE: Datenwort, der bei einem Schreibzugriff in dem Register mit der Nummer REGNO geschrieben werden soll.

VAR_OUTPUT

```
VAR_OUTPUT
  CTRL      : BYTE;
  DATAOUT  : WORD;
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  CURREGVALUE : WORD;
END_VAR
```

CONTROL: Kontrollbyte des Klemmenkanals.

DATAOUT: Datenausgangswort des Klemmenkanals.

BUSY: Bei der Aktivierung des Bausteines wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung der Funktion abgeschlossen wurde.

ERR: Sollte ein Fehler bei der Ausführung der Funktion auftreten, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die Fehlernummer.

CURREGVALUE: Bei einem erfolgreichen Lese- oder Schreibzugriff wird über die Variable der aktuelle Registerwert ausgegeben.

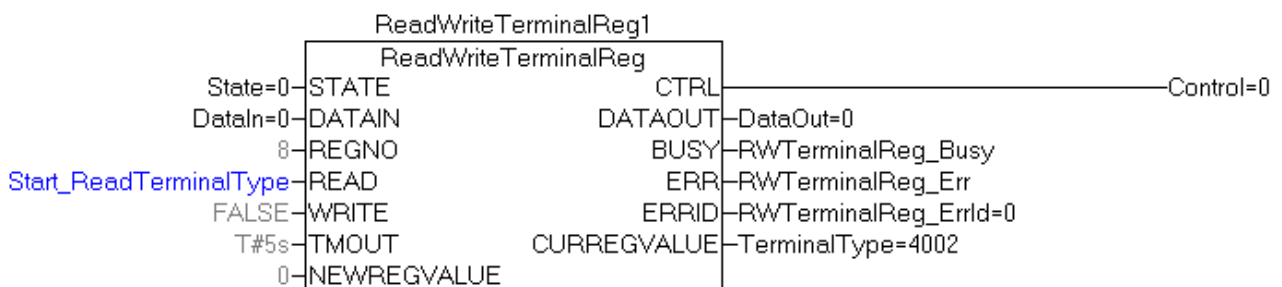
Fehlerbeschreibung:

Fehlernummer	Fehlerbeschreibung
0	kein Fehler
0x100	Timeout-Fehler. Die zulässige Ausführungszeit wurde überschritten.
0x200	Parameter Fehler (z.B. bei einer unzulässigen Registernummer).
0x300	Der gelesene Wert unterscheidet sich von dem geschriebenen Wert (Schreibzugriff auf diesen Register möglicherweise nicht erlaubt oder fehlgeschlagen)

Beispiele für einen Aufruf in FUP:

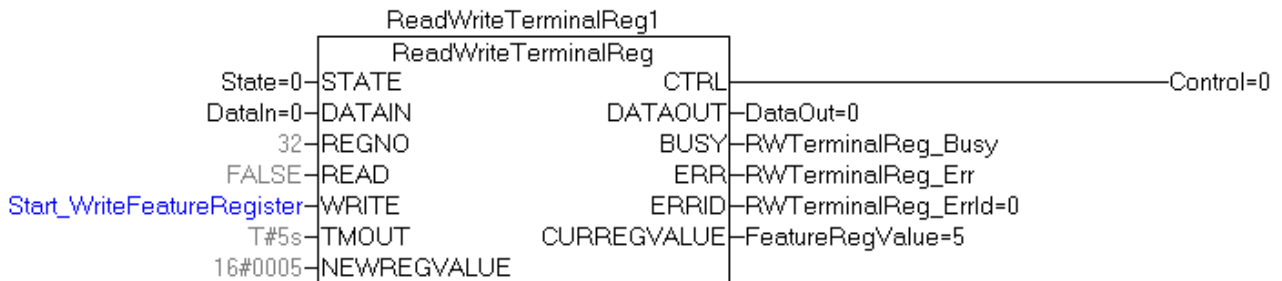
```
VAR
  ReadWriteTerminalReg1 : ReadWriteTerminalReg;
  State                 AT%IB0 : BYTE;
  Control               AT%QB0 : BYTE;
  DataIn               AT%IW1 : WORD;
  DataOut              AT%QW1 : WORD;
  Start_ReadTerminalType : BOOL;
  Start_WriteFeatureRegister : BOOL;
  RWTerminalReg_Busy   : BOOL;
  RWTerminalReg_Err    : BOOL;
  RWTerminalReg_ErrId  : UDINT;
  TerminalType         : WORD;
  FeatureRegValue      : WORD;
END_VAR
```

Beispiel 1



Im Beispiel 1 wird aus dem Register 8 einer analogen Ausgangsklemme die Klemmenbezeichnung ausgelesen. Die Variablen *State*, *Control*, *DataIn* und *DataOut* werden mit den entsprechenden IO-Variablen der Klemme im TwinCAT-Systemmanager verknüpft. Die Klemmenbezeichnung lautet: KL4002.

Beispiel 2

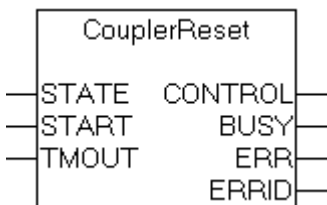


Im Beispiel 2 wird in dem Feature-Register (Register 32) einer analogen Ausgangsklemme KL4002 die Anwenderskalierung aktiviert. Der neue Wert des Feature-Registers wird dann von dem Funktionsbaustein gelesen und kann über die Ausgangsvariable **CURREGVALUE** überprüft werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version >= 2.7.0	PC (i386)	Alle Klemmen mit Status/Control-Byte (intelligente Klemmen)	Standard.Lib; TcPlcCoupler.Lib

3.2 CouplerReset



Mit dem Funktionsbaustein "CouplerReset" kann ein Reset des Kopplers über das 2 Byte SPS-Interface durchgeführt werden. Bei einem Koppler-Reset wird z.B. die aktuelle Klemmenkonfiguration über den K-Bus (Klemmenbus) von dem Koppler neu eingelesen und die Kommunikation auf dem K-Bus neu initialisiert. Vorhandene K-Bus Fehlermeldungen des Kopplers werden zurückgesetzt. Über die **STATE** und **CONTROL** Variablen wird ein Handshake während der Ausführung des Funktionsbausteins mit dem Koppler durchgeführt. Diese Variablen müssen daher mit den Status/Control IO-Variablen des 2 Byte SPS Interfaces im TwinCAT System Manager verknüpft werden.

VAR_INPUT

```
VAR_INPUT
    STATE      : PLCINTFSTRUCT;
    START      : BOOL;
    TMOUT      : TIME;
END_VAR
```

STATE: Statuswort [► 18] des 2 Byte SPS-Interfaces.

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

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung der Funktion nicht überschritten werden darf

VAR_OUTPUT

```
VAR_OUTPUT
  CONTROL : PLCINTFSTRUCT;
  BUSY : BOOL;
  ERR : BOOL;
  ERRID : UDINT;
END_VAR
```

CONTROL: Kontrollwort des 2 Byte SPS-Interfaces.

BUSY: Bei der Aktivierung des Bausteines wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung der Funktion abgeschlossen wurde.

ERR: Sollte ein Fehler bei der Ausführung der Funktion auftreten, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

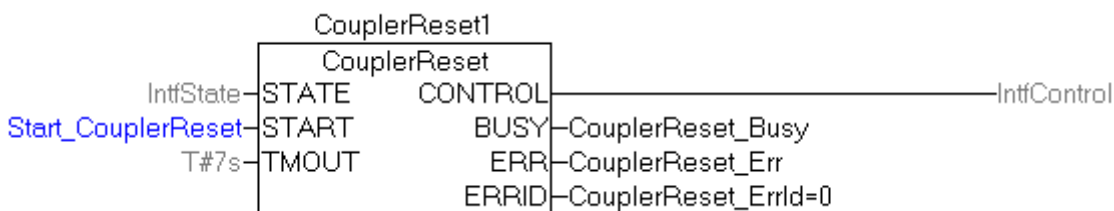
ERRID: Liefert bei einem gesetzten ERR-Ausgang die Fehlernummer ([Tabelle mit Fehlercodes](#) |▶ 211 |).

Fehlermeldungen

Fehlernummer	Fehlerbeschreibung
0	Kein Fehler
0x100	Fehler bei der Initialisierung der Kommunikation
0x200	Kommunikationsfehler
0x300	Zeitüberschreitung (Timeout-Fehler). Die vereinbarte Ausführungszeit wurde überschritten

Beispiel für einen Aufruf in FUP:

```
VAR
  IntfState AT%IW10 : PLCINTFSTRUCT;
  IntfControl AT%QW10 : PLCINTFSTRUCT;
  CouplerReset1 : CouplerReset;
  Start_CouplerReset : BOOL;
  CouplerReset_Busy : BOOL;
  CouplerReset_Err : BOOL;
  CouplerReset_ErrId : UDINT;
END_VAR
```

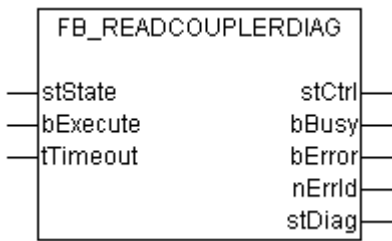


Die Variablen *IntfState* und *IntfControl* werden mit entsprechenden IO-Variablen im TwinCAT System Manager verknüpft.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version >= 2.7.0	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

3.3 FB_ReadCouplerDiag



Mit dem Funktionsbaustein FB_ReadCouplerDiag kann die erste und zweite Blinksequenz der Fehler-LED des Kopplers beim Auftreten eines Klemmbus- oder Koppler-Fehlers ausgelesen werden. Die Daten werden dabei über das 2 Byte SPS Interface zur SPS übertragen. Dies funktioniert aber nur dann, wenn die Kommunikation über den Feldbus aufrecht erhalten wurde. Die Daten müssen vom Koppler zur SPS über den Feldbus fehlerfrei übertragen werden können. Um einen aufgetretenen Kopplerfehler zu erkennen, kann das Statusbyte des Kopplers in der SPS zyklisch abgefragt werden und beim Eintreten eines Fehler wird der Funktionsbaustein aktiviert.

VAR_INPUT

```

VAR_INPUT
  stState      : PLCINTFSTRUCT;
  bExecute     : BOOL;
  tTimeout     : TIME;
END_VAR
  
```

stState : Statuswort [► 18] des 2 Byte SPS-Interfaces.

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

tTimeout : Gibt die Timeout-Zeit an, die bei der Ausführung der Funktion nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  stCtrl       : PLCINTFSTRUCT;
  bBusy        : BOOL;
  bError       : BOOL;
  nErrId       : UDINT;
  stDiag       : ST_CouplerDiag;
END_VAR
  
```

stCtrl : Kontrollwort des 2 Byte SPS-Interfaces.

bBusy : Bei der Aktivierung des Bausteines wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung der Funktion abgeschlossen wurde.

bError : Sollte ein Fehler bei der Ausführung der Funktion auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId : Liefert bei einem gesetzten bError-Ausgang die Fehlernummer. (Tabelle mit möglichen Fehlercodes [► 21]).

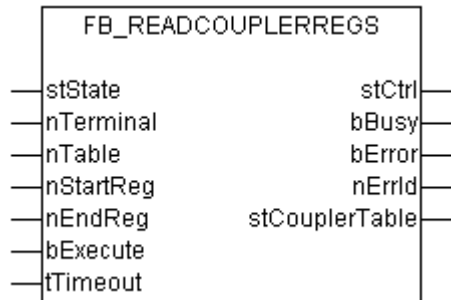
stDiag : Struktur [► 18] mit den Diagnoseinformationen des Kopplers (Fehlertyp und die erste und zweite Blinksequenz des Kopplers).

Voraussetzungen

Entwicklungsumgebung	Zielformat	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.8.0 Build > 735			

3.4 FB_ReadCouplerRegs



Mit dem Funktionsbaustein kann auf die Tabellenregister des Kopplers und die Register der intelligenten Klemmen lesend zugegriffen werden. Der Koppler selbst wird als Klemme 0 (Null) bezeichnet. Alle anderen Klemmen im Klemmenblock, außer den passiven Klemmen (z.B. Einspeiseklemme), werden (beginnend mit 1) fortlaufend gezählt. Es können alle Register oder nur ein Unterbereich (zwischen *nStartReg* und *nEndReg*) gelesen werden. Um alle Register (0.255) einer Tabelle des Kopplers zu lesen, werden mehrere Sekunden benötigt. Die erfolgreich gelesenen Registerwerte befinden sich in der Struktur *stCouplerTable*. Die Struktur ist ein Array mit Hi- und Lo-Bytes. Jedes Arrayelement entspricht einem Registerwert (z.B.: `stCouplerTable[5] == Register 5`).

VAR_INPUT

```
VAR_INPUT
  stState      : PLCINTFSTRUCT;
  nTerminal    : BYTE := TERM_COUPLER;
  nTable       : BYTE;
  nStartReg    : BYTE;
  nEndReg      : BYTE;
  bExecute     : BOOL;
  tTimeout     : TIME;
END_VAR
```

stState : Statuswort [► 18] des 2 Byte SPS-Interfaces.

nTerminal : Klemmennummer auf deren Tabellenregister zugegriffen werden soll. Der Koppler hat die Klemmennummer Null. Passive Klemmen dürfen nicht gezählt werden.

nTable : Tabellennummer deren Registerwerte gelesen werden sollen. Intelligente Klemmen besitzen nur eine Tabelle pro Klemmenkanal. Eine 4-Kanalige Klemme besitzt folgende Tabellennummern: 0-3. Eine intelligente Klemme besitzt aber nur maximal 64 Registerwerte pro Klemmenkanal!

nStartReg : Die Nummer des ersten Registers der gelesen werden soll.

nEndReg : Die Nummer des letzten Registers der gelesen werden soll.

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

tTimeout : Gibt die Timeout-Zeit an, die bei der Ausführung der Funktion nicht überschritten werden darf

VAR_OUTPUT

```
VAR_OUTPUT
  stCtrl       : PLCINTFSTRUCT;
  bBusy        : BOOL;
  bError       : BOOL;
  nErrId       : UDINT;
  stCouplerTable : ST_CouplerTable;
END_VAR
```

stCtrl : Kontrollwort des 2 Byte SPS-Interfaces.

bBusy : Bei der Aktivierung des Bausteines wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung der Funktion abgeschlossen wurde.

bError : Sollte ein Fehler bei der Ausführung der Funktion auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

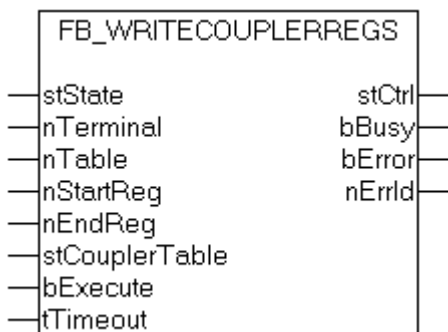
nErrId : Liefert bei einem gesetzten bError-Ausgang die Fehlernummer (Tabelle mit möglichen Fehlercodes [[► 21](#)]).

stCouplerTable : Struktur [[► 19](#)] mit den gelesenen Registerwerten der Klemme oder des Kopplers.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517 TwinCAT Version 2.8.0 Build > 735	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

3.5 FB_WriteCouplerRegs



Mit dem Funktionsbaustein kann auf die Tabellenregister des Kopplers und die Register der intelligenten Klemmen schreibend zugegriffen werden. Der Koppler selbst wird als Klemme 0 (Null) bezeichnet. Alle anderen Klemmen im Klemmenblock, außer den passiven Klemmen (z.B. Einspeiseklemme), werden (beginnend mit 1) fortlaufend gezählt. Es können alle Register oder nur ein Unterbereich (zwischen *nStartReg* und *nEndReg*) beschrieben werden. Um alle Register (0.255) einer Tabelle des Kopplers zu schreiben, werden mehrere Sekunden benötigt. Die zu schreibenden Registerwerte befinden sich in der Struktur *stCouplerTable*. Die Struktur ist ein Array mit Hi- und Lo-Bytes. Jedes Arrayelement entspricht einem Registerwert (z.B.: *stCouplerTable*[5] == Register 5).

VAR_INPUT

```

VAR_INPUT
  stState      : PLCINTFSTRUCT;
  nTerminal    : BYTE := TERM_COUPLER;
  nTable       : BYTE;
  nStartReg    : BYTE;
  nEndReg      : BYTE;
  bExecute     : BOOL;
  stCouplerTable : ST_CouplerTable;
  tTimeout     : TIME;
END_VAR
  
```

stState : Statuswort [[► 18](#)] des 2 Byte SPS-Interfaces.

nTerminal : Klemmennummer auf deren Tabellenregister zugegriffen werden soll. Der Koppler hat die Klemmennummer Null. Passive Klemmen dürfen nicht gezählt werden.

nTable : Tabellennummer deren Registerwerte geschrieben werden sollen. Intelligente Klemmen besitzen nur eine Tabelle pro Klemmenkanal. Eine 4-Kanalige Klemme besitzt folgende Tabellennummern: 0-3. Eine intelligente Klemme besitzt aber nur maximal 64 Registerwerte pro Klemmenkanal!

nStartReg : Die Nummer des ersten Registers der geschrieben werden soll.

nEndReg : Die Nummer des letzten Registers der geschrieben werden soll.

stCouplerTable : Array [▶ 19] der zu schreibenden Registerwerte.

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

tTimeout : Gibt die Timeout-Zeit an, die bei der Ausführung der Funktion nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  stCtrl      : PLCINTFSTRUCT;
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
END_VAR
```

stCtrl : Kontrollwort des 2 Byte SPS-Interfaces.

bBusy : Bei der Aktivierung des Bausteines wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Ausführung der Funktion abgeschlossen wurde.

bError : Sollte ein Fehler bei der Ausführung der Funktion auftreten, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

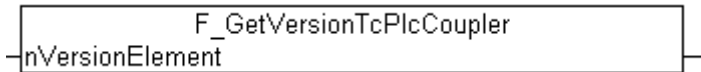
nErrId : Liefert bei einem gesetzten bError-Ausgang die Fehlernummer (Tabelle mit möglichen Fehlercodes [▶ 21]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 518 TwinCAT Version 2.8.0 Build > 737	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

4 Funktionen

4.1 F_GetVersionTcPlcCoupler



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

FUNCTION F_GetVersionTcPlcCoupler: UINT

```

VAR_INPUT
  nVersionElement : INT;
END_VAR
  
```

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

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

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT v2.9.0 Build > 1033	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib
TwinCAT v2.10.0 Build > 1257			

5 Datenstrukturen

5.1 PLCINTFSTRUCT

```

TYPE PLCINTFSTRUCT :
STRUCT
  Byte0 :BYTE;
  Byte1 :BYTE;
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version >= 2.7.0	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

5.2 E_CouplerErrType

```

TYPE E_CouplerErrType :
(
  CPLERR_NONE      := 0,      (* No error *)
  CPLERR_FIELDBUS  := 1,      (* Fieldbus error *)
  CPLERR_KBUS      := 2,      (* Terminal bus error (KBus)*)
  CPLERR_TERM_IO   := 4,      (* Terminal IO error *)
  CPLERR_COUPLER   := 8,      (* Coupler error *)
);
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib
TwinCAT Version 2.8.0 Build > 735			

5.3 ST_CouplerDiag

```

TYPE ST_CouplerDiag :
STRUCT
  eErrType      : E_CouplerErrType;
  stFlashCode   : ST_FlashCode;
END_STRUCT
END_TYPE

```

eErrType : Allgemeiner Fehlertyp [► 18].

stFlashCode : Die erste und zweite Sequenz [► 20] des Blinkcodes.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517 TwinCAT Version 2.8.0 Build > 735	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

5.4 ST_CouplerReg

```

TYPE ST_CouplerReg
STRUCT
    Lo : BYTE;
    Hi : BYTE;
END_STRUCT
END_TYPE
    
```

Ein Kopplerregister hat die Größe von einem Wort. In den Registern ist die Parametrierung und Konfiguration des Kopplers abgelegt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517 TwinCAT Version 2.8.0 Build > 735	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

5.5 ST_CouplerTable

```

TYPE ST_CouplerTable : ARRAY[ 0..255 ] OF ST_CouplerReg;
END_TYPE
    
```

Die Parameter und Konfiguration des Kopplers werden im EEPROM-Speicher des Kopplers abgelegt. Der Speicher ist in Tabellen unterteilt. Jede Tabelle besitzt maximal 256 Register [▶ 19](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517 TwinCAT Version 2.8.0 Build > 735	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

5.6 ST_FlashCode

```

TYPE ST_FlashCode :
STRUCT
  ErrType      : WORD;
  ErrLocation  : WORD;
END_STRUCT
END_TYPE

```

ErrType : Fehlertyp. Entspricht der ersten Blinksequenz des Kopplers.

ErrLocation : Fehlerstelle. Entspricht der zweiten Blinksequenz des Kopplers (Die Position der letzten Klemme vor der Fehlerstelle. Passive Klemmen werden nicht mitgezählt!).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	IO-Hardware	Einzubindende SPS-Bibliotheken
TwinCAT Version 2.7.0 Build > 517 TwinCAT Version 2.8.0 Build > 735	PC (i386)	Alle Koppler mit 2 Byte SPS- Interface	Standard.Lib; TcPlcCoupler.Lib

6 Anhang

6.1 Fehlercodes: 2 Byte PLC Interface

Fehlernummer	Fehlerbeschreibung
0	kein Fehler
0x100	Fehler bei der Initialisierung der Kommunikation über das 2 Byte SPS Interface.
0x200	Fehler während der Kommunikation
0x300	Timeout-Fehler. Die zulässige Ausführungszeit wurde überschritten.
0x400	Falscher Parameterwert bei Registernummer.
0x500	Falscher Parameterwert bei Tabellennummer.

Mehr Informationen:
www.beckhoff.de/tx1200

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

