

Handbuch | DE

TS6610

TwinCAT 2 | PLC S5/S7 Communication



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Einführung	10
4	Beschreibung der SPS-Datenbereich-Parameterwerte	12
5	Funktionsbausteine	14
5.1	FB_NLinkOpen.....	14
5.2	FB_NLinkClose	15
5.3	FB_NLinkFetch	16
5.4	FB_NLinkWrite	17
6	Funktionen	19
6.1	F_GetVersionTcS5S7Comm.....	19
7	Datentypen	20
7.1	E_NLinkArea	20
7.2	E_NLinkProtocol	20
7.3	E_NLinkTransport	20
7.4	E_NLinkOpState	21
7.5	ST_NLinkDebug.....	21
7.6	ST_NLinkSettings	21
7.7	T_HNLINK.....	22
8	Konstanten	23
9	Fehlercodes	24
9.1	Interne S5/S7 Kommunikations-Fehlercodes.....	24
9.2	Fehlersuche/Diagnose	26
10	Beispiele	28

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Übersicht

Mit den Funktionsbausteinen der SPS-Bibliothek **TcS5S7Comm.Lib** kann ein Datenaustausch zwischen der lokalen TwinCAT SPS und einer S5/S7 Steuerung über einen Ethernet-LAN-Adapter der Fa. IBHsoftec realisiert werden. Dieser Adapter ist nicht Bestandteil des Beckhoff Produktes und muss direkt bei IBHsoftec gekauft werden.

Systemvoraussetzungen

Unterstützte Adapter

- **IBH Link S5++** Adapter der Firma IBHsoftec (**ab Firmwareversion 1.39**, die Firmwareversion können Sie mit der mitgelieferten Konfigurationssoftware auslesen).
Ältere Adapter (ohne ++) werden zur Zeit nicht unterstützt.
- **IBH Link S7++** Adapter der Firma IBHsoftec (**ab Firmwareversion NL50MPI V02.004 05.06.09**, die Firmwareversion können Sie mit der mitgelieferten Konfigurationssoftware auslesen).

Programmierungsumgebung

- Windows NT (XP, XPe/WES, Vista, W7, WES7, ...);
- TwinCAT Installation: Level TwinCAT PLC oder höher;
- TwinCAT System Version 2.11.0 Build >= 1537 oder höher;

Zielplattform

- Industrie PC oder Embedded PC/CX (x86, ARM);
- Betriebssystem:
 - Windows NT (XP, XPe/WES, Vista, W7, WES7, ...);
 - Windows CE (x86) (image v3.08c oder höher);
 - Windows CE (ARM) (image v2.22c oder höher);
- TwinCAT SPS-Laufzeitsystem Version 2.10.0 oder höher;

Produktkomponenten

- **TcS5S7Comm.Lib** (Bibliothek mit den Kommunikationsbausteinen die Sie in Ihrer SPS-Applikation einbinden müssen);
- TcIpl.Lib (implementiert basic TCP/IP- und UDP-Funktionen);
- TcSocketHelper.Lib (implementiert TCP/IP-Hilfsfunktionen);
- TwinCAT TCP/IP Connection Server (TwinCAT Server);

Installation

Windows NT (XP, XPe/WES, Vista, W7, WES7, ...)

Die SPS-Bibliotheken werden in den ...\\TwinCAT\\PLC\\Lib-Ordner kopiert. Der TwinCAT TCP/IP Connection Server wird in die Liste der TwinCAT Server eingetragen. Beim TwinCAT Start wird der TCP/IP Connection Server automatisch gestartet und beim TwinCAT Stop gestoppt.

Windows CE

Führen Sie folgende Schritte aus, wenn Sie eine Produktversion für Windows CE erworben haben:

- Installieren Sie das Produkt zuerst wie gewohnt auf Ihrem Programmier-PC. Die SPS-Bibliotheken werden in den ...\\TwinCAT\\PLC\\Lib-Ordner kopiert.
- X86 CPU (CX1000, CX1020, ...):
 - Nach der Installation finden Sie im Ordner: ...\\TwinCAT\\CE\\TCPIP\\Install\\ ein Cabinet-File für das CE-Laufzeitsystem.
 - Kopieren Sie die darin befindliche Datei: **TcTCPIPSvrCe.I586.CAB** in einen Ordner auf dem CE-Laufzeitsystem.

- ARM CPU (CX9000, ...):
 - Nach der Installation finden Sie im Ordner: ...**TwinCAT\CE\TCPIP\Install**\ ein Cabinet-File für das CE-Laufzeitsystem.
 - Kopieren Sie die darin befindliche Datei: **TcTCPIPSvrCe.ARMV4I.CAB** in einen Ordner auf dem CE-Laufzeitsystem.
- Auf dem CE-System: Installieren Sie (durch einen Doppelklick auf das Cabinet-File) die CE-Komponenten.
- Führen Sie ein Reboot des CE-Geräts durch. Der TwinCAT TCP/IP Connection Server wird mit dem CE-Betriebssystem automatisch gestartet.

3 Einführung

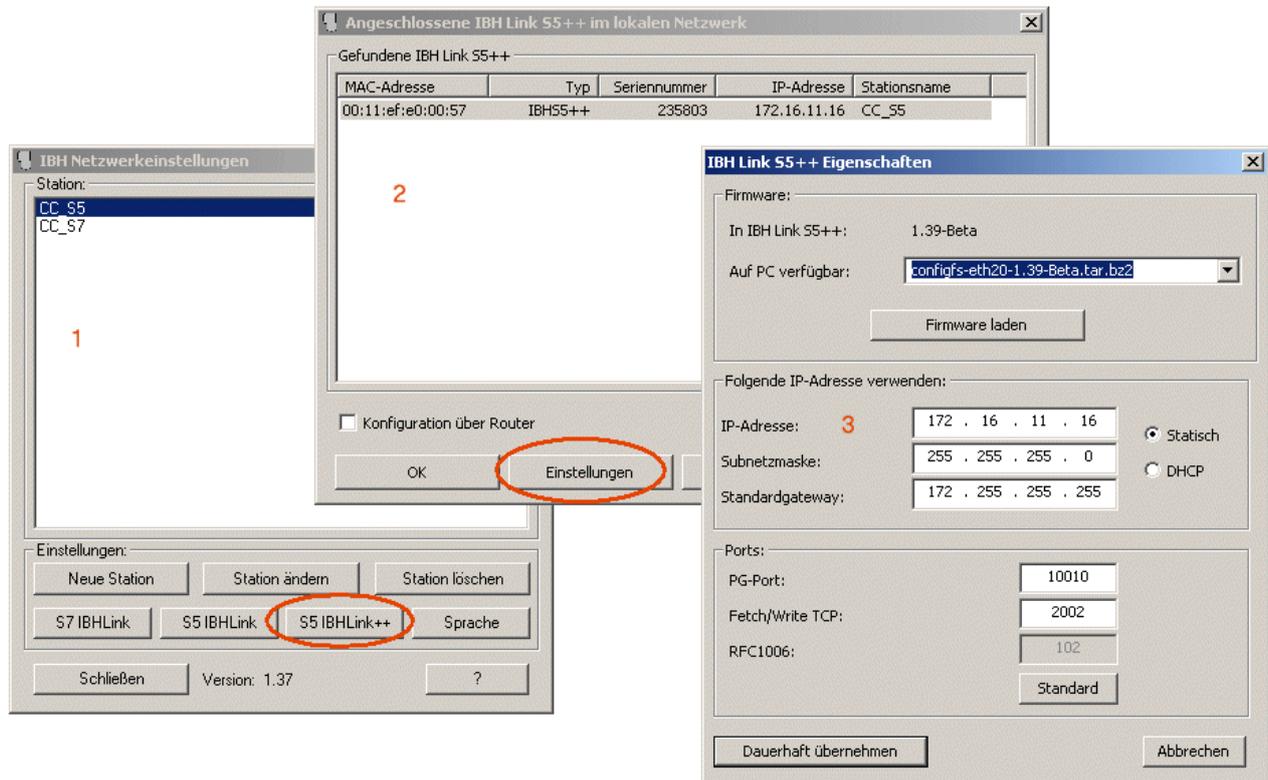
In der Herstellerdokumentation finden Sie eine detaillierte Beschreibung zu Konfiguration der Module.

Installationshinweise finden Sie auf der [Übersichtsseite](#) [► 8].

Auf der [Beispiele](#) [► 28]-Seite finden Sie einfache SPS-Projektbeispiele inklusive aller Quellen.

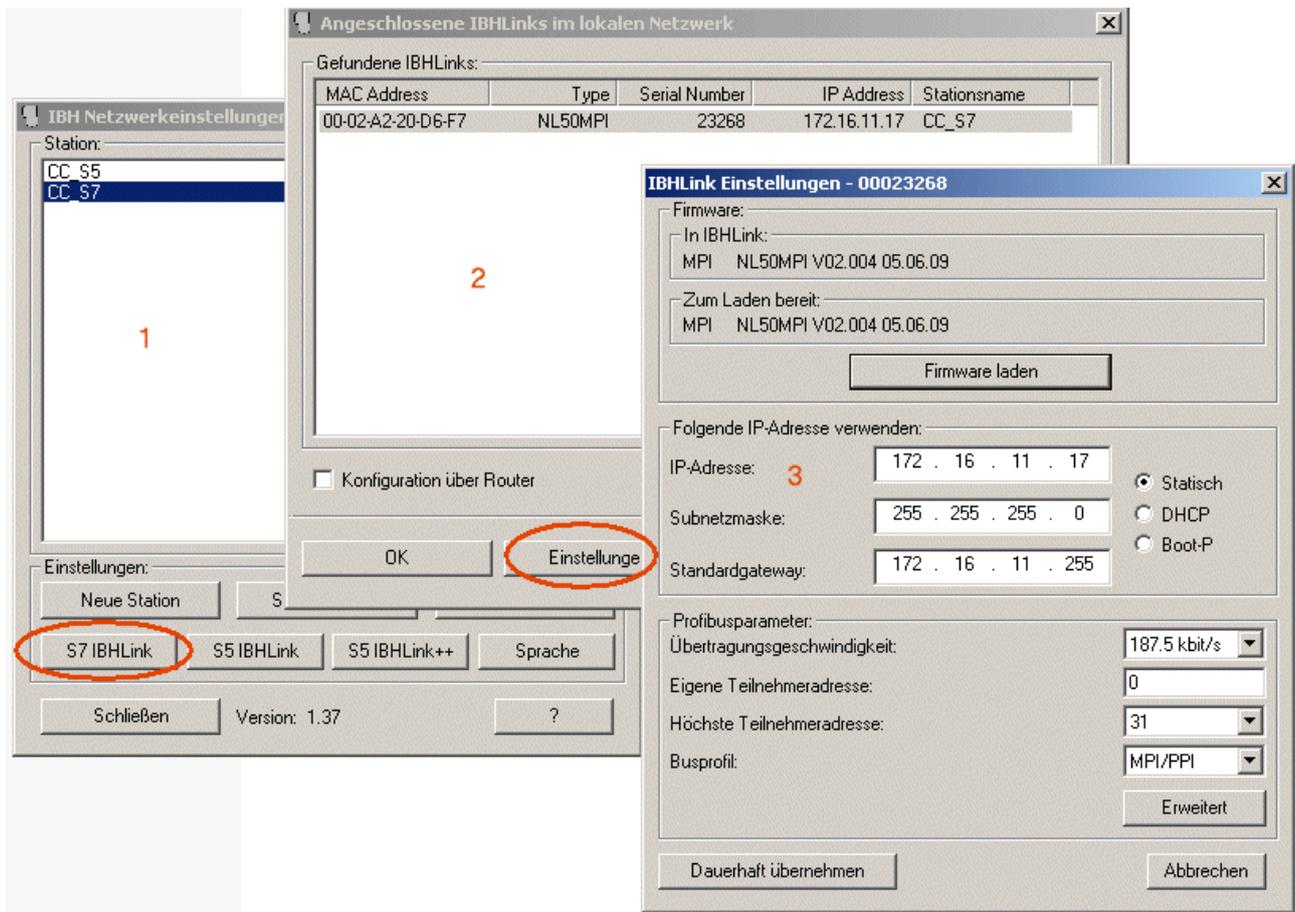
Konfiguration eines IBH Link S5++ Adapters

Im folgenden Beispiel wurde eine feste IP-Adresse: '172.16.11.16' eingestellt:



Konfiguration eines IBH Link S7++ Adapters

Im folgenden Beispiel wurde eine feste IP-Adresse: '172.16.11.17' eingestellt:



4 Beschreibung der SPS-Datenbereich-Parameterwerte

Abhängig von dem Typ der verwendeten Fremdsteuerung, des Adapters oder Kommunikationsprotokolls unterscheiden sich die zu verwendenden Parameterwerte für den Schreib-/Lesebefehl. Einige Datenbereiche sind möglicherweise nicht verfügbar oder die Daten müssen wortweise statt byteweise gelesen/geschrieben werden. Weiterführende Informationen finden Sie in der Dokumentation der Fremdsteuerung.

Format-Konvertierung notwendig

Beim Lesezugriff auf Wort-Variablen in der S5/S7 Steuerung liefert diese die Daten im Motorola-Format (Hi und Lo-Bytes) vertauscht. Die gelesenen Daten müssen für die Auswertung zuerst in das im TwinCAT System üblichen Intel-Format konvertiert werden. Daten, die vom TwinCAT System zur S5/S7-Steuerung geschrieben werden sollen, müssen vor dem Schreibzugriff zuerst in das Motorola-Format konvertiert werden! Das BCD-Zahlensystem wird in der S5/S7-Steuerung zur Darstellung der Zähler/Timer-Variablen benutzt. Die Daten müssen dann ggf. auch entsprechend konvertiert werden.



Die TwinCAT SPS-Bibliothek: TcUtilities.Lib bietet entsprechende Funktionen zur Konvertierung zwischen Motorola und Intel-Format (z.B.: BE16_TO_HOST, HOST_TO_BE16...).

Folgende Tabelle gibt eine Übersicht über die verfügbaren Kommunikationswege:

Steuerung	LAN-Adapter	Transportprotokoll	LAN-Adapter-Kommunikationsprotokoll	Beschreibung
S5	IBH Link S5++	TCP/IP	IBH S5 Lan Link Protokoll = FETCH/ WRITE-Telegramme [▶ 12]	Keine Verbindungsparametrierung in der Fremdsteuerung erforderlich!
S7	IBH Link S7++	TCP/IP	NetLink MPI-Protokoll (NL-MPI) der Fa. Hilscher [▶ 13]	Keine Verbindungsparametrierung in der Fremdsteuerung erforderlich!

IBH S5 Lan Link Protokoll = FETCH/WRITE-Telegramme

- Dieses Protokoll kann zurzeit nur mit S5-Steuerungen verwendet werden;
- Die TCP/IP-Portnummer hat den Defaultwert: 2002;
- Der Parameter devAddr wird nicht ausgewertet;
- Die Datenbausteinnummer (nDB) wird nur beim Zugriff auf Datenbereich: Datenbausteine (DB) und erweiterte Datenbausteine (DX) ausgewertet. Zulässiger Bereich: 1..255;
- Der Bereich P (Peripherie) und OS (Operating state) wird nicht unterstützt;

Datenbereich (eArea)	Startadresse (nOffset)	Zulässige Datenlänge (cbBuffer)
DB	Datenwort-Offset: 0..2047	1..2048 Worte lesen/schreiben
M	Merkerbyte-Offset: 0..255	1..256 Bytes lesen/schreiben
I	Eingangsbyte-Offset: 0..127	1..128 Bytes lesen
Q	Ausgangsbyte-Offset: 0..127	1..128 Bytes schreiben
P	I/O Peripheriebyte-Offset: 0..127 (digital I/Os), 128..255 (analog I/Os)	1..256 Bytes lesen/schreiben
C	Zähler Start-Offset: 0..255	1 Wort lesen/schreiben
T	Timer Start-Offset: 0..255	1 Wort lesen/schreiben
DX	Datenwort-Offset	1..2048 Worte lesen/schreiben

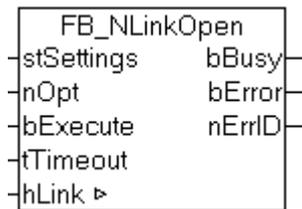
NetLink MPI-Protokoll (NL-MPI) der Fa. Hilscher

- Dieses Protokoll kann nur mit S7 Steuerungen verwendet werden;
- Die TCP/IP-Portnummer hat den Defaultwert: 1099
- Die MPI-Adresse (devAddr) hat den Defaultwert: 2. Zulässige Werte: 1..126;
- Die Datenbausteinnummer (nDB) wird nur beim Zugriff auf Datenbereich: Datenbausteine (DB) ausgewertet. Zulässige Werte: 0..65534;
- Der Bereich P (Peripherie) und DX (erweiterter Datenbaustein) wird nicht unterstützt;
- Beim Parameter für die Datenlänge (cbBuffer) muss immer Bytelänge angegeben werden;
- Beim Bereich OS (Operating state) ist nur lesender Zugriff möglich;

Datenbereich (eArea)	Startadresse (nOffset)	Zulässige Datenlänge (cbBuffer)
DB	Datenbyte-Offset: 0..65534	max. 222 Bytes lesen und 216 schreiben
M	Merkerbyte-Offset: 0..65534	
I	Eingangsbyte-Offset: 0..65534	
Q	Ausgangsbyte-Offset: 0..65534	
C	Zähler Start-Offset: 0..65534	Max. 111 Wort-Zähler lesen und max. 108 Zähler schreiben
T	Timer Start-Offset: 0..65534	Max. 111 Wort-Timer lesen und max. 108 Timer schreiben
OS	Operating state: Irrelevant	1 Wort (0=STOP, 1=START, 2=RUN, 3=UNKNOWN)

5 Funktionsbausteine

5.1 FB_NLinkOpen



Dieser Funktionsbaustein baut (aktiv) eine Netzwerkverbindung zum Fremdsystem auf. Beim Erfolg werden die internen Parameter der *hLink*-Variablen initialisiert. Die selbe *hLink*-Variableninstanz muss dann an die anderen Instanzen der Kommunikationsbausteine: [FB_NLinkFetch \[▶ 16\]](#), [FB_NLinkWrite \[▶ 17\]](#) oder [FB_NLinkClose \[▶ 15\]](#) übergeben werden. In einer SPS-Applikation wird pro Netzwerkverbindung (TwinCAT System<->Ethernet-LAN-Adapter<->Fremdstuerung) eine Instanz von dieser Variablen benötigt.

VAR_IN_OUT

```
VAR_IN_OUT
  hLink : T_HNLINK;
END_VAR
```

hLink : [Netzwerkverbindungshandle \[▶ 22\]](#). Die Instanz dieser Variable wird von den anderen Kommunikationsbausteinen benötigt und wird an sie auch per VAR_IN_OUT übergeben.

VAR_INPUT

```
VAR_INPUT
  stSettings : ST_NLinkSettings := ( sHost := '127.0.0.1', nPort := 2002 ); (* Link/
connection parameter/settings *)
  nOpt       : DWORD := 0; (* Additional options (reserved) *)
  bExecute   : BOOL; (* Rising edge at this input starts command execution *)
  tTimeout   : TIME := T#45s; (* Maximum time allowed for the execution of the command *)
END_VAR
```

stSettings: [Struktur \[▶ 21\]](#) mit Konfigurationseinstellungen der Kommunikationsverbindung. Besetzen Sie die Membervariablen dieser Struktur mit den Parametern wie Portnummer, Hostname usw.

nOpt: Reserviert für zukünftige Anwendungen.

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

tTimeout: Gibt die Timeout-Zeit an, die bei der Annahme des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

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

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Kommandos, sondern nur dessen Annahme zeitlich überwacht wird.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die [Fehlernummer \[▶ 24\]](#).

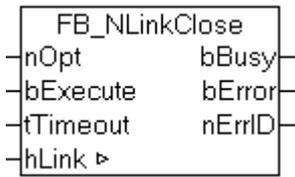
Beispiel:

Siehe unter [Beispiele \[▶ 28\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

5.2 FB_NLinkClose



Dieser Funktionsbaustein baut die Netzwerkverbindung zum Fremdsystem ab.

VAR_IN_OUT

```

VAR_IN_OUT
    hLink : T_HNLINK;
END_VAR
  
```

hLink : Netzwerkverbindungshandle [▶ 22]. Hier übergeben Sie die Instanz der Variablen die bei der Verbindungsherstellung von dem FB_NLinkOpen [▶ 14]-Funktionsbaustein verwendet wurde.

VAR_INPUT

```

VAR_INPUT
    nOpt      : DWORD := 0; (* Additional options (reserved) *)
    bExecute  : BOOL; (* Rising edge at this input starts command execution *)
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
    (* Maximum time allowed for the execution of the command *)
END_VAR
  
```

nOpt: Reserviert für zukünftige Anwendungen.

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

tTimeout: Gibt die Timeout-Zeit an, die bei der Annahme des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

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

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Kommandos, sondern nur dessen Annahme zeitlich überwacht wird.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die Fehlernummer [▶ 24].

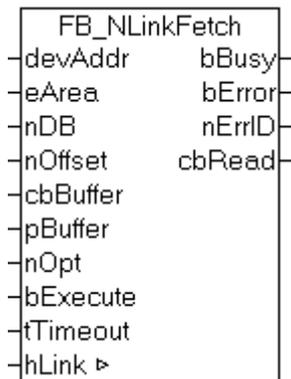
Beispiel:

Siehe unter Beispiele [▶ 28].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

5.3 FB_NLinkFetch



Dieser Funktionsbaustein liest Daten vom Fremdsystem und schreibt diese ins lokale TwinCAT System (FETCH-Dienst).

VAR_IN_OUT

```
VAR_IN_OUT
  hLink : T_HNLINK;
END_VAR
```

hLink : [Netzwerkverbindungshandle](#) [▶ 22]. Hier übergeben Sie die Instanz der Variablen die bei der Verbindungsherstellung von dem [FB_NLinkOpen](#) [▶ 14]-Funktionsbaustein verwendet wurde.

VAR_INPUT

```
VAR_INPUT
  devAddr      : BYTE := 2;
  (* Device address, MPI address of the remote component or its PROFIBUS address *)
  eArea        : E_NLinkArea := eNLinkArea_DB; (* PLC data area *)
  nDB          : WORD := 0; (* Data block number from which the data is read (DB, DX only). *)
  nOffset      : WORD := 0; (* Start byte/word address from which the data are taken to *)
  cbBuffer     : UDINT(0..MAX_NLink_DATALENGTH) := 0; (* Contains the max. number of destination
  bytes to be received *)
  pBuffer      : DWORD := 0; (* Contains the address of the destination buffer for the received
  data *)
  nOpt         : DWORD := 0; (* Additional options (reserved) *)
  bExecute     : BOOL; (* Rising edge at this input starts command execution *)
  tTimeout     : TIME := DEFAULT_ADS_TIMEOUT;
  (* Maximum time allowed for the execution of the command *)
END_VAR
```

devAddr: Geräteadresse, MPI-Adresse der Komponente auf dem Fremdsystem.

eArea: SPS [Datenbereich](#) [▶ 20] auf dem Fremdsystem (Eingänge, Ausgänge, Datenbaustein usw.) aus dem Daten gelesen werden sollen.

nDB: Datenbausteinnummer auf dem Fremdsystem. Aus diesem Datenbaustein sollen Daten gelesen werden (nur relevant beim Zugriff auf Datenbausteine und erweiterte Datenbausteine).

nOffset: Die Start-Byteadresse oder Wortadresse ab der in dem Fremdsystem Daten gelesen werden sollen. Ob Byte- oder Wortadresse angegeben werden soll ist abhängig von dem Typ der verwendeten Fremdsteuerung/Protokoll und Datenbereich auf den Sie zugreifen möchten. Weitere Informationen finden Sie hier: [Beschreibung der SPS-Datenbereich-Parameterwerte](#) [▶ 12].

cbBuffer: Maximale Anzahl der Datenbytes, die gelesen werden soll. Auch beim Zugriff auf Wortadressen wird hier immer die Bytelänge angegeben!

pBuffer: Pointer/Adresse des Zieldatenpuffers auf dem TwinCAT System. In diesen Puffer werden die gelesenen Daten geschrieben. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Zieldatenpuffer darf nicht kleiner sein als die angegebene Länge der zu lesenden Daten.

nOpt: Reserviert für zukünftige Anwendungen.

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

tTimeout: Gibt die Timeout-Zeit an, die bei der Annahme des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError  : BOOL;
  nErrID  : UDINT;
  cbRead  : UDINT := 0; (* Number of recend realy returned data bytes *)
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Kommandos, sondern nur dessen Annahme zeitlich überwacht wird.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die Fehlernummer [▶ 24].

cbRead: Bei Erfolg Anzahl der erfolgreich gelesenen Datenbytes.

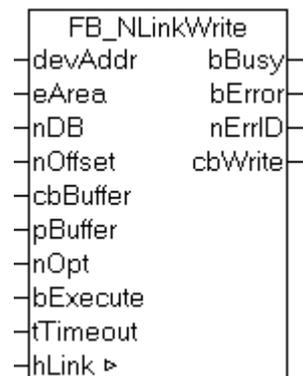
Beispiel:

Siehe unter Beispiele [▶ 28].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

5.4 FB_NLinkWrite



Dieser Funktionsbaustein sendet Daten vom lokalen TwinCAT System zum Fremdsystem (WRITE-Dienst).

VAR_IN_OUT

```
VAR_IN_OUT
  hLink : T_HNLINK;
END_VAR
```

hLink : Netzwerkverbindungshandle [▶ 22]. Hier übergeben Sie die Instanz der Variablen die bei der Verbindungsherstellung von dem FB_NLinkOpen [▶ 14]-Funktionsbaustein verwendet wurde.

VAR_INPUT

```
VAR_INPUT
  devAddr      : BYTE := 2;
  (* Device address, MPI address of the remote component or its PROFIBUS address *)
  eArea       : E_NLinkArea := eNLinkArea_DB; (* PLC data area *)
  nDB        : WORD := 0; (* Data block number to which the data is written (DB, DX only) *)
  nOffset     : WORD := 0; (* Start byte/word address to which the data is send *)
```

```

cbBuffer      : UDINT(0..MAX_NLink_DATALENGTH) := 0;
(* Contains the number of source bytes to be send *)
pBuffer       : DWORD := 0; (* Contains the address of the source buffer containing the data to
be send *)
nOpt          : DWORD := 0; (* Additional options (reserved) *)
bExecute      : BOOL; (* Rising edge at this input starts command execution *)
tTimeout      : TIME := DEFAULT_ADS_TIMEOUT;
(* Maximum time allowed for the execution of the command *)
END_VAR

```

devAddr: Geräteadresse, MPI-Adresse der Komponente auf dem Fremdsystem.

eArea: SPS [Datenbereich](#) [► 20] auf dem Fremdsystem (Eingänge, Ausgänge, Datenbaustein usw.) in den Daten geschrieben werden sollen.

nDB: Datenbausteinnummer auf dem Fremdsystem. In diesen Baustein sollen Daten geschrieben werden (nur relevant beim Zugriff auf Datenbausteine und erweiterte Datenbausteine).

nOffset: Die Start-Byteadresse oder Wortadresse ab der auf dem Fremdsystem Daten geschrieben werden sollen. Ob Byte- oder Wortadresse angegeben werden soll ist abhängig von dem Typ der verwendeten Fremdsteuerung/Protokoll und dem Datenbereich auf den Sie zugreifen möchten. Weitere Informationen finden Sie hier: [Beschreibung der SPS-Datenbereich-Parameterwerte](#) [► 12].

cbBuffer: Maximale Bytelänge der Daten, die vom Quelldatenpuffer auf dem lokalen TwinCAT System zum Fremdsystem übertragen werden soll. Auch beim Zugriff auf Wortadressen wird hier die Bytelänge angegeben!

pBuffer: Pointer/Adresse auf den Quelldatenpuffer. Daten aus diesem Puffer werden zum Fremdsystem übertragen. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Quelldatenpuffer darf nicht kleiner sein als die angegebene Länge der zu sendenden Daten.

nOpt: Reserviert für zukünftige Anwendungen.

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

tTimeout: Gibt die Timeout-Zeit an, die bei der Annahme des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrID     : UDINT;
  cbWrite    : UDINT := 0; (* Number of successfull written data bytes *)
END_VAR

```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Kommandos, sondern nur dessen Annahme zeitlich überwacht wird.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die [Fehlernummer](#) [► 24].

cbWrite: Bei Erfolg Anzahl der erfolgreich geschriebenen Datenbytes.

Beispiel:

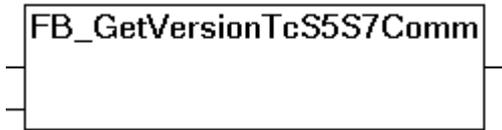
Siehe unter [Beispiele](#) [► 28].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

6 Funktionen

6.1 F_GetVersionTcS5S7Comm



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

FUNCTION F_GetVersionTcS5S7Comm : 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	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC oder CX (x86, ARM)	TcS5S7Comm.Lib

7 Datentypen

7.1 E_NLinkArea

```

TYPE E_NLinkArea:
(
  eNLinkArea_DB := 16#01, (* Source/dest. data from/to data block in main memory. *)
  eNLinkArea_M := 16#02, (* Source/dest. data from/to flag area. *)
  eNLinkArea_I := 16#03, (* Source/dest. data from/to process image of the inputs. *)
  eNLinkArea_Q := 16#04, (* Source/dest. data from/to process image of the outputs. *)
  eNLinkArea_P := 16#05, (* Source/dest. data from/to in I/O modules (S7 only PI, PQ). *)
  eNLinkArea_C := 16#06, (* Source/dest. data from/to counter cells. *)
  eNLinkArea_T := 16#07, (* Source/dest. data from/to timer cells. *)
  eNLinkArea_DX := 16#0A (* Source/dest. data from/to extended data block. *)
  eNLinkArea_OS := 16#100 (* Read PLC operational status information *)
);
END_TYPE

```

SPS Datenbereich (Eingänge, Ausgänge, Merker, Datenbaustein usw.) auf den zugegriffen werden soll. Abhängig von dem Typ der verwendeten Fremdsteuerung, des Adapters oder Kommunikationsprotokolls unterscheiden sich die zu verwendenden Parameterwerte. Einige Datenbereiche sind möglicherweise nicht verfügbar oder die Daten müssen wortweise statt byteweise gelesen/geschrieben werden.

Genauere Informationen zu diesem Thema finden Sie hier: [Beschreibung der SPS-Datenbereich-Parameterwerte](#) |> [12](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.2 E_NLinkProtocol

```

TYPE E_NLinkProtocol:
(
  eNLinkProtocol_IBH_SX_PP := 0, (* IBH Link protocol (used by IBH Link S5++ adater) *)
  eNLinkProtocol_MPI (* Hilscher NL-MPI protocol (used only by IBH Link S7++ adapter) *)
);
END_TYPE

```

Kommunikationsprotokoll (hängt von der verwendeten Hardware ab).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.3 E_NLinkTransport

```

TYPE E_NLinkTransport:
(
  eNLinkTransport_TCPIP := 0
);
END_TYPE

```

Transportprotokoll (bestimmt wie Daten transportiert werden sollen). Zur Zeit wird nur das streamorientierte TCP/IP-Protokoll unterstützt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.4 E_NLinkOpState

```

TYPE E_NLinkOpState:
(
  eNLinkOpState_STOP      := 0,
  eNLinkOpState_START     := 1,
  eNLinkOpState_RUN       := 2,
  eNLinkOpState_UNKNOWN   := 3
);
END_TYPE
    
```

Operational state (nur S7);

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.5 ST_NLinkDebug

Über diese Parameter können bei der Fehlersuche die Debugausgaben aktiviert/deaktiviert werden (Standardeinstellung: deaktiviert).

Die Debugausgaben werden im Logview vom TwinCAT System Manager ausgegeben.

```

TYPE ST_NLinkDebug:
STRUCT
  eTx      : E_DbgDirection := eDbgDirection_OFF; (* Enable/disable debug output of data-requests/
responses. *)
  eRx      : E_DbgDirection := eDbgDirection_OFF; (* Enable/disable debug output of data-indications/
confirmations. *)
  bState   : BOOL           := TRUE; (* TRUE => Enable debug output *)
END_STRUCT
END_TYPE
    
```

eTx: Aktiviert/deaktiviert die Debugausgabe des Sendepuffers.

eRx: Aktiviert/deaktiviert die Debugausgabe des Empfangspuffers.

bState: Aktiviert/deaktiviert Statusmeldungen beim Aufbauen oder Abbauen der Socketverbindung.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.6 ST_NLinkSettings

```

TYPE ST_NLinkSettings:
STRUCT
  sSrvNetID      : T_AmsNetID := ''; (* The network address of the TcpIpServer.exe. If empty st
ring=>server runs on local system *)
  eProtocol       : E_NLinkProtocol := eNLinkProtocol_IBH_SX_PP; (* Lan link protocol type *)
  eTransport      : E_NLinkTransport := eNLinkTransport_TCPIP; (* Used transport protocol *)
  sHost          : T_IPv4Addr := '127.0.0.1'; (* Remote (server) address. String containing an
(Ipv4) Internet Protocol dotted address. *)
  nPort          : UDINT := 2002; (* Remote (server) Internet Protocol (IP) port. *)
  local_T_selector : STRING := '00 01'; (* local-transport-selector *)
  remote_T_selector : STRING := '00 01'; (* remote-transport-selector *)
  bServer         : BOOL := FALSE; (* Reserved: TRUE = server connection, FALSE = client connect
ion *)
  bThrottle       : BOOL := FALSE; (* If set reduces the number of polling socket read requests
*)
  debug           : ST_NLinkDebug;
END_STRUCT
END_TYPE
    
```

sSrvNetID: Die TwinCAT Netzwerkadresse des TwinCAT TCP/IP Connection servers (default = Leerstring).

eProtocol: Über diesen Parameter [► 20] wird das zu verwendende Kommunikationsprotokoll eingestellt.

eTransport: Über diesen [Parameter \[► 20\]](#) wird das zu verwendende Transportprotokoll eingestellt.

sHost: String mit der Netzwerkadresse (Ipv4) des Kommunikationspartners (default = '127.0.0.1').

nPort: Internet Protocol (IP) port (default = 2002).

local_T_selector : Lokale TSAP-Adresse (reserviert).

remote_T_selector : Remote TSAP-Adresse (reserviert).

bServer: FALSE = Client (aktive) Verbindung. TRUE = Server (passive) Verbindung. Zur Zeit werden nur Clientverbindungen zugelassen.

bThrottle: Wenn dieser Parameter gesetzt ist, dann reduziert sich die Anzahl der pollenden Socket-Aufrufe wenn keine oder Wenig Daten empfangen werden. D.h. der Poll-Zyklus wird verlangsamt und dadurch die Systemauslastung verringert (default = FALSE).

debug: Optionale [Einstellungen \[► 21\]](#) der Debugausgabe.

Hintergrundinformation zur Konfiguration der eProtocol- und eTransport-Parameter:

Der IBHsoftec S7++ Adapter kann theoretisch über mehr als einen Kommunikationsweg (Protokoll) mit der S7-Hardware kommunizieren.

- Über das Hilscher MPI-Protokoll + TCP/IP als Transportprotokoll.
- Über das IBHsoftec S5++/S7++-Protokoll + RFC1006 als Transportprotokoll.

Die TwinCAT S5/S7 Kommunikationsbibliothek unterstützt zurzeit nur den ersten Kommunikationsweg, da zur Zeit nur das TCP/IP als Transportprotokoll unterstützt wird.

D.h für die Kommunikation mit S7 Geräten bitte folgende Einstellung setzen:

eProtocol := eNLinkProtocol_MPI;

eTransport := eNLinkTransport_TCPIP;

Für die Kommunikation mit den S5 Geräten bitte folgende Einstellung setzen:

eProtocol := eNLinkProtocol_IBH_SX_PP;

eTransport := eNLinkTransport_TCPIP;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

7.7 T_HNLINK

Eine Variable von diesem Typ repräsentiert ein Netzwerkverbindungshandle. Die Strukturelemente sollen nicht direkt beschrieben oder verändert werden. Variablen von diesem Typ werden für den internen Datenaustausch benutzt.

Pro Netzwerkverbindung (TwinCAT System<->Ethernet-LAN-Adapter<->Fremdsteuerung) wird eine Instanz benötigt. Dieselbe Instanz wird dann als VAR_IN_OUT-Parameter an andere Kommunikationsbausteine übergeben.

```
TYPE T_HNLINK :
STRUCT
END_STRUCT
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

8 Konstanten

Name	Wert	Beschreibung
MAX_NLINK_BUFFER_SIZE	16#FFFF	Max. interne Tx/Rx-Puffergröße (64K)
MAX_NLINK_DATALEN	2048	Max. Länge der zu übertragenen Daten die von den SPS-Bausteinen unterstützt wird. Die tatsächliche max. Länge der zu übertragenen Daten kann aber durch Geräteeigenschaften kleiner sein!
MAX_NLINK_HEADERLEN	16	Max. Länge des Frame-Headers beim NL-MPI und IBHNet-Protokoll.
MAX_NLINK_MPI_CONNECTIONS	16	Max. Anzahl der offenen MPI-Verbindungen. Beim Überschreiten dieser Anzahl werden die ältesten Verbindungen automatisch wieder geschlossen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC or CX (x86, ARM)	TcS5S7Comm.Lib

9 Fehlercodes

Voraussetzungen

Fehlercodes (Hex)	Fehlercodes (Dez)	Fehlerquelle	Beschreibung
0x00000000-0x00007800	0-30720	TwinCAT System Fehler	TwinCAT System Fehler (ADS-Fehlercodes inklusive)
0x00008000-0x000080FF	32768-33023	Interne TwinCAT TCP/IP Connection Server Fehler	Interne Fehler des TwinCAT TCP/IP Connection Servers
0x00008300-0x000085FF	33536-34303	Interne TwinCAT S5/S7 Communication Fehler [► 24]	Interne Fehler im S5/S7 Ethernet/Lan Modul oder der TwinCAT SPS-Bibliothek
0x80070000-0x8007FFFF	2147942400-2148007935	Fehlerquelle = Fehlercode - 0x80070000 = Win32 System Fehler	Win32 Systemfehler (Windows Sockets Fehlercodes inklusive)

9.1 Interne S5/S7 Kommunikations-Fehlercodes

Fehlercode (Hex)	Fehlercode (Dez)	Symbolische Konstante	Beschreibung
		General error codes	Offset 0x8300 + Fehlercode
0x00000000	0	NLINK_COMMERR_NOERROR	Kein Fehler
0x00008300	33536	NLINK_COMMERR_TIMEOUT	Timeout während der Kommunikation (Transaktion)
0x00008301	33537	NLINK_COMMERR_TXOVERFLOW	Sendepuffer Überlauf
0x00008302	33538	NLINK_COMMERR_RXOVERFLOW	Empfangspuffer Überlauf
0x00008303	33539	NLINK_COMMERR_FATAL	Fataler interner Fehler
0x00008304	33540	NLINK_COMMERR_SYSTEMID	Fehlerhafte System-ID
0x00008305	33541	NLINK_COMMERR_HEADERLEN	Fehlerhafter Header-Längenparameter
0x00008306	33542	NLINK_COMMERR_OPCODE	Fehlerhafter OP-Code, OP-Codelänge oder OP-Codewort
0x00008307	33543	NLINK_COMMERR_INVALIDSIZE	Fehlerhafte Datenlänge
0x00008308	33544	NLINK_COMMERR_INVALIDSTATE	Gerät oder Verbindung befindet sich im fehlerhaften/unzulässigen Zustand
0x00008309	33545	NLINK_COMMERR_INVALIDVALUE	Es wurde ein falscher Parameterwert angegeben
0x0000830A	33546	NLINK_COMMERR_INVALIDAREA	Falscher/nicht unterstützter Datenbereich
0x0000830B	33547	NLINK_COMMERR_UNEXPTECTED	Unerwarteter Fehler
		IBH Link S5++ specific error codes	Offset 0x8400 + Fehlercode
0x00008402	33794	NLINK_COMMERR_IBH_AREA_NOT_AVAILABLE	Datenbereich in der Steuerung nicht vorhanden, Datenbaustein nicht vorhanden
0x00008403	33795	NLINK_COMMERR_IBH_AREA_TO_SMALL	Angeforderter Bereich ist zu klein
0x00008406	33798	NLINK_COMMERR_IBH_INVALID_HEADER	Falsches Frame-/Header-Format

Fehlercode (Hex)	Fehlercode (Dez)	Symbolische Konstante	Beschreibung
0x00008407	33799	NLINK_COMMERR_IBH_DATA_OVE RFLOW	Zulässige Datengröße überschritten, Datenpuffer zu klein
0x00008409	33801	NLINK_COMMERR_IBH_TIMEOUT	Timeout während der Kommunikation mit der Steuerung
0x0000840A	33802	NLINK_COMMERR_IBH_DATA_UND ERFLOW	Es wurden zu wenig Daten empfangen
		Hilscher NL-MPI interface error codes	Offset 0x8500 + MPI Response Message Fehlercode
0x00008501	34049	NLINK_COMMERR_MPI_UE	Timeout from remote station remote station remote station has not responded within 1 sec.timeout
0x00008502	34050	NLINK_COMMERR_MPI_RR	Resource unavailable remote station remote station has no left buffer space for the requested service
0x00008503	34051	NLINK_COMMERR_MPI_RS	Requested function of master is not activated within the remote station. The connection seems to be closed in the remote station. Try to send command again
0x00008511	34065	NLINK_COMMERR_MPI_NA	No response of the remote station remote station check network wiring, check remote address, check baud rate
0x00008512	34066	NLINK_COMMERR_MPI_DS	Master not into the logical token ring network in general check master DP-Address or highest-station-Address of other masters. Examine bus wiring to bus short circuits
0x00008514	34068	NLINK_COMMERR_MPI_LR	Resource of the local FDL controller not available or not sufficient. HOST too many messages. no more segments in DEVICE free
0x00008515	34069	NLINK_COMMERR_MPI_IV	The specified msg.data_cnt parameter invalid HOST check the limit of 222 bytes (read) respectively 216 bytes (write) in msg.data_cnt
0x00008530	34096	NLINK_COMMERR_MPI_TO	Timeout, the request message was accepted but no indication is sent back by the remote station remote station MPI protocol error, or station not presenter
0x00008539	34105	NLINK_COMMERR_MPI_SE	Sequence fault, internal state machine error. Remote station does not react like awaited or a reconnection was retried while connection is already

Fehlercode (Hex)	Fehlercode (Dez)	Symbolische Konstante	Beschreibung
			open or device has no SAPs left to open connection channel
0x00008585	34181	NLINK_COMMERR_MPI_REJ_IV	Specified offset address out of limits or not known in the remote station HOST please check msg.data_adr if present or offset parameter in request message
0x00008586	34182	NLINK_COMMERR_MPI_REJ_PDU	Wrong PDU coding in the MPI response of the remote station DEVICE contact hotline
0x00008587	34183	NLINK_COMMERR_MPI_REJ_OP	Specified length to write or to read results in an access outside the limits HOST please check msg.data_cnt length in request message
0x00008588	34184	NLINK_COMMERR_MPI_REJ_OP	Specified address not defined in the remote station, please check msg.data_adr in the request message
0x00008589	34185	NLINK_COMMERR_MPI_REJ_OP	MPI remote station not in the right operational mode, bring S7 into RUN-P Mode

Voraussetzungen

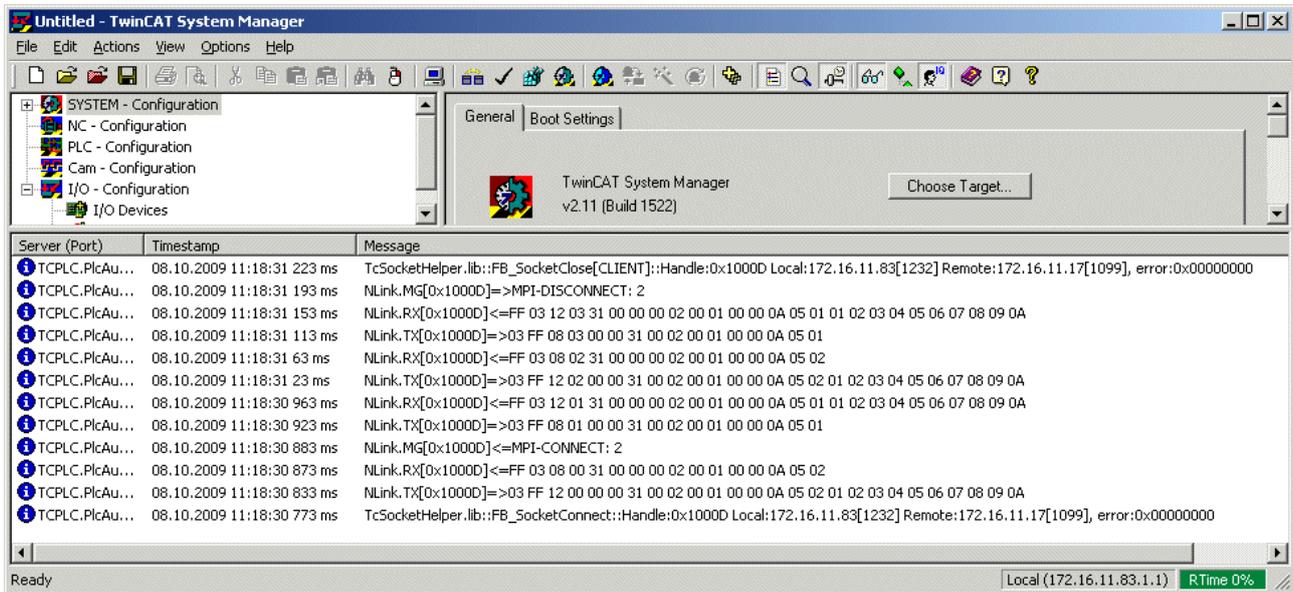
Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.11.0 Build > 1536	PC oder CX (x86, ARM)	TcS5S7Comm.Lib

9.2 Fehlersuche/Diagnose

- Bei Verbindungsproblemen kann der PING-Befehl dazu benutzt werden, um festzustellen ob die Fremdsteuerung über die Netzwerkverbindung erreichbar ist. Wenn dies nicht der Fall ist überprüfen Sie die Konfiguration/Stromversorgung und den Netzwerkanschluss des Adaptermoduls. Hinweise zur Konfiguration der Adaptermodule finden Sie in der Herstellerdokumentation des Adapters.
- Eine komplette Aufzeichnung der Netzwerkkommunikation kann mit Sniffer-Tools wie Wireshark durchgeführt werden. Die Aufnahme kann dann vom Beckhoff-Supportpersonal analysiert werden.
- Überprüfen Sie die in dieser Dokumentation beschriebenen Hardware- und Softwareanforderungen (Adapter-Hersteller, -Model, -Firmwareversion).
- Wenn Sie über den IBH Link S5++ oder IBH Link S7++ Adapter in Verbindung mit TCP/IP als Transportprotokoll auf die Fremdsteuerung zugreifen dann müssen in der Fremdsteuerung keinerlei Verbindungen konfiguriert/parametriert werden! Wenn Sie trotzdem Probleme beim Datenzugriff haben, dann überprüfen Sie ob die entsprechenden Datenbereiche (Merker, Datenbausteine usw.) in der Fremdsteuerung richtig konfiguriert und vorhanden sind.
- Überprüfen Sie ob die Eingangsparameter, die Sie an die Funktionsbausteine übergeben, richtig sind (Netzwerkadresse, MPI-Adresse, Datenbausteinnummer usw..). Überprüfen Sie ob der Funktionsbaustein einen Fehlercode ausgibt. Die Dokumentation zu den Fehlercodes finden Sie hier: [Übersicht der Fehlercodes](#) [► 24].
- In der TwinCAT SPS-Bibliothek ist bereits eine einfache Debugausgabe integriert. Diese muss vor dem Herstellen der Verbindung am FB NLinkOpen [► 14]-Funktionsbaustein aktiviert werden. Öffnen Sie den TwinCAT System Manager und aktivieren das LogView-Fenster. Auf folgendem Bild sehen Sie einige Debug-Ausgabestrings.

Beispiel:

Kommunikation über den IBH Link S7++ (NL-MPI-Protokoll über TCP/IP). Es werden immer komplette Frames geloggt (16 Byte header + SPS-Daten aus dem entsprechenden Datenbereich). D.h. ab Byte 16 folgen die gelesenen bzw. geschriebenen SPS-Daten.



Einige Erklärungen zu den Meldungen:

Nr	Meldung	Bedeutung
12	TcSocketHelper.lib::FB_SocketClose[CLIENT]::Handle:0x1000D Local:172.16.11.83[1232] Remote:172.16.11.17[1099], error:0x00000000	TCP/IP-Verbindung (internes Handle: 0x100D) wird Fehlerfrei geschlossen.
11	NLink.MG[0x1000D]>=>MPI-DISCONNECT: 2	Die interne MPI-Verbindung zum Gerät mit der MPI-Adresse 2 wird geschlossen.
10	NLink.RX[0x1000D]<=FF 03 12 03 31 00 00 00 02 00 01 00 00 0A 05 01 01 02 03 04 05 06 07 08 09 0A	READ-Response wurde empfangen...
9	NLink.TX[0x1000D]>=>03 FF 08 03 00 00 31 00 02 00 01 00 00 0A 05 01	READ-Request wurde gesendet...
8	NLink.RX[0x1000D]<=FF 03 08 02 31 00 00 00 02 00 01 00 00 0A 05 02	WRITE-Response wurde empfangen...
7	NLink.TX[0x1000D]>=>03 FF 12 02 00 00 31 00 02 00 01 00 00 0A 05 02 01 02 03 04 05 06 07 08 09 0A	WRITE-Request wurde gesendet...
6	NLink.RX[0x1000D]<=FF 03 12 01 31 00 00 00 02 00 01 00 00 0A 05 01 01 02 03 04 05 06 07 08 09 0A	READ-Response wurde empfangen (10 Byte: 01 02 03 04 05 06 07 08 09 0A erfolgreich empfangen).
5	NLink.TX[0x1000D]>=>03 FF 08 01 00 00 31 00 02 00 01 00 00 0A 05 01	READ-Request wurde gesendet.
4	NLink.MG[0x1000D]<=MPI-CONNECT: 2	Die interne MPI-Verbindung zum Gerät mit der MPI-Adresse 2 wird geöffnet.
3	NLink.RX[0x1000D]<=FF 03 08 00 31 00 00 00 02 00 01 00 00 0A 05 02	WRITE-Response wurde empfangen (nur Quittierung).
2	NLink.TX[0x1000D]>=>03 FF 12 00 00 00 31 00 02 00 01 00 00 0A 05 02 01 02 03 04 05 06 07 08 09 0A	WRITE-Request wurde gesendet (10 Byte: 01 02 03 04 05 06 07 08 09 0A werden in die Fremdsteuerung geschrieben).
1	TcSocketHelper.lib::FB_SocketConnect::Handle:0x1000D Local:172.16.11.83[1232] Remote:172.16.11.17[1099], error:0x00000000	TCP/IP-Verbindung (internes Handle: 0x100D) wird Fehlerfrei hergestellt.

10 Beispiele

Die Kommunikationsparameter in den Beispielprojekten müssen passend zu Ihrer Konfiguration eingestellt werden.

SPS Project	Fremdsystem	Adapter	Kommunikationsparameter	Datenbereich-Parameterwerte	Beschreibung
https://infosys.beckhoff.com/content/1031/tcplclibs5s7com/Resourcen/11426458507.zip	S5	Ethernet-Adapter: IBH Link S5++	<ul style="list-style-type: none"> • Protokoll: IBH S5 Lan Link Protokoll • Transport: TCP/IP • IP-Adresse: '172.16.11.16' • Portnummer: 2002 	<ul style="list-style-type: none"> • Datenbereich: DB • Datenbaustein nummer: 16 • 1 Wort (2 Bbyte) schreiben ab Startadresse (Wortoffset): 9 • 10 Worte (20 Byte) lesen ab Startadresse (Wortoffset): 0 	Es wird auf die Daten vom Datenbaustein Wortweise zugegriffen.
https://infosys.beckhoff.com/content/1031/tcplclibs5s7com/Resourcen/11426459915.zip	S7	Ethernet-Adapter: IBH Link S7++	<ul style="list-style-type: none"> • Protokoll: NetLink MPI-Protokoll (NL-MPI) • Transport: TCP/IP • IP-Adresse: '172.16.11.17' • Portnummer: 1099 • MPI-Adresse: 2 	<ul style="list-style-type: none"> • Datenbereich: DB • Datenbaustein nummer: 1 • 10 Byte schreiben ab Startadresse (Byteoffset): 0 • 10 Byte lesen ab Startadresse (Byteoffset): 0 	Es wird auf die Daten vom Datenbaustein Byteweise zugegriffen. Zusätzlich zu dem angewählten Datenbereich wird auch der Betriebsstatus der Steuerung ausgelesen (STOP, RUN, START...).

Mehr Informationen:

www.beckhoff.com/de-de/produkte/automation/twincat/tsxxx-twincat-2-supplements/

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

