

Handbuch | DE

TS6300

TwinCAT 2 FTP Client

Supplement | 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	Voraussetzungen	9
4	Installation	10
5	Installation Windows CE	13
6	Verbindungsart	15
7	SPS-Bibliothek	16
7.1	Funktionsbausteine	17
7.1.1	FB_FTP_HostResolve	17
7.1.2	FB_FTP_Open	18
7.1.3	FB_FTP_OpenEx.....	19
7.1.4	FB_FTP_Close.....	20
7.1.5	FB_FTP_CloseAll.....	21
7.1.6	FB_FTP_Info.....	21
7.1.7	FB_FTP_FileUpload.....	22
7.1.8	FB_FTP_FileUploadEx	23
7.1.9	FB_FTP_FileDownload	24
7.1.10	FB_FTP_FileDownloadEx.....	25
7.1.11	FB_FTP_DirCreate	26
7.1.12	FB_FTP_DirRemove.....	27
7.1.13	FB_FTP_FileList	28
7.1.14	FB_FTP_FileListEx	29
7.1.15	FB_FTP_FileExist	31
7.1.16	FB_FTP_FileRemove.....	32
7.1.17	FB_FTP_FileRename	33
7.1.18	FB_GetStateTcFTPClient	34
7.2	Funktionen	35
7.2.1	F_GetVersionTcFTPClient.....	35
7.3	Datentypen.....	35
7.3.1	T_HFTP.....	35
7.3.2	ST_FTP_ConnInfo	36
7.3.3	ST_FTP_FileDetails	36
7.3.4	E_FTP_ConnMode	37
7.4	Konstanten	37
7.4.1	Globale Variablen.....	37
8	Beispiele	38
8.1	TwinCAT FTP Client: Hochladen einer Datei auf einen FTP Server.....	38
8.2	TwinCAT FTP Client: Herunterladen einer Datei von einem FTP Server auf ein CX-Gerät	38
8.3	TwinCAT FTP Client: Auslesen der Verbindungsinformationen mit FB_FTP_Info	40

8.4 TwinCAT FTP Client: Löschen einer Datei vom FTP Server 40

8.5 TwinCAT FTP Client: Auslesen einer Dateiliste von einem FTP Server 42

9 Anhang 44

9.1 Erzeugen einer Fehlerloggdatei 44

9.2 Einstellen des Portbereiches 45

9.3 Return Codes 45

9.3.1 Übersicht der Fehlercodes des TwinCAT FTP Client 45

9.3.2 ADS Return Codes 46

9.3.3 FTP Client Return Codes 49

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® 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

Der TwinCAT FTP Client ist ein Supplement und bietet die Möglichkeit, aus dem TwinCAT System auf einfache und unkomplizierte Art und Weise mit einem FTP Server zu kommunizieren.

Verschiedene Funktionen stehen zur Verfügung:

Verbindungsaufbau:

- Mit Authentifizierung

Übertragungsfunktionen:

- Hochladen von Dateien
- Herunterladen von Dateien

Dateifunktionen:

- Umbenennen von Dateien
- Löschen von Dateien
- Suchen nach Dateien
- Verzeichnisse erstellen
- Verzeichnisse löschen

Alle Funktionalitäten können mit Hilfe von Funktionsblöcken aus der SPS heraus angestoßen werden. Es können mehrere Verbindungen zu unterschiedlichen Servern aufgebaut werden, die mit so genannten Handles unterschieden und benutzt werden können. So müssen Sie nicht bei jedem Funktionsaufruf die Authentifizierungsdaten mit angeben.

3 Voraussetzungen

Der TwinCAT FTP Client basiert auf einer Implementation die das .NET2.0 Framework (bzw. für CE auf dem Compact Framework) benötigt.

Das Produkt ist somit auf alle Plattformen lauffähig, welches folgende Komponenten bietet:

- Minimum TwinCAT PLC
- Microsoft .NET Framework 2.0 (oder höher) bzw. Microsoft .NET Compact Framework 2.0 (oder höher)
Fast alle Beckhoff CE Geräte sind mit dem .NET Compact Framework per Default ausgestattet

Beispiel:

- Der Beckhoff CX9001 liefert das .NET CF 2.0
- Der Beckhoff CX9000 bietet zwar TwinCAT aber kein .NET - somit ist der FTP-Client hier nicht lauffähig

4 Installation

Dieser Teil der Dokumentation führt den Benutzer Schritt-für-Schritt durch den Installationsvorgang des TwinCAT FTP-Client Supplements für Windows XP basierte Betriebssysteme. Es wird hierbei auf die folgenden Themen eingegangen:

- Herunterladen der Setup-Datei
- Starten der Installation

Herunterladen der Setup-Datei

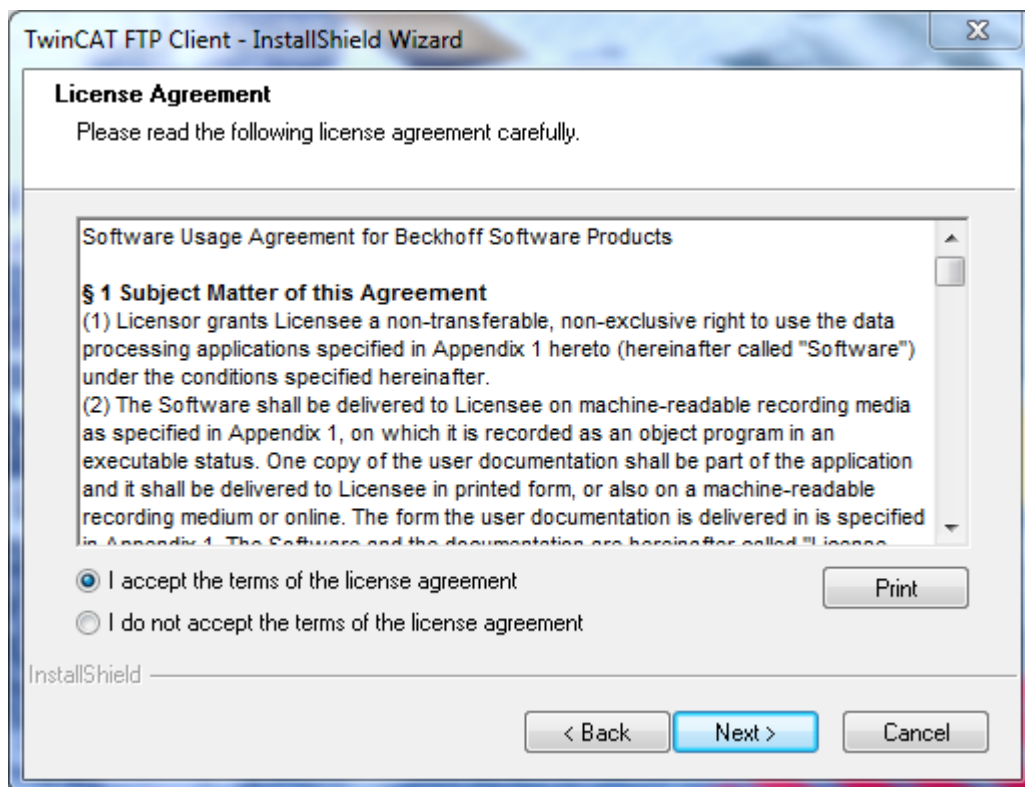
Wie viele andere TwinCAT Supplement-Produkte auch, steht TwinCAT FTP-Client als Download auf dem Beckhoff FTP-Server zur Verfügung. Es handelt sich hierbei um die jeweils aktuellste Version des Produkts. Führen Sie die folgenden Schritte durch, um die Setup-Datei zu downloaden:

1. Öffnen Sie eine FTP-Client Software Ihrer Wahl, z.B. FileZilla oder den Total Commander.
2. Öffnen Sie eine Verbindung zu [.TS6300 | TwinCAT FTP Client](#)
3. Wählen Sie TS6300 TwinCAT FTP Client und starten Sie den Download über den Warenkorb.
4. (Optional) Übertragen Sie die Datei auf das TwinCAT-Laufzeitsystem, auf welchem Sie das Supplement installieren möchten.

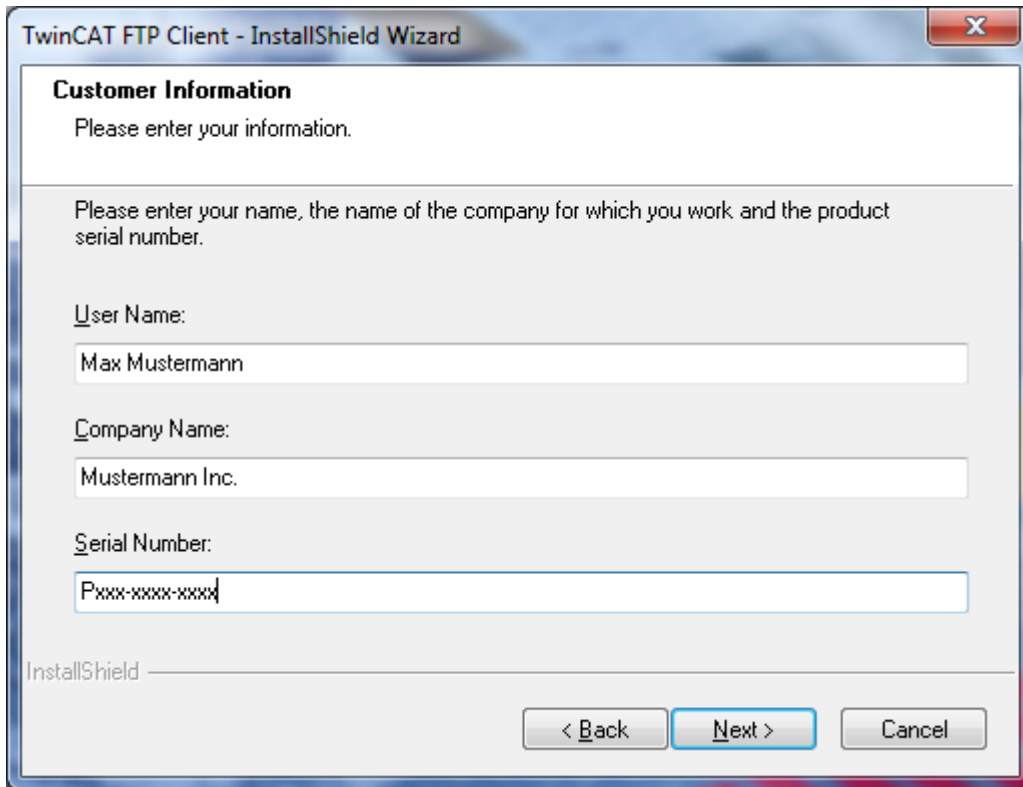
Starten der Installation

Um das Supplement zu installieren, führen Sie die folgenden Schritte durch:

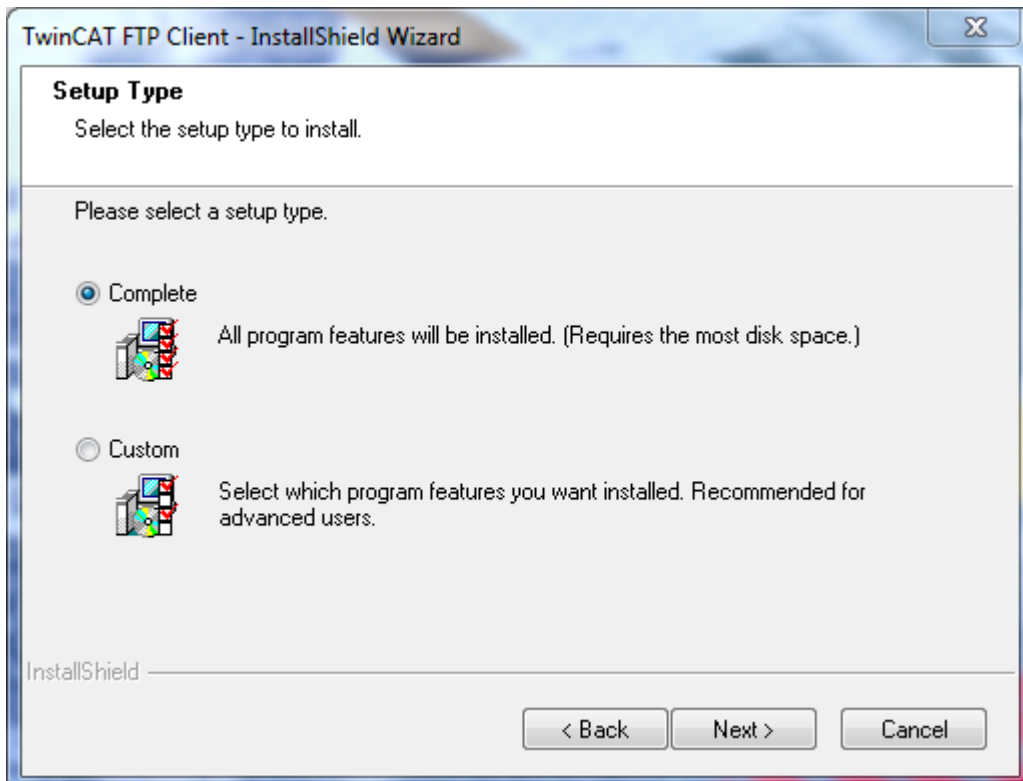
1. Führen Sie einen Doppelklick auf die heruntergeladene Datei aus.
Starten Sie die Installation unter Windows 7 32-bit/64-bit per "Als Administrator ausführen", indem Sie die Setup-Datei mit der rechten Maustaste anklicken und die entsprechende Option im Kontextmenü auswählen.
2. Wählen Sie eine **Sprache** aus, in der Sie die Software installieren möchten.
3. Klicken Sie auf "Next" und akzeptieren Sie dann die **Endbenutzervereinbarung**.



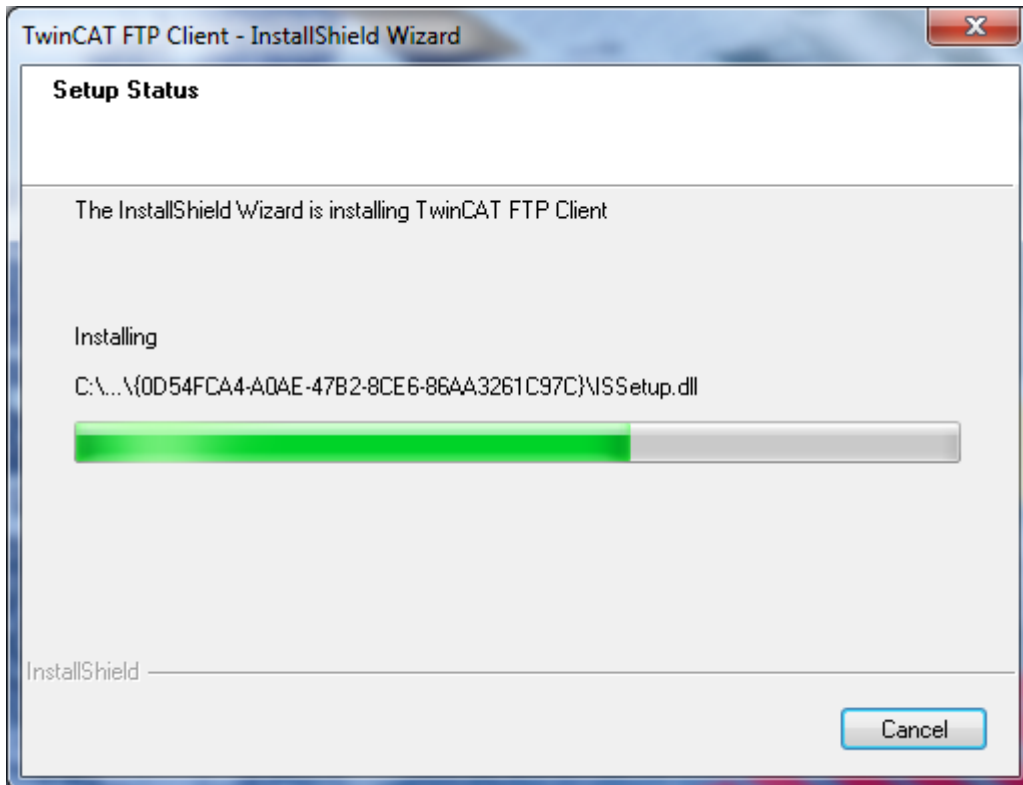
4. Geben Sie Ihre **Benutzerdaten** ein. Alle sichtbaren Felder sind hierbei Pflichtfelder.



5. Wählen Sie **Complete** und klicken Sie auf **Next**.



6. Klicken Sie auf **Install** um die Installation zu starten



7. Zum Abschluss der Installation starten Sie Ihren Computer neu.

5 Installation Windows CE

Die Installation unter Windows CE besteht aus insgesamt 2 Schritten:

Schritt 1: Installation der TwinCAT FTP-Client Supplement Software auf einem Host-PC.

Dies wird in einer separaten Installationsanleitung beschrieben und soll auf dieser Seite nicht weiter behandelt werden.

Durch die Installation der Supplement-Software werden die benötigten Dateien für Windows CE automatisch in das TwinCAT-Unterverzeichnis **..\TwinCAT\CE\TwinCAT FTP Client CE** auf dem Host-PC installiert.

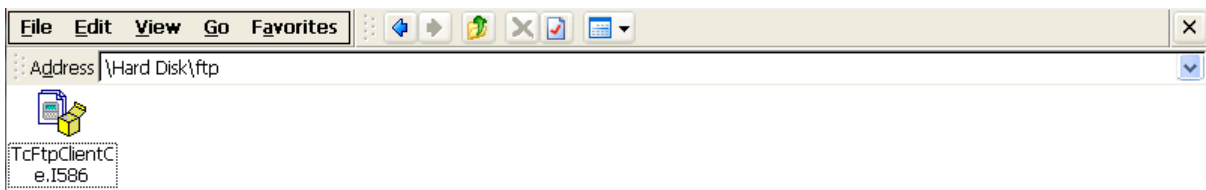
Schritt 2: Transfer der CAB-Dateien für Windows CE auf das CE-Gerät

Transferieren Sie die entsprechend benötigte CAB-Datei aus dem TwinCAT-Unterverzeichnis **..\TwinCAT\CE\TwinCAT FTP Client CE** auf das CE-Gerät (z.B. per MemoryStick/FTP/Public-Folder/CF-Adapter...). Welche CAB-Datei Sie benötigen hängt von der jeweiligen CPU-Architektur Ihres CE-Geräts ab:

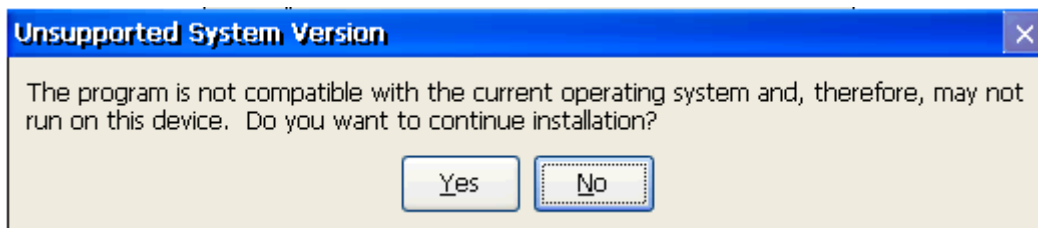
- **TcFtpClientCe.I586.cab**: TwinCAT FTP-Client für x86 basierte CPUs (wie z.B. CX10xx, CP62xx, C69xx, ...)
- **TcFtpClientCe.ARM.cab**: TwinCAT FTP-Client für ARM basierte CPUs (wie z.B. CX9001, CX9010, CP6608, ...)

Bitte führen Sie die folgenden Schritte durch, nachdem Sie die CAB-Datei auf dem Gerät übertragen haben:

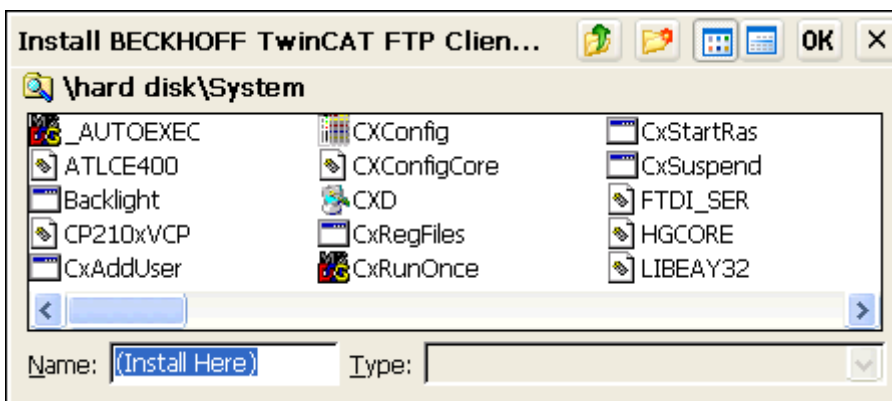
- Navigieren Sie zu dem Ordner, in den Sie die Installationsdatei übertragen haben



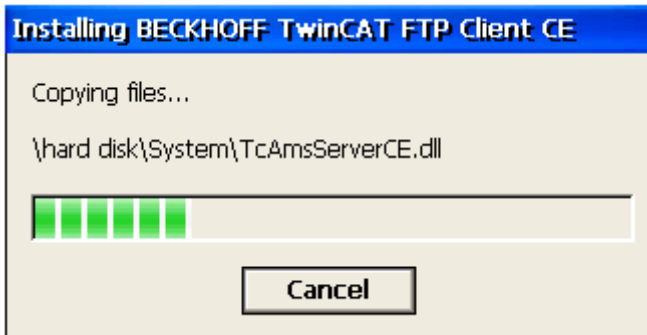
- Führen Sie einen Doppelklick auf die CAB-Datei aus. Sollten Sie eine MessageBox mit dem Hinweis "Program is not compatible with current operating system" erhalten, so überprüfen Sie, ob Sie die korrekte CAB-Datei (ARM, I586) für Ihre Plattform verwendet haben.
- Sind Sie sich sicher, dass die CAB-Datei korrekt ist, bestätigen Sie die Meldung mit **"Yes"**.



- Stellen Sie sicher, dass als Zielverzeichnis **"\Hard Disk\System"** ausgewählt ist



- Zum Starten der Installation klicken Sie in der rechten oberen Ecke auf **"Ok"**



Nach der Installation löscht sich die Installationsdatei automatisch.



Bitte beachten Sie, dass der TwinCAT FTP-Client erst nach dem nächsten Neustart des Systems verfügbar ist.



Bitte beachten Sie, dass der TcFTPClient unter Windows CE erst dann automatisch startet, wenn Sie die "StartUp.exe" ab der Version 1.35 verwenden. Andernfalls werden Sie bei jedem Aufruf den ADS-Fehler 6 zurückbekommen.

Installationsumfang auf dem CE Gerät:

- "\\Hard Disk\System"
 - TcFTPClientCe.exe
 - TcAmsServerCe.dll
 - TwinCAT.Ads.dll
 - TwinCAT.Ads.Server.dll

6 Verbindungsart

Zum Senden und Empfangen von Dateien sowie zur Übertragung von Verzeichnislisten (der Standard-Port dafür ist Port 20) wird pro Vorgang jeweils eine separate TCP-Verbindung verwendet. FTP kennt für den Aufbau solcher Verbindungen zwei Modi: "**Aktives FTP**" und "**Passives FTP**".

Der TwinCAT FTP Client unterstützt ab Version 1.0.8 beide Verbindungsmodi. Bei älteren Versionen wird nur "Passives FTP" unterstützt.

Aktives FTP

Beim aktiven FTP (auch „Active Mode“) öffnet der Client einen zufälligen Port und teilt dem Server diesen sowie die eigene IP-Adresse mittels des PORT-Kommandos mit. Dies ist typischerweise ein Port des Clients, der jenseits von 1023 liegt, kann aber auch ein anderer Server sein, der seinerseits in den Passive Mode geschaltet wurde, also auf eine Verbindung wartet (so genanntes FXP). Die Datenübertragung auf der Server-Seite erfolgt dabei über Port 20. Die Kommunikation mit Befehlen erfolgt ausschließlich auf dem Control Port. Man spricht auch von der Steuerung „Out of Band“. Somit bleibt es möglich, dass während der Datenübertragung die Partner noch immer miteinander kommunizieren können.

Passives FTP

Beim Passiven FTP (auch „Passive Mode“) sendet der Client ein PASV-Kommando, der Server öffnet einen Port und übermittelt diesen mitsamt IP-Adresse an den Client. Hier wird auf der Client-Seite ein Port jenseits 1023 verwendet und auf der Server-Seite der vorher an den Client übermittelte Port. Diese Technik wird eingesetzt, wenn der Client für den Server nicht erreichbar ist. Dies ist beispielsweise der Fall, wenn der Client sich hinter einem Router befindet, der die Adresse des Clients mittels NAT umschreibt, oder wenn eine Firewall das Netzwerk des Clients vor Zugriffen von außen abschirmt.

7 SPS-Bibliothek

Übersicht

Die TcFTPClient.lib Bibliothek beinhaltet Funktionsblöcke zum Steuern und Konfigurieren des TwinCAT FTP Clients.

Funktionsblöcke

Name	Beschreibung
FB_GetStateTcFTPClient [► 34]	Statusinformationen abrufen.
FB_FTP_HostResolve [► 17]	Löst den angegebenen Hostnamen auf und gibt die entsprechende IPv4 Adresse aus
FB_FTP_Open [► 18]	Öffnet eine Verbindung zu einem FTP Server (Passiv)
FB_FTP_OpenEx [► 19]	Öffnet eine Verbindung zu einem FTP Server (Passive / Aktiv)
FB_FTP_Close [► 20]	Schließt eine Verbindung zu einem FTP Server
FB_FTP_CloseAll [► 21]	Schließt alle Verbindungen zu einem FTP Server
FB_FTP_Info [► 21]	Gibt Informationen von den geöffneten Verbindungen zurück
FB_FTP_FileUpload [► 22]	Kopiert eine Quelldatei auf einen FTP Server über eine bereits geöffnete Verbindung
FB_FTP_FileUploadEx [► 23]	Kopiert eine Quelldatei auf einen FTP Server
FB_FTP_FileDownload [► 24]	Kopiert eine Quelldatei von einem FTP Server auf einen Zielrechner über eine bereits geöffnete Verbindung
FB_FTP_FileDownloadEx [► 25]	Kopiert eine Quelldatei von einem FTP Server
FB_FTP_DirCreate [► 26]	Erzeugt ein Verzeichnis auf einen FTP Server
FB_FTP_DirRemove [► 27]	Löscht ein Verzeichnis von einem FTP Server
FB_FTP_FileList [► 28]	Liest Datei- und Ordernamen vom FTP Server und gibt sie als Array zurück.
FB_FTP_FileListEx [► 29]	Liest Datei- und Ordernamen mit zusätzlichen Detailinformationen vom FTP Server und gibt sie als Array zurück.
FB_FTP_FileExist [► 31]	Überprüft ob eine bestimmte Datei auf dem FTP Server existiert
FB_FTP_FileRemove [► 32]	Löscht eine Datei vom FTP Server
FB_FTP_FileRename [► 33]	Benennt eine Datei auf dem FTP Server um

Funktionen

Name	Beschreibung
F_GetVersionTcFTPClient [► 35]	Versionsinformationen abrufen.

Datentypen

Name	Beschreibung
T_HFTP [► 35]	Handle des FTP Servers.

Name	Beschreibung
ST_FTP_ConnInfo [▶ 36]	Informationen einer bestehenden FTP Verbindung.
ST_FTP_FileDetails [▶ 36]	Dateiinformatoren zurückgeliefert vom FTP "LIST" Kommando.
E_FTP_ConnMode [▶ 37]	FTP Verbindungsarten (Passiv / Aktiv).

7.1 Funktionsbausteine

7.1.1 FB_FTP_HostResolve



Mit dem Funktionsbaustein FB_FTP_HostResolve können Hostnamen von FTP Servern aufgelöst werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId    := '';
  sHostname   : T_MaxString   := '';
  bExecute    : BOOL;
  tTimeout    : TIME          := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

sHostname : Hostname der in die dazugehörige IPv4 Adresse aufgelöst werden soll.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrID      : UDINT;
  sIPv4Addr   : T_IPv4Addr;
END_VAR
```

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

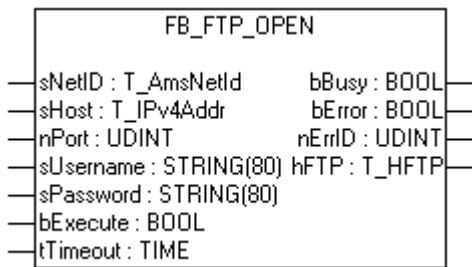
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [▶ 45].

sIPv4Addr : Liefert die passende IPv4 Adresse zum angegebenen Hostnamen zurück.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.2 FB_FTP_Open



Mit dem Funktionsbaustein FB_FTP_Open kann eine Verbindung zu einem FTP-Server hergestellt werden. Mit dem zurück gelieferten Handle können dann weitere Aktionen am FTP-Server durchgeführt werden. Die verwendete Verbindungsart ist Passives-FTP.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId      := '';
  sHost       : T_IPv4Addr      := '127.0.0.1';
  nPort       : UDINT           := 21;
  sUsername   : STRING          := '';
  sPassword   : STRING          := '';
  bExecute    : BOOL;
  tTimeout    : TIME           := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

sHost : Ist ein String, der die IPv4 Adresse des FTP-Server enthält.

nPort : FTP Port (default 21).

sUsername : Benutzername für FTP Server Authentifizierung

sPassword : Passwort für FTP Server Authentifizierung

bExecute : Mit der steigenden Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

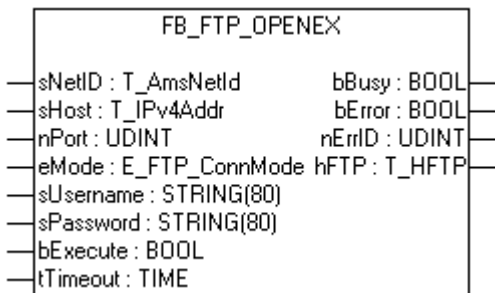
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

hFTP : Handle [► 35] der FTP-Verbindung

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.3 FB_FTP_OpenEx



Mit dem Funktionsbaustein FB_FTP_OpenEx kann eine Verbindung zu einem FTP-Server hergestellt werden. Mit dem zurück gelieferten Handle können dann weitere Aktionen am FTP-Server durchgeführt werden. Zusätzlich kann die Verbindungsart, Passives oder Aktives FTP, ausgewählt werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId      := '';
  sHost       : T_IPv4Addr      := '127.0.0.1';
  nPort       : UDINT           := 21;
  eMode       : E_FTP_ConnMode  := eConnMode_PASSIVE;
  sUsername   : STRING          := '';
  sPassword   : STRING          := '';
  bExecute    : BOOL;
  tTimeout    : TIME           := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

sHost : Ist ein String, der die IPv4 Adresse des FTP-Server enthält.

nPort : FTP Port (default 21).

eMode : FTP Verbindungsart [► 37] (Aktiv / Passiv).

sUsername : Benutzername für FTP Server Authentifizierung

sPassword : Passwort für FTP Server Authentifizierung

bExecute : Mit der steigenden Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

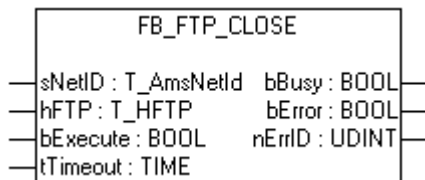
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

hFTP : Handle [► 35] der FTP-Verbindung

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib (Ab TcFTPClient Version 1.0.8)
TwinCAT v2.10.0	CX (ARM)	

7.1.4 FB_FTP_Close



Mit dem Funktionsbaustein FB_FTP_Close kann eine bestehende Verbindung zu einem FTP-Server geschlossen werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId   := '';
  hFTP        : T_HFTP;
  bExecute    : BOOL;
  tTimeout    : TIME        := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : Handle [► 35] des FTP Servers.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

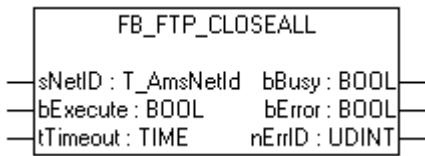
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.5 FB_FTP_CloseAll



Mit dem Funktionsbaustein FB_FTP_CloseAll können alle bestehenden Verbindungen zu einem FTP-Server geschlossen werden.

VAR_INPUT

```
VAR_INPUT
    sNetID      : T_AmsNetId    := '';
    bExecute    : BOOL;
    tTimeout    : TIME          := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

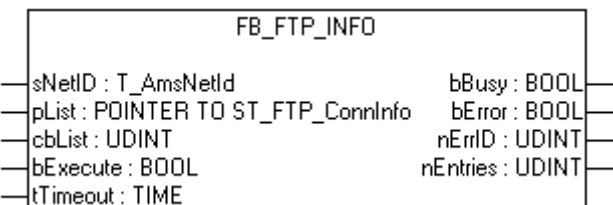
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.6 FB_FTP_Info



Mit dem Funktionsbaustein FB_FTP_Info können Informationen zu den erzeugten FTP-Server-Verbindungen abgerufen werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId      := '';
  pList       : POINTER TO ST_FTP_ConnInfo;
  cbList      : UDINT           := 0;
  bExecute    : BOOL;
  tTimeout    : TIME           := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

pList : Pointer Adresse zu einer Variablen vom Typ ST_FTP_ConnInfo [► 36]. Diese Variable kann auch ein Array vom Typ ST_FTPConnInfo sein.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeoutzeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

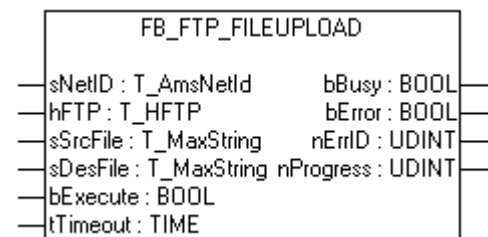
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

nEntries : Anzahl der bestehenden Verbindungen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.7 FB_FTP_FileUpload

Mit dem Funktionsbaustein FB_FTP_FileUpload können Dateien vom ADS-Device auf einen FTP-Server übermittelt werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId      := '';
  hFTP        : T_HFTP;
  sSrcFile    : T_MaxString     := '';
  sDesFile    : T_MaxString     := '';
  bExecute    : BOOL;
  tTimeout    : TIME           := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle \[► 35\]](#) des FTP Servers.

sSrcFile : Quelldatei die auf den FTP Server kopiert werden soll

sDesFile : Zieldateiname mit Pfadangabe

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den [ADS Error Code. \[► 46\]](#)

nProgress : Zeigt den aktuellen Status der Übermittlung in Prozent an.

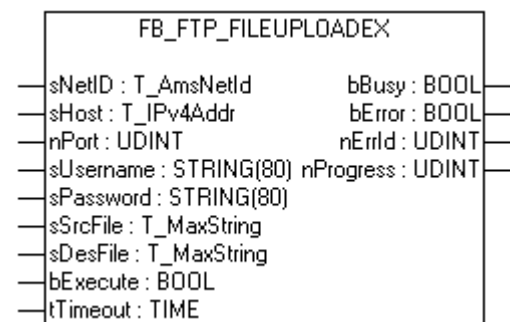
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Sehen Sie dazu auch

- Übersicht der Fehlercodes des TwinCAT FTP Client [\[► 45\]](#)

7.1.8 FB FTP_FileUploadEx



Mit dem Funktionsbaustein FB FTP_FileUploadEx können Dateien vom ADS-Device auf einen FTP-Server übermittelt werden. Die verwendete Verbindungsart ist Passives-FTP.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  sHost       : T_IPv4Addr := '127.0.0.1';
  nPort       : UDINT      := 21;
  sUsername   : STRING     := '';
  sPassword   : STRING     := '';
  sSrcFile    : T_MaxString := '';
```

```

sDesFile   : T_MaxString   := '';
bExecute   : BOOL;
tTimeout   : TIME         := T#15s;
END_VAR

```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

sHost : Ist ein String, der die IPv4 Adresse des FTP-Servers enthält.

nPort : FTP Port (default 21).

sUsername : Benutzername für FTP Server Authentifizierung

sPassword : Passwort für FTP Server Authentifizierung

sSrcFile : Quelldatei die auf den FTP Server kopiert werden soll

sDesFile : Zieldateiname mit Pfadangabe

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

```

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

```

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

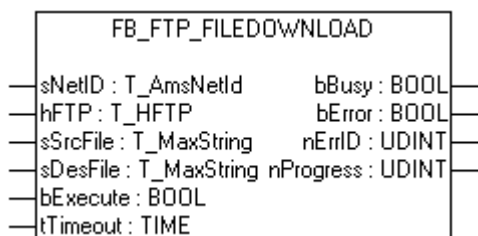
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

nProgress : Zeigt den aktuellen Status der Übermittlung in Prozent an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.9 FB_FTP_FileDownload



Mit dem Funktionsbaustein FB_FTP_FileDownload können Dateien vom FTP-Server auf ein ADS-Device übermittelt werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP Server.

VAR_INPUT

```

VAR_INPUT
  sNetID     : T_AmsNetId   := '';
  hFTP       : T_HFTP
  sSrcFile   : T_MaxString   := '';

```



```
sDesFile      : T_MaxString    := '';
bExecute     : BOOL;
tTimeout     : TIME           := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : Handle [► 35] des FTP Servers.

sSrcFile : Quelldatei, die vom FTP Server kopiert werden soll

sDesFile : Zieldateiname mit Pfadangabe

bExecute : Mit der steigenden Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

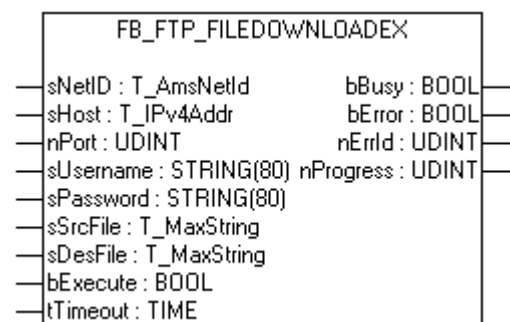
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

nProgress : Zeigt den aktuellen Status der Übermittlung in Prozent an. Bei **CE FTP Server** werden keine Zwischenwerte zurückgeliefert nur 0% und 100%.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.10 FB FTP_FileDownloadEx



Mit dem Funktionsbaustein FB FTP_FileDownloadEx können Dateien von einem FTP-Server auf ein ADS-Device übermittelt werden. Die verwendete Verbindungsart ist Passives-FTP.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId    := '';
  sHost       : T_IPv4Addr    := '127.0.0.1';
  nPort       : UDINT         := 21;
  sUsername   : STRING        := '';
```

```

sPassword  : STRING      := '';
sSrcFile   : T_MaxString := '';
sDesFile   : T_MaxString := '';
bExecute   : BOOL;
tTimeout   : TIME       := T#15s;
END_VAR

```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

sHost : Ist ein String, der die IPv4 Adresse des FTP-Server enthält.

nPort : FTP Port (default 21).

sUsername : Benutzername für FTP Server Authentifizierung

sPassword : Passwort für FTP Server Authentifizierung

sSrcFile : Quelldatei, die vom FTP Server kopiert werden soll

sDesFile : Zielfeldname mit Pfadangabe

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

```

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

```

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

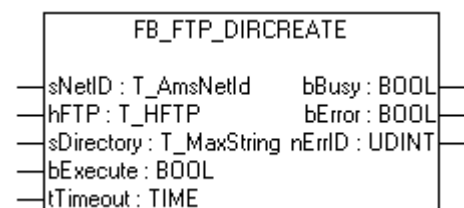
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

nProgress : Zeigt den aktuellen Status der Übermittlung in Prozent an. Bei **CE FTP Server** werden keine Zwischenwerte zurückgeliefert nur 0% und 100%.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.11 FB_FTP_DirCreate



Mit dem Funktionsbaustein FB_FTP_DirCreate können Verzeichnisse auf einen FTP-Server erstellt werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId   := '';
  hFTP        : T_HFTP       := 0;
  sDirName    : T_MaxString  := '';
  bExecute    : BOOL;
  tTimeout    : TIME         := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle](#) [► 35] des FTP Servers.

sDirectory : Der Name des zu erstellenden Verzeichnisses.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

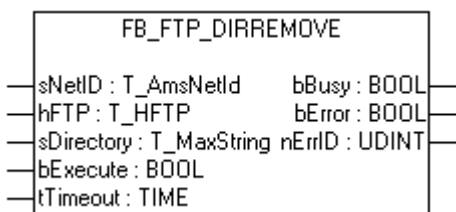
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen [Fehlercode](#) [► 45].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.12 FB_FTP_DirRemove



Mit dem Funktionsbaustein FB_FTP_DirRemove können Verzeichnisse vom FTP-Server gelöscht werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId   := '';
  hFTP        : T_HFTP;
  sDirectory  : T_MaxString  := '';
  bExecute    : BOOL;
  tTimeout    : TIME         := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle \[▶ 35\]](#) des FTP Servers.

sDirectory : Der Name des Verzeichnisses, welches gelöscht werden soll.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

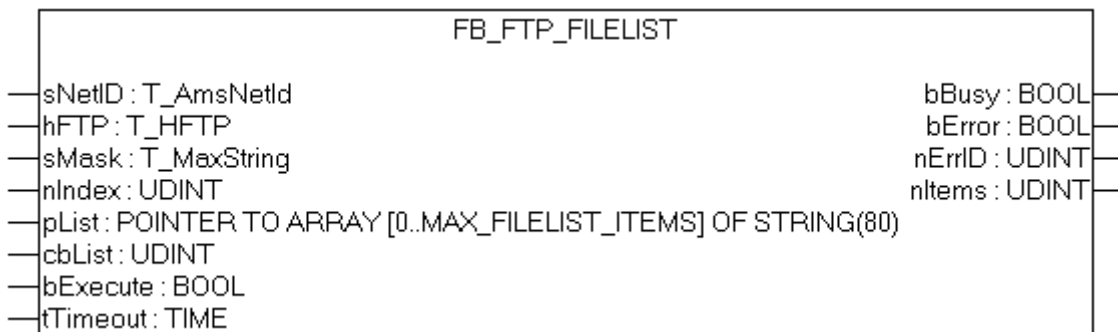
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [\[▶ 45\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.13 FB FTP_FileList



Mit dem Funktionsbaustein FB FTP_FileList können Datei- und Ordernamen vom FTP-Server gelesen bzw. gesucht werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP       : T_HFTP;
  sMask      : T_MaxString := '';
  nIndex     : UDINT;
  pList      : POINTER TO ARRAY [0..MAX_FILELIST_ITEMS] OF STRING(80);
  cbList     : UDINT;
  bExecute   : BOOL;
  tTimeout   : TIME := T#20s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle \[▶ 35\]](#) des FTP Servers.

sMask : Suchmaske zum Filtern entsprechender Dateien.

Beispiel	
.	Liest alle Dateien aus
*	Liest alle Dateien und alle Ordernamen aus
*.txt	Liest alle Dateien mit der Endung "TXT" aus
\\Test*.*	Liest alle Dateien aus dem Unterordner Test aus

nIndex : Gibt den Index des ersten zu lesenden Dateinamens an

pList : Gibt die Adresse des Stringarrays an in das die Dateinamen geschrieben werden soll

cbList : Gibt die Größe des Stringarrays in Byte an

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den ADS Error Code. [► 46]

nItems : Liefert die Gesamtanzahl der Dateilisteinträge zurück.

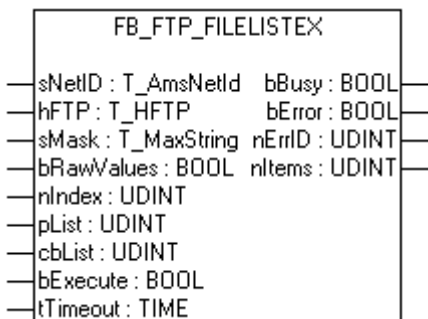
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Sehen Sie dazu auch

- Übersicht der Fehlercodes des TwinCAT FTP Client [► 45]

7.1.14 FB FTP FileListEx



Mit dem Funktionsbaustein FB FTP FileListEx können Datei- und Ordernamen vom FTP-Server gelesen bzw. gesucht werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server. Im Gegensatz zu dem Baustein FB FTP FileList werden auch Details zu den Dateien/Ordern zurückgeliefert.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId := '';
  hFTP        : T_HFTP;
  sMask       : T_MaxString := '';
  bRawValues  : BOOL;
  nIndex     : UDINT;
  pList       : UDINT;
  cbList      : UDINT;
  bExecute    : BOOL;
  tTimeout    : TIME := T#20s;
END_VAR

```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle \[► 35\]](#) des FTP Servers.

sMask : Suchmaske zum Filtern entsprechender Dateien.

Beispiel	
**	Liest alle Dateien aus
*	Liest alle Dateien und alle Ordernamen aus
*.txt	Liest alle Dateien mit der Endung "TXT" aus
\Test*.*	Liest alle Dateien aus dem Unterordner Test aus

bRawValues : Gibt an in welcher Form die Dateiliste zurückgeliefert werden soll.

bRawValues := TRUE Informationen werden in Form eines T_MaxString Arrays zurückgeliefert.

bRawValues := FALSE Informationen werden in Form eines [ST FTP FileDetails \[► 36\]](#) Arrays zurückgeliefert. (Wird nicht von jedem FTP Server unterstützt)

nIndex : Gibt den Index der ersten zu lesenden Datei an

pList : Gibt die Adresse des Arrays an in das die Dateiliste geschrieben werden soll. Maximale Arraygröße 256 Elemente.

cbList : Gibt die Größe des Arrays in Byte an

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

```

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

```

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den [ADS Error Code. \[► 46\]](#)

nItems : Liefert die Gesamtanzahl der Dateilisteinträge zurück.

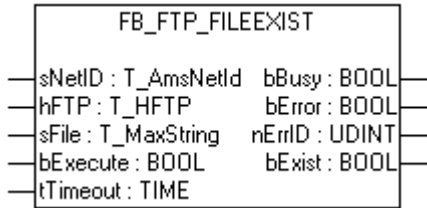
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

Sehen Sie dazu auch

Übersicht der Fehlercodes des TwinCAT FTP Client [▶ 45]

7.1.15 FB FTP_FileExist



Mit dem Funktionsbaustein FB FTP_FileExist kann überprüft werden ob eine bestimmte Datei auf dem FTP-Server vorhanden ist. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetId   := '';
  hFTP        : T_HFTP;
  sFile       : T_MaxString  := '';
  bExecute    : BOOL;
  tTimeout    : TIME        := T#15s;
END_VAR
  
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : Handle [▶ 35] des FTP Servers.

sFile : Dateiname nach dem gesucht wird.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout Zeit an.

VAR_OUTPUT

```

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

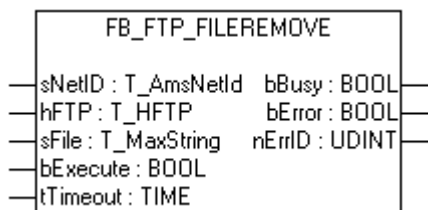
nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [▶ 45].

bExist : Wird TRUE, wenn die angegebene Datei gefunden wurde.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.16 FB_FTP_FileRemove



Mit dem Funktionsbaustein FB_FTP_FileRemove können Dateien auf einem FTP-Server gelöscht werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId   := '';
  hFTP        : T_HFTP;
  sFile       : T_MaxString  := '';
  bExecute    : BOOL;
  tTimeout    : TIME        := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle \[► 35\]](#) des FTP Servers.

sFile : Der Name der Datei, welche gelöscht werden soll.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

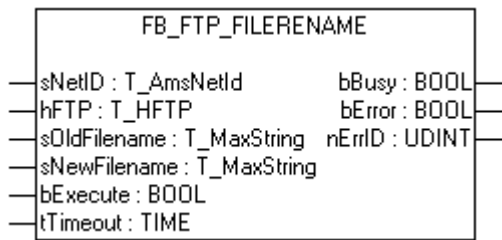
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen [Fehlercode \[► 45\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.17 FB_FTP_FileRename



Mit dem Funktionsbaustein FB_FTP_FileRename können Dateien auf einem FTP-Server umbenannt werden. Dieser Baustein benutzt eine bereits geöffnete Verbindung zu einem FTP-Server.

VAR_INPUT

```
VAR_INPUT
    sNetID      : T_AmsNetId    := '';
    hFTP        : T_HFTP;
    sOldFilename : T_MaxString  := '';
    sNewFilename : T_MaxString  := '';
    bExecute    : BOOL;
    tTimeout    : TIME         := T#15s;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

hFTP : [Handle](#) [► 35] des FTP Servers.

sOldFilename : Der alte Dateiname der zu ändernden Datei.

sNewFilename : Der neue Dateiname.

bExecute : Mit der steigende Flanke wird der Funktionsblock ausgeführt.

tTimeout : Gibt die Timeout-Zeit an.

VAR_OUTPUT

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

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

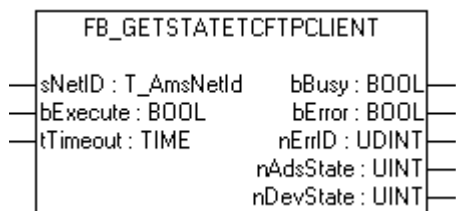
bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen [Fehlercode](#) [► 45].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.1.18 FB_GetStateTcFTPClient



Mit dem Funktionsbaustein FB_GetStateTcFTPClient kann der aktuelle Status des FTP Clients abgefragt werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  bExecute    : BOOL;
  tTimeout    : TIME;
END_VAR
```

sNetID : Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.

bExecute : Mit der steigende Flanke wird das Kommando ausgeführt.

tTimeout : Gibt die Timeoutzeit an.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrID      : UDINT;
  nAdsState   : UINT;
  nDevState   : UINT;
END_VAR
```

bBusy : Kommando wird gerade per ADS übertragen. Solange bBusy auf TRUE, wird kein neues Kommando angenommen.

bError : Wird TRUE, sobald ein Fehler eintritt.

nErrID : Liefert bei einem gesetzten bError-Ausgang den dazugehörigen Fehlercode [► 45].

nAdsState : Enthält die Zustandskennzahl des ADS-Zielgerätes. Die hier zurück gelieferten Codes sind festgelegt für alle ADS-Server:

- ADSSTATE_INVALID =0 ;
- ADSSTATE_IDLE =1 ;
- ADSSTATE_RESET =2 ;
- ADSSTATE_INIT =3 ;
- ADSSTATE_START =4 ;
- ADSSTATE_RUN =5 ;
- ADSSTATE_STOP =6 ;
- ADSSTATE_SAVECFG =7 ;
- ADSSTATE_LOADCFG =8 ;
- ADSSTATE_POWERFAILURE =9 ;
- ADSSTATE_POWERGOOD =10 ;
- ADSSTATE_ERROR =11;

nDevState : Enthält die spezifische Zustandskennzahl des ADS-Zielgerätes. Die hier zurück gelieferten Codes sind Zusatzinformationen, die für das ADS-Gerät spezifisch sind.

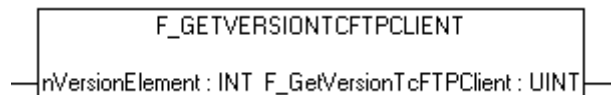
- 1 = TwinCAT FTP Client gestartet

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.2 Funktionen

7.2.1 F_GetVersionTcFTPClient



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

FUNCTION F_GetVersionTcFTPClient: 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 v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3 Datentypen

7.3.1 T_HFTP

VAR_INPUT

```
TYPE T_HFTP :
STRUCT
    hClient      :UDINT;
END_STRUCT
END_TYPE
```

hClient : Gibt den Handle der FTP Verbindung an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.2 ST_FTP_ConnInfo

VAR_INPUT

```

TYPE ST_FTP_ConnInfo :
STRUCT
  sHost      : T_IPv4Addr;
  nPort      : UDINT;
  hFTP       : T_HFTP;
  sUsername  : STRING;
END_STRUCT
END_TYPE

```

sHost : Gibt die IP-Adresse des FTP Servers an.

nPort : Gibt den Port der FTP Verbindung an.

hFTP : Gibt den Handle der FTP Verbindung an.

sUsername : Gibt den Namen des angemeldeten Benutzers an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.3 ST_FTP_FileDetails

VAR_INPUT

```

TYPE ST_FTP_FileDetails:
STRUCT
  bDir      : BOOL;
  sPermission : STRING(10);
  nSize     : UDINT;
  nFilecode : UDINT;
  tTimestamp : DT;
  sOwner    : STRING(79);
  sGroup    : STRING(79);
  sFilename : STRING(79);
END_STRUCT
END_TYPE

```

bDir : Gibt an ob es sich um eine Datei oder ein Verzeichnis handelt. (wird nicht von jedem FTP Server unterstützt)

sPermission : Gibt die Berechtigungsparameter der Datei/Verzeichnis an. (wird nicht von jedem FTP Server unterstützt)

nSize : Gibt die Größe der Datei an

nFilecode : Gibt den Dateicode zurück. (wird nicht von jedem FTP Server unterstützt)

tTimestamp : Gibt den Zeitstempel der Datei an.

sOwner : Gibt den Benutzer zurück. (wird nicht von jedem FTP Server unterstützt)

sGroup : Gibt die zugehörige Gruppe an. (wird nicht von jedem FTP Server unterstützt)

sFilename : Gibt den Namen der Datei/Verzeichnis an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

7.3.4 E_FTP_ConnMode

```

TYPE E_FTP_ConnMode : (
    eConnMode_PASSIVE := 0,
    eConnMode_ACTIVE := 1
);
END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib (Ab TcFTPClient Version 1.0.8)
TwinCAT v2.10.0	CX (ARM)	

7.4 Konstanten

7.4.1 Globale Variablen

```

VAR_GLOBAL CONSTANT

    AMSPORT_FTPADSSRV      : UDINT      := 10900;

    FTPADS_IGR_CONNOOPEN   : UDINT      := 16#100;
    FTPADS_IGR_RESOLVEDNS  : UDINT      := 16#101;
    FTPADS_IGR_CONNCLOSE   : UDINT      := 16#200;
    FTPADS_IGR_CONNCLOSEALL : UDINT      := 16#201;

    FTPADS_IGR_CONNINFO    : UDINT      := 16#300;

    FTPADS_IGR_FILEUPLOAD   : UDINT      := 16#10000;
    FTPADS_IGR_FILEDOWNLOAD : UDINT      := 16#20000;

    FTPADS_IGR_DIRREMOVE   : UDINT      := 16#30000;
    FTPADS_IGR_DIRCREATE    : UDINT      := 16#30001;

    FTPADS_IGR_FILEEXIST    : UDINT      := 16#40000;
    FTPADS_IGR_FILERENAME   : UDINT      := 16#40001;
    FTPADS_IGR_FILEREMOVE   : UDINT      := 16#40002;
    FTPADS_IGR_FILELIST     : UDINT      := 16#40003;
    FTPADS_IGR_FILELISTEX   : UDINT      := 16#40004;

    MAX_FTP_CONNECTIONS    : UDINT      := 25;
    MAX_FILELIST_ITEMS     : UDINT      := 255;

    DEFAULT_FTP_PORT        : UDINT      := 21;

END_VAR
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.10.0	PC oder CX (x86)	TcFTPClient.Lib
TwinCAT v2.10.0	CX (ARM)	

8 Beispiele

8.1 TwinCAT FTP Client: Hochladen einer Datei auf einen FTP Server

In diesem Beispiel wird eine Datei von einem ADS-Gerät auf einen FTP Server hochgeladen.

Um dieses Beispiel ausführen zu können, sollten Sie einen Benutzer mit dem Namen "TestUser" und dem Passwort "TestPwd123" bei Ihrem FTP Server einrichten.

Wenn Sie nun an den Eingang "bExecute" eine positive Flanke anlegen mit Hilfe der Variable "startstop", so wird die Datei "\\Program Files\TestFile1.txt" auf den FTP Server geladen. Auf dem Server hat die Datei dann den Namen "\\TestFolder\File1.txt". Ist der Ordner "TestFolder" nicht vorhanden wird dieser erzeugt.

Achten Sie darauf, dass der TestUser Schreib- und Leserechte besitzt.

Variablendeklaration

```
PROGRAM MAIN
VAR
  FB_FTPFileUpload1      : FB_FTP_FileUploadEx;
  startstop              : BOOL;
  busy                   : BOOL;
  err                    : BOOL;
  errid                  : UDINT;
  progress               : UDINT;
END_VAR
```

SPS Programm

```
FB_FTPFileUpload1(
  sNetID:= ,
  sHost:= '172.16.235.100',
  nPort:= DEFAULT_FTP_PORT,
  sUsername:= 'TestUser',
  sPassword:= 'TestPwd123',
  sSrcFile:= '\\Program Files\TestFile1.txt',
  sDesFile:= '\\TestFolder\File1.txt',
  bExecute:= startstop,
  tTimeout:= T#15s,
  bBusy=> busy,
  bError=> err,
  nErrId=> errid,
  nProgress=> progress);
```

Einzubindende Bibliothek ist die "TcFTPClient.lib", "TcSystem.lib", "TcBase.lib" und "STANDARD.lib"

8.2 TwinCAT FTP Client: Herunterladen einer Datei von einem FTP Server auf ein CX-Gerät

In diesem Beispiel wird eine Datei von einem FTP Server auf ein CX-Gerät heruntergeladen.

Um dieses Beispiel ausführen zu können, sollten Sie einen Benutzer mit dem Namen "TestUser" und dem Passwort "TestPwd123" bei Ihrem FTP Server eingerichtet haben.

Mit einer positiven Flanke an der startstop Variablen wird das Herunterladen gestartet.

Zunächst wird eine Verbindung zum FTP Server mit dem FB_FTP_Open erzeugt. Danach wird die angegebene Datei, "\\TestFolder\File1.txt", mit dem Baustein FB_FTP_Upload vom FTP Server geladen. Der folgende Pfad "\\Program Files\TestFile1.txt", ist der Speicherort der Datei auf dem CX.

Zum Schluss wird noch die Verbindung zum FTP Server mit dem Baustein FB_FTP_Close geschlossen.

Achten Sie darauf, dass der TestUser Schreib- und Leserechte besitzt.

Variablendeklaration

```

PROGRAM MAIN
VAR
  RisingEdge          : R_TRIG;
  startstop           : BOOL;

  state               : BYTE;

  FB_FTP_Open1        : FB_FTP_Open;
  FB_FTP_FileDownload1 : FB_FTP_FileDownload;
  FB_FTP_Close1       : FB_FTP_Close;

  busy                : BOOL;
  err                 : BOOL;
  errid               : UDINT;

  handle              : T_HFTP;

  progress            : UDINT;
END_VAR

```

SPS Programm

```

RisingEdge(CLK:= startstop);
IF RisingEdge.Q THEN
  state := 1;
END_IF

CASE state OF
  0:
    ;

  1:
    FB_FTP_Open1(
      sNetID:= '5.0.252.142.1.1',
      sHost:= '172.16.9.223',
      nPort:= 21,
      sUsername:= 'TestUser',
      sPassword:= 'TestPwd123',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrId=> errid,
      hFTP=> handle);

    IF NOT busy AND NOT err THEN
      FB_FTP_Open1(bExecute:= FALSE);
      state := 2;
    END_IF

  2:
    FB_FTP_FileDownload1(
      sNetID:= '5.0.252.142.1.1',
      hFTP:= handle,
      sSrcFile:= '\TestFolder\File1.txt',
      sDesFile:= '\Program Files\TestFile1.txt',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy => busy,
      bError => err,
      nErrId => errid,
      nProgress => progress);

    IF NOT busy AND NOT err THEN
      FB_FTP_FileDownload1(bExecute:= FALSE);
      state := 3;
    END_IF

  3:
    FB_FTP_Close1(
      sNetID:= '5.0.252.142.1.1',
      hFTP:= handle,
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy => busy,
      bError => err,
      nErrId => errid);

```

```

IF NOT busy AND NOT err THEN
    FB_FTP_Close1(bExecute:= FALSE);
    state := 0;
END_IF

```

```
END_CASE
```

Einzubindende Bibliothek ist die "TcFTPClient.lib", "TcSystem.lib", "TcBase.lib" und "STANDARD.lib"

8.3 TwinCAT FTP Client: Auslesen der Verbindungsinformationen mit FB_FTP_Info

In diesem Beispiel wird gezeigt, wie Verbindungsinformationen vom TwinCAT FTP Client ausgelesen werden können.

Variablendeklaration

```

PROGRAM MAIN
VAR
    FB_FTP_Info1 : FB_FTP_Info;
    list : ARRAY [0..5] OF ST_FTP_ConnInfo;
    startstop : BOOL;
    busy : BOOL;
    err : BOOL;
    errid : BOOL;
    entries : UDINT;
END_VAR

```

SPS Programm

```

FB_FTP_Info1(
    sNetID:= ,
    pList:= ADR(list),
    cbList:= SIZEOF(list),
    bExecute:= startstop,
    tTimeout:= T#15s,
    bBusy=> busy,
    bError=> err,
    nErrID=> errid,
    nEntries=> entries);

```

Einzubindende Bibliothek ist die "TcFTPClient.lib", "TcSystem.lib", "TcBase.lib" und "STANDARD.lib"

Im folgenden Bild sehen Sie eine mögliche Ausgabe:

```

| FB_FTP_Info1
| list
| list[0]
|   sHost = '172.16.9.223'
|   nPort = 21
|   hFTP
|     hClient = 1096041956
|     sUsername = 'TestUser'
| list[1]
| list[2]
| list[3]
| list[4]
| list[5]
| startstop = TRUE
| busy = FALSE
| err = FALSE
| errid = 0
| entries = 1

```

8.4 TwinCAT FTP Client: Löschen einer Datei vom FTP Server

In diesem Beispiel wird gezeigt, wie eine Datei vom FTP Server gelöscht wird.

Zunächst wird mit dem Baustein FB_FTP_Open eine Verbindung zum FTP Server hergestellt. Danach wird überprüft, ob die Datei, welche gelöscht werden soll, überhaupt auf dem FTP Server existiert. Dies geschieht mit dem Baustein FB_FTP_FileExist. Der Löschvorgang wird mit dem Baustein FB_FTP_FileRemove durchgeführt. Zum Schluss wird dann nur noch die Verbindung zum Server geschlossen.

Sie starten das Beispiel, indem Sie eine Positive Flanke mit der Variablen "startstop" erzeugen.

Variablen Deklaration

```
PROGRAM MAIN
VAR
  RisingEdge      : R_TRIG;
  startstop       : BOOL;

  state           : BYTE;

  FB_FTP_Open1    : FB_FTP_Open;
  FB_FTP_FileExist1 : FB_FTP_FileExist;
  FB_FTP_FileRemove1 : FB_FTP_FileRemove;
  FB_FTP_Close1   : FB_FTP_Close;

  busy            : BOOL;
  err             : BOOL;
  errid           : UDINT;

  handle          : T_HFTP;

  exist           : BOOL;
END_VAR
```

SPS Programm

```
RisingEdge(CLK:=startstop);
IF RisingEdge.Q THEN
  state := 1;
END_IF

CASE state OF
  0:
    ;
  1:
    FB_FTP_Open1(
      sNetID:= '5.0.252.142.1.1',
      sHost:= '172.16.9.223',
      nPort:= DEFAULT_FTP_PORT,
      sUsername:= 'TestUser',
      sPassword:= 'TestPwd123',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrID=> errid,
      hFTP=> handle);

    IF NOT busy AND NOT err THEN
      state := 2;
      FB_FTP_Open1(bExecute:=FALSE);
    END_IF
  2:
    FB_FTP_FileExist1(
      sNetID:= '5.0.252.142.1.1',
      hFTP:= handle,
      sFile:= '\\TestFolder\File1.txt',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrID=> errid,
      bExist=> exist);

    IF NOT busy AND NOT err THEN
      IF exist THEN
        state := 3;
      ELSE
        state := 4;
      END_IF
      FB_FTP_FileExist1(bExecute:= FALSE);
    END_IF
END_CASE
```

```

    END_IF
3:
    FB_FTP_FileRemove1(
        sNetID:= '5.0.252.142.1.1',
        hFTP:= handle,
        sFile:= '\TestFolder\File1.txt',
        bExecute:= TRUE,
        tTimeout:= T#15s,
        bBusy=> busy,
        bError=> err,
        nErrID=> errid);

    IF NOT busy AND NOT err THEN
        state := 4;
        FB_FTP_FileRemove1(bExecute:=FALSE);
    END_IF
4:
    FB_FTP_Close1(
        sNetID:= '5.0.252.142.1.1',
        hFTP:= handle,
        bExecute:= TRUE,
        tTimeout:= T#15s,
        bBusy=> busy,
        bError=> err,
        nErrID=> errid);

    IF NOT busy AND NOT err THEN
        state := 0;
        FB_FTP_Close1(bExecute:=FALSE);
    END_IF
END_CASE

```

Einzubindende Bibliothek ist die "TcFTPClient.lib", "TcSystem.lib", "TcBase.lib" und "STANDARD.lib"

8.5 TwinCAT FTP Client: Auslesen einer Dateiliste von einem FTP Server

In diesem Beispiel wird eine Dateiliste von einem FTP Server gelesen.

Um dieses Beispiel ausführen zu können, sollten Sie einen Benutzer mit dem Namen "TestUser" und dem Passwort "TestPwd123" bei Ihrem FTP Server eingerichtet haben.

Mit einer positiven Flanke an der startstop Variablen wird eine Dateiliste erzeugt.

Zunächst wird eine Verbindung zum FTP Server mit dem FB_FTP_Open erzeugt. Danach werden alle Dateien ausgegeben die die Endung "*.txt" haben, mit dem Baustein FB_FTP_FileList vom FTP Server geladen.

Zum Schluss wird noch die Verbindung zum FTP Server mit dem Baustein FB_FTP_Close geschlossen.

Achten Sie darauf, dass der TestUser Schreib- und Leserechte besitzt.

Variablendeklaration

```

PROGRAM MAIN
VAR
    RisingEdge          : R_TRIG;
    startstop           : BOOL;

    state               : BYTE;

    FB_FTP_Open1        : FB_FTP_Open;
    FB_FTP_FileList1    : FB_FTP_FileList;
    FB_FTP_Close1       : FB_FTP_Close;

    busy                : BOOL;
    err                  : BOOL;
    errid                : UDINT;

    handle               : T_HFTP;

```

```

FileList      : ARRAY [0..MAX_FILELIST_ITEMS] OF STRING;
Items         : UDINT;
END_VAR

```

SPS Programm

```

RisingEdge(CLK:= startstop);
IF RisingEdge.Q THEN
  state := 1;
END_IF

CASE state OF
  0:
    ;

  1:
    FB_FTP_Open1(
      sNetID:= '5.0.252.142.1.1',
      sHost:= '172.16.9.223',
      nPort:= 21,
      sUsername:= 'TestUser',
      sPassword:= 'TestPwd123',
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrId=> errid,
      hFTP=> handle);

    IF NOT busy AND NOT err THEN
      FB_FTP_Open1(bExecute:= FALSE);
      state := 2;
    END_IF

  2:
    FB_FTP_FileList1(
      sNetID:= '5.0.252.142.1.1',
      hFTP:= handle,
      sMask:= '*.txt',
      nIndex:= 0,
      pList:= ADR(FileList),
      cbList:= SIZEOF(FileList),
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy=> busy,
      bError=> err,
      nErrID=> errid,
      nItems=> Items);

    IF NOT busy AND NOT err THEN
      FB_FTP_FileList1(bExecute:= FALSE);
      state := 3;
    END_IF

  3:
    FB_FTP_Close1(
      sNetID:= '5.0.252.142.1.1',
      hFTP:= handle,
      bExecute:= TRUE,
      tTimeout:= T#15s,
      bBusy => busy,
      bError => err,
      nErrId => errid);

    IF NOT busy AND NOT err THEN
      FB_FTP_Close1(bExecute:= FALSE);
      state := 0;
    END_IF
END_CASE

```

Einzubindende Bibliothek ist die **"TcFTPClient.lib"**, **"TcSystem.lib"**, **"TcBase.lib"** und **"STANDARD.lib"**

9 Anhang

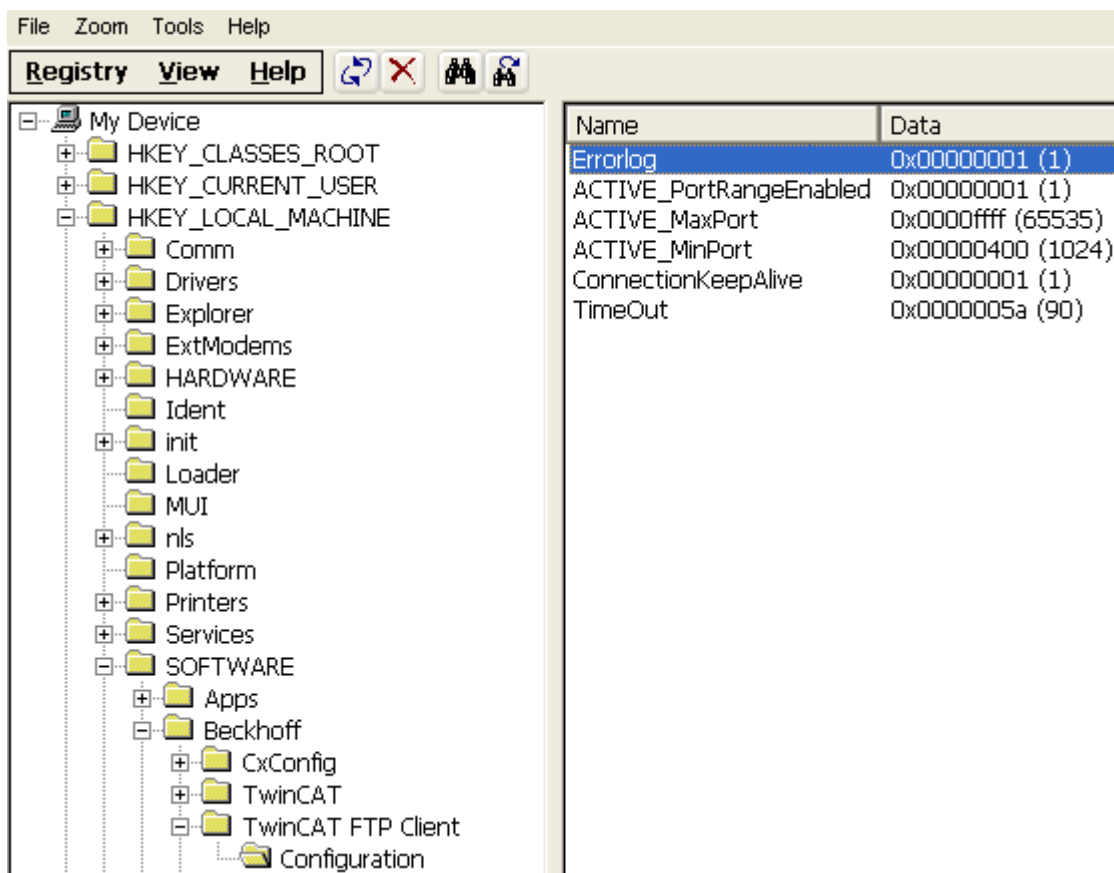
9.1 Erzeugen einer Fehlerloggdatei

Um auftretende Fehler besser lokalisieren zu können und eine bessere Beschreibung zum Fehlerfall zu erhalten bietet der TwinCAT FTP Client die Möglichkeit, auftretende Fehler in einer Textdatei zu protokollieren.

Die Textdatei mit dem Namen "**TcFTPErrorLog.txt**" wird immer in dem Ordner erzeugt, in dem auch die "EXE" des TwinCAT FTP Clients liegt. Standardmäßig im Ordner "\Hard Disk\System".

Sie können die Funktionalität aktivieren durch das Erzeugen des folgenden Registry Keys:

"HKEY_LOCAL_MACHINE\Software\Beckhoff\TwinCAT FTP Client\Configuration\ErrorLog"



Mit dem Wert 1 wird die Funktionalität nach einem Restart des TwinCAT Systems gestartet.

Mit dem Wert 0 wird die Funktionalität nach einem Restart des TwinCAT Systems gestoppt.

HINWEIS

Schädigung des Flashmediums

Das zyklische Schreiben auf das Flashmedium kann dieses beschädigen. Des Weiteren besteht die Möglichkeit, dass der Speicher des Flashmediums durch das zyklische Schreiben vollgeschrieben wird.

HINWEIS

Schädigung des Flashmediums

Das Fehlerloggen nur beim Testen verwenden!

9.2 Einstellen des Portbereiches

Beim Aktiven-FTP Modus wird der Port für die Verbindung mit dem FTP Server vom Client aus vorgegeben. Soll der TwinCAT FTP Client im Aktiven FTP Modus betrieben werden, kann ein Portbereich definiert werden, aus dem ein freier Port für den Verbindungsaufbau verwendet werden soll.

Das Einstellen des Portbereiches erfolgt über die folgenden drei Registrykeys.

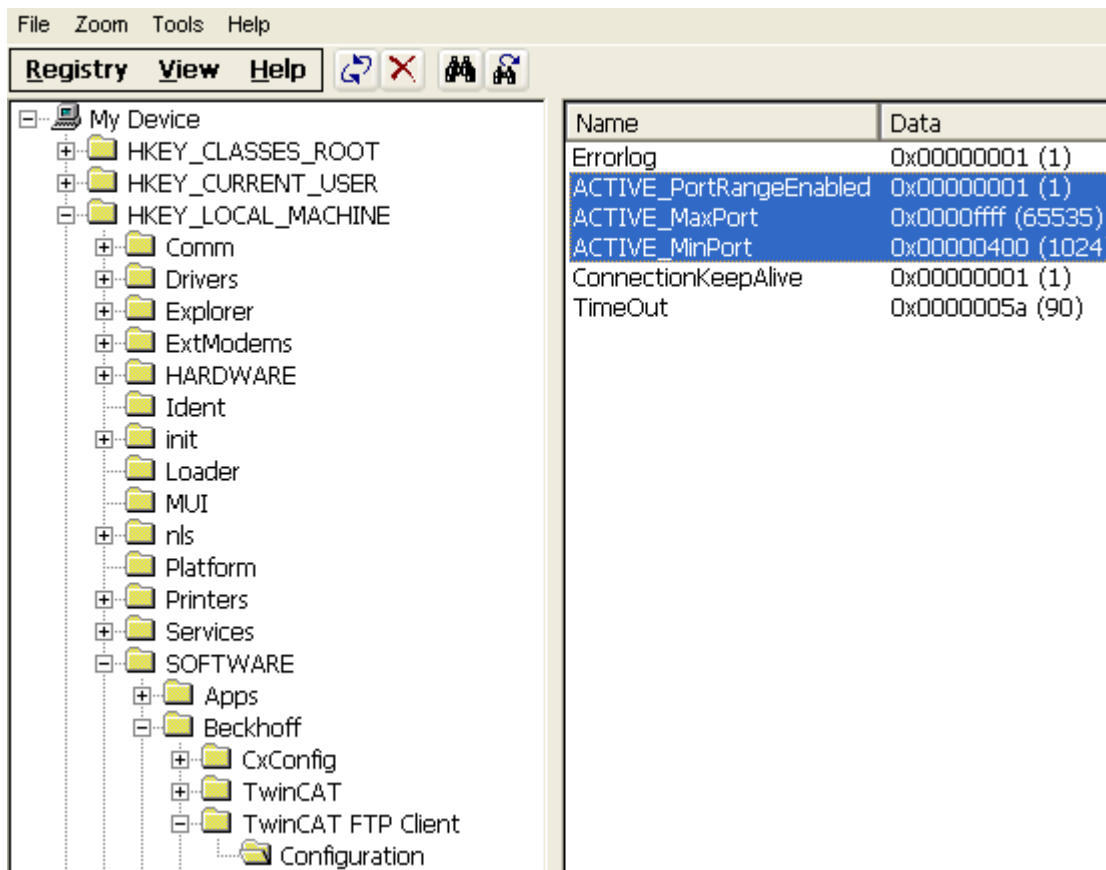
"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\
\\Configuration\\ACTIVE_PortRangeEnabled"

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\
\\Configuration\\ACTIVE_MaxPort"

"HKEY_LOCAL_MACHINE\\Software\\Beckhoff\\TwinCAT FTP Client\\Configuration\\ACTIVE_MinPort"

Zum Aktivieren des Portbereiches muss der Registrykey "ACTIVE_PortRangeEnabled" auf 1 gesetzt werden. Andernfalls wird der TwinCAT FTP Client einen beliebigen freien Port für den Verbindungsaufbau verwenden.

Mit den Registrykeys "ACTIVE_MaxPort" und "ACTIVE_MinPort" können die obere bzw. untere Grenze des Portbereiches definiert werden.



9.3 Return Codes

9.3.1 Übersicht der Fehlercodes des TwinCAT FTP Client

Voraussetzungen

Offset + Fehler-code	Bereich	Beschreibung
0x00000000 + TwinCAT System Fehler [► 46]	0x00000000-0x0000 7800	TwinCAT System Fehler (ADS-Fehlercodes inklusive)

Offset + Fehler-code	Bereich	Beschreibung
0x00008000 + <u>Interner TwinCAT</u> <u>FTP Client Fehler</u> [▶ 49]	0x00008000-0x0000 81C4	Interne Fehler des TwinCAT FTP Clients

9.3.2 ADS Return Codes

Error codes: [0x000... \[▶ 46\]](#), [0x500... \[▶ 46\]](#), [0x700... \[▶ 46\]](#), [0x1000... \[▶ 46\]](#), [0x274C... \[▶ 46\]](#)

Global Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x0	0	no error		
0x1	1	Internal error		
0x2	2	No Rtime		
0x3	3	Allocation locked memory error		
0x4	4	Insert mailbox error	No ADS mailbox was available to process this message.	Reduce the number of ADS calls (e.g ADS-Sum commands or Max Delay Parameter)
0x5	5	Wrong receive HMSG		
0x6	6	target port not found	ADS Server not started	
0x7	7	target machine not found	Missing ADS routes	
0x8	8	Unknown command ID		
0x9	9	Bad task ID		
0xA	10	No IO		
0xB	11	Unknown ADS command		
0xC	12	Win 32 error		
0xD	13	Port not connected		
0xE	14	Invalid ADS length		
0xF	15	Invalid AMS Net ID		
0x10	16	Low Installation level		
0x11	17	No debug available		
0x12	18	Port disabled		
0x13	19	Port already connected		
0x14	20	ADS Sync Win32 error		
0x15	21	ADS Sync Timeout		
0x16	22	ADS Sync AMS error		
0x17	23	ADS Sync no index map		
0x18	24	Invalid ADS port		
0x19	25	No memory		
0x1A	26	TCP send error		
0x1B	27	Host unreachable		
0x1C	28	Invalid AMS fragment		

Router Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x500	1280	ROUTERERR_NOLOCKEDMEMORY	No locked memory can be allocated	
0x501	1281	ROUTERERR_RESIZEMEMORY	The size of the router memory could not be changed	
0x502	1282	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages. The current sent message was rejected	Check the connection between the communication partners
0x503	1283	ROUTERERR_DEBUGBOXFULL	The mailbox has reached the maximum number of possible messages. The sent message will not be displayed in the debug monitor	Check the connection to the debug monitor
0x504	1284	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown	
0x505	1285	ROUTERERR_NOTINITIALIZED	Router is not initialised	
0x506	1286	ROUTERERR_PORTALREADYINUSE	The desired port number is already assigned	
0x507	1287	ROUTERERR_NOTREGISTERED	Port not registered	
0x508	1288	ROUTERERR_NOMOREQUEUES	The maximum number of Ports reached	
0x509	1289	ROUTERERR_INVALIDPORT	The port is invalid.	
0x50A	1290	ROUTERERR_NOTACTIVATED	TwinCAT Router not active	
0x50B	1291	ROUTERERR_FRAGMENTBOXFULL		
0x50C	1292	ROUTERERR_FRAGMENTTIMEOUT		
0x50D	1293	ROUTERERR_TOBEREMOVED		

General ADS Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x700	1792	error class <device error>		
0x701	1793	Service is not supported by server		
0x702	1794	invalid index group		
0x703	1795	invalid index offset		
0x704	1796	reading/writing not permitted		
0x705	1797	parameter size not correct		
0x706	1798	invalid parameter value(s)		
0x707	1799	device is not in a ready state		
0x708	1800	device is busy		
0x709	1801	invalid context (must be in Windows)		
0x70A	1802	out of memory		
0x70B	1803	invalid parameter value(s)		
0x70C	1804	not found (files, ...)		
0x70D	1805	syntax error in command or file		
0x70E	1806	objects do not match		

Hex	Dec	Description	Possible Causes	Solution
0x70F	1807	object already exists		
0x710	1808	symbol not found		
0x711	1809	symbol version invalid	Onlinechange	Release handle and get a new one
0x712	1810	server is in invalid state		
0x713	1811	AdsTransMode not supported		
0x714	1812	Notification handle is invalid	Onlinechange	Release handle and get a new one
0x715	1813	Notification client not registered		
0x716	1814	no more notification handles		
0x717	1815	size for watch too big		
0x718	1816	device not initialized		
0x719	1817	device has a timeout		
0x71A	1818	query interface failed		
0x71B	1819	wrong interface required		
0x71C	1820	class ID is invalid		
0x71D	1821	object ID is invalid		
0x71E	1822	request is pending		
0x71F	1823	request is aborted		
0x720	1824	signal warning		
0x721	1825	invalid array index		
0x722	1826	symbol not active	Onlinechange	Release handle and get a new one
0x723	1827	access denied		
0x724	1828	missing license		Activate license for TwinCAT 3 function
0x72c	1836	exception occurred during system start		Check each device transistions
0x740	1856	Error class <client error>		
0x741	1857	invalid parameter at service		
0x742	1858	polling list is empty		
0x743	1859	var connection already in use		
0x744	1860	invoke ID in use		
0x745	1861	timeout elapsed		Check ADS routes of sender and receiver and your <u>firewall setting</u>
0x746	1862	error in win32 subsystem		
0x747	1863	Invalid client timeout value		
0x748	1864	ads-port not opened		
0x750	1872	internal error in ads sync		
0x751	1873	hash table overflow		
0x752	1874	key not found in hash		
0x753	1875	no more symbols in cache		
0x754	1876	invalid response received		
0x755	1877	sync port is locked		

RTime Error Codes

Hex	Dec	Description	Possible Causes
0x1000	4096	RTERR_INTERNAL	Internal fatal error in the TwinCAT real-time system
0x1001	4097	RTERR_BADTIMERPERIODS	Timer value not valid
0x1002	4098	RTERR_INVALIDTASKPTR	Task pointer has the invalid value ZERO
0x1003	4099	RTERR_INVALIDSTACKPTR	Task stack pointer has the invalid value ZERO
0x1004	4100	RTERR_PRIOEXISTS	The demand task priority is already assigned
0x1005	4101	RTERR_NOMORETCB	No more free TCB (Task Control Block) available. Maximum number of TCBs is 64
0x1006	4102	RTERR_NOMORESEMAS	No more free semaphores available. Maximum number of semaphores is 64
0x1007	4103	RTERR_NOMOREQUEUEES	No more free queue available. Maximum number of queue is 64
0x1008	4104	TwinCAT reserved.	
0x1009	4105	TwinCAT reserved.	
0x100A	4106	TwinCAT reserved.	
0x100B	4107	TwinCAT reserved.	
0x100C	4108	TwinCAT reserved.	
0x100D	4109	RTERR_EXTIRQALREADYDEF	An external synchronisation interrupt is already applied
0x100E	4110	RTERR_EXTIRQNOTDEF	No external synchronisation interrupt applied
0x100F	4111	RTERR_EXTIRQINSTALLFAILED	The apply of the external synchronisation interrupt failed
0x1010	4112	RTERR_IRQNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in BIOS.
0x1019	4121	RTERR_VMXCONTROLSMISSING	Missing feature in Intel VT-x extension.
0x101A	4122	RTERR_VMXENABLEFAILS	Enabling Intel VT-x fails.

TCP Winsock Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x274c	10060	A socket operation was attempted to an unreachable host	Host unreachable	Check network connection via ping
0x274d	10061	A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.	Host unreachable	Check network connection via ping
0x2751	10065	No connection could be made because the target machine actively refused it		
		Further Winsock error codes: Win32 Error Codes		

9.3.3 FTP Client Return Codes

Hex	Dez	Beschreibung
0x00008001	32768 + 1 := 32769	Interner Fehler TwinCAT FTP Client.

Hex	Dez	Beschreibung
0x00008002	32768 + 2 := 32770	Datei Fehler (z.B. Datei nicht gefunden, Zugriff verweigert)
0x00008003	32768 + 3 := 32771	Übertragungsfehler (z.B. Verbindung abgebrochen)
0x00008004	32768 + 4 := 32772	Fehler beim Verbinden auf den Server. Verbindung konnte nicht hergestellt werden.
0x00008005	32768 + 5 := 32773	Keinen Response vom FTP Server empfangen.
0x000081F4	32768 + 500 := 33268	Syntax Fehler, Kommando unbekannt. Dies kann den Fehler "Kommandozeile zu lang" beinhalten.
0x000081F5	32768 + 501 := 33269	Syntax Fehler bei Parametern oder Argumenten.
0x000081F6	32768 + 502 := 33270	Kommando nicht implementiert.
0x000081F7	32768 + 503 := 33271	Falsche Abfolge der Kommandos.
0x000081F8	32768 + 504 := 33272	Kommando nicht für diesen Parameter implementiert.
0x00008212	32768 + 530 := 33298	Nicht eingeloggt.
0x00008214	32768 + 532 := 33300	Benötigt Account um Dateien zu speichern.
0x00008226	32768 + 550 := 33318	Angeforderte Aktion nicht ausgeführt. Datei nicht erreichbar (z.B. Datei nicht gefunden, keinen Zugriff).
0x00008227	32768 + 551 := 33319	Angeforderte Aktion abgebrochen, Seitentyp unbekannt.
0x00008228	32768 + 552 := 33320	Angeforderte Dateiaktion abgebrochen. Zugewiesenen Speicher überschritten (für aktuelles Verzeichnis oder Datei).
0x00008229	32768 + 553 := 33321	Angeforderte Aktion nicht ausgeführt. Dateiname nicht erlaubt.
0x000081A5	32768 + 421 := 33189	Service nicht erreichbar, schließt Control Verbindungen. Dies ist vermutlich die Rückantwort auf jedes Kommando welches einen Neustart verursacht.
0x000081A9	32768 + 425 := 33193	Kann keine Datenverbindung öffnen.
0x000081AA	32768 + 426 := 33194	Verbindung geschlossen; Transfer abgebrochen.
0x000081C2	32768 + 450 := 33218	Angeforderte Dateiaktion nicht ausgeführt.
0x000081C3	32768 + 451 := 33219	Angeforderte Aktion abgebrochen. Lokaler Fehler während der Abarbeitung.
0x000081C4	32768 + 452 := 33220	Angeforderte Aktion nicht ausgeführt. Unzureichender Speicherplatz im System. Datei nicht erreichbar (z.B. Datei in Benutzung).

Mehr Informationen:
www.beckhoff.de/ts6300

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

