



## Anwendungshinweis

### Bibliothek zum Dateitransfer mit FTP für XControl-Steuerungen

---

**08/05 AN2700K34D** V1.0

© Moeller GmbH, Bonn

Autor: Matthias Blümling

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelfalter.

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Anwendungshinweises darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Moeller GmbH, Bonn, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Stand: April 2001

Änderungen vorbehalten.

## Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>ALLGEMEINES .....</b>  | <b>3</b>  |
| 1.1      | FUNKTION .....  | 3         |
| 1.2      | EINSATZBEREICH .....  | 3         |
| 1.3      | HARDWAREVORAUSSETZUNGEN .....                                     | 4         |
| 1.4      | SOFTWAREVORAUSSETZUNG .....                                       | 4         |
| <b>2</b> | <b>DIE FTP-BIBLIOTHEK .....</b>                                   | <b>5</b>  |
| 2.1      | EINBINDEN DER FTP-BIBLIOTHEK .....                                | 5         |
| 2.2      | FUNKTIONSBAUSTEINE UND PARAMETER .....                            | 5         |
| 2.2.1    | <i>FTP_Connect</i> .....  | 6         |
| 2.2.2    | <i>FTP_Username</i> .....   | 6         |
| 2.2.3    | <i>FTP_Password</i> .....   | 7         |
| 2.2.4    | <i>FTP_Disconnect</i> .....                                       | 7         |
| 2.2.5    | <i>FTP_ChangeDirectory</i> .....                                  | 8         |
| 2.2.6    | <i>FTP_WorkingDirectory</i> .....                                 | 8         |
| 2.2.7    | <i>FTP_MakeDirectory</i> .....                                    | 9         |
| 2.2.8    | <i>FTP_DeleteDirectory</i> .....                                  | 9         |
| 2.2.9    | <i>FTP_DeleteFile</i> .....                                       | 10        |
| 2.2.10   | <i>FTP_UploadFile</i> .....                                       | 10        |
| 2.2.11   | <i>FTP_DownloadFile</i> .....                                     | 11        |
| 2.2.12   | <i>FTP_GetLastError</i> .....                                     | 11        |
| 2.2.13   | <i>FTP_GetLastServerResponse</i> .....                            | 12        |
| 2.2.14   | <i>Globale Parameter</i> .....                                    | 12        |
| <b>3</b> | <b>INBETRIEBNAHME .....</b>                                       | <b>13</b> |
| 3.1      | EINBINDUNG DER FTP ANKOPPLUNG UNTER STEUERUNGEN DER X-REIHE ..... | 13        |
| 3.2      | ANWENDUNG DER FTP-BIBLIOTHEK .....                                | 13        |
| 3.2.1    | <i>Anmelden am Server</i> .....                                   | 13        |
| 3.2.2    | <i>Verzeichnisfunktionen</i> .....                                | 13        |
| 3.2.3    | <i>Dateifunktionen</i> .....                                      | 14        |
| 3.2.4    | <i>Abmelden vom FTP-Server</i> .....                              | 14        |
| 3.2.5    | <i>Die Funktion FTP_GetLastError</i> .....                        | 14        |
| 3.2.6    | <i>Die Funktion FTP_GetLastServerResponse</i> .....               | 16        |
| <b>4</b> | <b>VERSIONSINDEX .....</b>  | <b>17</b> |

# 1 Allgemeines

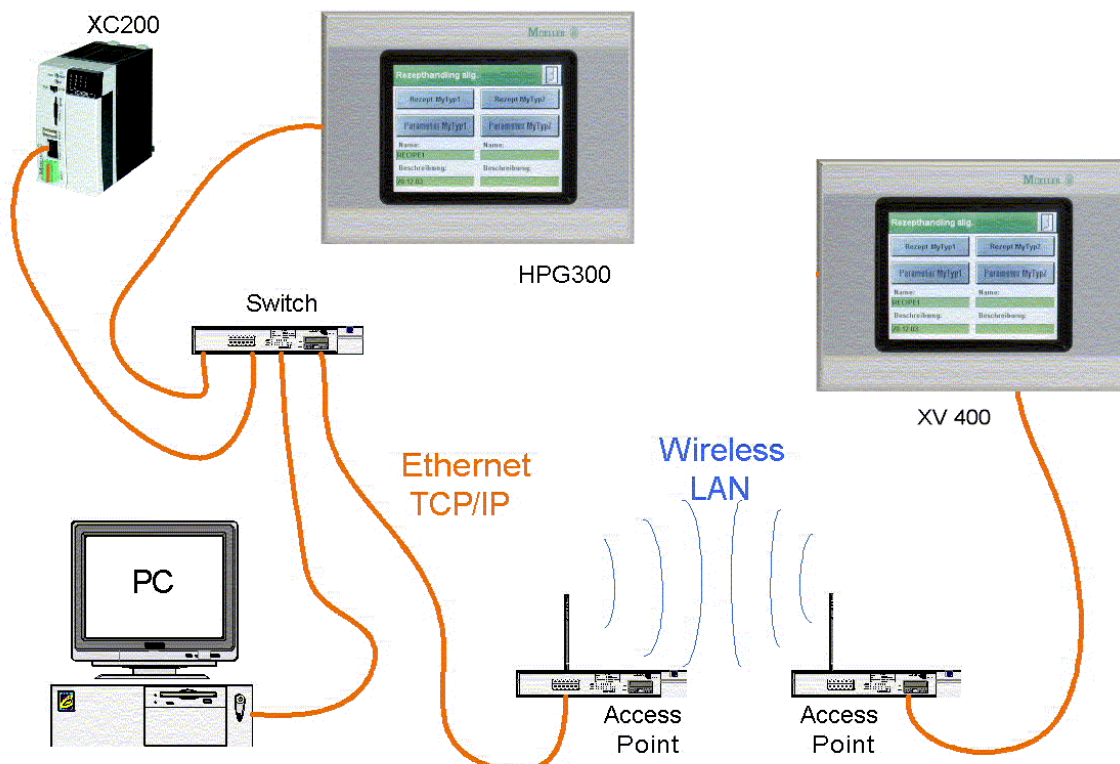
## 1.1 Funktion

FTP-Client, anwendbar mit XControl Steuerungen und MC-HPG.

- An- und Abmelden am FTP-Server
- Up- und Download von Dateien
- Löschen von Dateien
- Abfragen des aktuellen Verzeichnisses
- Erstellen und Löschen von Verzeichnissen
- Wechsel des Arbeitsverzeichnisses auf dem Remote-Host

## 1.2 Einsatzbereich

Dateitransfer zwischen SPS (auch zum oder vom PC), aus der Applikation heraus gesteuert.  
Bsp.: Up- und Download von Messwertdateien, Rezepten, Parametrierungen usw.



### 1.3 Hardwarevoraussetzungen

| XControl Steuerung | ab OS Version |
|--------------------|---------------|
| MC-HPG 200         | V 1.34        |
| MC-HPG 300         | V 1.34        |
| XC 200             | V 1.03.02     |
| XV 400             | V 2.04        |
| XVC 600            | V 1.34        |
| XCC 600            | V 1.34        |

### 1.4 Softwarevoraussetzung

Es wird die XSoft ab Version 2.3.3 benötigt.

#### Benötigte Bibliotheken:

- Standard.lib
- SysLibCallback.lib
- SysLibSockets.lib      ab Version 2.4.0.6
- SysLibFile.lib

## 2 Die FTP-Bibliothek

### 2.1 Einbinden der FTP-Bibliothek

Die Bibliothek FTP\_Client.lib wird über den Bibliotheksverwalter in die Applikation eingefügt und von dort aus aufgerufen. Die zusätzlich benötigten Bibliotheken Standard.lib, SysLibSockets.lib, SysLibCallback.lib und SysLibFile.lib müssen manuell hinzugefügt werden, sofern sie nicht automatisch von der XSoft hinzugefügt wurden.

### 2.2 Funktionsbausteine und Parameter

**Achtung:** Die internen Funktionen dürfen nur durch die Funktionen der FTP-Bibliothek aufgerufen werden. Sie sind nicht vom Anwender aufzurufen!

**Achtung:** Es darf pro Zyklus immer nur eine FTP-Funktion aktiv (busy) sein! Vor dem Aufruf einer weiteren FTP-Funktion, muss sichergestellt sein, dass die vorherige Funktion vollständig abgearbeitet wurde.  
Bei Funktionen mit Ausgabevariable xBusy muss gewartet werden, bis diese Variable den Wert FALSE aufweist. Erst dann darf die nächste FTP-Funktion gestartet werden.

Alle Funktionsblöcke werden mit einer positiven Flanke an der boolschen Eingangsvariable xExecute gestartet.

Die Ausgabevariable xBusy ist TRUE, solange der Funktionsbaustein beschäftigt ist. In dieser Zeit darf kein weiterer FTP-Befehl ausgeführt werden!

Die Ausgangsvariablen xError und xDone zeigen für einen Zyklus das Ergebnis des Funktionsaufrufs an, wenn xExecute schon wieder den Wert FALSE hat. Falls xExecute noch TRUE ist, werden xDone und xError mit der fallenden Flanke von xExecute auf FALSE zurück gesetzt.

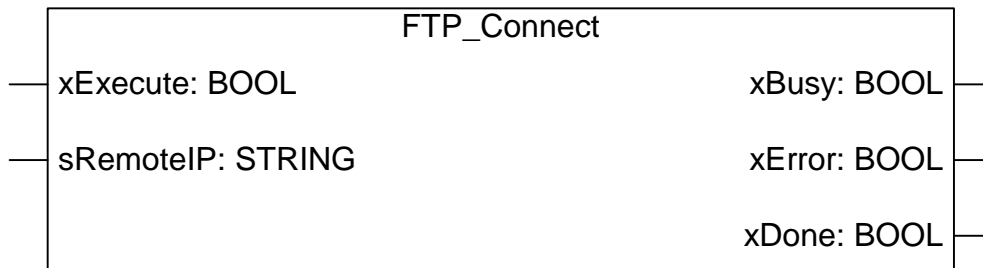
Die Dauer der Ausführung ist vom jeweiligen Baustein abhängig. Weitere Faktoren sind der eingestellte Task-Intervall und die Größe der zu übertragenden Datei.

Die Größe der zu übertragenden Datei ist theoretisch nur durch den Speicher der PLC bzw. MMC oder USB-Stick begrenzt.

Die Ausgabevariable xError zeigt einen Fehler an. Zusammen mit xError wird auch eine globale Error-ID mit FTP\_Client\_wErrorID ausgegeben. Der 2-stellige Code definiert die Fehlerursache. Damit wird die Suche nach einem eventuell aufgetretenen Fehler erleichtert. (siehe auch 3.2.5)

### 2.2.1 FTP\_Connect

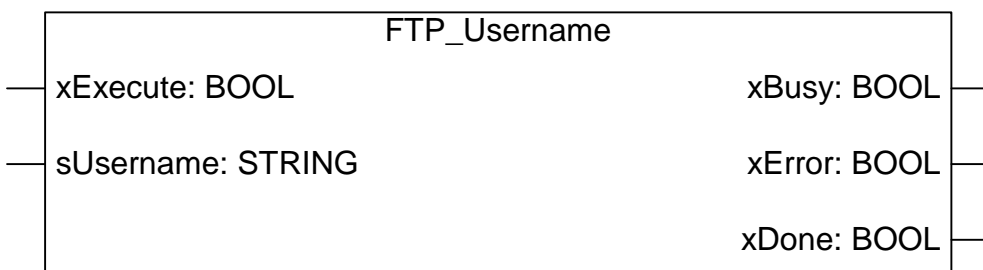
Diese Funktion dient dem Aufbau der Verbindung zwischen der Steuerung und einem FTP-Server.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet  |
| sRemoteIP  | IP des FTP-Servers im Standard-Dotted-Format  |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist   |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an |
| xError     | Zeigt einen Fehler an                         |

### 2.2.2 FTP Username

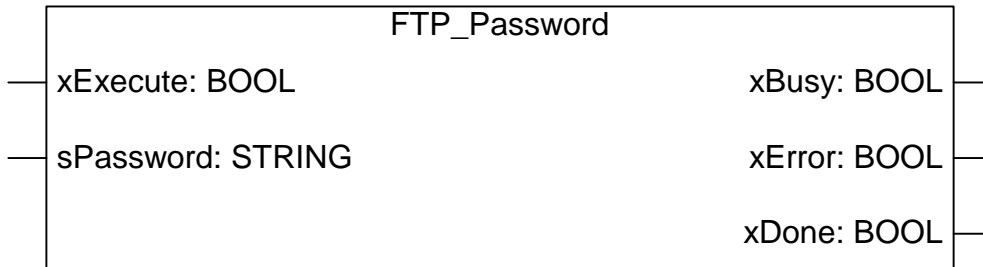
Diese Funktion dient dem Übertragen eines User-Namens an den Server. Diese Funktion wird zur Anmeldung benötigt.



| Var_Input  |   |
|------------|---|
| XExecute   | Funktion wird mit positiver Flanke gestartet  |
| sUsername  | Username. Maximal 80 Zeichen                  |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist   |
| xError     | Zeigt einen Fehler an                         |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an |

### 2.2.3 FTP\_Password

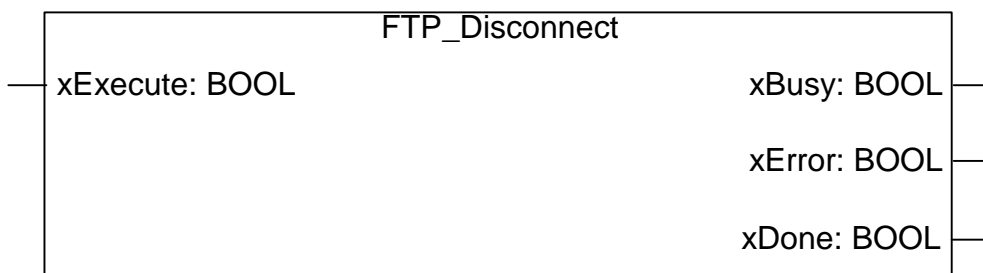
Diese Funktion wird zur Übertragung eines Passwortes an den Server verwendet. Ein Passwort wird nur dann benötigt, wenn nach der Ausführung der Funktion FTP\_User der Server einen Responsecode = 331 gesendet wird („username ok; send password“). Wurde der Response-Code 230 gesendet, so ist kein Passwort erforderlich („login ok“).



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet  |
| sPassword  | Passwort                                      |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist   |
| xError     | Zeigt einen Fehler an                         |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an |

### 2.2.4 FTP\_Disconnect

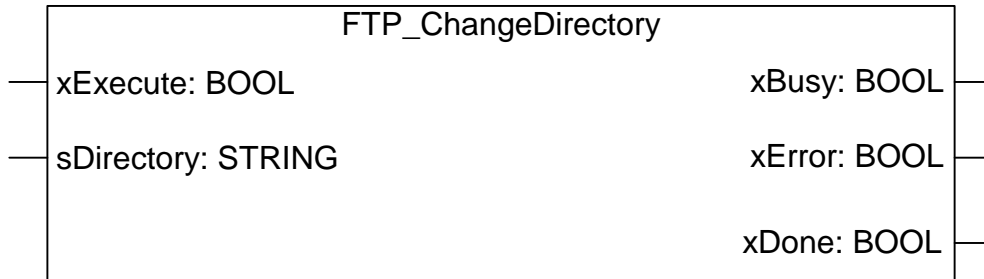
Bei Aufruf dieser Funktion wird die bestehende FTP-Verbindung beendet.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet  |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist   |
| xError     | Zeigt einen Fehler an                         |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an |

### 2.2.5 FTP\_ChangeDirectory

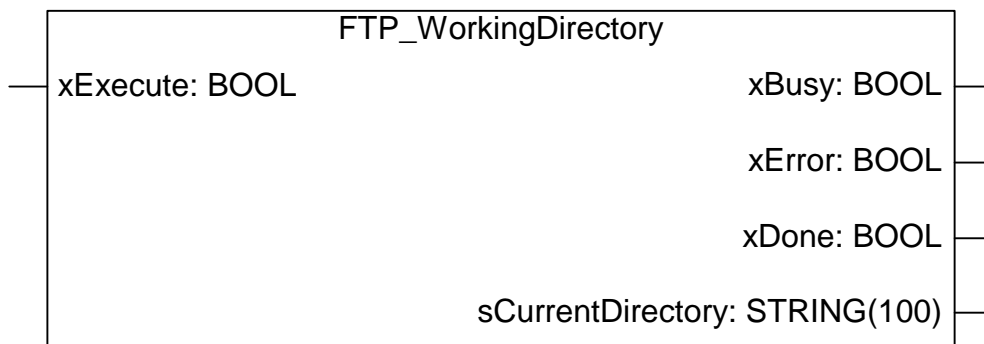
Diese Funktion wird zum wechseln des Arbeitsverzeichnisses auf dem Remote-Host verwendet.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet  |
| sDirectory | Verzeichnisname. Maximal 80 Zeichen           |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist   |
| xError     | Zeigt einen Fehler an                         |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an |

### 2.2.6 FTP\_WorkingDirectory

Gibt das aktuelle Arbeitsverzeichnis des FTP-Servers aus.

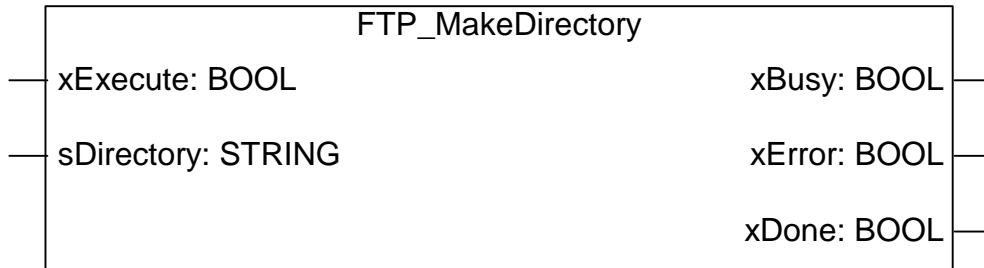


| Var_Input         |   |
|-------------------|---|
| xExecute          | Funktion wird mit positiver Flanke gestartet  |
| Var_Output        |   |
| xBusy             | Zeigt an, dass die Funktion beschäftigt ist   |
| xError            | Zeigt einen Fehler an                         |
| xDone             | Zeigt erfolgreiche Ausführung der Funktion an |
| sCurrentDirectory | Ausgabe des aktuellen Arbeitsverzeichnisses   |



### 2.2.7 FTP\_MakeDirectory

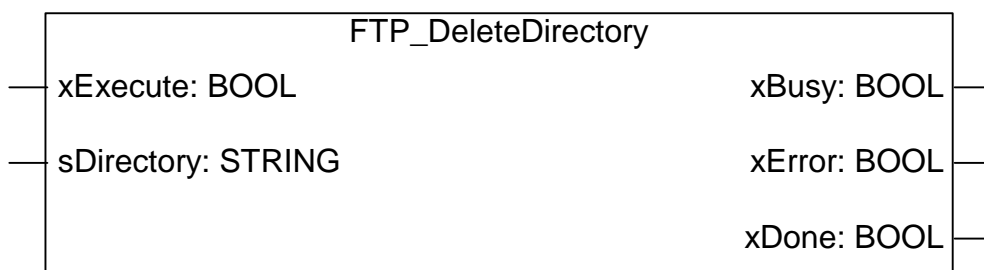
FTP\_MakeDirectory wird zum erzeugen von Verzeichnissen verwendet.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet      |
| sDirectory | Name des Verzeichnisses, das erstellt werden soll |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist       |
| xError     | Zeigt einen Fehler an                             |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an     |

### 2.2.8 FTP\_DeleteDirectory

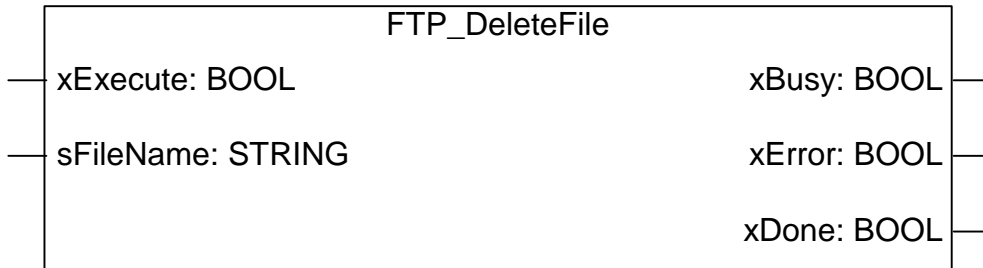
FTP\_DeleteDirectory wird zum löschen von Verzeichnissen verwendet.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet      |
| sDirectory | Name des Verzeichnisses, das gelöscht werden soll |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist       |
| xError     | Zeigt einen Fehler an                             |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an     |

### 2.2.9 FTP\_DeleteFile

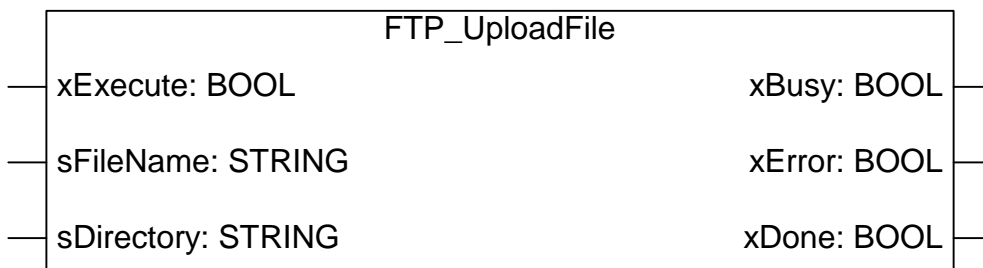
FTP\_Delete wird zum Löschen von Dateien auf dem Server verwendet.



| Var_Input  |   |
|------------|---|
| xExecute   | Funktion wird mit positiver Flanke gestartet            |
| sFileName  | Name der Datei auf dem Server, die gelöscht werden soll |
| Var_Output |   |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist             |
| xError     | Zeigt einen Fehler an                                   |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an           |

### 2.2.10 FTP\_UploadFile

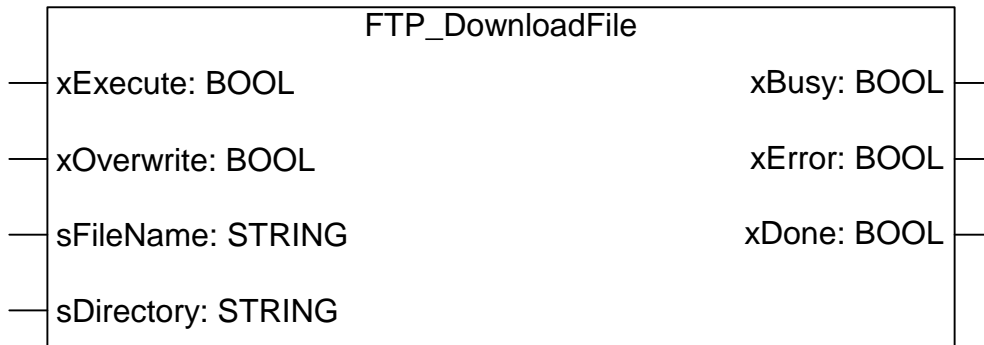
Diese Funktion wird zum Upload einer Datei auf den Server verwendet.



| Var_Input  |  |
|------------|--|
| xExecute   | Funktion wird mit positiver Flanke gestartet   |
| sFileName  | Name der Datei, die auf den Server geladen werden soll   |
| sDirectory | Verzeichnis der Datei, die auf den Server geladen werden soll (unbedingt mit „\“ abschließen!) |
| Var_Output |  |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist  |
| xError     | Zeigt einen Fehler an  |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an  |

### 2.2.11 FTP\_DownloadFile

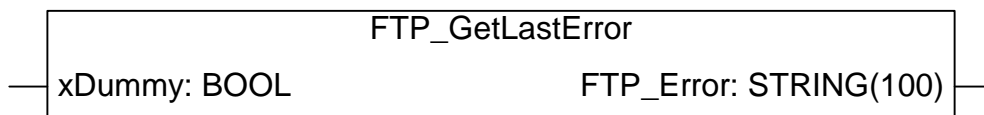
Diese Funktion dient zum Download einer Datei vom FTP-Server.



| Var_Input  |  |
|------------|--|
| xExecute   | Funktion wird mit positiver Flanke gestartet   |
| xOverwrite | Wenn TRUE, dann wird eine eventuell bestehende Datei überschrieben   |
| sFileName  | Name der Datei, die vom Server geladen werden soll   |
| sDirectory | Verzeichnis, in dem die vom Server übertragene Datei gespeichert werden soll. Unbedingt mit „\“ abschließen! |
| Var_Output |  |
| xBusy      | Zeigt an, dass die Funktion beschäftigt ist  |
| xError     | Zeigt einen Fehler an  |
| xDone      | Zeigt erfolgreiche Ausführung der Funktion an  |

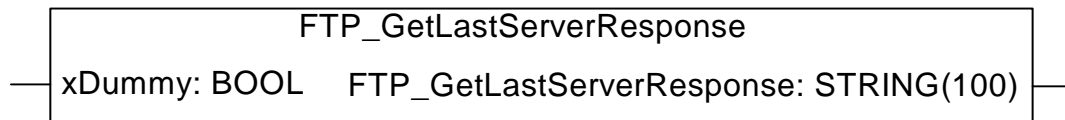
### 2.2.12 FTP\_GetLastError

Diese Funktion wird verwendet, um einen Error-String im Klartext zu erhalten. Es wird keine Inputvariable benötigt. Es dieser Funktion eine Dummyvariable mitgegeben werden, weil der Compiler mindestens eine Inputvariable erwartet. Sie hat keine weitere Funktion.



### 2.2.13 *FTP\_GetLastServerResponse*

Diese Funktion gibt die Bedeutung des Code FTP\_Client\_uiLastServerResponse vom letzten Server-Response im Klartext aus. Es wird keine Inputvariable benötigt. Es ist dieser Funktion eine Dummyvariable mitgegeben werden, weil der Compiler mindestens eine Inputvariable erwartet. Sie hat keine weitere Funktion.



### 2.2.14 *Globale Parameter*

|                                  |   |
|----------------------------------|---|
| FTP_Client_xActive:              | Wenn diese Variable TRUE ist, dann besteht eine aktive Verbindung zu einem FTP_Server. Es kann keine weitere Verbindung aufgebaut werden, da der FTP_Client für eine Instanz ausgelegt ist.                   |
| FTP_Client_wErrorID:             | In dieser Variable wird der aktuelle Fehlercode abgelegt. Sie wird als Parameter der Funktion FTP_Error verwendet, um eine Fehlerbeschreibung im Klartext zu erhalten.  |
| FTP_Client_xInit:                | Diese Variable wird nach erfolgreicher Registrierung der Callback-Funktion auf TRUE gesetzt.  |
| FTP_Client_uiLastServerResponse: | In dieser Variable wird der letzte Response-Code des Servers abgelegt. Sie wird als Parameter der Funktion FTP_ServerResponse verwendet, um die Bedeutung des Server-Response im Klartext anzeigen zu können. |

Alle weiteren globalen Variablen haben für die Anwendung der FTP-Bibliothek keine Bedeutung.

## 3 Inbetriebnahme

### 3.1 Einbindung der FTP Ankopplung unter Steuerungen der X-Reihe

Die Bibliothek FTP\_Client.lib ist mit dem Bibliotheksverwalter einzufügen. Es werden, soweit vorhanden, die Bibliotheken Standard.lib, SysLibSockets.lib und SysLibCallback.lib automatisch eingefügt. Wird dies von der XSoft nicht automatisch vorgenommen, so müssen diese Bibliotheken manuell eingefügt werden.

**Achtung:** Der FTP-Client wurde für eine Instanz ausgelegt. Aus diesem Grund dürfen **keine** Instanzen von diesem Baustein aufgerufen werden.

**Achtung:** Funktionen im Ordner ‚Internal Functions‘ sind nicht vom Anwender aufzurufen!

**Achtung:** Die FTP-Funktionen sollten in einer extra Task ausgeführt werden. Der Watchdog ist auszuschalten! Taskeinstellungen können abhängig von der Anforderung optimiert werden. Sollen in einem größeren zeitlichen Abstand z.B. Messwertdateien übertragen werden, so kann der FTP-Task ereignisgesteuert aufgerufen werden. Ansonsten empfiehlt sich ein Taskintervall von rund 100 ms oder größer. Die Priorität sollte niedrig gewählt werden.

### 3.2 Anwendung der FTP-Bibliothek

#### 3.2.1 Anmelden am Server

Die Verbindung zu einem FTP-Server wird mit der Funktion FTP\_Connect hergestellt. Als Eingabeparameter wird die IP-Adresse des FTP-Server im Standard-Dotted-Format benötigt (Bsp.: 192.168.119.300).

Laut Definition des FTP-Protokolls ist eine Anmeldung am FTP-Server Pflicht. Nach dem Öffnen der Verbindung muss aus diesem Grund ein Username mit der Funktion FTP\_UserName übermittelt werden. Für öffentliche Server hat sich der Name Anonymous zum Standard entwickelt. Hat der Server den Usernamen akzeptiert, muss anschließend der Server-Response in der globalen Variablen FTP\_Client\_uiLastServerResonse geprüft werden. Ist dieser Code = 230 („login ok“), so ist die Anmeldung am Server abgeschlossen. Wurde allerdings der Code 331 gesendet („username ok; send password“), so erwartet der Server zur vollständigen Anmeldung noch ein Passwort. Dieses wird mit der Funktion FTP\_Password übermittelt. Bei Verwendung des Benutzernamens Anonymous, kann das Passwort beliebig gewählt werden. Hat der Server das Passwort akzeptiert, ist die Anmeldung am Server abgeschlossen.

#### 3.2.2 Verzeichnisfunktionen

Die Funktion FTP\_ChangeDirectory wird zum Verzeichniswechsel auf dem Remote-Host verwendet. Als Eingabeparameter wird der Verzeichnisname benötigt, in den gewechselt werden soll. Diese Funktion funktioniert wie die bekannte MS-DOS Funktion CD.

Die Funktion FTP\_MakeDirectory erstellt in neues Verzeichnis. Dabei ist die Länge des gewünschten Verzeichnisnamens zu beachten. Manche Geräte (z.B. HPG 300) lassen nur 8 Zeichen zu. Als Verzeichnisnamen können absolute und relative Bezeichnungen eingegeben werden.

Die Funktion FTP\_DeleteDirectory dient zum löschen von Verzeichnissen. Als Verzeichnisnamen können absolute und relative Bezeichnungen eingegeben werden.

Die Funktion `FTP_WorkingDirectory` liefert nach dem Aufruf das aktuelle Arbeitsverzeichnis auf dem FTP-Server zurück.

**Achtung:** Die Ausgangsvariable behält bis zum nächsten Aufruf dieser Funktion ihren aktuellen Wert. Auch wenn zwischenzeitlich die Funktion `FTP_ChangeDirectory` angewendet wurde.

### 3.2.3 Dateifunktionen

Die Funktion `FTP_UploadFile` dient dem Upload einer Datei auf den Server. Als Eingangsparameter werden der Dateiname und das Verzeichnis der Datei benötigt, die auf den Server geladen werden soll. Die Inputvariable `sDirectory` muss unbedingt mit einem Backslash abgeschlossen werden, z.B.: `,disk_mmc\daten\'` oder für das Rootverzeichnis `,\'`. Die Abarbeitungsdauer dieser Funktion ist abhängig von der Dateigröße und von dem eingestellten Taskintervall.

Die Funktion `FTP_DownloadFile` wird zum Dateidownload verwendet. Als Eingabeparameter sind der Name der Datei auf dem Server und der Name des Verzeichnisses, in welches die Datei gespeichert werden soll, notwendig. Für die Inputvariable `xDirectory` gelten die selben Vorschriften wie für `FTP_UploadFile`. Der Eingangsparameter `xOverwrite` stellt einen Schreibschutz dar. Wenn dieser Wert auf `FALSE` gesetzt wird, so kann eine existierende Datei nicht überschrieben werden und die Funktion kehrt mit einer entsprechenden Fehlermeldung zurück. Wird dies jedoch gewünscht, so muss diese Eingangsvariable auf `TRUE` gesetzt werden.

Diese Funktion benötigt ebenfalls, in Abhängigkeit von der Dateigröße, mehrere Zyklen.

**Achtung:** Wird bei der Funktion `FTP_DownloadFile` kein Verzeichnis angegeben, so wird die Datei in das aktuelle Verzeichnis auf der PLC abgelegt. Da zurzeit keine lokalen Verzeichnisfunktionen zur Verfügung stehen, kann dies ein x-beliebiges Verzeichnis sein. Aus diesem Grund muss immer das Verzeichnis immer mit angegeben werden.

Die Funktion `FTP_DeleteFile` wird zum Löschen von Dateien auf dem Server verwendet. Im Gegensatz zum PC ist dieser Vorgang nicht reversibel. Als Inputvariable wird der Dateiname in `sFileName` eingefügt. Der Dateiname darf inklusive Dateikennung maximal 30 Zeichen lang sein. Mit einer positiven Flanke an `xExecute` wird der Vorgang gestartet. Bei erfolgreicher Ausführung wird die Ausgabevariable `xDone` für einen Zyklus auf `TRUE` gesetzt. Im Fehlerfall wird `xError` auf `TRUE` gesetzt.

### 3.2.4 Abmelden vom FTP-Server

Die Abmeldung vom Server erfolgt nach einem vorgegebenen Prinzip. Dafür steht die Funktion `FTP_Disconnect` zur Verfügung. Sie sollte nach erledigter Dateioperation aufgerufen werden, um die FTP-Sitzung zu beenden. Wird sie nicht aufgerufen und ist der FTP-Client für eine gewisse Zeit inaktiv, so unterbrechen manche FTP-Server die Verbindung.

### 3.2.5 Die Funktion `FTP_GetLastError`

Mit dieser Funktion kann mithilfe der globalen Variablen `FTP_Client_wErrorID` ein String bezogen werden, der die Fehlerursache im Klartext enthält.

Bsp.: `sErrorString:=FTP_GetLastError(xDummy:=TRUE);`

Error-Codes und die zugehörigen Strings:

|    |   |
|----|---|
| 0  | : 'No Error';   |
| 10 | : 'unable to retrieve host-address';  |
| 11 | : 'unable to get local host-name';  |
| 12 | : 'socket creation failed';   |
| 13 | : 'receiving server response timed out';  |
| 14 | : 'transmitting PORT command failed';   |
| 15 | : 'unexpected server response';   |
| 16 | : 'server is shutting down';  |
| 17 | : 'transmitting CWD command timed out';   |
| 18 | : 'transmitting CWD command failed';  |
| 19 | : 'transmitting QUIT command failed';   |
| 20 | : 'unexpected server response. connection closed';                              |
| 21 | : 'creating file failed';   |
| 22 | : 'FileName already exists! overwrite isn't allowed';                           |
| 23 | : 'listen for data connection failed';  |
| 24 | : 'port function failed';   |
| 25 | : 'transmitting RETR command failed';   |
| 26 | : 'opening file failed';  |
| 27 | : 'accepting data connection failed';   |
| 28 | : 'writing data to file failed, disk full';                                     |
| 29 | : 'FTP command connection already open';  |
| 30 | : 'registering callback-functions failed';                                      |
| 31 | : 'connecting to remote host failed';   |
| 32 | : 'transmitting PASS command timed out';  |
| 33 | : 'transmitting PASS command failed';   |
| 34 | : 'opening file failed';  |
| 35 | : 'transmitting USER command timed out';  |
| 36 | : 'transmitting USER command failed';   |
| 37 | : 'transmitting XPWD command failed';   |
| 38 | : 'server closed connection; command socket closed too; FTP-Client not active'; |
| 39 | : 'transmitting QUIT command timed out';  |
| 40 | : 'transmitting file failed';   |
| 41 | : 'transmitting PORT command timed out';  |
| 42 | : 'receiving server response timed out; connection closed';                     |
| 43 | : 'transmitting RETR command timed out';  |
| 44 | : 'waiting for incoming connection request timed out';                          |
| 45 | : 'transmitting STOR command timed out';  |
| 46 | : 'transmitting STOR command failed';   |
| 47 | : 'transmitting XPWD command timed out';  |
| 48 | : 'transmitting DELE command timed out';  |
| 49 | : 'transmitting DELE command failed';   |
| 50 | : 'transmitting TYPE command timed out';  |
| 51 | : 'transmitting TYPE command failed';   |
| 52 | : 'transmitting RMD command timed out';   |
| 53 | : 'transmitting RMD command failed';  |
| 54 | : 'transmitting MKD command timed out';   |
| 55 | : 'transmitting MKD command failed';  |

Alle anderen Codes werden als 'unknown error' quittiert.

### 3.2.6 Die Funktion *FTP\_GetLastServerResponse*

Diese Funktion liefert mit Hilfe der globalen Variablen `FTP_Client_uiLastServerResponse` die Bedeutung des letzten Response-Codes im Klartext zurück. Es ist keine Inputvariable erforderlich.

Bsp.: `sResponse:=FTP_GetLastServerResponse(xDummy:=TRUE);`

Response-Codes und die dazugehörigen Strings:

|      |   |
|------|---|
| 0:   | 'no server response already received';                |
| 120: | 'server not ready yet. try again sometime later';     |
| 150: | 'file status ok; about to open data connection';      |
| 200: | 'command ok';   |
| 202: | 'command not implemented';                            |
| 220: | 'server ready; send username';                        |
| 221: | 'server is closing control connection';               |
| 226: | 'server is closing data connection';                  |
| 230: | 'login ok';   |
| 250: | 'requested file action ok';                           |
| 257: | 'requested directory action ok';                      |
| 331: | 'username ok; send password';                         |
| 332: | 'need account for login';                             |
| 421: | ' !!! server is shutting down !!! ';                  |
| 425: | 'can't open data connection';                         |
| 426: | 'connection closed; transfer aborted';                |
| 500: | 'unknown command';                                    |
| 501: | 'error in parameter';                                 |
| 502: | 'command not supplied';                               |
| 503: | 'wrong command sequence';                             |
| 504: | 'parameter not allowed';                              |
| 530: | 'user not registered';                                |
| 532: | 'user account needed for storing files on server';    |
| 550: | 'command not executed; unknown file';                 |
| 551: | 'command not executed';                               |
| 552: | 'command not executed; not enough memory';            |
| 553: | 'command not executed; invalid filename';             |
| 554: | 'command not executed; invalid file-mark';            |
| 555: | 'command not executed; invalid TYPE or STRU command'; |

Alle anderen Response-Codes werden mit 'unknown response' quittiert.



## 4 Versionsindex

| Release; Implementierung | Änderung       | Bearbeiter        |
|--------------------------|----------------|-------------------|
| 01; 00                   | Ersterstellung | Matthias Blümling |
|                          |                |                   |
|                          |                |                   |
|                          |                |                   |