



Setup of FTP client in CODESYS

Abstract:

This application note provides an example how to set up an FTP client within CODESYS, using the free OSCAT library. It describes the integration of the OSCAT-library and the configuration of an FTP client in your PLC project.

Setup of FTP client in CODESYS

Hardware reference

No.	Component name	Article No.	Hardware / Firmware version
1	UC20-WL2000-AC	1334950000	u-OS 2.0.0
2	IoT-GW30	2682620000 2682630000	u-OS 2.0.0

Software reference

No.	Software name	Article No.	Software version
1	CODESYS Development System		SP18 Patch 4
2	CODESYS Runtime App		4.7.0.0-2
3	CODESYS Control SL for Weidmueller u-OS		4.7.0.0
4	OSCAT NETWORK Library for CODESYS V3 http://www.oscat.de/		1.3.5.2
5	Mozilla FTP Server		

File reference

No.	Name	Description	Version
1	Example_FTP_Client.project	Example CODESYS project	

Contact

Weidmüller Interface GmbH & Co. KG
Klingenbergstraße 26
32758 Detmold, Germany
www.weidmueller.com

For any further support please contact your
local sales representative:
<https://www.weidmueller.com/countries>

Content

1	Warning and Disclaimer.....	4
2	Introduction.....	5
3	Add the OSCAT Library to CODESYS.....	6
4	FTP client in CODESYS	7
4.1	Function Block.....	8
4.2	Function Block Error	9
5	Example Program.....	10
5.1	Download a file from the server to the client.	10
5.2	Upload of a file to the server	11

1 Warning and Disclaimer

Warning

Controls may fail in unsafe operating conditions, causing uncontrolled operation of the controlled devices. Such hazardous events can result in death and / or serious injury and / or property damage. Therefore, there must be safety equipment provided / electrical safety design or other redundant safety features that are independent from the automation system.

Disclaimer

This Application Note / Quick Start Guide / Example Program does not relieve you of the obligation to handle it safely during use, installation, operation and maintenance. Each user is responsible for the correct operation of his control system. By using this Application Note / Quick Start Guide / Example Program prepared by Weidmüller, you accept that Weidmüller cannot be held liable for any damage to property and / or personal injury that may occur because of the use.

Note

The given descriptions and examples do not represent any customer-specific solutions, they are simply intended to help for typical tasks. The user is responsible for the proper operation of the described products. Application notes / Quick Start Guides / Example Programs are not binding and do not claim to be complete in terms of configuration as well as any contingencies. By using this Application Note / Quick Start Guide / Example Program, you acknowledge that we cannot be held liable for any damages beyond the described liability regime. We reserve the right to make changes to this application note / quick start guide / example at any time without notice. In case of discrepancies between the proposals Application Notes / Quick Start Guides / Program Examples and other Weidmüller publications, like manuals, such contents have always more priority to the examples. We assume no liability for the information contained in this document. Our liability, for whatever legal reason, for damages caused using the examples, instructions, programs, project planning and performance data, etc. described in this Application Note / Quick Start Guide / Example is excluded.

Security notes

In order to protect equipment, systems, machines and networks against cyber threats, it is necessary to implement (and maintain) a complete state-of-the-art industrial security concept. The customer is responsible for preventing unauthorized access to his equipment, systems, machines and networks. Systems, machines and components should only be connected to the corporate network or the Internet if necessary and appropriate safeguards (such as firewalls and network segmentation) have been taken.

2 Introduction

The File Transfer Protocol is a network protocol for transferring files over IP networks. It is used to copy files from the client to the server (upload), from the server to the client (download) or client-controlled between two FTP servers (File Exchange Protocol). In addition, FTP can be used to create and read out directories and to rename or delete directories and files.

Currently, there is no native CODESYS FTP client available, but it is possible to use a free library for CODESYS called *OSCAT Network Library*. This library contains an FTP client function block. The FTP_CLIENT function block is used to transfer files from the PLC to an FTP server and to transfer files from the FTP server to the PLC.

This application note explains how to install the OSCAT library (chapter 3), how to add the FTP client FB, configure and use it (chapter 4). For testing, a free ftp server can be used, as described in chapter 5. Additionally, there is a CODESYS project attached to this document, containing an example application of the FTP client.

3 Add the OSCAT Library to CODESYS

1. Open the CODESYS Store web page and search for OSCAT.
2. Download the current OSCAT Network library.

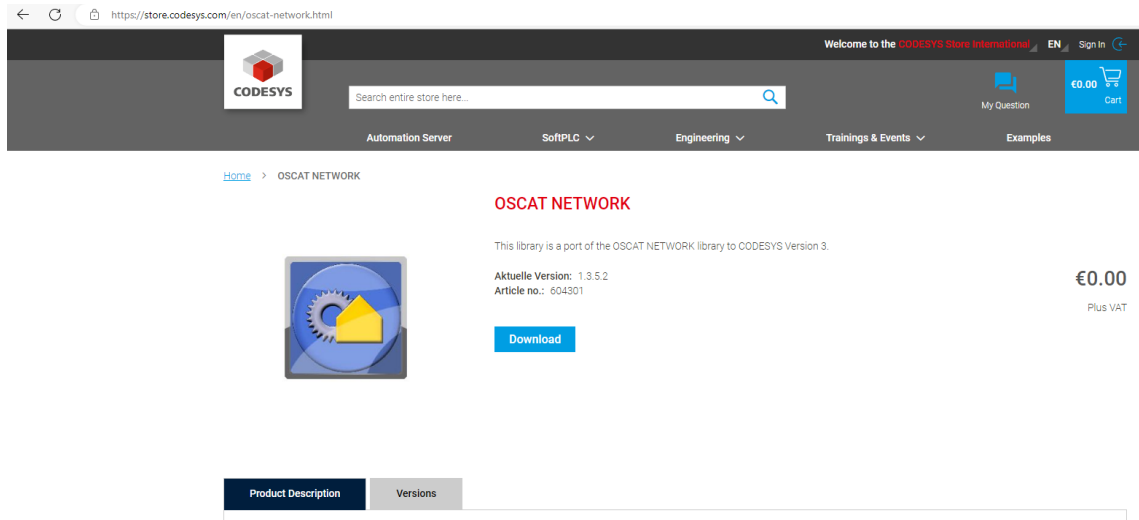


Figure 1: OSCAT library



Note:

You need a CODESYS account to download the library from the store.

3. Open the explorer and navigate to the OSCAT Network.package download file, double-click it to open the installation, or open the installed CODESYS Installer and select the downloaded file. Install the package.
4. Open CODESYS and check if the library is available inside the Library Repository.

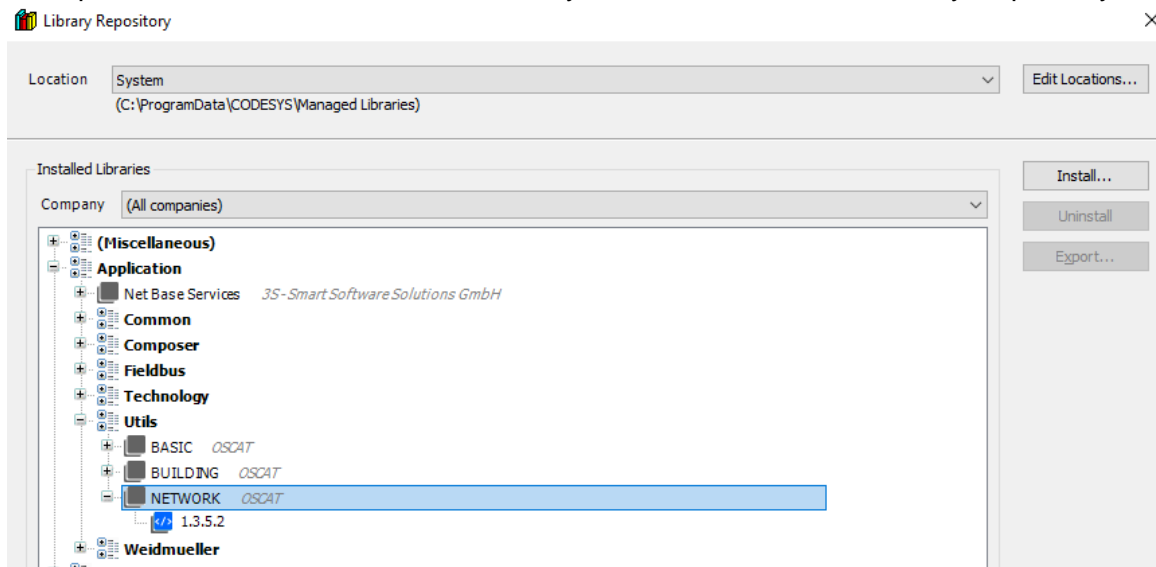


Figure 2: OSCAT Network Library

4 FTP client in CODESYS

1. Open a CODESYS project and open the “Library Manager”.
2. Then insert the OSCAT Network library. ⇒ Library manager ⇒ Add library... ⇒ Network

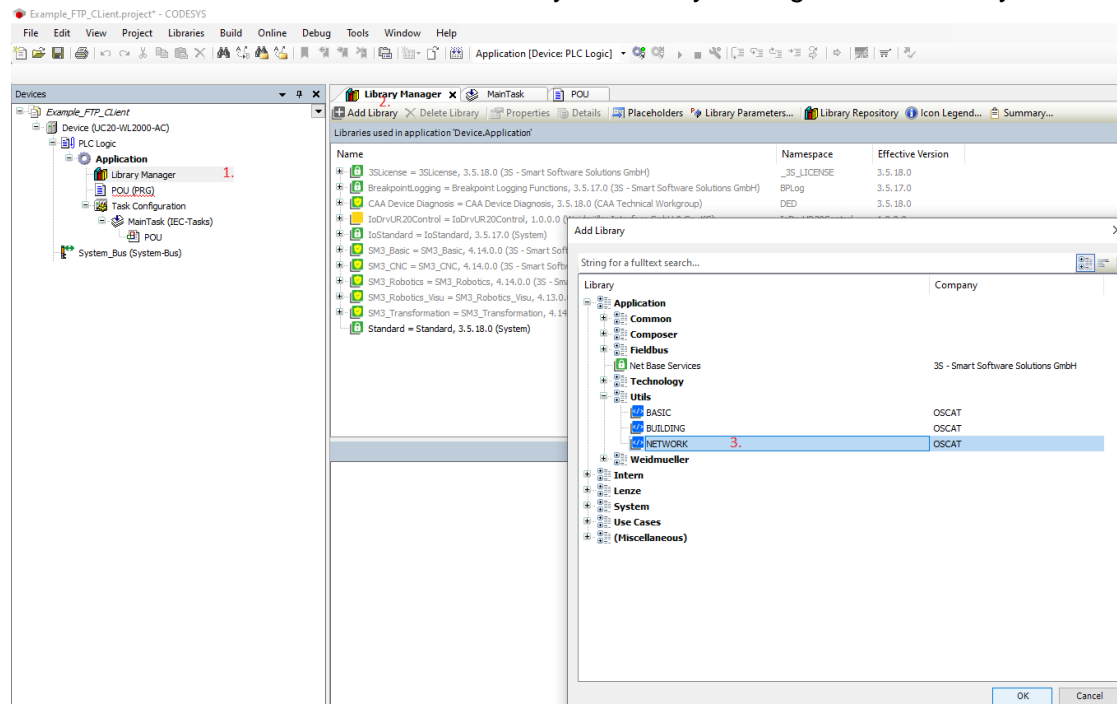


Figure 3: Add OSCAT library

3. Check if all sub-libraries can be found and referenced, or whether some libraries may need to be installed. Install missing libraries via download button “Download Missing Libraries”.

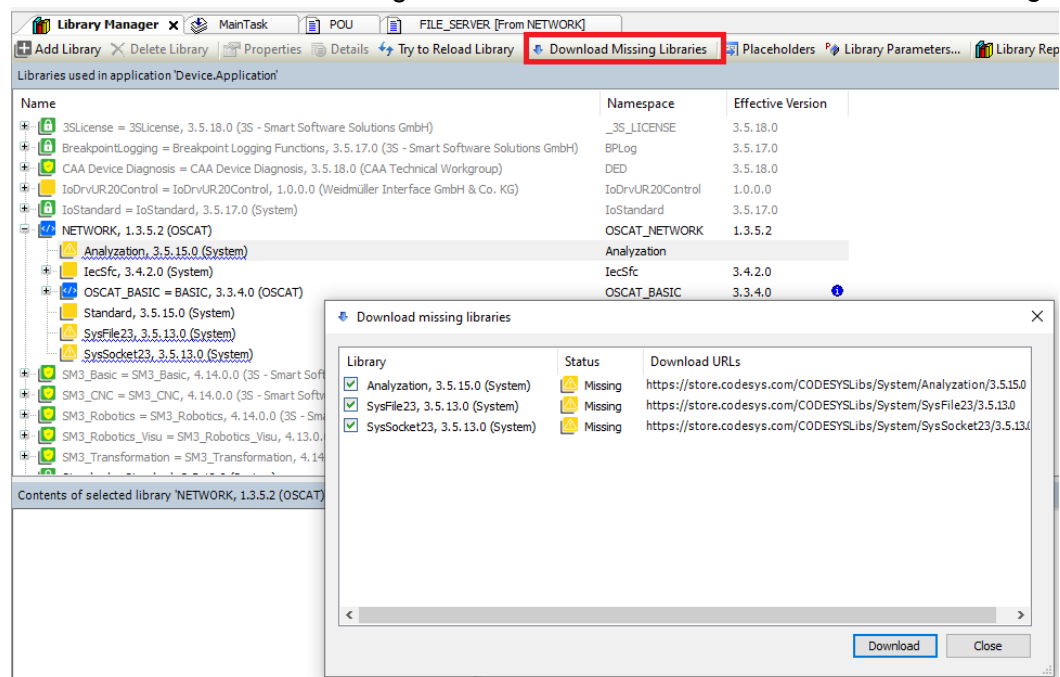


Figure 4: Download missing libraries

4. Open or create a program (PRG) or function block (FB) and use the FTP_CLIENT function block from the OSCAT network library.

4.1 Function Block

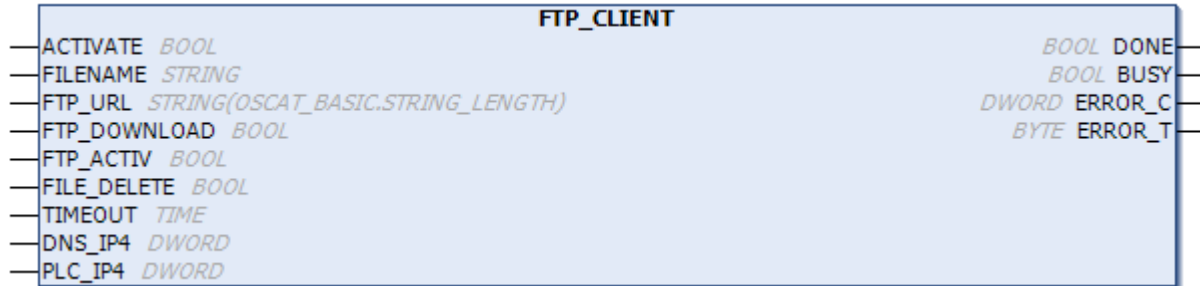


Figure 5: Function Block

- A positive edge at ACTIVATE starts the transfer process.
- FTP_URL contains the name of the FTP server and optionally passes username and password, an access path, and an additional port number for the data channel. If no username or password is passed, the module will automatically try to connect as "Anonymous".
URL examples:
ftp://username:password@servername:portnummer/directory/
ftp://username:password@servername
ftp://username:password @servername/directory/
ftp://servername
ftp://username:password@192.168.1.1/directory/
ftp://192.168.1.1
- FTP_DOWNLOAD specified the transmission direction.
- The parameter FTP_ACTIV determines whether the FTP server is operated in active or passive mode. In the ACTIV mode, the FTP server tries to establish the data channel for control, however these may cause problems by security software, firewall, etc. because these could block the connection request. For this purpose, in the firewall a corresponding exception rule has to be defined. In the passive mode, this problem is alleviated since the controller establishes the connection and can easily pass through the firewall. The control channel is always set up on port 20, and the data channel via standard PORT21, but this depends on whether active or passive mode is used, or optional PORT number in the FTP-URL is specified.
- the parameter FILE_DELETE can be determines whether the source file should be deleted after successful transfer. This works on FTP and even on the control side.
- In specifying FTP directories, the behavior depends on FTP server. Directories might already exist or are created automatically. Normally, they should be already available.
- The size of files is not explicitly limited, but there are practical limits, such as free space on PLC, FTP storage and the transmission time.

- With DNS_IP4 the IP address of the DNS server must be specified, if the FTP URL contains a DNS name. Alternatively, an IP address can be entered in the FTP URL. At parameters PLC_IP4 the controller's IP address must be provided.
- If errors occur during transmission these are passed to the outputs ERROR_C and ERROR_T. While the transfer is running, BUSY = TRUE, and after a successful completion of the operation, DONE = TRUE. Once a new transfer is started, DONE, ERROR_T and ERROR_C are reset.

4.2 Function Block Error

ERROR_T:

Value Properties

- 1 Problem: DNS_CLIENT
The exact meaning of ERROR_C can be read at module DNS_CLIENT
- 2 Problem: FTP control channel
The exact meaning of ERROR_C can be read at module IP_CONTROL
- 3 Problem: FTP data channel
The exact meaning of ERROR_C can be read at module IP_CONTROL
- 4 Problem: FILE_SERVER
The exact meaning of ERROR_C can be read at block FILE_SERVER
- 5 Problem: END - TIMEOUT
ERROR_C contains the left WORD of the step number, and the right WORD has the response code received by the FTP server.
The parameters must be considered first as a HEX value, divided into two WORDS, and then be considered as a decimal value.
Example:
ERROR_T = 5
ERROR_C = 0x0028_00DC
End-step number 0x0028 = 40
Response-Code 0x00DC = 220



Note:

Please also refer to the OSCAT manual and read the function block description.

Documentation Link:

<http://www.oscat.de/de/component/jdownloads/category/3-oscat-network.html?Itemid=0>

5 Example Program

For the example, a free available FTP Server with the program “Mozilla FTP Server” is used. This Server program is installed and running on a PC and connected to a u-OS device via Ethernet.

The server has the IP-Address: 192.168.1.100

The client device has the IP-Address: 192.168.1.101

The Server has the following user credentials:

User: admin

Password: Detmold



Note:

The example does not explain how to configure and execute the server. For example, setting up the server might require deactivating the firewall on the FTP Server device.

5.1 Download a file from the server to the client.

- Adjust the settings like shown in Figure 6: Example Program Download to download the file with the name: Test_FTP21.txt

Expression	Type	Value
FTP_CLIENT	FTP_CLIENT	
ftp_activate	BOOL	TRUE
ftp_done	BOOL	TRUE
ftp_busy	BOOL	FALSE
ftp_error_c	DWORD	0
ftp_error_t	BYTE	0
ftp_url	STRING	'ftp://admin:Detmold@192.168.1.100'
ftp_filename	STRING	'Test_FTP21.txt'
ftp_download	BOOL	TRUE
ftp_activ	BOOL	FALSE
file_delete	BOOL	FALSE
ftp_timeout	TIME	T#30s
zeit	TIME	T#30s
stop	BOOL	FALSE
start	BOOL	TRUE
reset	BOOL	FALSE
error	DWORD	0
ok	DWORD	1
COUNT_DR_1	OSCAT_NETWORK...	
COUNT DR 2	OSCAT NETWORK...	


```

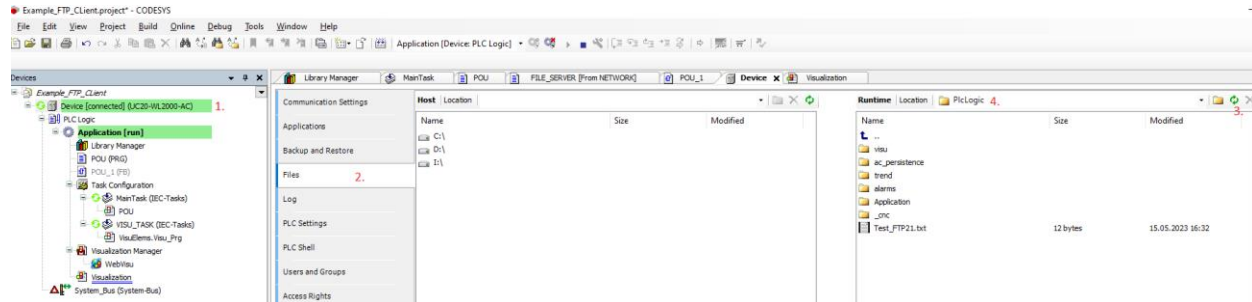
47  *)
48  (***** )
49
50
51  (*Count error *)
52  COUNT_DR_1(UP:=stop[FALSE],CNI:=0=>error:=0);
53
54  (*Count read/send *)
55  COUNT_DR_2(UP:=ftp_done[TRUE],CNI:=1=>ok:=1);
56
57  FTP_CLIENT(
58    ACTIVATE[TRUE]:=ftp_activate[TRUE], (* ACTIVATE: BOOL (positive edge starts the query) *)
59    FILENAME[Test_FTP21]:=ftp_filename[Test_FTP21], (* FILE NAME: STRING (file path/ filename) *)
60    FTP_URL[ftp://admin:=ftp_url[ftp://admin], (* FTP URL: STRING(STRING LENGTH) (FTP access path) *)
61    FTP_DOWNLOAD[TRUE]:=ftp_download[TRUE], (* FTP_DOWNLOAD : BOOL (UPLOAD = 0 / DOWNLOAD = 1) *)
62    FTP_ACTIV[FALSE]:=ftp_activ[FALSE], (* FTP_ACTIV : BOOL (PASSIV = 0 / ACTIV = 1) *)
63    FILE_DELETE[FALSE]:=file_delete[FALSE], (* FILE_DELETE: BOOL (delete files after transfer) *)
64    TIMEOUT[T#30s]:=ftp_timeout[T#30s], (* TIMEOUT: TIME (time) *)
65    DNS_IP4[192.168.1.100]:=OSCAT_NETWORK.OSCAT_BASIC.DWORD_OF_BYTE(8,8,8,8), (*Dns_ip4: DWORD (IP4 address of the DNS server)*)
66    PLC_IP4[323235877]:=OSCAT_NETWORK.OSCAT_BASIC.DWORD_OF_BYTE(192,168,1,101), (*Plc_ip4: DWORD (IP4 address of own ip address) *)
67    DONE[TRUE]=>ftp_done[TRUE], (*DONE: BOOL (Transfer completed without error) *)
68    BUSY[FALSE]=>ftp_busy[FALSE], (* BUSY: BOOL (Transfer active) *)
69    ERROR_C[0]=>ftp_error_c[0], (* ERROR_C: DWORD (Error code) *)
70    ERROR_T[0]=>ftp_error_t[0]; (* ERROR_T: BYTE (Problem type) *)
71
72  RETURN
  
```

Figure 6: Example Program Download

Setup of FTP client in CODESYS

After a successful download, the file should be stored on the controller. You can check the file system of the device if the file is available.

Open file browser of CODESYS ⇒ Device ⇒ Files ⇒ Network ⇒ Refresh ⇒ Location ⇒ /PLCLogic ⇒ yourfile.txt



5.2 Upload a file to the server

- Adjust the settings like shown in Figure 7: Upload file to server to upload the file with the name: TestFileFTP.txt that is contained inside the PLCLogic Folder of the controller.

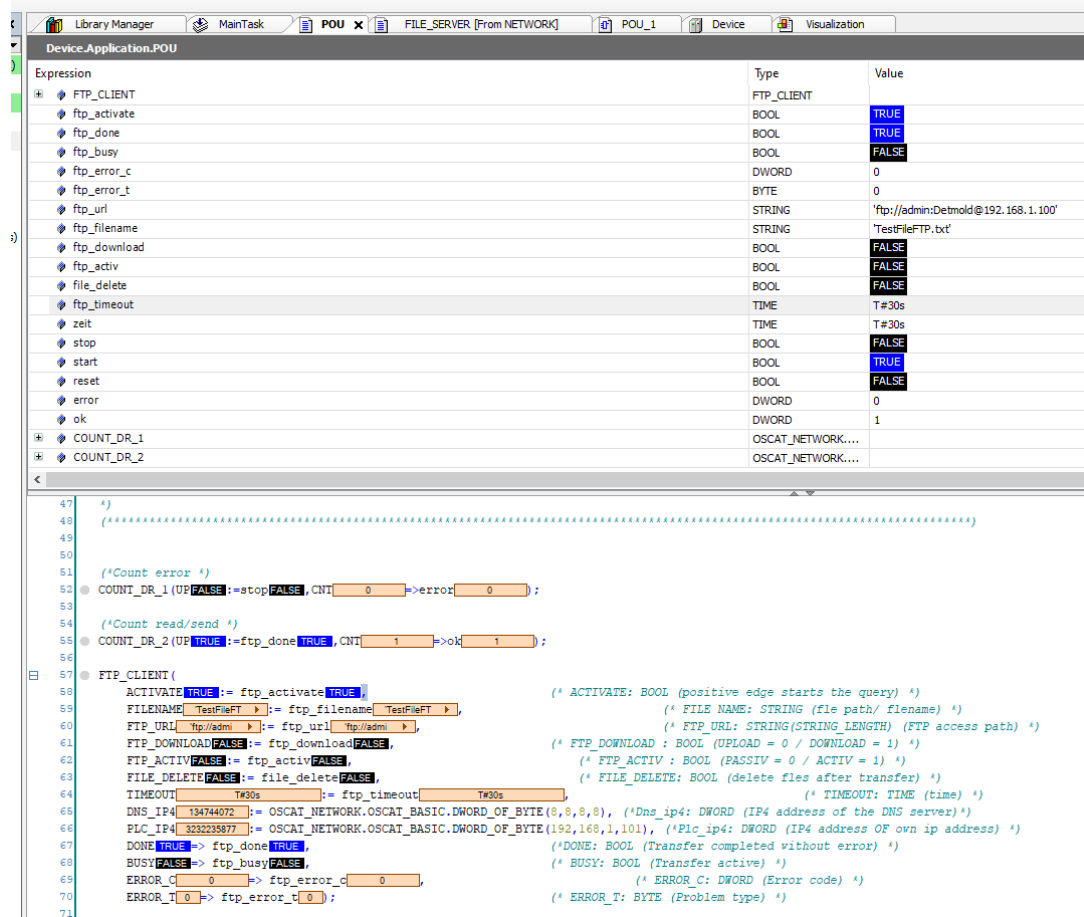


Figure 7: Upload file to server