

Application: “Sending an E-Mail as a Process Event Message with SMTP Communication Blocks”

SIMATIC STEP 7

[Application Description](#) • January 2011

Applications & Tools

Answers for industry.

SIEMENS

Industry Automation and Drive Technologies Service & Support Portal

This document is taken from the Service Portal of Siemens AG, Industry Automation and Drive Technologies. The following link takes you directly to the download page of this document.

<http://support.automation.siemens.com/WW/view/en/46817803>

If you have any questions concerning this document please e-mail us to the following address:

online-support.automation@siemens.com

S

SIMATIC

Sending an E-Mail as a Process Event Message with SMTP Communication Blocks

Communication with S7-300

Automation Problem

1

Automation Solution

2

**Functional Mechanisms
of this Application**

3

Installation and Startup

4

**Operation of the
Application**

5

Appendices

6

References

7

History

8

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are correctly used. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that Siemens cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – then the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc. described in this application example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health. The above provisions do not imply a change in the burden of proof to your detriment.

It is not permissible to transfer or copy these application examples or excerpts of them without first having prior authorization from Siemens Industry Sector in writing.

Table of Contents

Warranty and Liability	4
Table of Contents.....	5
1 Automation Problem	6
2 Automation Solution	7
2.1 Overview of the overall solution	7
2.2 Description of the core functionality	7
2.3 Hardware and software components used.....	8
3 Functional Mechanisms of this Application	10
3.1 Overview of the CPU configuration	10
3.2 Functionality of FB 490.....	10
3.2.1 Dependencies	10
3.2.2 Program details on block FB 490	11
3.2.3 Parameter list of FB 490.....	12
3.2.4 Reset of FB 490	15
3.2.5 E-mail transmission	16
3.2.6 Error messages for FB 490: Values of the “STATUS” and “SFC_STATUS” status variables	17
4 Installation and Startup.....	21
4.1 Hardware configuration (example)	21
4.2 Installing the STEP 7 configuration	21
4.3 Establishing an Internet connection to the SMTP server	21
4.4 Extending the local data memory	22
5 Monitoring and Modifying the Example	23
5.1 Using data blocks and the variable table	23
5.2 Sending an e-mail with the aid of the SMTP block FB 490	27
6 Appendices	29
6.1 SMTP commands: Minimum command set	29
6.2 SMTP connection process between client and server	29
6.3 Status and error messages of an SMTP server	32
7 References	34
7.1 References	34
7.2 Internet links	34
7.2.1 Relevant RFCs	35
8 History.....	36

1 Automation Problem

Introduction

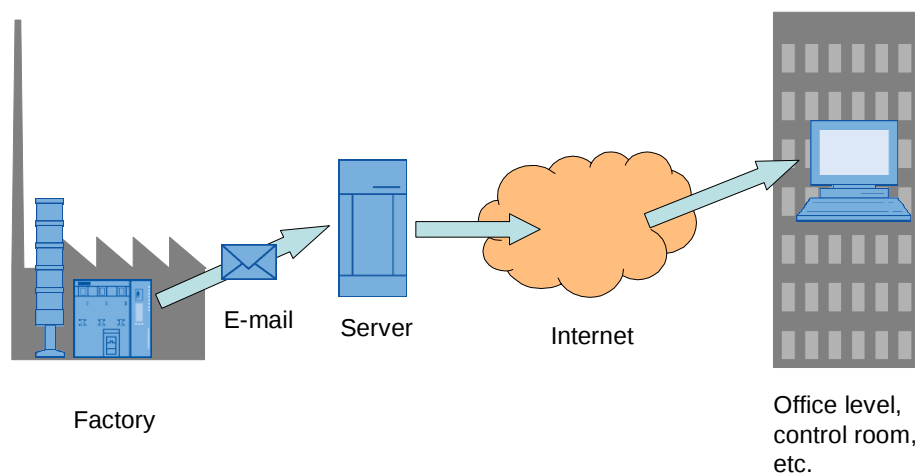
The growing degree of networking in the automation industry also increases the demand for the exchange of messages beyond the limits of local area networks (LANs).

One possibility for such communications is the use of e-mail services via the Internet.

Overview of the automation problem

The figure below provides an overview of the automation problem.

Figure 1-1



In an automation or process system, the sending of an e-mail is to be triggered depending on any operational event.

Within the scope of the configuration, any recipient can be configured. The content of the e-mail is also to be freely configurable and is to include, for example, a plain-text message of the event that has occurred or even specific operating variables.

Increased operational safety supports different mail server authentication methods. This prevents the communication from being influenced by unauthorized third parties.

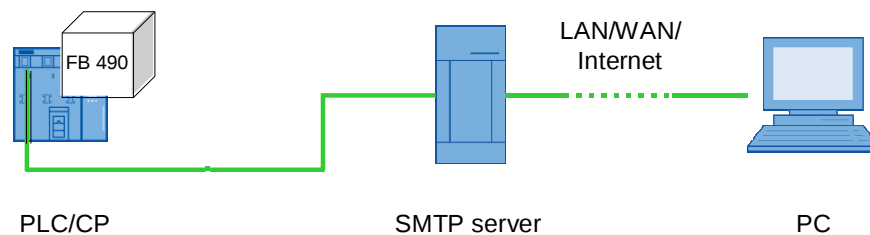
2 Automation Solution

2.1 Overview of the overall solution

Diagrammatic representation

The diagrammatic representation below shows the most important components of the solution:

Figure 2-1



The core of the automation solution is FB490, which, in conjunction with other SIMATIC blocks for open communication, controls the sending of e-mails via the Internet.

While the open communication blocks control the basic TCP/IP services, FB490 operates the higher-level e-mail transmission services and communicates with the mail server. The current Internet standard SMTP ("Simple Mail Transfer Protocol") is used as a protocol.

Once the SMTP mail server has received the message, it automatically sends this message to a PC with any recipient's address via the connected network (LAN, WAN or Internet).

Scope

This application does not include a description of TCP/IP communication basics or the use of Simple Mail Transfer Protocols. Basic knowledge of these fields is assumed or can be found in the documents listed in the appendix.

The application describes only the use and properties of the used communication blocks. It is assumed that all other components are available, in particular a correctly configured SMTP server. The configuration of these components is not part of this application.

2.2 Description of the core functionality

FB 490 is an asynchronous communication block. It allows to send an e-mail from an S7 CPU that supports open communication to any SMTP server to which the PLC is connected via Industrial Ethernet (not necessarily via PROFINET) at any time. This means that the PLC can be programmed so that the sending of the e-mail is always triggered as a response to any operational event (messages, values of operating variables, expiry of deadlines, ...).

Among other things, the configuration enables the user to set the following parameters:

- IP address of the SMTP server
- User name and password for authentication to the SMTP server if required
- Recipient's address and where necessary CC recipient addresses
- Subject line
- Priority
- ASCII plain text and where necessary an e-mail attachment

FB 490 is not intended for *receiving* messages (POP).

Advantages of this solution

The solution presented here offers you the following advantages:

- It is based on the most commonly used standard for message transmission via the Internet.
- SMTP is a powerful, established, flexible method.
- An Ethernet structure is available in nearly every location, i.e. there are hardly any additional costs.
- SMTP mail services are available worldwide.
- Recipients have numerous options to check the e-mails (PC, smartphone, ...).

2.3 Hardware and software components used

Hardware components

FB 490 and the underlying open communication blocks can be executed on the following hardware platforms:

- CPU 41x-3 PN/DP
- CPU 315-2 PN/DP and CPU 317-2 PN/DP (with firmware V 2.5 or higher)
- CPU 319-3 PN/DP
- IM 151-8 PN/DP CPU
- IM 154-8 CPU
- WinAC RTX (F)

For more details regarding open communication, see link \8\.

Standard software components

The blocks are executable in STEP 7 V5.5 and higher with S7-SCL 5.3+SP5 and higher. On WinAC computers, the WinAC application version 2.6 or higher can be operated.

Table 2-1

Component	Qty.	MLFB / order number	Note
STEP 7 V5.5	1	6ES7810-4AA00-0YX0	Or higher version
S7-SCL V5.3+SP5	1	6ES7811-1AA00-0YX0	Or higher version

NOTICE

If an error message is displayed in your project in HW Config when compiling, you have to update the HSB files to be able to start up the application.

To do so, follow the instructions in FAQs 23183356 (link \6\) or 22374837 (link \7\) of the Automation Support portal!

Sample files and projects

The following list contains all files and projects that are used in this example.

Table 2-2

Component	Note
46817803_SMTP-Mailversand_CODE.zip	Sample project and function block
46817803_SMTP-Mailversand_DOKU_v10_en.zip	This document.

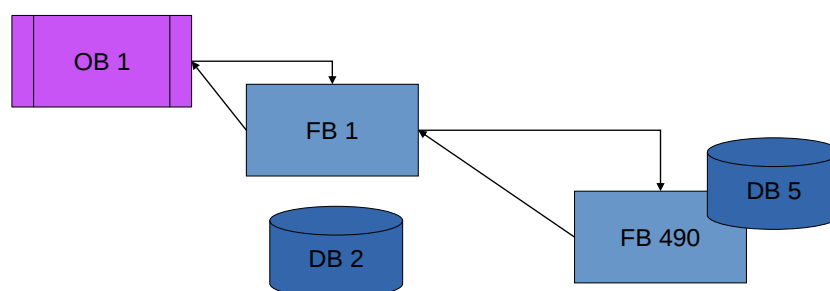
NOTE

Please note that the different blocks of the project are supplied with Know-How Protect. This means that you can use these blocks in your projects, but you can neither view nor change their source texts.

3 Functional Mechanisms of this Application

3.1 Overview of the CPU configuration

Figure 3-1



The configuration in the application example is kept very simple and only used to explain the basic function.

OB 1 cyclically calls FB 1, which in turn calls FB 490 in each cycle. The parameters for calling FB 490 are stored in DB 2 while DB 5 is the instance data block for FB 490.

All communication with the SMTP server takes place via FB 490.

Except for serving as a wrapper for FB 490, the only function of FB 1 is to transfer the transient error codes (*STATUS*, *SFC_STATUS*, see 3.2.3) to permanent flags if an error occurs so that they are available for later analysis.

3.2 Functionality of FB 490

3.2.1 Dependencies

FB 490 needs the following function blocks and SFCs from the STEP 7 standard library ("IEC Function Blocks" folder):

Table 3-1

FC	SFC
FC 10 "EQ_STRING"	SFC 1
FC 11 "FIND"	SFC 20
FC 17 "INSERT"	SFC 24
FC 20 "LEFT"	SFC 51
FC 21 "LEN"	SFC 58
FC 32 "RIGHT"	SFC 59
	SFC 64
	SFC 131
	SFC 132
	SFC 133
	SFC 134

These blocks must be copied to the project to be able to operate FB 490.

In the sample project, DB 5 is configured as an instance data block.

3.2.2 Program details on block FB 490

FB 490 is an asynchronous function block. Since processing the sending of an e-mail covers several CPU cycles, the FB must be called at regular intervals (ideally in each CPU cycle) at least in the time between triggering the sending of the e-mail and terminating the dialog with the SMTP server.

Supported authentication methods

FB 490 supports two e-mail sending methods:

- Anonymous, i.e. without user authentication. Nowadays, this method is rarely used for data security reasons. "Open mail relays" that do not require user identification are considered as potential channels for SPAM and malware.
- SMTP AUTH, currently considered as the standard for configuring mail servers. With Extended SMTP, it is available on almost all common mail servers.

SMTP AUTH offers four different protocols for handling the authentication: *login*, *plain*, *CRAM-MD5* and *NTLM*.

FB 490 only supports the *login* protocol.

Syntax checks

For performance reasons, the syntax of transferred e-mail addresses is not checked for correctness. This applies particularly to the `TO_S`, `CC` and `FROM` parameters listed below.

3.2.3 Parameter list of FB 490

When calling, the following parameters are transferred to FB 490. The respective parameter type (in, in/out, out), the data type and the mandatory or optional assignment of the parameter are listed in brackets.

REQ (in, BOOL, mandatory)

This parameter ("Send REQuest") controls the overall e-mail transmission process.

A rising edge 0 → 1 initiates a new e-mail transmission process; in all other cases, a transmission process that has already started is continued or – if it is no longer necessary to process a transmission job – the block returns immediately.

(However, see also COM_RST).

For details on the transmission process, see 3.2.5.

ADR_MAIL_SERVER (in, DWORD, mandatory)

The absolute IP address of the SMTP server to which the FB transfers the e-mail transmission job.

The address is transferred as a string of four bytes; the most significant byte of the DWORD contains the first byte of the IP address, the least significant byte of the DWORD includes the last byte of the IP address. A parameter value

"DW#16#C0A800C8" thus represents IP address 192.168.0.200. (16#C0 = 192, etc.)

WATCH_DOG_TIME (in, TIME, mandatory)

Specifies the maximum time that may pass until a connection is successfully established.

If a connection to the specified SMTP server cannot be established within this specified time, the block automatically disconnects the connection.

When using a direct Ethernet connection, the connection is usually established within a few seconds.

Please note that after the watchdog timeout, the FB still needs time to disconnect the connection. This means that you should continue the FB calls, even in case of a watchdog timeout with the respective error value.

USERNAME (in, ANY, optional)

PASSWORD (in, ANY, optional)

Name and associated password for a user account on the selected SMTP server.

USERNAME is independent of the e-mail address of the sender of the message to be transmitted.

If no user name is entered or if USERNAME is an empty string "", the FB attempts to transmit the e-mail without authentication. This will fail if the SMTP has not been set up as an "open" relay.

TO_S (in, ANY, mandatory)

Recipient's e-mail address as an ASCII string with a maximum length of 240 characters.

The parameter starts with the keyword "TO:", followed by a space and the e-mail address between inequality signs, for example: "TO:

<john.doe@nowhere.com>"

CC (in, ANY, optional)

E-mail address of the recipient of an e-mail copy as an ASCII string with the same format as TO_S.

Although SMTP basically allows a list of CC recipients, only one address can be transferred to FB 490.

FROM (in, ANY, mandatory)

E-mail address of the sender as an ASCII string like TO_S, but starting with the keyword "FROM:".

The FROM address is not identical to the above USERNAME.

If the recipient replies to this e-mail using "Reply", the reply will be sent to the address specified in this parameter. The validity of the FROM address is not verified, i.e. it is basically possible to enter any information.

SUB (in, ANY, optional)

Subject line of the e-mail as an ASCII string such as "Fault in the system". (The keyword "SUBJECT:" is completed by FB 490.)

If SUB is missing, the e-mail will be sent with a blank subject line.

X_PRIORITY (in, WORD, optional)

Parameter that contains the e-mail transmission urgency.

The X-PRIORITY or IMPORTANCE fields of the e-mail are set with the aid of this parameter.

Valid values for X_PRIORITY are 1, 3 and 5:

Table 3-2: Equivalent of the values for priority and importance

X-PRIORITY	IMPORTANCE
1	"high"
3	--
5	"low"

Priority 3 corresponds to "normal" importance; when using this value, the IMPORTANCE field of the e-mail is normally not included in the e-mail.

For all other values (or when not using this parameter), no X-PRIORITY/IMPORTANCE fields are entered in the message.

Please note that mail servers are not committed to prioritizing e-mail transmission depending on X-PRIORITY or IMPORTANCE; in fact, most mail servers completely ignore these fields.

SENDER_TIMEZ (in, STRING[6], optional)

The time zone of the sending client as a string with up to 6 ASCII characters, e.g. "+0200".

While the FB independently determines the time and date of sending,¹ the CPU has no possibility to independently determine in which time zone it is. Therefore, this data must be transferred when calling the FB.

The time zone indicates the difference between local time and “Universal Coordinated Time” (UTC). The string consists of one mandatory “+” or “-” sign, followed by two digits for the hours and two additional digits for the minute difference from UTC.

During the standard time period Germany is in zone “+0100”, during the daylight saving time period the country is in zone “+0200”.

If no `SENDER_TIMEZ` is transferred or if the length of the transferred string differs from 5 characters, the FB takes on the default value “+0100”.

There is no check for implausible values (e.g., “+3599”).

TEXT (in, ANY, optional)

ASCII string of the message text to be transmitted.

The string must not exceed a maximum length of 240 characters.

The parameter may be omitted, but it is non-permissible to transfer an empty string “”.

ATTACHMENT (in, ANY, optional)

Pointer to a memory area that is attached to the message as a binary MIME attachment².

The attachment must not exceed a maximum length of 65534 bytes.

The parameter may be omitted, but it is non-permissible to transfer an empty string “”.

DEVICE_ID (in, BYTE, optional)

Hardware ID used to identify the slot of a WinAC computer via which the connection to the SMTP server is established.

Depending on the used hardware, different slots with additional Ethernet interfaces are available on WinAC computers. The relevant used slot whose interface is used to establish the connection to the SMTP server must be known to the FB to ensure trouble-free SMTP communication.

Several options are available:

- If the FB is used on a WinAC computer, this parameter must be transferred. Its value then equals the index of the slot that has been configured for SMTP communication in Station Configurator.
- In all other cases, the parameter remains empty and the FB uses the device MLFB to determine the interface configuration.

¹ The time when the e-mail is transferred to FB 490.

² MIME = “Multipurpose Internet Mail Enhancement”

COM_RST (in/out, BOOL)

Flag that triggers a reset of the FB.

After processing the reset, the FB independently resets the flag.

For reset details, see 3.2.4.

BUSY (out, BOOL)

Flag that indicates whether an e-mail is currently being transmitted.

From the moment the transmission procedure starts until it stops, BUSY has the value "1"; otherwise, it has the value "0".

DONE (out, BOOL)

Flag "1" indicates that the e-mail has been successfully transmitted.

ERROR (out, BOOL)

Flag that indicates whether an error has occurred during e-mail processing.

If an error has occurred, it has the value "1" only for one cycle and is then reset to "0".

STATUS or SFC_STATUS (see below) provide detailed error codes.

STATUS (out, INT)

Status code of FB 490

This value indicates the current status of FB 490. This can be both a normal status and an error message within the scope of the e-mail transmission or an error message received by the SMTP server that has been passed on.

This value is only present for one cycle and then reset to "0".

SFC_STATUS (out, INT)

Status code of a called SFC

Passed on status of an SFC called by FB 490. This code can also be used for error analysis. A memory area transferred as an attachment that is too large may, for example, result in an error when calling SFC 20 "BLMOV".

This value is only present for one cycle and then reset to "0".

3.2.4 Reset of FB 490

Setting the COM_RST parameter initiates a reset of function block 490. This reset has the following functions:

- Before using the FB for the first time, the reset must be performed to initialize the FB.
- Furthermore, it may be necessary to reset the FB if the connection has been aborted during SMTP communication or if another unexpected event has occurred. (In the case of most errors, however, the FB independently responds with a reset, at the latest after the transmission period set in the watchdog has elapsed.)
- Finally, a reset forces the acceptance of new parameter sets.

For performance and data consistency reasons, the complete parameter set transferred to the FB is not accepted to the instance data block of the FB each time FB 490 is called.³ Instead, the following parameters are updated in the instance DB only when the FB is called with a reset:

- USERNAME
- PASSWORD
- CC
- TEXT
- ATTACHMENT
- DEVICE_ID

3.2.5 E-mail transmission

Transmission procedure

E-mail transmission is initiated once FB 490 detects a change of the parameter value from 0 to 1.

While the e-mail is being transmitted, `BUSY` has the value 1 and `STATUS` or `SFC_STATUS` indicate the relevant current values of FB 490 or system blocks called by this FB.

Once the transmission has been completed, `BUSY` returns to 0 and `DONE` and if necessary `ERROR` are set.

The following table shows an overview of possible combinations of the `BUSY`, `DONE` and `ERROR` values:

Table 3-3

BUSY	DONE	ERROR	Status
0	0	0	No send job is pending
1	--	--	Send job is being processed
0	1	0	Send job successfully completed
0	0	1	Send job has been aborted with an error message (see values of <code>STATUS</code> and <code>SFC_STATUS</code>)

Check whether the transmission was successful

The FB transfers the e-mail to the SMTP server for further transmission and this transfer completes the send operation of FB 490.

This does not mean that the e-mail has been successfully transmitted. In fact, the FB receives no feedback whatsoever on whether or when the e-mail has reached the recipient(s). The only way to find out whether sending has been successful or not is to check the recipient's e-mailbox (or the SMTP server log).

³ Since processing an e-mail send job covers several CPU cycles, there would be a great risk of data inconsistencies if, for example, the contents of the buffers for texts or attachments were changed during e-mail transmission.

3.2.6 Error messages for FB 490: Values of the “STATUS” and “SFC_STATUS” status variables

The following table provides an overview of the values the `STATUS` and `SFC_STATUS` parameters can take on. In addition, it provides information on how to respond to errors that have occurred.

3 Functional Mechanisms of this Application

3.2 Functionality of FB 490

Table 3-4

Return value STATUS (W#16#...)	Return value SFC_STATUS (W#16#...)	Explanation	Notes
0000		Execution of FB 490 has been completed without error.	Completing FB 490 without error does not automatically mean that the sent e-mail arrives at the intended recipient (see 3.2.5)
7001		FB 490 is active (BUSY = 1).	
7002	7002	FB 490 is active (BUSY = 1).	
8xxx	xxxx	Execution of FB 490 has been completed with an error code of the communication SFCs called internally or SFC 20 "BLKMOV".	For detailed information on the evaluation of the SFC_STATUS parameter, please refer to the documentation /2/ "Open Communication via Industrial Ethernet" or the documentation "Copying Memory Areas with SFC 20 "BLKMOV"" (see \5).
800z	xxxx	The error message originates from SFC 20 "BLKMOV" and indicates an error that has occurred when copying one of the parameters. The last digit identifies the responsible parameter: z = 1 TO_S z = 2 CC z = 3 FROM z = 4 SUB z = 5 TEXT z = 6 ATTACHMENT z = 7 USERNAME z = 8 PASSWORD	See also the SFC 20 "BLKMOV" documentation (see \5).
8010	xxxx	Error while establishing the connection.	COM_RST may not have been set after loading the instance DB, see 3.2.4.
8011	xxxx	Error while sending data.	For more information on the evaluation of SFC_STATUS, please refer to the description of the "FB 65 TCON", "FB 63 TSEND", "FB 64 TRCV", "FB 66 TDISCON" blocks in the documentation /2/ "Open Communication via Industrial Ethernet".

3 Functional Mechanisms of this Application

3.2 Functionality of FB 490

Return value STATUS (W#16#...)	Return value SFC_STATUS (W#16#...)	Explanation	Notes
8012	xxxx	Error while receiving data.	Among other things, this error results from an incorrectly configured <code>DEVICE_ID</code> , see 3.2.3, or if no block reset (3.2.4) was performed after changing the parameters.
8013	xxxx	Error while establishing the connection.	
8014		Connection cannot be established.	You may have entered an incorrect mail server IP address (<code>ADDR_MAIL_SERVER</code>) or a period that is too short (<code>WATCH_DOG_TIME</code>) for connection establishment. It is also possible that the CPU is not connected to the network or that the CPU configuration is incorrect.
82xx, 84xx, 85xx		Error messages from this number range originate from the mail server and, except for the leading digit "8", correspond to the error number of SMTP. The following lines list several error codes that may occur when using FB 490:	See also chapter 6.3 in the appendix.
8450		Action not performed: Mailbox not available/cannot be accessed.	Try again later.
8451		Action aborted: Local error during processing	Try again later.
8500		Syntax error: Error not detected. This also includes the error of a command string that is too long. This may also be caused by the fact that the e-mail server does not support the LOGIN authentication method.	Check the parameters of FB 490. Try sending an e-mail without authentication. To do so, replace the <code>USERNAME</code> parameter by an empty string. Make sure that your SMTP server is configured so that it accepts LOGIN authentications. (See 3.2.2)
8501		Syntax error: Wrong parameter or argument	You may have entered an address with incorrect syntax for <code>TO_S</code> or <code>CC</code> . (Missing inequality signs or similar incorrect entries, see 3.2.3.)
8502		Unknown or non-implemented error.	Check your entries, particularly the <code>FROM</code> parameter. It may be incomplete and you may have forgotten "@" or ".", see 3.2.3.
8535		SMTP authentication incomplete.	You may have entered a non-existing user name or an incorrect password for the account.

3 Functional Mechanisms of this Application

3.2 Functionality of FB 490

Return value STATUS (W#16#...)	Return value SFC_STATUS (W#16#...)	Explanation	Notes
8550		Mail server cannot be accessed, you have no access rights.	You may have entered an incorrect user name or password or the mail server does not support your LOGIN. Another cause could be the incorrect entry of the domain name after "@" in TO_S or CC.
8552		Action aborted: Allotted memory size exceeded	Try again later.
8554		Transmission failed.	Try again later.

4 Installation and Startup

4.1 Hardware configuration (example)

Figure 4-1

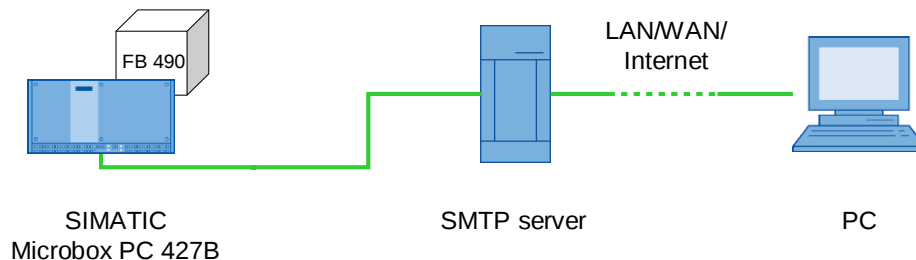


Figure 4-1 shows an example of the hardware configuration for the supplied application. Your configuration may differ considerably from the one shown here.

4.2 Installing the STEP 7 configuration

Install the STEP 7 project on your system as usual. If the hardware you are using is identical to the sample project, no further actions are required.

Otherwise, copy FB 490 and the blocks depending on it to your new project and adjust the configuration as necessary.

NOTICE

Do not download a changed configuration to the PLC while an e-mail is being transmitted (i.e. while `REQ` or `BUSY` do not equal 0). Otherwise, the connection may not be established correctly so that communication resources remain permanently occupied. This can in turn cause an undefined state for the TCP/IP communication functions and prevent the connection from being established.

4.3 Establishing an Internet connection to the SMTP server

The function of FB 490 requires that an Internet connection be established to an SMTP server you have selected. Establish this connection using one of the commonly used methods.

Make sure that the SMTP server has an account whose access data matches the data you provide FB 490 with. This is particularly important if the SMTP server is not "open", i.e. if it does not allow anonymous relaying of messages.

NOTE

For details on configuring your Internet connection, please refer to the documentation of your Internet service provider (ISP) and follow the relevant instructions!

NOTE

Make sure that your selected SMTP server is configured so that it accepts one of the authentication methods supported by FB 490 (see 3.2.2)!

4.4 Extending the local data memory

NOTE

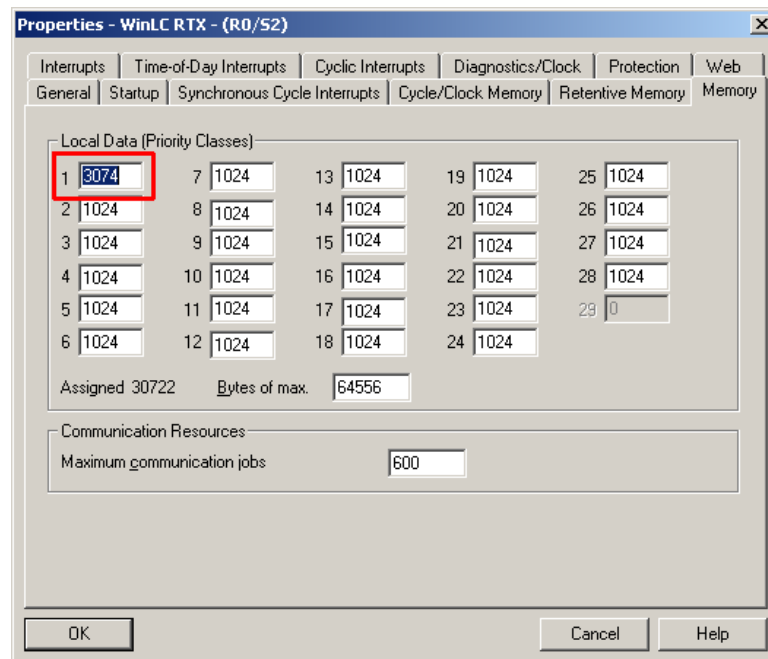
This step is only necessary if you are using a WinLC computer as a CPU.

To ensure that FB 490 can be executed on a WinLC computer, the local data memory for priority class 1 must be extended to at least 3074 bytes.

To do so, open HW Config in your project and open the WinLC RTX Properties dialog box by double-clicking on the respective entry.

Select the "Memory" tab and in the "1" field of "Local Data (Priority Classes)" enter the value "3074" (or higher).

Figure 4-2



The default settings of the other values do not have to be changed.

Save your changes and transfer the changed configuration to the WinLC computer.

In the sample project, this adjustment has already been made.

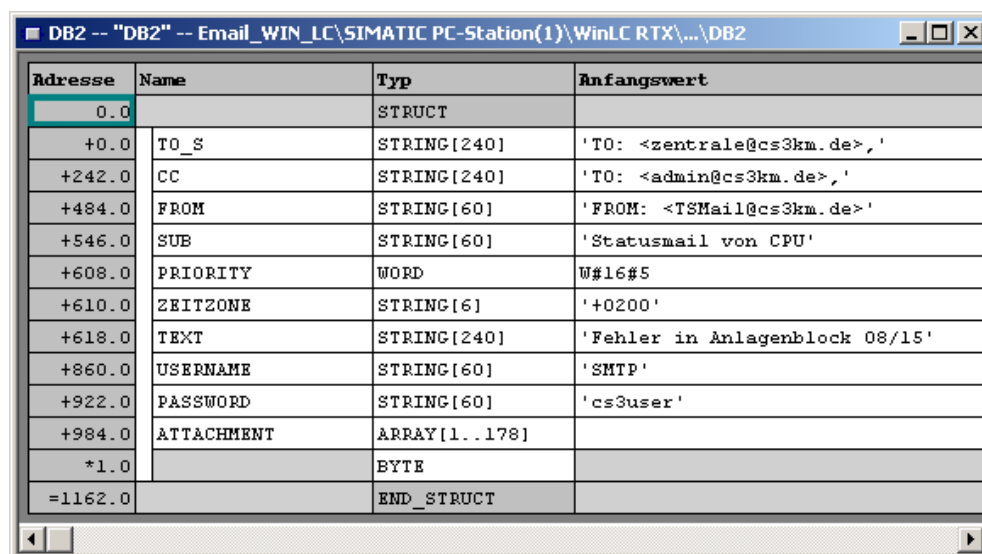
5 Monitoring and Modifying the Example

5.1 Using data blocks and the variable table

You can modify the supplied configuration example by adjusting the values of DB 2 or the flags provided in variable table VAT_2. In addition, variable table VAT_2 enables you to follow the processing of the e-mail transmission and to check the results.

FB 1 uses the contents of DB 2 as parameters for calling FB 490. For the individual formats of the parameters, please refer to chapter 3.2.3.

Figure 5-1: DB 2



Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	TO_S	STRING[240]	'TO: <zentrale@cs3km.de>,'
+242.0	CC	STRING[240]	'TO: <admin@cs3km.de>,'
+484.0	FROM	STRING[60]	'FROM: <TSMail@cs3km.de>'
+546.0	SUB	STRING[60]	'Statusmail von CPU'
+608.0	PRIORITY	WORD	W#16#5
+610.0	ZEITZONE	STRING[6]	'+0200'
+618.0	TEXT	STRING[240]	'Fehler in Anlagenblock 08/15'
+860.0	USERNAME	STRING[60]	'SMTP'
+922.0	PASSWORD	STRING[60]	'cs3user'
+984.0	ATTACHMENT	ARRAY[1..178]	
*1.0		BYTE	
=1162.0		END_STRUCT	

Your values for USERNAME and PASSWORD probably differ from the values shown here and must match the accounts that have been set up on the SMTP server you are using.

A number of parameters for FB 490 are not taken from DB 2, but transferred directly, including the flag that triggers the sending of an e-mail (REQ, filled from M 100.0) and the SMTP mail server address (ADR_MAIL_SERVER).

Figure 5-2: FB 1

```
CALL "PN_MAIL" , "DB5"
REQ      :=M100.0           //Auftragsanstoß bei steigender Flanke
ADR_MAIL_SERVER:=DW#16#AC102A2C //entspricht der IP-Adresse 172.16.1.1 im HEX-Format
WATCH_DOG_TIME :=T#2M      //Zeitspanne von 2 Min. in der der Versand abgeschl. sein sollte
USERNAME      :=DB2.USERNAME //Zeiger auf Benutzer-Name des Absenders
PASSWORD      :=DB2.PASSWORD //Zeiger auf Kennwort. Wird DB2.USERNAME weggelassen,d.h. es wird kein Parameter zugewiesen
TO_S          :=DB2.TO_S     //Zeiger auf Empfänger-Adresse
CC           :=
FROM         :=DB2.FROM      //Zeiger auf Absender-Adresse
SUB          :=DB2.SUB       //Zeiger auf das Mail-Thema
X_PRIORITY   :=DB2.PRIORITY  //Zeiger auf Mail-Befoerderungssvorrangstufe
SENDER_TIMEZ :=DB2.ZEITZONE  //Zeiger auf den Mailtext (optional)
TEXT         :=DB2.TEXT      //Zeiger auf Attachment-Text (optional, Bsp. bezieht sich auf obigen DB)
ATTACHMENT   :=P#DB20.DEX0.0 BYTE 38 //Bei WinAC: benutzter Interface-Stackplatz, sonst 0
DEVICE_ID    :=MB52
BUSY         :=M46.0         //Ausgangs-/Zustandsparameter BUSY
DONE         :=M46.1         //Ausgangs-/Zustandsparameter DONE
ERROR        :=M46.2         //Ausgangs-/Zustandsparameter ERROR
STATUS       :=MW48          //Ausgangs-/Zustandsparameter STATUS
SFC_STATUS   :=MW50          //Ausgangs-/Zustandsparameter SFC_STATUS
COM_RST      :=M46.3         // Reset erzwingen
// SENDER_TIMEZ ist optionaler Parameter; wird er nicht uebergeben, so wird der default
// "+0100" (entspricht der CET Winterzeit) benutzt

//Prüfung der Aufrufergebnisse auf das Ende der Bearbeitung
U      M      46.0           //ist BUSY ==0?
SPB    Ende
U      M      46.1           //nein, BUSY ==1, AS_MAIL bearbeitet den Auftrag noch
SPB    Ende
U      M      46.2           //Wurde die Bearbeitung des
SPB    Ende
U      M      46.2           //FE 490 "PN_MAIL" fehlerfrei beendet?
SPB    Ende
SPBN    Ende
S      M      60.0           // Hilfsmerker
L      MW     48             // STATUS
T      MW     62             //
L      MW     50             //SFC Status
T      MW     64             //
NOP     0                   //Nein, bei der Bearbeitung ist ein Fehler aufgetreten, BUSY ==0, BIE

//==0
Ende: BE
```

The program section following the call of FB 1 is used to permanently freeze error messages that are normally only present for one cycle and that would otherwise have been overwritten after the next CPU cycle: If an error has occurred (M 46.2), flag M 60.0 is set and the values for STATUS and SFC_STATUS are copied to the words MW 62 or MW 64.

Figure 5-3 shows variable table VAT_2 while processing an e-mail transmission job.

Figure 5-3: Variable table VAT_2 (excerpt)

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	M 100.0		BOOL	true	true
2	M 46.0		BOOL	true	
3	M 46.1		BOOL	false	
4	M 46.2		BOOL	false	
5	M 46.3		BOOL	false	false
6	MB 52		HEX	B#16#01	B#16#01
7	MW 62		HEX	VW#16#0000	
8	M 60.0		BOOL	false	
9	MW 64		HEX	VW#16#0000	
10	DB5.DBW 2		HEX	VW#16#AC10	
11	DB5.DBW 4		HEX	VW#16#2A2C	
12	DB2.DBW 860		HEX	VW#16#3C04	
13	DB2.DBW 862		ZEICHEN	'SM'	
14	DB2.DBW 864		ZEICHEN	'TP'	
15	DB2.DBW 866		ZEICHEN	VW#16#0000	
16	DB2.DBW 868		ZEICHEN	VW#16#0000	
17	DB2.DBW 870		ZEICHEN	VW#16#0000	
18	DB2.DBW 922		HEX	VW#16#3C07	
19	DB2.DBW 924		ZEICHEN	'cs'	
20	DB2.DBW 926		ZEICHEN	'3u'	
21	DB2.DBW 928		ZEICHEN	'se'	
22	DB2.DBW 930		ZEICHEN	VW#16#7200	
23	DB2.DBW 932		ZEICHEN	VW#16#0000	
24	DB2.DBW 934		ZEICHEN	VW#16#0000	
25	DB2.DBW 936		ZEICHEN	VW#16#0000	
26	DB2.DBW 938		ZEICHEN	VW#16#0000	
27	DB2.DBW 608	"DB2".PRIORITY	DEZ	5	1
28	DB5.DB 390		ZEICHEN	'250.'	
29	DB5.DB 394		ZEICHEN	'Jana'	
30	DB5.DB 398		ZEICHEN	'-Ser'	
31	DB5.DB 402		ZEICHEN	'ver\$ir'	
32	DB5.DB 406		ZEICHEN	'\$1250'	
33	DB5.DB 410		ZEICHEN	'-STA'	
34	DB5.DB 414		ZEICHEN	'RTTL'	
35	DB5.DB 382		ZEICHEN	DVW#16#00000000	
36	DB5.DB 386		ZEICHEN	DVW#16#00000000	
37	DB5.DB 72		ZEICHEN	'□□+0'	
38	DB5.DB 76		ZEICHEN	DVW#16#32303000	
39	DB2.DB 610		ZEICHEN	'□□+0'	
40	DB2.DB 614		ZEICHEN	DVW#16#32303000	
41					

The following table lists the meaning of the individual variable table entries:

Table 5-1

Line	Memory area	Meaning	In the example
1	M 100.0	REQ=TRUE → trigger the start of a new e-mail transmission process	Transmission started
2	M 46.0	BUSY=TRUE → e-mail transmission active	Transmission active
3	M 46.1	DONE=TRUE → e-mail transmission completed	Transmission not yet completed
4	M 46.2	ERROR=TRUE → e-mail transmission completed with error	No error has occurred
5	M 46.3	COM_RST=TRUE → perform FB reset	
6	MB 52	DEVICE_ID of the used hardware	"1"
7	MW 62	Permanent copy of STATUS	No error
8	M 60.0	Permanent copy of ERROR	No error
9	MW 64	Permanent copy of SFC_STATUS	No error
10-11	DB5.DBW 2-5	ADDR_MAIL_SERVER, IP address of the SMTP server	"172.16.42.44"
12-17	DB2.DBW 860-871	USERNAME used for login to the SMTP server	"SMTP"
18-26	DB2.DBW 922-939	PASSWORD for login to the SMTP server	"cs3user"
27	DB2.DBW 608	X_PRIORITY of the e-mail to be sent	"5" (low)
28-34	DB5.DBD 390-417	Status message of the SMTP server as plain text	"250- Jana-Server 250- STARTTL ..."
35-36	DBD5.DBD 382-389	--	
37-38	DBD5.DBD 72-79	TIMEZONE for the e-mail to be sent, copy in the instance data block	" +0200"
39-40	DB2.DBD 610-617	TIMEZONE for the e-mail to be sent, original value in the data block of the calling FB	" +0200"

For the meaning of the individual parameters of FB 490, please also refer to 3.2.3.

5.2 Sending an e-mail with the aid of the SMTP block FB 490

To send an e-mail to the SMTP server after configuring it accordingly, follow the steps in the table below:

Table 5-2

1.	In FB 1, transfer the IP address of your SMTP server as the <code>ADR_MAIL_SERVER</code> parameter and download the relevant configuration to the CPU.
2.	Fill DB 2 with the addresses and contents you want to send (see 3.1 and 3.2.3).
3.	If necessary, set the hardware ID (MB 52) and e-mail priority level (DB2.DBW 608) if you want to change their default values.
4.	Perform a reset of FB 490 by setting flag bit 46.3 to <code>TRUE</code> (see chapter 3.2.4). After a brief moment, M 46.3 is reset to <code>FALSE</code> .
5.	Set flag bit 100.0 to <code>TRUE</code> to start the sending of the e-mail. For a few moments, M 46.0 (<code>BUSY</code>) changes to <code>TRUE</code> . Flag words 62 and 64 display different status values generated by the FB and the SMTP server. Memory area DB5.DBD 390-417 displays the SMTP server messages as plain text.
6.	After completing the transmission, M 46.0 (<code>BUSY</code>) is reset to <code>FALSE</code> . (M 46.1 [<code>DONE</code>] is set to <code>TRUE</code> for one cycle and then cleared; this process is probably too fast to be viewed in the variable table.)
7.	If an error preventing e-mail transmission has occurred, M 46.2 (<code>ERROR</code>) is set to <code>TRUE</code> for one cycle and M 60.0 is permanently set to <code>TRUE</code> . In this case, flag words 62 and 64 contain the last valid values for <code>STATUS</code> or <code>SFC_STATUS</code> .
8.	Use the SMTP server log to check whether the e-mail has been successfully transmitted.
9.	To send another e-mail, reset M 100.0 to <code>FALSE</code> and then back to <code>TRUE</code> . (The rising edge of M 100.0 triggers the sending of the e-mail.)
10.	If the recipient addresses or anything similar change between two e-mails, repeat step 4 and then perform a block reset by setting M 46.3 (see chapter 3.2.4) before you clear M 100 and set it again.

Figure 5-4 shows an e-mail that has been generated and transmitted with the aid of FB 490.

Figure 5-4: Example of a transmitted e-mail

```
X-UIDL-JANA-SERVER: 76843341.1290520549@Jana-Server
FROM: <TSMail@cs3km.de>
TO: <zentrale@cs3km.de>
SUBJECT: Statusmail von CPU
X-PRIORITY: 5
IMPORTANCE: low
DATE: Tue, 23 Nov 2010 14:57:59 +0200
MIME-Version: 1.0
Content-TYPE: multipart/mixed; boundary="--"

---
Content-TYPE: text/plain;

Fehler in Anlagenblock 08/15
---
Content-Type: application/octet-stream; name="Attachment.bin"
Content-Transfer-Encoding:BASE64
Content-Disposition: attachment; filename="Attachment.bin"

AA8/rDyfgRVEYXMgaXN0IGVpbIBNZWxkZXRleHQAAAAAAGCACYC=|
```

6 Appendices

6.1 SMTP commands: Minimum command set

The command set that SMTP servers must understand is limited. The minimum implementation is already possible with only eight different commands:

Table 6-1: Minimum command set for an SMTP server

SMTP command	Description
HELO/EHLO (Hello/Extended Hello)	HELO or EHLO starts the SMTP session and identifies the client to the server.
MAIL	MAIL initiates the e-mail transmission and also provides the sender's address.
RCPT (Recipient)	RCPT indicates the address of one or several recipients. This command can be executed multiple times.
DATA	DATA initiates the transmission of the e-mail message. The end of the e-mail message is marked by "CRLF.CRLF".
RSET (Reset)	RSET aborts an already initiated e-mail transmission. The connection between client and server remains established.
VRTY (Verify)	The recipient's address can be verified using VRFY.
EXPN (Expand)	Most MTAs treat EXPN like VRFY.
NOOP	NOOP causes a response from the server. This prevents disconnection due to a timeout.

6.2 SMTP connection process between client and server

The table below shows an example of the message traffic that is generated when sending an e-mail between the client and its server.

The commands and the actual e-mail contents are exchanged as ASCII text commands. This (and the use of relatively long timeouts) enables the user to "manually" send an e-mail by establishing a Telnet connection to the server, for which the following mail commands are entered via the terminal.

Table 6-2: SMTP dialog process between client and server when sending an e-mail without authentication

Client	Server	Explanation
	220 smtp.example.com ESMTP Postfix	
HELO relay.example.org		If at this point the client sends EHLO instead of HELO, the server will respond with a list of supported authentication methods and an authentication dialog can start.
	250 Hello relay.example.org, I am glad to meet you	
MAIL FROM:<bob@example.org>		Account of the sender of the message. If respective authentication methods are used, this account must exist on the server.
	250 Ok	
RCPT TO:<alice@example.com>		A list of recipients' accounts starts.
	250 Ok	Each recipient's address is acknowledged by the server.
RCPT TO:<theboss@example.com>		
	250 Ok	
DATA		The list of recipients ends, the main part of the e-mail starts.
	354 End data with <CR><LF>.<CR><LF>	Information from the server: The main part continues until a line is transferred that consists only of the decimal point ".".
From: "Bob Example" <bob@example.org>		Sender and recipient fields follow. These fields are for information only and do not necessarily have to match the above recipients' accounts!
To: "Alice Example" <alice@example.com>		
Cc: theboss@example.com		

Client	Server	Explanation
Date: Tue, 15 Jan 2008 16:02:43 - 0500		Date and time of the message as provided by the client. PRIORITY and IMPORTANCE fields can be inserted here.
Subject: Test message		
Hello Alice.		
This is a test message with 5 header fields and 4 lines in the message body.		
Your friend,		
Bob		
.		Line with only one decimal point as the end of the e-mail body.
	250 Ok: queued as 12345	Acknowledgement of the server.
QUIT		
	221 Bye	

6.3 Status and error messages of an SMTP server

When processing the e-mail traffic, the SMTP server sends all status or error messages as a three-digit decimal number, followed by the description in plain text. Such a message looks, for example, as follows:

```
354 End data with .
```

The number “352” describes the server status (in this case: The server is ready to receive the body of the e-mail) while the rest of the line indicates a possible human operator on the client side. (Here: The transmission of the e-mail body is completed by entering a line that only consists of one decimal point “.”.)

The status number is sufficient to evaluate the SMTP server status.

The first digit indicates the status number category:

- 2xx: Positive response, the command was executed.
- 3xx: Positive response, but the server needs more information for successful execution.
- 4xx: “Temporary” error, for example no connection available. When trying again, it is possible that the error no longer occurs.
- 5xx: Permanent error, for example command not understood. No improvement by repeating the same command.

Table 6-3: SMTP server status codes

Status code	English description
211	System status, or system help reply.
214	Help message.
220	<i>Domain</i> service ready. Ready to start TLS.
221	<i>Domain</i> service closing transmission channel.
250	OK, queuing for node <i>node</i> started. Requested mail action okay, completed.
251	OK, no messages waiting for node <i>node</i> . User not local, will forward to <i>forwardpath</i> .
252	OK, pending messages for node <i>node</i> started. Cannot VRFY user (e.g., info is not local), but will take message for this user and attempt delivery.
253	OK, <i>messages</i> pending messages for node <i>node</i> started.
354	Start mail input; end with <CRLF>.<CRLF>.
355	Octet-offset is the transaction offset.
421	<i>Domain</i> service not available, closing transmission channel.
432	A password transition is needed.
450	Requested mail action not taken: mailbox unavailable. ATRN request refused. ⁴
451	Requested action aborted: local error in processing. Unable to process ATRN request now.
452	Requested action not taken: insufficient system storage.
453	You have no mail.
454	TLS not available due to temporary reason. Encryption required for requested authentication mechanism.
458	Unable to queue messages for node <i>node</i> .

⁴ ATRN: “Authenticated Turn”, a method for sending e-mails when using dynamic IP addresses. Also referred to as ODMR (“On-Demand Mail Relay”).

Status code	English description
459	Node <i>node</i> not allowed: <i>reason</i> .
500	Command not recognized: <i>command</i> . Syntax error.
501	Syntax error, no parameters allowed.
502	Command not implemented.
503	Bad sequence of commands.
504	Command parameter not implemented.
521	<i>Machine</i> does not accept mail.
530	Must issue a STARTTLS command first. Encryption required for requested authentication mechanism.
534	Authentication mechanism is too weak.
538	Encryption required for requested authentication mechanism.
550	Requested action not taken: mailbox unavailable.
551	User not local; please try <i>forwardpath</i> .
552	Requested mail action aborted: exceeded storage allocation.
553	Requested action not taken: mailbox name not allowed.
554	Transaction failed.

7 References

7.1 References

This list is by no means complete and only presents a selection of related references.

Table 7-1

	Topic	Title
/1/	STEP7	Automating with STEP7 in STL and SCL Hans Berger Publicis Corporate Publishing ISBN 3-89578-113-4
/2/	Open communication	"Open Communication via Industrial Ethernet", Siemens documentation A5E00711636-01

7.2 Internet links

This list is by no means complete and only presents a selection of appropriate information.

Table 7-2

	Topic	Title
\1\	Reference to the document	http://support.automation.siemens.com/WW/view/en/46817803
\2\	Siemens I IA/DT Customer Support	http://support.automation.siemens.com
\3\	Manual: TeleService S7-Library V6.1	http://support.automation.siemens.com/WW/view/en/24781774
\4\	Manual: S7-CPs for Industrial Ethernet Configuring and Commissioning, Part A (see chapter 8)	http://support.automation.siemens.com/WW/view/en/30374198
\5\	Manual: System and Standard Functions for S7-300/400 Volume 1 and Volume 2, (see chapter 3.1)	http://support.automation.siemens.com/WW/view/en/44240604
\6\	FAQ: STEP 7 Hardware Support Packages (HSPs – as at: 12/2010) for installing hardware updates in the Hardware Configuration	http://support.automation.siemens.com/WW/view/en/23183356
\7\	FAQ: How can you update the hardware catalog without direct Internet access?	http://support.automation.siemens.com/WW/view/en/22374877

	Topic	Title
181	FAQ: Which communication services are supported by CPUs with an integrated PN interface?	http://support.automation.siemens.com/WW/view/en/18909487

7.2.1 Relevant RFCs

SMTP was formalized in a series of RFCs ("Requests for Comments"; directives, definitions and suggestions for standardization published by the Internet Engineering Task Force).

The following table lists the most important relevant RFCs:

Table 7-3

RFC no.	Topic	URL
--	Overview of all published RFCs	http://tools.ietf.org/rfc
821	SMTP definition	http://tools.ietf.org/html/rfc821
822	Format for "ARPA Internet Text Messages", e-mail predecessor	http://tools.ietf.org/html/rfc822
2195	"CRAM-MD5" authentication	http://tools.ietf.org/html/rfc2195
2554	SMTP extension by SMTP authentication method	http://tools.ietf.org/html/rfc2554
2595	"PLAIN" and "LOGIN" authentication methods	http://tools.ietf.org/html/rfc2595
2821	Extension of SMTP, obsoletes RFC 821	http://tools.ietf.org/html/rfc2821
2822	E-mail format ⁵ , obsoletes RFC 822	http://tools.ietf.org/html/rfc2822
2831	"DIGEST-MD5" authentication method	http://tools.ietf.org/html/rfc2831
5321	Definition of SMTP with further updates, obsoletes RFC 2821	http://tools.ietf.org/html/rfc5321

⁵ Here still referred to as "Internet Message Format".

8 History

Table 8-1

Version	Date	Modification
V1.0	01/12/11	First edition