Application description • 05/2015

Sending and Receiving of Text Messages via serial CPs/CMs and GPRS/GSM Modem MD720-3

SIMATIC S7-300/400/1500, SIMATIC ET 200S/ET 200SP, SINAUT MD720 3
STEP 7 V13

https://support.industry.siemens.com/cs/ww/en/view/25545680

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 used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we 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 − 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”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.industry.siemens.com.

Table of contents

1 Task

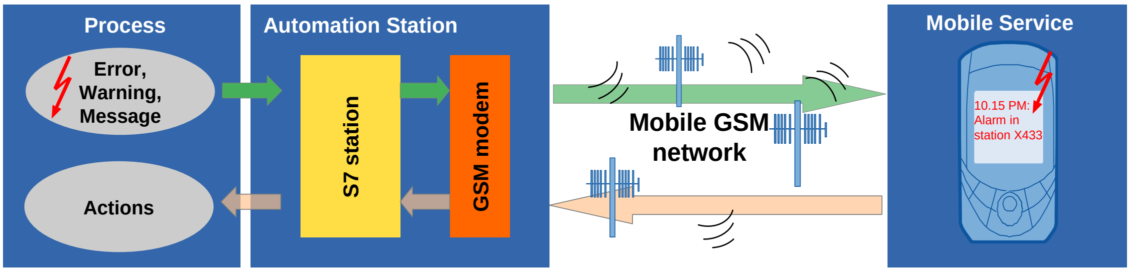
Introduction

In this application, we will show you how you can implement a simple system for wireless signaling and switching based on SMS messaging.

1.1 Overview of the automation problem

The figure below provides an overview of the automation task.

Figure 1-1



1.2 Description of the automation task

The aim of this solution is to realize the following scenarios:

The GSM Modem SINAUT MD720-3 shall be used for

- sending an SMS message to a configurable recipient,
- sending an SMS message with an attached coefficient (e.g. an analog process value) to a configurable recipient,
- receiving an SMS message from a mobile GSM device, evaluating it, and initiating a control action,
- receiving a remote query of a value via SMS.

It shall be possible to employ the following peer devices for communication with the GSM modem:

- SIMATIC S7-300 CPU
- SIMATIC S7-400 CPU
- SIMATIC S7-1500 CPU
- SIMATIC ET 200S
- SIMATIC ET 200SP

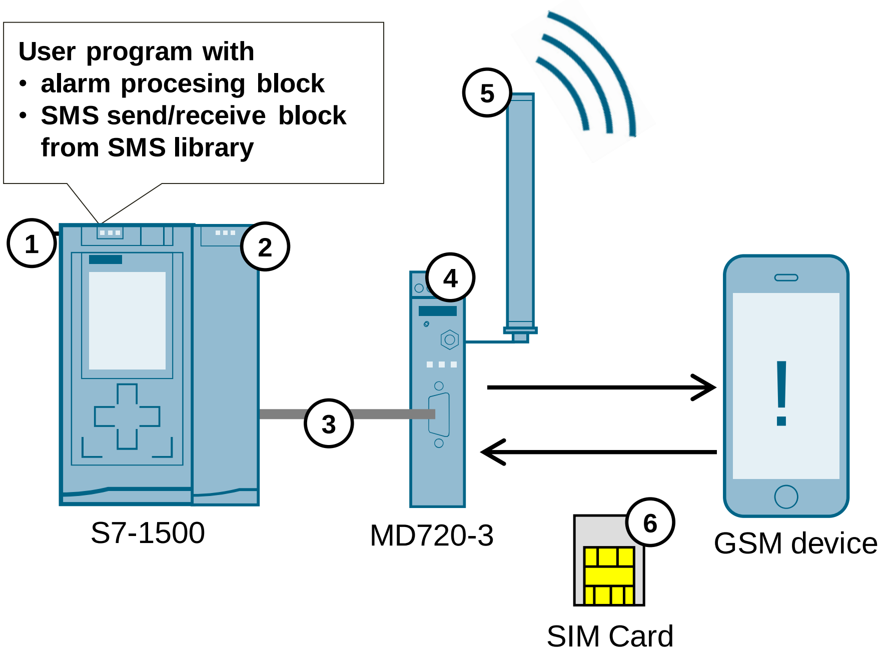
2 Solution

2.1 Overview of general solution

Schematic layout

The figure below shows the most important components of the solution with an S7-1500 CPU with CM PtP (alternatively, the solution can be implemented with communication modules CP340, CP341, CP441, 1SI and CM PtP – see chapter 4.1):

Figure 2-1



Components included

Table 2-1 Hardware components

No.	Component	Description
1	S7 station	An S7-300, S7-400, S7-1500, ET 200S or ET 200SP station
2	Serial communications processor	Depending on the S7 CPU, the respective CP/CM is connected.
3	Serial cable	CP or CM and GSM modem are connected via a serial cable.
4	GSM modem	MD720-3
5	Quad-band antenna	ANT794-4MR
6	SIM card	from service provider

Table 2-2 Software components

Block	Function	Comment
FB “SMS_Meld”	• Generation of an SMS message according to a predefined logic • Send/receive management with SMS library block • Evaluation of a received SMS message	Individual user block in SCL
FB “SMS_sr_xxxx”	Coordinated data exchange between S7 CPU, serial CP/CM and MD720-3	Universal SMS library block in SCL

Topics not covered

This application does not include any basic information

- about GSM radio communication; for more information, refer to document \12\.
- about the programming languages LAD / FBD / STL / SCL.

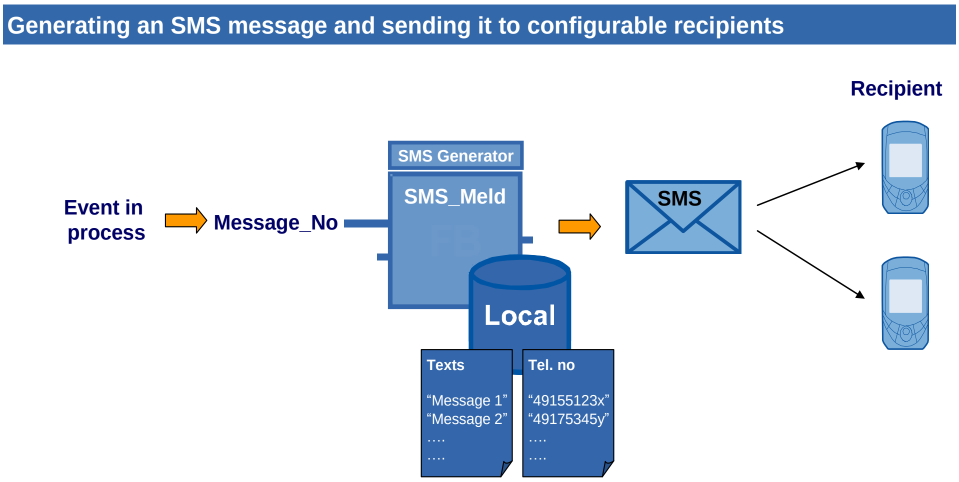
Basic knowledge of these topics is assumed.

2.2 Description of the core functionality

In this example, the required scenarios are implemented with a user block (**FB “SMS_Meld”**) and with the aid of a universal SMS library block (**FB “SMS_sr_xxxx”**) both programmed in SCL.

2.2.1 Scenario: “Generating and sending an SMS message”

Figure 2-2

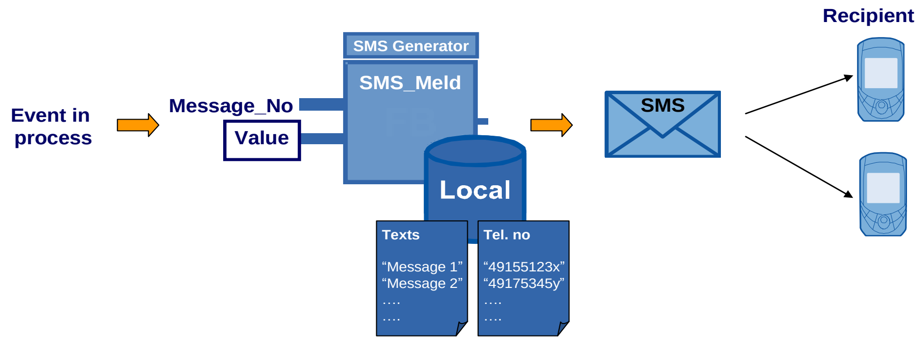

Generating an SMS message and sending it to configurable recipients

When the S7 CPU (e.g. via I/O sensors) detects an event in the process, this event will be assigned an individually defined message number. In the **FB “SMS_Meld”** user block, a text and a recipient are assigned to the message number. The **FB “SMS_Meld”** block coordinates the sending to the recipient.

2.2.2 Scenario: “Sending an SMS message with attached coefficient”

Figure 2-3

Generating an SMS message with attached coefficient and sending it to configurable recipients

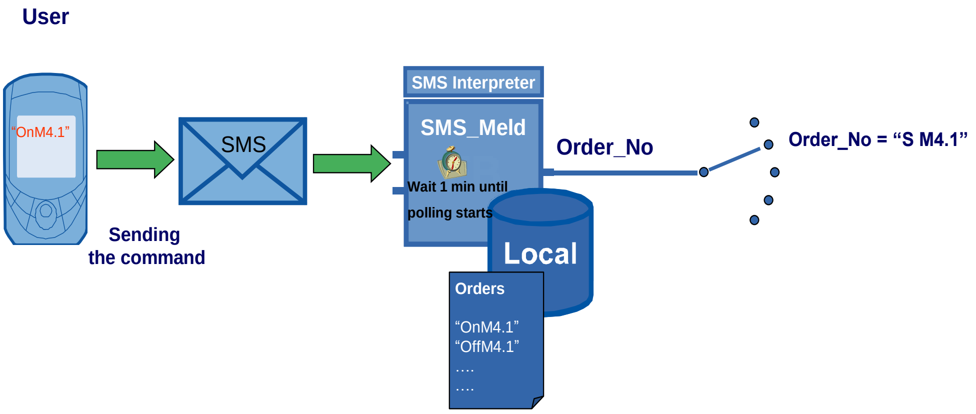


When the S7 CPU (for example, via I/O sensors) detects an event in the process, this event will be assigned an individually defined message number and any coefficient (e.g., an analog process value). In the **FB “SMS_Meld”** user block, a text and a recipient are assigned to the message number. The block coordinates the sending of the text with the coefficient to the recipient.

2.2.3 Scenario: “Receiving and evaluating an SMS order”

Figure 2-4

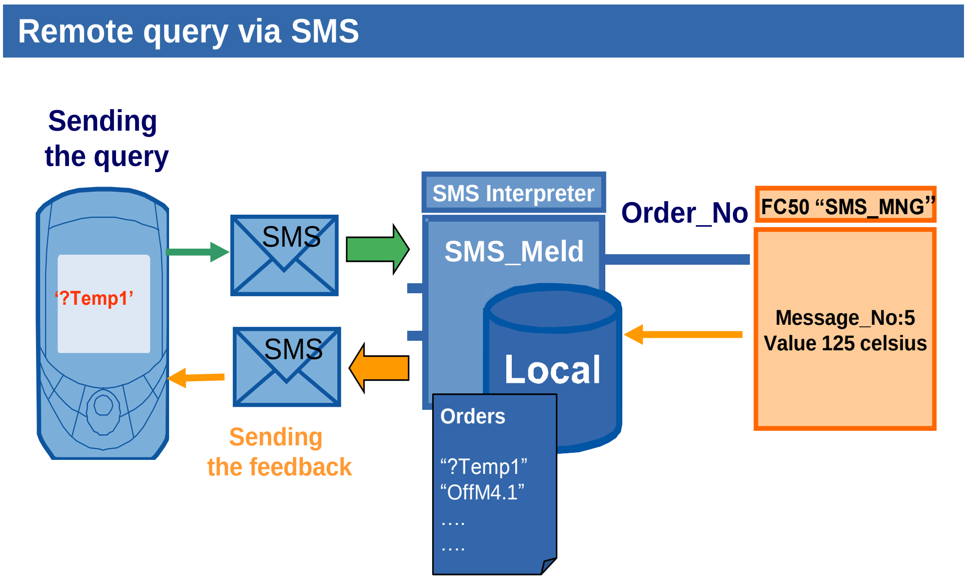
Receiving and evaluating an incoming SMS message



When a user sends a defined string to the controller via SMS, the **FB “SMS_Meld”** user block outputs a permanently assigned order number (Order_No) once it has found this order in its configurable order list. The user can then individually use this order number in his/her program to control the process.

2.2.4 Scenario: “Receiving a remote query of a value”

Figure 2-5



In this scenario, the user wants to query a specific value from the controller. To do so, he/she sends a specific command string to the controller by SMS.

The controller replies with the aid of the **FB “SMS_Meld”** user block and the **FC “SMS_MNG”** user function.

FB “SMS_Meld” interprets the receive text of the SMS message and assigns an **Order_No** to the command string. This number is transferred to **FC “SMS_MNG”**, which transfers the desired value and the appropriate **Message_No** back to the **FB “SMS_Meld”** block.
An SMS message is then sent back to the user.

Advantages of these solutions

The solutions presented here offer you the following advantages:

- The code of the **FB “SMS_Meld”** block can be used as a basis and adjusted to your individual signaling or command scenarios (for example, any SMS recipients, any text messages, any command strings (note chapter 3.4), etc.).
- The **FB “SMS_sr_CPxxx”** SMS block for sending and receiving SMS messages used in the solutions is available in a separate library for each serial communications processor (CP 340, CP 341, CP 441-2, ET 200S 1SI, CM PtP, ET 200SP CM PtP).
- The SMS block from the library can be used for your own solutions even without the **FB “SMS_Meld”** block. For more information, please refer to the library description for this block (see link \1\).

2.3 Hardware and software components used

This application has been generated with the following components:

2.3.1 Hardware for the SIMATIC station

Hardware for the S7-1500 station

The following components are necessary if you want to set up the example with an S7-1500 station.

Table 2-3

Component	No.	Order number	Note
PS 25W 24VDC	1	6ES7505-0KA00-0AB0	
CPU 1516-3 PN/DP	1	6ES7516-3AN00-0AB0	Alternatively, you can also employ a different S7-1500 CPU.
CM PtP RS232 HF alternative: CM PtP RS232 BA	1	6ES7541-1AD00-0AB0 alternative: 6ES7540-1AD00-0AA0	

Alternative hardware for the S7-300 station

The following components are necessary if you want to set up the example with an S7-300 station.

Table 2-4

Component	No.	MLFB / order number	Note
PS307 5A	1	6ES7307-1EA00-0AA0	
CPU 315-2 PN/DP	1	6ES7315-2EH14-0AB0	Alternatively, you can also employ a different S7-300 CPU.
CP 341	1	6ES7341-1AH01-0AE0	Alternatively, the CP 340 can also be used (6ES7340-1AH02-0AE0)
MPI connecting cable	1	6ES7 901-0BF00-0AA0	For loading the CPU

Alternative hardware for the S7-400 station

The following components are necessary if you want to set up the example with an S7-400 station.

Table 2-5

Component	No.	Order number	Note
PS407 10A	1	6ES7407-0KA02-0AA0	
CPU 414-2	1	6ES7414-2XG04-0AB0	Alternatively, you can also employ a different S7-400 CPU.
CP 441-2	1	6ES7441-2AA04-0AE0	
RS232 interface module	1	6ES7963-1AA00-0AA0	

Alternative hardware for the ET 200S station

The following components are necessary if you want to set up the example with an ET 200S station.

Table 2-6

Component	No.	Order number	Note
PS307 5A	1	6ES7307-1EA00-0AA0	
CPU 315-2 PN/DP	1	6ES7315-2EH14-0AB0	Alternatively, you can also use a different controller/master-capable S7-300 CPU
Interface module IM151-3 PN STANDARD	1	6ES7151-3AA20-0AB0	Alternatively, you can also use a different head-end (PROFIBUS, for example).
PM-E DC24V	1	6ES7138-4CA01-0AA0	
ET 200S 1SI 3964/ASCII	1	6ES7138-4DF01-0AB0	
ET 200S, TERM.-MOD. TM-E15S24-01	1	6ES7193-4CB20-0AA0	
ET 200S, TERM.-MOD. TM-P15S23-A0	1	6ES7193-4CD20-0AA0	

Alternative hardware for the ET 200SP station

The following components are necessary if you want to set up the example with an ET 200SP station.

Table 2-7

Component	No.	Order number	Note
PS 25W 24VDC	1	6ES7505-0KA00-0AB0	
CPU 1516-3 PN/DP	1	6ES7516-3AN00-0AB0	Alternatively, you can also employ a different S7-1500 CPU.
IM 155-6 PN ST	1	6ES7155-6AU00-0BN0	Alternatively, you can use the PROFIBUS head of the ET 200SP.
Base unit	1	6ES7193-6BP00-0DA0	
CM PtP	1	6ES7137-6AA00-0BA0	
Server module	1	6ES7193-6PA00-0AA0	

2.3.2 GSM components

Table 2-8

Component	No.	Order number	Note
MD720-3	1	6NH9720-3AA00	
GSM antenna	1	6NH9860-1AA00	
SIM card	2		
Cable with 9-pole Sub-D connectors.	1		

2.3.3 Software components

Table 2-9

Component	No.	Order number
SIMATIC STEP 7 PROFESSIONAL V13 SP1 update 1	1	6ES7822-1AA03-0YA5

Sample files and projects

The following table contains all files and projects used in this example.

Table 2-10

Component	Note
SMS_MD720-3_Library_V13_SP1.zip	This zip file contains the library blocks.
SMS_Example_V13_SP1.zip	This zip file contains the sample project.
25545680_SMS_MD720-3_DOKU__V13_V1_0_en.pdf	This document
25545680_SMS_MD720-3_Library_V13_V1_0_en.pdf	Library description for STEP 7 V13

2.4 Basic performance data

This chapter provides you with an overview of the performance data of the components and the performance of the application.

Communications processors

This application was tested on the following serial CPs/CMs of SIMATIC S7-300/ -400/-1500/ET 200S/ET 200SP:

Table 2-11

Module	Driver	Possible baud rates in combination with MD720-3
CP 340 –RS232C	ASCII driver	2400,4800,9600 Bits/s
CP 341 –RS232C	ASCII driver	1200,2400,4800,9600,19200,38400,57600 Bits/s
CP 441-2	RS232 module (ASCII driver)	1200,2400,4800,9600,19200,38400,57600 Bits/s
ET 200S 1SI/ ASCII	ASCII driver	1200,2400,4800,9600,19200,38400,57600 Bits/s
S7-1500 CM PtP	Freeport protocol (corresponds to ASCII)	1200,2400,4800,9600,19200 Bits/s
ET 200SP CM PtP	Freeport protocol (corresponds to ASCII)	1200,2400,4800,9600,19200,38400,57600 Bits/s

MD720-3

Table 2-12

	Property	Performance data
X1 Interface	Standard connection	RS232, jack; D-SUB 9-pin
	Default transmission rate	9600Bits/ s
	Control via	AT commands
	Possible transmission rate	1200, 2400, 4800, 9600, 19200, 38400, 57600 Baud
Radio interface	GSM module	GPRS / CSD / quad band
	GPRS	• up to 2 uplinks • up to 4 downlinks (max. 5 slots)
	Transmitter power	• GSM 850 MHz (max. 2W) • GSM 900 MHz (max. 2W) • DCS 1800 MHz (max. 1W) • PCS 1900 MHz (max. 1W)

Application software

The following table shows the performance data of the **FB “SMS_Meld”** user block.

Table 2-13

Criterion	Performance data	Note
Transmission rate with ASCII driver	9600 bits/s	Can be changed in the hardware configuration
Number of message texts	20	Can be changed in the block
Number of telephone numbers	5	
Maximum text length	30 characters	
Maximum order length	10 characters	
Coefficient length	6 characters	A variable of the INTEGER type is used as a coefficient. In the user program, this variable is converted into a STRING.

Measured data/empirical values

The following table shows typical key data and reflects the test environment for the development of this application example. It provides only a rough basis.

Table 2-14

Criterion	Empirical value	Remark
Request time for sending an SMS message	10 sec	Average time from send trigger command to end of send operation
Modem initialization duration	25 sec	Average time from initialization trigger command to end of initialization.

3 Functional Mechanisms of this Application

Introduction

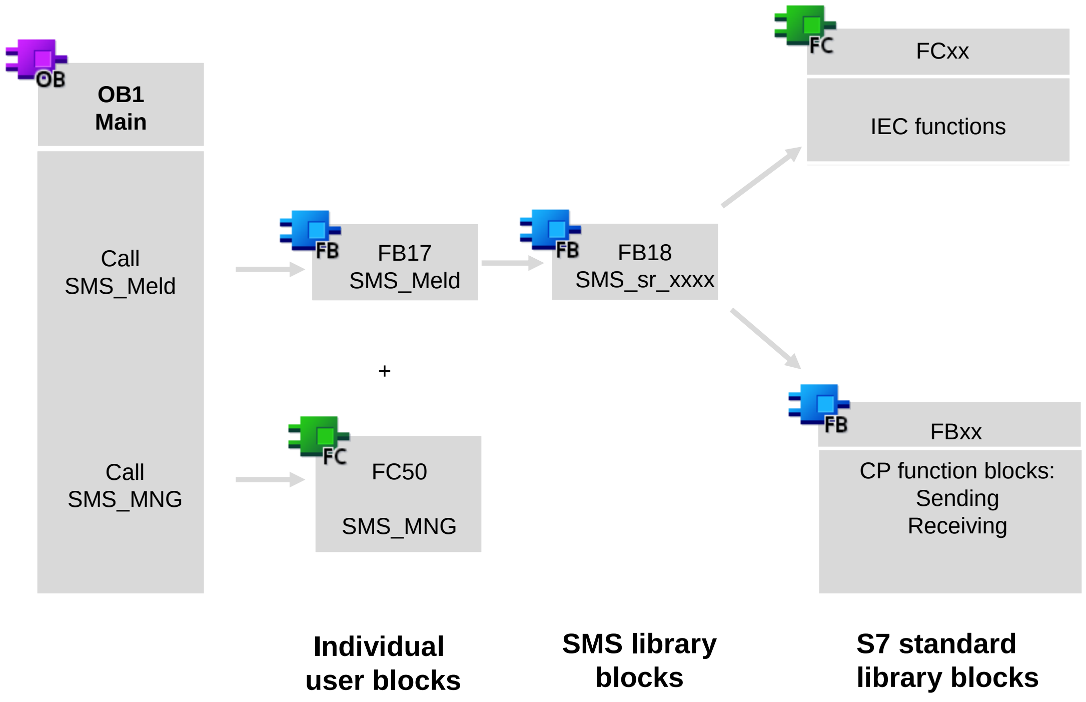
The following sections provide detailed explanations of the functionalities of the different scenarios

- Generating and sending an SMS message
- Sending an SMS message with attached coefficient
- Receiving and evaluating SMS orders and
- Receiving a remote query of a value.

3.1 Program overview

The graphic below shows the program structure of the entire STEP 7 project.

Figure 3-1



Call of SMS_Meld in OB1

The following figure and table show the call interface of the core user block **FB “SMS_Meld”** in OB1

Figure 3-2

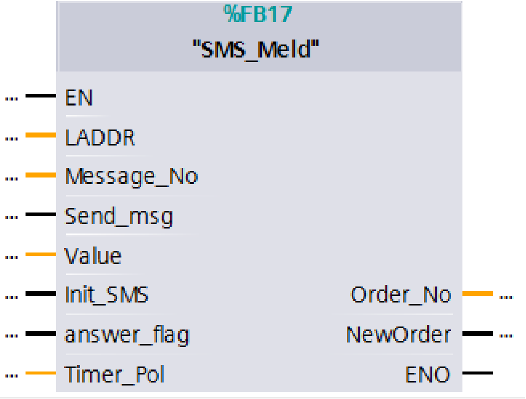


Table 3-1

Symbol	Data type	Explanation
HW_ID	INT/PORT	Specifies the hardware address of the used communication module. S7-300/S7-400/ET 200S: start address of the inputs S7-1500/ET 200SP: HW identifier from HWCN
Message_No	INT	Transfers the number of the SMS message to be generated
Send_msg	BOOL	Send trigger command
Value	INT	Optional coefficient
Init_SMS	BOOL	Initialization trigger command
answer_flag	BOOL	sets telephonnumber to the one last SMS was received from
Order_No	INT	Output order number
NewOrder	BOOL	Set to TRUE when a new SMS message is received.
Timer_Pol	TIME	Once the timer has elapsed, the block polls the received SMS from MD720-3.

Function block **FB “SMS_Meld”** is called cyclically in OB1 and internally calls function block **“SMS_sr_xxxx”** in return.

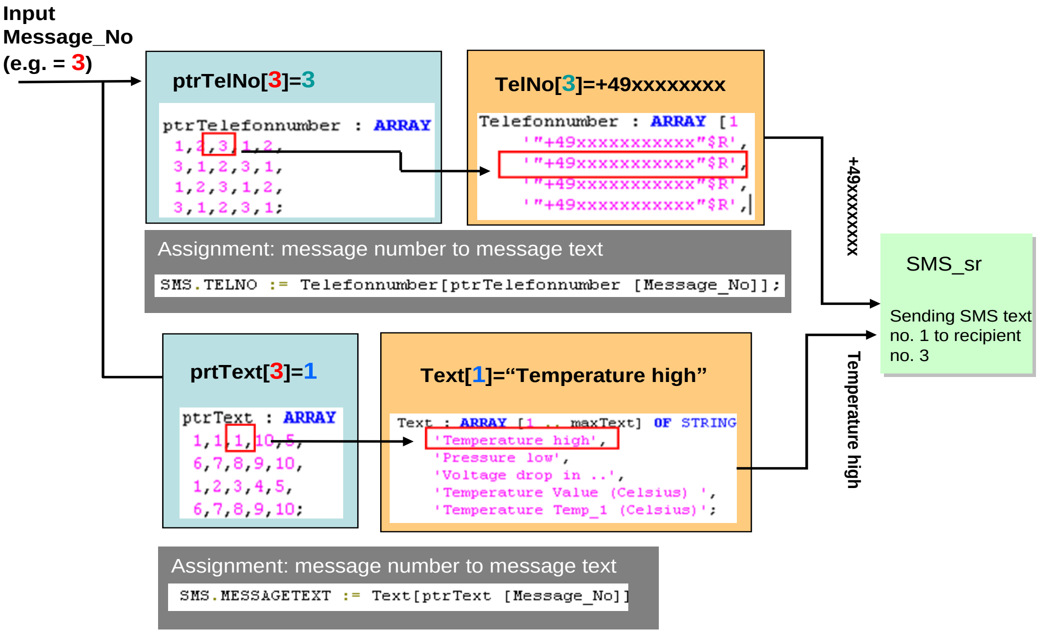
3.2 Scenario: “Generating and sending an SMS message”

Function block **FB “SMS_Meld”** generates a configurable text message via an internal data array using the **Message_No** input variable and sends it to a recipient assigned to this variable.

3.2.1 Diagram for the “Generating and sending an SMS message” scenario

The diagrammatic representation below shows how **FB “SMS_Meld”** generates a text message from a message number.

Figure 3-3


Input
Message_No
(e.g. = 3)
ptrTelNo[3]=3
TelNo[3]=+49xxxxxxxx
+49xxxxxxxx
SMS_sr
Sending SMS text no. 1 to recipient no. 3
prtText[3]=1
Text[1]=“Temperature high”
Temperature high
Assignment: message number to message text
Assignment: message number to message text

```
ptrTelefonnumber : ARRAY
 1,2,3,1,2,
 3,1,2,3,1,
 1,2,3,1,2,
 3,1,2,3,1;
```

```
Telefonnumber : ARRAY [1
   '"+49xxxxxxxxxxx"$R',
   '"+49xxxxxxxxxxx"$R',
   '"+49xxxxxxxxxxx"$R',
   '"+49xxxxxxxxxxx"$R',
```

```
SMS.TELNO := Telefonnumber[ptrTelefonnumber [Message_No]];
```

```
ptrText : ARRAY
 1,1,1,10,5,
 6,7,8,9,10,
 1,2,3,4,5,
 6,7,8,9,10;
```

```
Text : ARRAY [1 .. maxText] OF STRING
   'Temperature high',
   'Pressure low',
   'Voltage drop in ..',
   'Temperature Value (Celsius) ',
   'Temperature Temp_1 (Celsius)';
```

```
SMS.MESSAGETEXT := Text[ptrText [Message_No]]
```

The **Message_No** variable contains the message number that is necessary to generate an SMS message. Using this variable, the associated telephone number and message text are determined from a total of four arrays.

Two of these lists are linked to one another:

- The “ptrTelefonnumber” and “Telefonnumber” lists are used to determine the telephone number.
  Hint: If you use the INPUT “answer_flag” the used telephone number is the one out of RECV_TELNO.
- The “ptrText” and “Text lists” are used to determine the message text.

Depending on the value of the **Message_No** variable, the number is now determined from the **ptrTelefonnumber** and **ptrText** lists, which then defines the position in the list in **Telefonnumber** and **Text**.
In this way, the information necessary for the SMS message is determined (telephone number and message text).

You can create any allocation and individually design the SMS message by reconfiguring the indices and contents of the arrays.

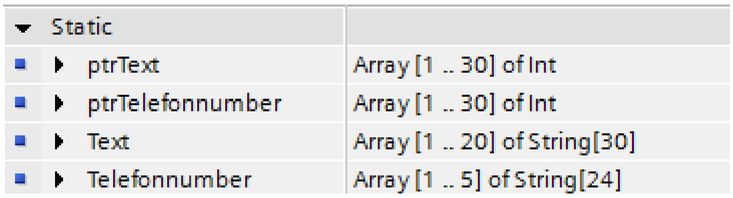
3.2.2 Program details for the “Generating and sending an SMS message” scenario

This chapter illustrates the most important code fragments of this function block from the documented source code of this example project.

Variables for SMS allocation

You may adapt the length of the arrays to your individual requirements.

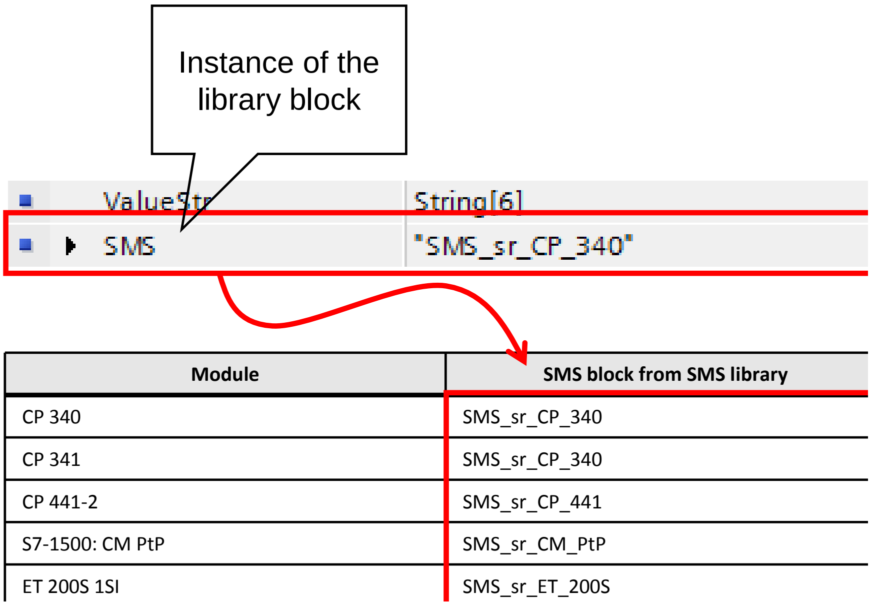
Figure 3-4



Static	
ptrText	Array [1 .. 30] of Int
ptrTelefonnumber	Array [1 .. 30] of Int
Text	Array [1 .. 20] of String[30]
Telefonnumber	Array [1 .. 5] of String[24]

General variables

Figure 3-5



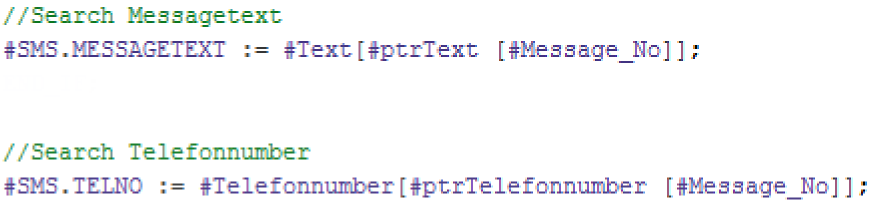
ValueStr	String[6]
SMS	"SMS_sr_CP_340"

Module	SMS block from SMS library
CP 340	SMS_sr_CP_340
CP 341	SMS_sr_CP_340
CP 441-2	SMS_sr_CP_441
S7-1500: CM PtP	SMS_sr_CM_PtP
ET 200S 1SI	SMS_sr_ET_200S

Message assignment code fragment

The following code lines illustrate how Message_No is assigned to SMS text and telephone number in SCL.

Abbildung 3-6



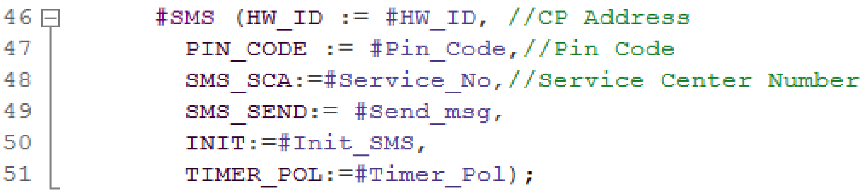
```
//Search Messagetext
#SMS.MESSAGETEXT := #Text[#ptrText [#Message_No]];


//Search Telefonnumber
#SMS.TELNO := #Telefonnumber[#ptrTelefonnumber [#Message_No]];
```

Call of SMS_sr_xxxx

To send the SMS message, the **FB “SMS_Meld”** function block calls the **FB “SMS_sr_xxxx”** library block.

The screenshot below shows the call of the **FB “SMS_sr_xxxx”** function block.

Figure 3-7



```
    #SMS (HW_ID := #HW_ID, //CP Address
      PIN_CODE := #Pin_Code,//Pin Code
      SMS_SCA:=#Service_No,//Service Center Number
      SMS_SEND:= #Send_msg,
      INIT:=#Init_SMS,
      TIMER_POL:=#Timer_Pol);
```

3.3 Scenario: “Sending an SMS message with attached coefficient”

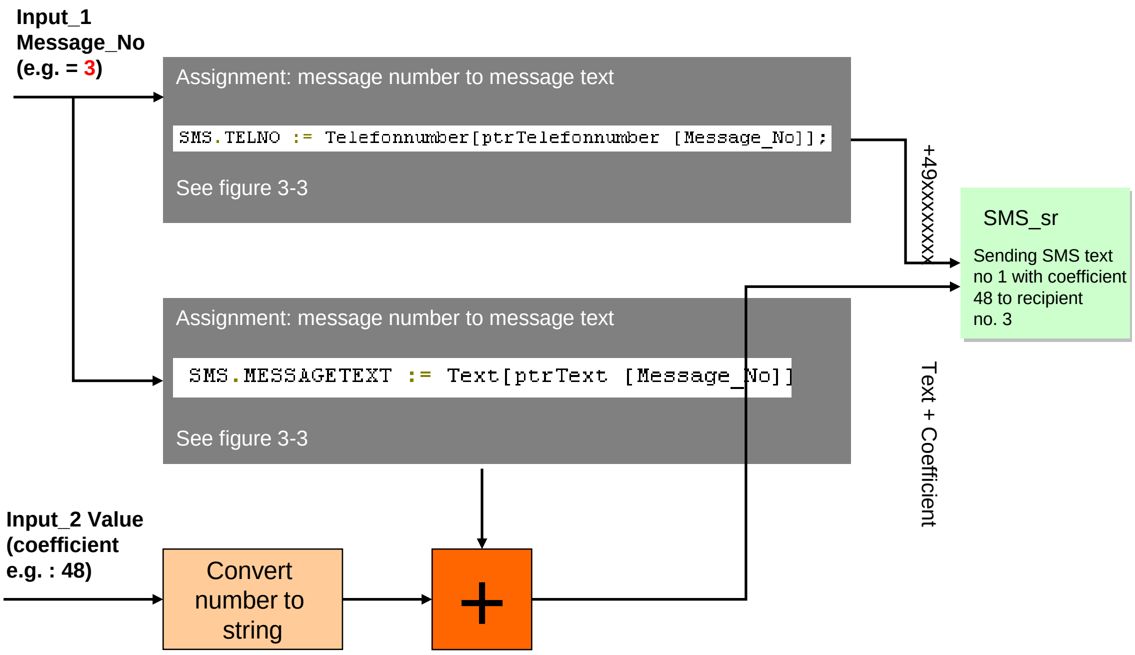
This scenario works like the “Generating and sending an SMS message” scenario. The only difference is that the message text can be sent with an attached coefficient.

The coefficient is stored in the **“Value”** variable (type:INT) and, using the **“ValueStr”** variable, must first be converted to the STRING data type for further processing.

3.3.1 Diagram for the “Sending an SMS message with attached coefficient” scenario

The diagrammatic representation below shows how **FB “SMS_Meld”** generates a text message from a message number and sends it to the recipient with an attached coefficient.

Figure 3-8



```
SMS.TELNO := Telefonnumber[ptrTelefonnumber [Message_No]];
```

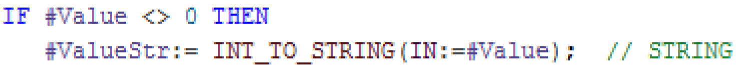
```
SMS.MESSAGETEXT := Text[ptrText [Message_No]]
```

3.3.2 Program details for the “Sending an SMS message with attached coefficient” scenario

This chapter illustrates the most important code fragments of this function block from the documented source code of this example project.

Converting the “Value” variable from number to text

Figure 3-9

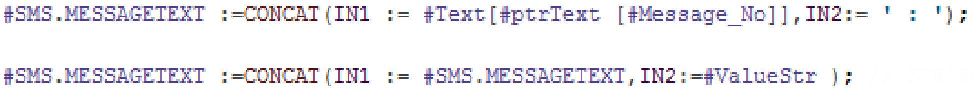


```
IF #Value <> 0 THEN
   #ValueStr:= INT_TO_STRING(IN:=#Value);  // STRING
```

Message assignment code fragment

The following code lines illustrate how to attach a coefficient to a message text in SCL.

Figure 3-10



```
#SMS.MESSAGETEXT :=CONCAT(IN1 := #Text[#ptrText [#Message_No]],IN2:= ' : ');

#SMS.MESSAGETEXT :=CONCAT(IN1 := #SMS.MESSAGETEXT,IN2:=#ValueStr );
```

3.4 Scenario: “Receiving and evaluating SMS orders”

The **FB “SMS_Meld”** function block checks all incoming SMS messages for special “keywords”. When checking the received text with the keyword is positive, the respective order number will be output on the “**Order-No**” parameter.

The receive buffer of the GSM modem is checked at regular intervals. The time value is defined by the value of the **Timer_Polling** parameter (standard value: 1 minute). After **receiving** a new message, the information contained in the message is stored in the receive data area.

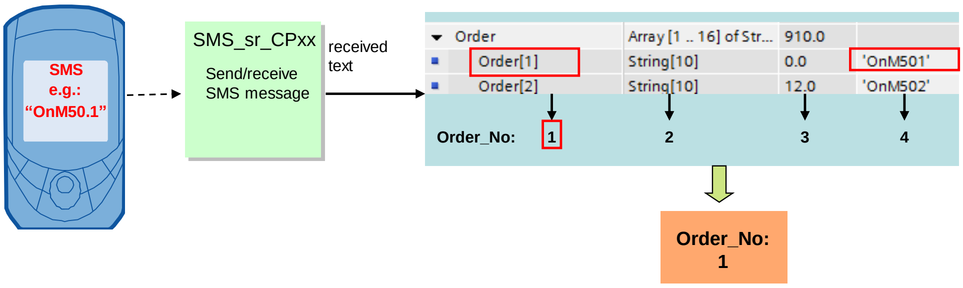
CAUTION	**Do not send an SMS message containing the key word ‘ERROR’ or ‘OK’, since MD720-3 evaluates these words when a faulty behavior occurs during sending from MD720-3 to the communication module.**

NOTE	For sending SMS to the modem it is recommended to use only alphabetic characters and figures. This is in order to avoid display and recognition errors of the STEP 7 application program.

3.4.1 Diagram for the “Evaluating SMS orders” scenario

The diagrammatic representation below shows how **FB “SMS_Meld”** receives and evaluates an incoming SMS message. The keywords are stored in the **Order** array.

Abbildung 3-11



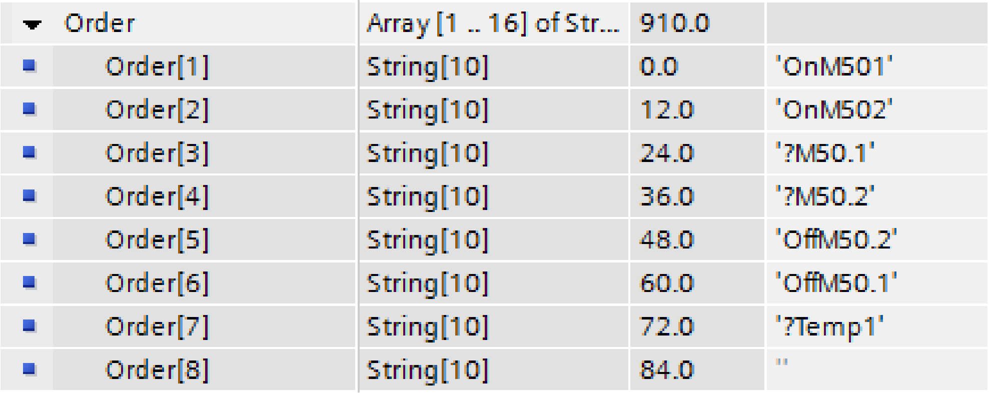
Assignment: message text to order number

3.4.2 Program details for the “Evaluating SMS orders” scenario

Variables for order allocation

The following code fragment shows the “keywords” for which an SMS message is checked in this example. You can enter any special “keywords” in the **FB “SMS_Meld”** SCL source.

Figure 3-12

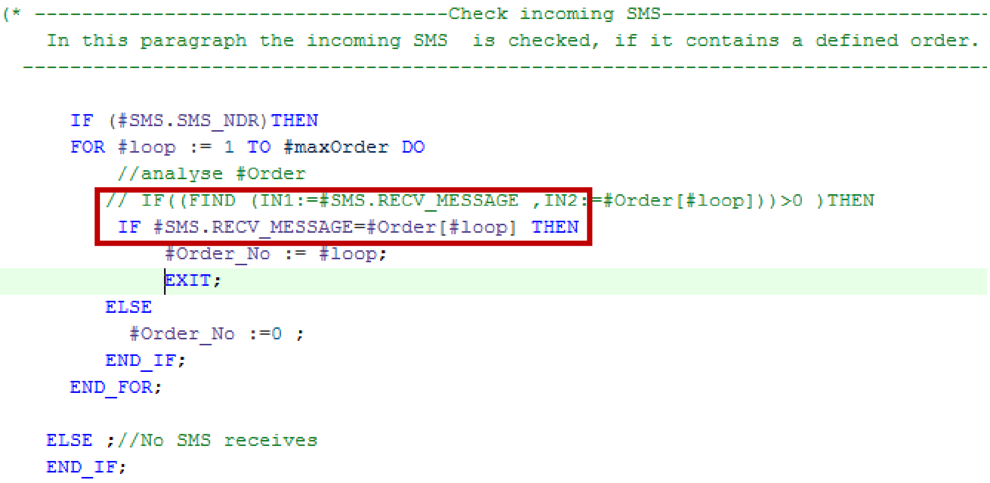


▼ Order	Array [1 .. 16] of Str...	910.0	
■ Order[1]	String[10]	0.0	'OnM501'
■ Order[2]	String[10]	12.0	'OnM502'
■ Order[3]	String[10]	24.0	'?M50.1'
■ Order[4]	String[10]	36.0	'?M50.2'
■ Order[5]	String[10]	48.0	'OffM50.2'
■ Order[6]	String[10]	60.0	'OffM50.1'
■ Order[7]	String[10]	72.0	'?Temp1'
■ Order[8]	String[10]	84.0	''

Order assignment code fragment

The following section of the documented source code shows how the incoming SMS message is checked for special “keywords” in a search loop.

Figure 3-13



```
(* -----------------------------------Check incoming SMS----------------------------
    In this paragraph the incoming SMS  is checked, if it contains a defined order.
  ----------------------------------------------------------------------------------

      IF (#SMS.SMS_NDR)THEN
      FOR #loop := 1 TO #maxOrder DO
          //analyse #Order
         // IF((FIND (IN1:=#SMS.RECV_MESSAGE ,IN2:=#Order[#loop]))>0 )THEN
          IF #SMS.RECV_MESSAGE=#Order[#loop] THEN
              #Order_No := #loop;
              EXIT;
         ELSE
           #Order_No :=0 ;
         END_IF;
      END_FOR;

    ELSE ;//No SMS receives
    END_IF;
```

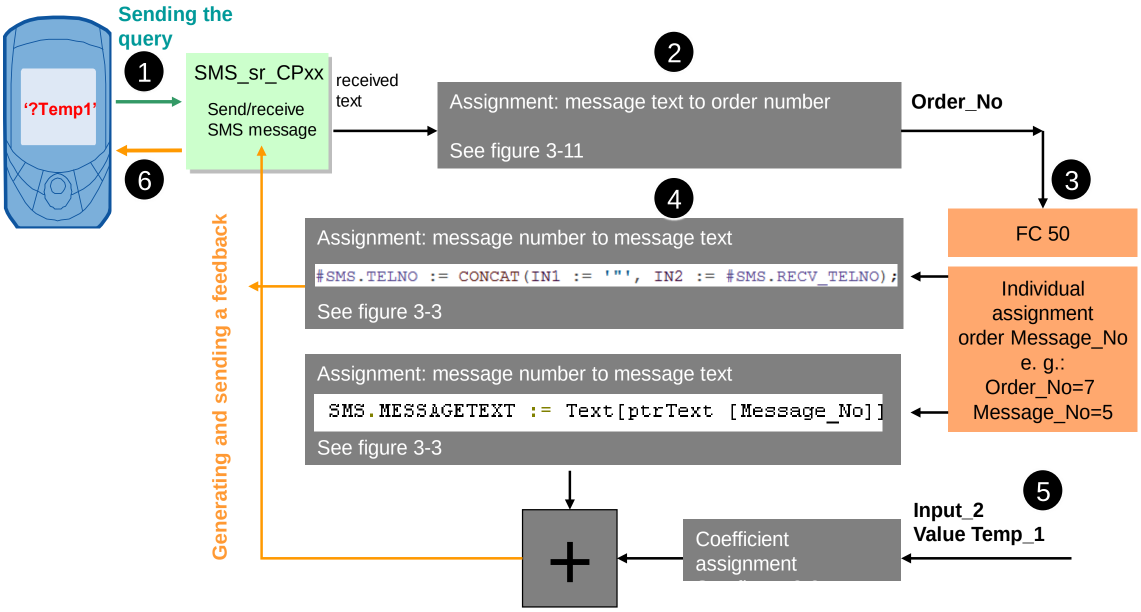
3.5 Scenario: “Receiving a remote query of a value”

Using the **FB “SMS_Meld”** block, SMS messages for querying a value (e.g., the analog value of a temperature) can be interpreted. An automatic feedback message is sent with the aid of the **FC “SMS_MNG”** function.

3.5.1 Diagram for the “Receiving a remote query of a value” scenario

The diagrammatic representation below shows how **FC “SMS_MNG”** sends an answer to a query of a value back to the sender.

Figure 3-14


Sending the query
1
‘?Temp1’
6
SMS_sr_CPxx
Send/receive SMS message
received text
2
Assignment: message text to order number
See figure 3-11
Order_No
3
FC 50
Individual assignment order Message_No e. g.: Order_No=7 Message_No=5
4
Assignment: message number to message text
See figure 3-3
Assignment: message number to message text
See figure 3-3
Generating and sending a feedback
+
Coefficient assignment
5
Input_2
Value Temp_1

```
#SMS.TELNO := CONCAT(IN1 := '"', IN2 := #SMS.RECV_TELNO);
```

```
SMS.MESSAGETEXT := Text[ptrText [Message_No]]
```

Table 3-2

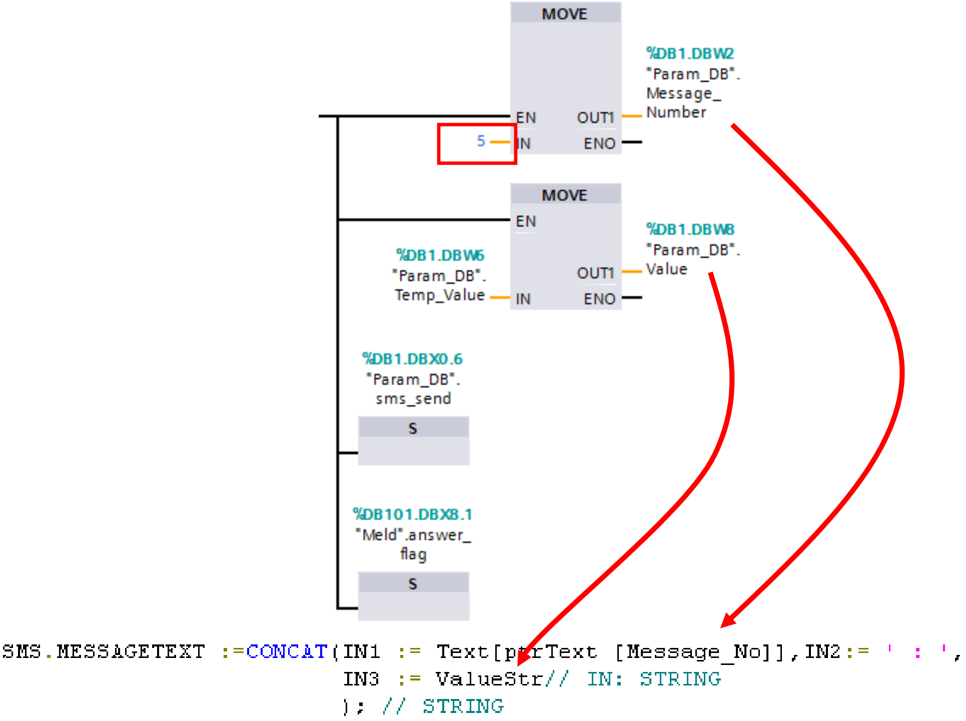
No.	Description
1.	A query is sent to the controller.
2.	The received text is evaluated and assigned to an order number.
3.	With the aid of FC 50, the order number is assigned to a message number.
4.	Message text is generated from this message number, recipient number is extracted from received SMS.
5.	The coefficient is attached to the message text.
6.	The feedback message is generated and sent to the user.

3.5.2 Program details for the “Receiving a remote query of a value” scenario

The screenshots below shows the functionality of **FC “SMS_MNG”**.
To implement this scenario, the block is called in OB1.

When receiving an SMS with the text “?Temp_1”, the current temperature is returned to the sender.

Figure 3-15



```
SMS.MESSAGETEXT :=CONCAT(IN1 := Text[ptrText [Message_No]],IN2:= ' : ',
                         IN3 := ValueStr// IN: STRING
                         ); // STRING
```

Message_No and value are used in the SCL source for generating the feedback.

4 Installation

4.1 Hardware installation of the S7 station

Overview

This application example includes the following S7 project variants:

Table 4-1

Versions
S7-1500: CM PtP HW Config and S7 program for the variant with CM PtP
CP_340 HW Config and S7 program for the variant with CM 340
CP_341 HW Config and S7 program for the variant with CM 341
ET 200S _1SI HW Config and S7 program for variant with ET 200S SI module
CP_441-2 HW Config and S7 program for the variant with CM 441-2
Et 200SP: CM PtP HW Config and S7 program for the variant with CM PtP

NOTICE

The following applies to all project variants:
Before you switch on the power supply, complete and check the installation!

4.1.1 S7-1500 station with CM PtP

Figure 4-1

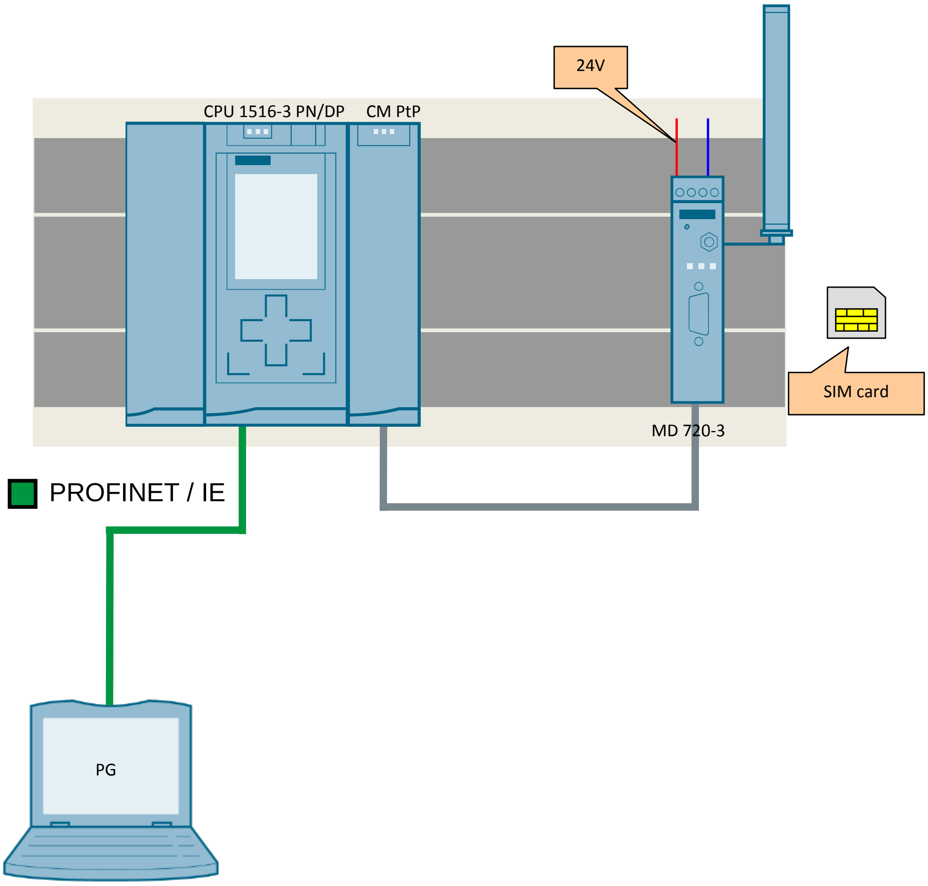


Table 4-2

No.	Action	Note
1.	Insert the individual modules in a suitable rack.	List of components Table 2-3.
2.	Use a backplane bus connector to connect CPU and CP.	
3.	Connect all respective components to a 24 V direct current source. Connect the direct current source to the power grid (230 V AC).	Take note of correct polarity.
4.	Connect the PG with your S7-1500 CPU.	
5.	Connect the MD720-3 modem to the CM via a cable with 9-pole D-Sub connectors.	

4.1.2 S7-300 station with CP 340

The figure below shows the hardware configuration of the project variant with CP340.

Figure 4-2

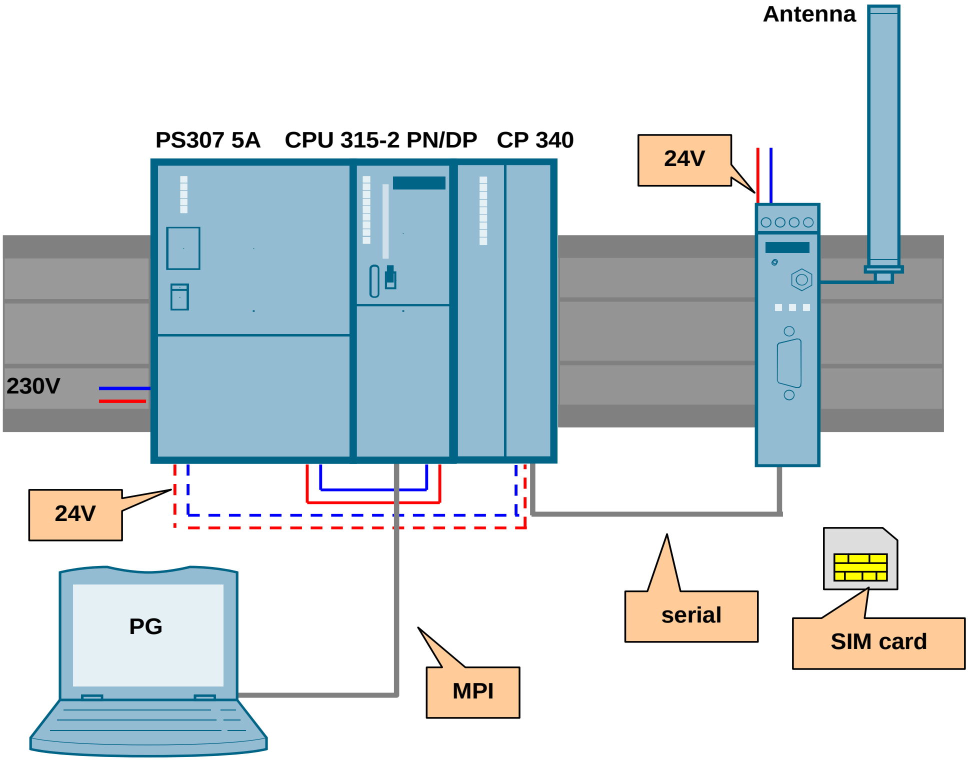


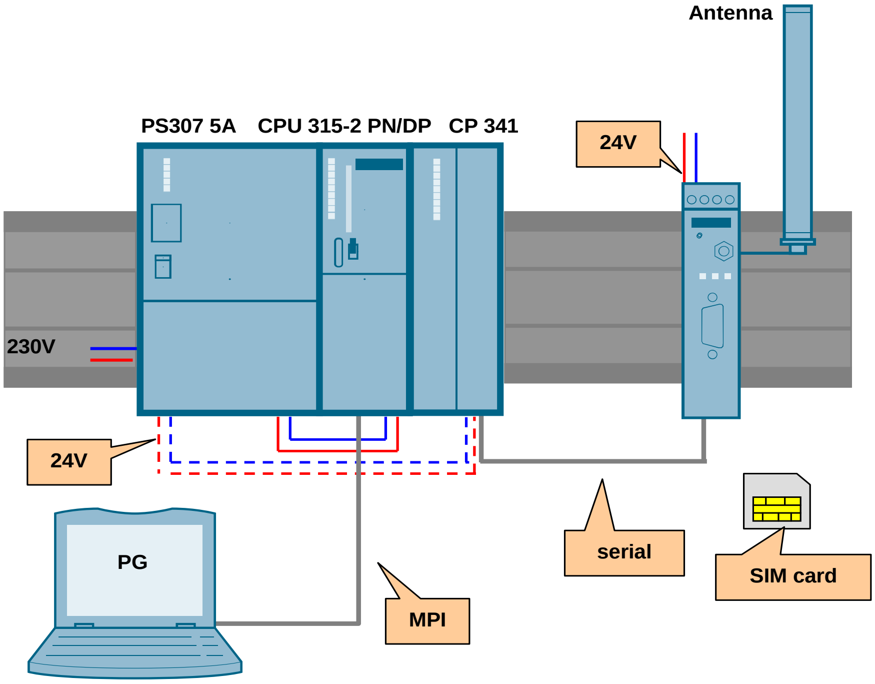
Table 4-3

No.	Action	Note
1.	Insert the individual modules in a suitable rack.	List of components Table 2-4
2.	Use a backplane bus connector to connect CPU and CP.	
3.	Connect all respective components to a 24 V direct current source (PS307). Connect the PS307 to the power grid (230 V AC).	Take note of correct polarity.
4.	Connect the MPI of the engineering PG to the MPI of the S7 CPU.	You can also connect the CPU to the PG via an Ethernet cable.
5.	Connect the MD720-3 modem to the CP via a cable with 9-pole D-Sub connectors.	

4.1.3 S7-300 station with CP 341

The figure below shows the hardware configuration of the project variant with CP341.

Figure 4-3



Use the CP341 to configure the S7 station as shown in Table 4-3.

4.1.4 S7-300 station with ET 200S with 1SI module

The figure below shows the configuration of the S7-300 / ET 200S station.

Figure 4-4

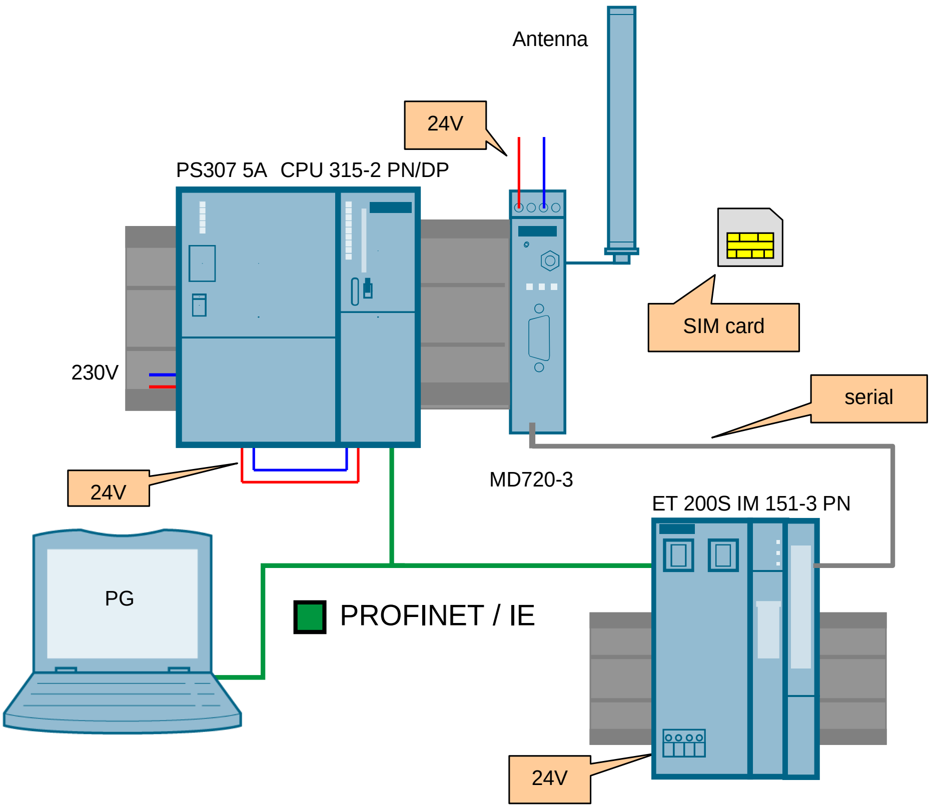
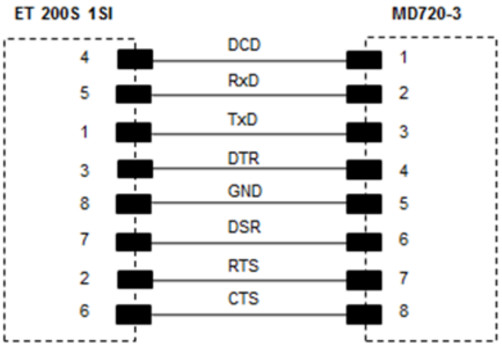


Table 4-4

No.	Action	Note
1.	Insert the individual modules in a suitable rack.	List of components Table 2-6
2.	Connect the PS307 to the network. (230 V AC)	Take note of correct polarity.
3.	Connect the following Profinet interfaces: • Engineering PG to the S7 CPU. • S7 CPU to the ET 200S.	
4.	Connect the serial cable to the 1SI module. Connect the other side to the MD 720-3 (see manual \3\).	 ET 200S 1SI MD720-3 4 DCD 1 5 RxD 2 1 TxD 3 3 DTR 4 8 GND 5 7 DSR 6 2 RTS 7 6 CTS 8

4.1.5 S7-400 station with CP 441-2

The figure below shows the hardware configuration of the project variant with CP441-2.

Figure 4-5

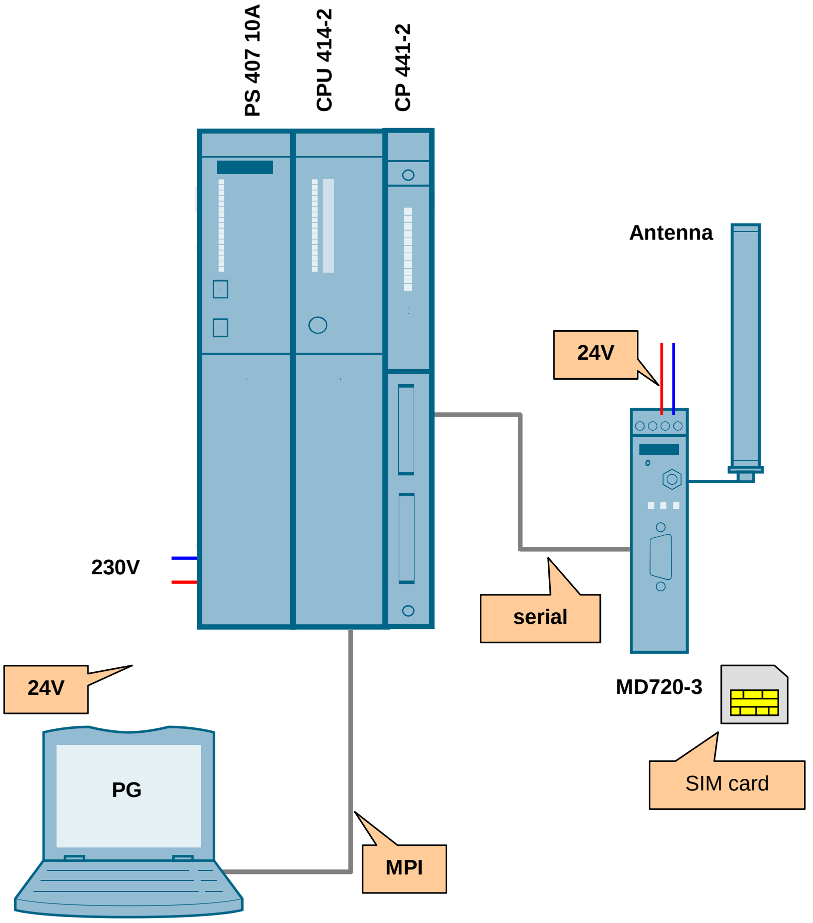


Table 4-5

No.	Action	Note
1.	Insert the individual modules in a suitable rack.	List of components Table 2-5
2.	Connect the PS407 to the network. (230 V AC)	Take note of correct polarity.
3.	Connect the MPI of the engineering PG to the MPI of the S7 CPU.	
4.	Connect the MD720-3 modem to the CP via a cable with 9-pole D-Sub connectors.	

4.1.6 S7-1500 station with ET 200SP with CM PtP

Figure 4-6

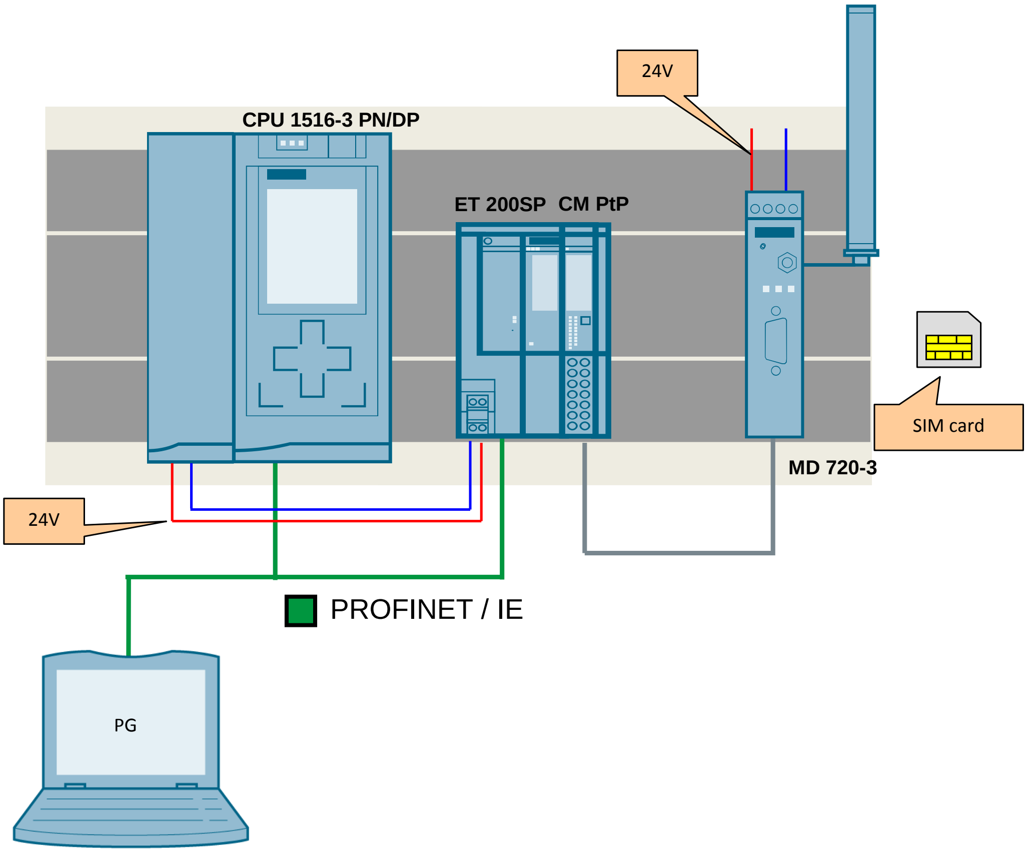
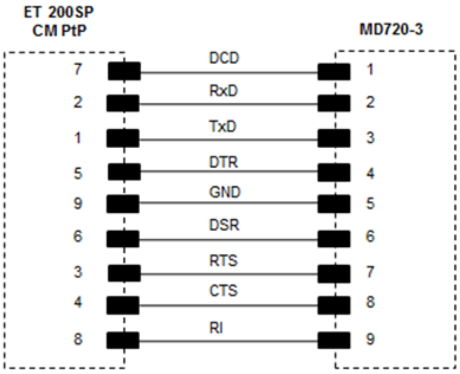


Table 4-6

No.	Action	Note
1.	Insert the individual modules in a suitable rack.	List of components Table 2-7.
2.	Connect the PS307 to the network. (230 V AC)	Take note of correct polarity.
3.	Connect the following Profinet interface: • Engineering PG to the S7 CPU. • S7 CPU to the ET 200S.	
4.	Connect the serial cable to the CM PtP module of the ET 200SP. Connect the other side to the MD 720-3 (see manual \3\).	 ET 200SP CM PtP MD720-3 7 DCD 1 2 RxD 2 1 TxD 3 5 DTR 4 9 GND 5 6 DSR 6 3 RTS 7 4 CTS 8 8 RI 9

4.2 MD720-3 hardware configuration

Table 4-7

No.	Action	Note
1.	Open the casing of the MD720-3 and insert the SIM card. Close the MD720-3.	Please note the Manual for MD720-3
2.	Connect the antenna to the respective socket.	
3.	Connect the MD720-3 to a 24 V direct current source.	
4.	Connect the serial cable to the interface of the MD720-3 via a 9-pole D-Sub connector.	

Note The installation guidelines for the installation of all components always have to be observed.

4.3 Installation of the software

The Engineering station is used as the PG for projecting the the S7 station.

Table 4-8

Action	Note
Install STEP 7 as of V13.	Follow the instructions of the system manual

5 Commissioning the Application

5.1 General preparations

5.1.1 Installation of the application software

The following table shows all steps required for the installation of the sample code.

Table 5-1

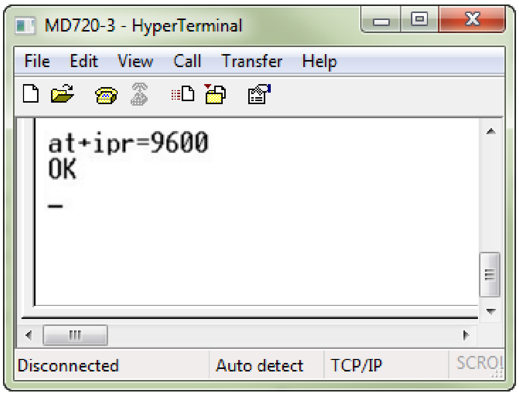
Step	Procedure
1.	Download the SMS_Example.zip file to your engineering station and unzip the folder.
2.	In the extracted program folder, double-click on the "SMS_Example.ap12" icon. Now the project opens in TIA V13.

5.1.2 Setting the baud rate of the MD720-3

In this project, the baud rate of the CPs is always set to 9600 bps. CP and MD720-3 must support the same baud rate.

The rate at the MD720-3 is modified using a terminal program.

Table 5-2

Step	Procedure
1.	Connect a PC to the serial interface of the MD720-3.
2.	Start a terminal program, for example HyperTerminal.
3.	Select the appropriate COM interface to which the MD720-3 modem has been connected.
4.	Set the character format and baud rate to the same values as the serial interface of the MD720-3. The factory settings of the MD720-3 are as follows: Baud rate: 19200 bits/s Character format: 8N1. The baud rate is changed via AT command: **AT+IPR=<baud rate>**. Enter this command in the terminal program and press the return key.  MD720-3 - HyperTerminal File Edit View Call Transfer Help Disconnected Auto detect TCP/IP SCROL at+ipr=9600 OK

.

Note The modem is only accessed by AT commands when it is in the command phase. If this is not the case, the modem must be reset to factory settings (manual: \3\).

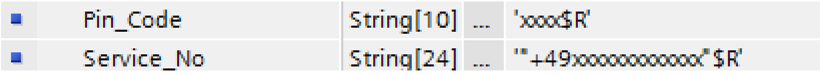
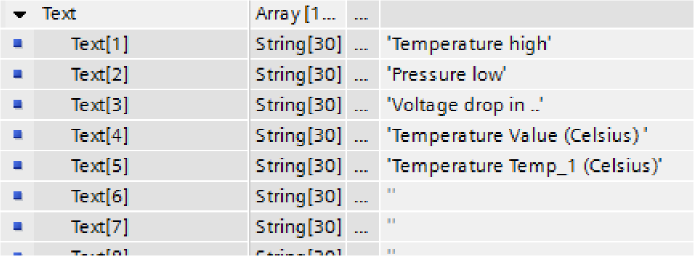
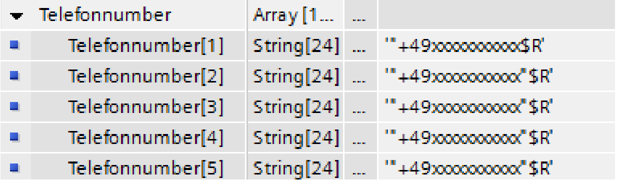
5.1.3 Adjust the SMS_Meld block

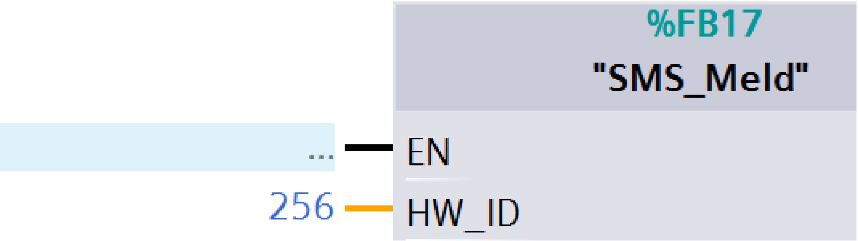
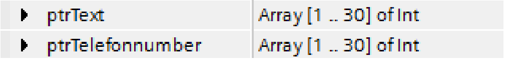
Before you can send and receive SMS messages, you have to store the following information for the initialization of the MD720-3 modem:

- PIN of the SIM card in the modem
- Number of the short message service center (see library description “25545680_Library_SMS_MD720_DOKU_V13_V1_0_e.pdf”, table 4-1 “SMS_SCA”).

In addition, the text to be sent and the telephone number of the recipient must be entered.

Table 5-3

No.	Procedure
1.	Open the **SMS_Meld** block.
2.	For the **Pin_code** parameter, enter the PIN of the SIM card you have inserted into the MD720-3 and for the **Service_No** parameter, enter the short message service center number. For the number of the short message service center, please contact your provider, for example.  Pin_Code String[10] ... 'xxxx$R' Service_No String[24] ... '"+49xxxxxxxxxxxx"$R' Make sure that you replace only the telephone number (+49xxxxxxxxxx) or PIN (xxxxx) respectively; the “$R” control character must **NOT** be deleted!
3.	In the **Text** array, you can enter the individual message texts that are to be sent for the assigned Message_No (to the array index).  Text Array [1... ... Text[1] String[30] ... 'Temperature high' Text[2] String[30] ... 'Pressure low' Text[3] String[30] ... 'Voltage drop in ..' Text[4] String[30] ... 'Temperature Value (Celsius) ' Text[5] String[30] ... 'Temperature Temp_1 (Celsius)' Text[6] String[30] ... '' Text[7] String[30] ... '' Text[8] String[30] ... ''
4.	In the **Telefonnumber** array, enter the possible recipient numbers to which the SMS messages are to be sent.  Telefonnumber Array [1... ... Telefonnumber[1] String[24] ... '"+49xxxxxxxxxx$R' Telefonnumber[2] String[24] ... '"+49xxxxxxxxxx"$R' Telefonnumber[3] String[24] ... '"+49xxxxxxxxxx"$R' Telefonnumber[4] String[24] ... '"+49xxxxxxxxxx"$R' Telefonnumber[5] String[24] ... '"+49xxxxxxxxxx"$R' Make sure that you replace only the telephone number (+49xxxxxxxxxx). The “$R” control character must **NOT** be deleted!

No.	Procedure
5.	In OB1 you enter the logic address of the CP at the **HW_ID** parameter. 
6.	In the **ptrText** and **ptrTelefonnumber** allocation arrays, you set the values so any message can be sent to any recipient.  ptrText Array [1 .. 30] of Int ptrTelefonnumber Array [1 .. 30] of Int
7.	Compile your user program.

5.1.4 Configuring the serial CPs

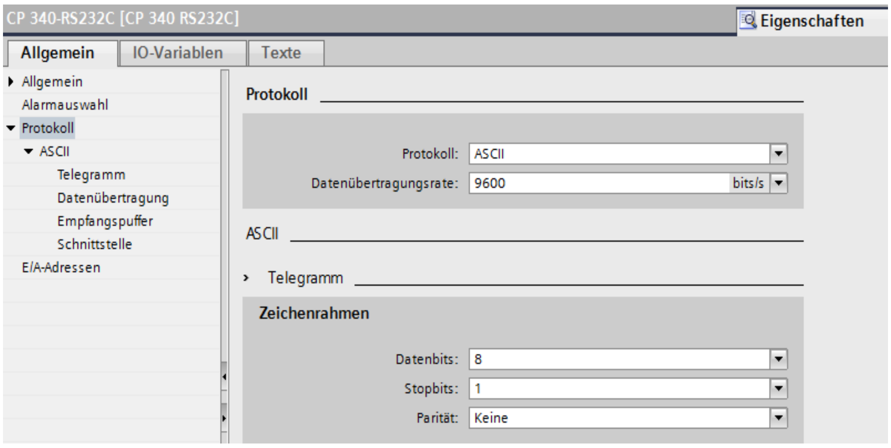
Configuration for CM PtP/ CP 340/ CP 341/ ET 200S 1SI

In this example, the serial CP uses the ASCII driver for the interface. This allows sending or receiving AT commands and mere text code to or from a communications partner via the interface.

The properties of the ASCII driver are to be as follows:

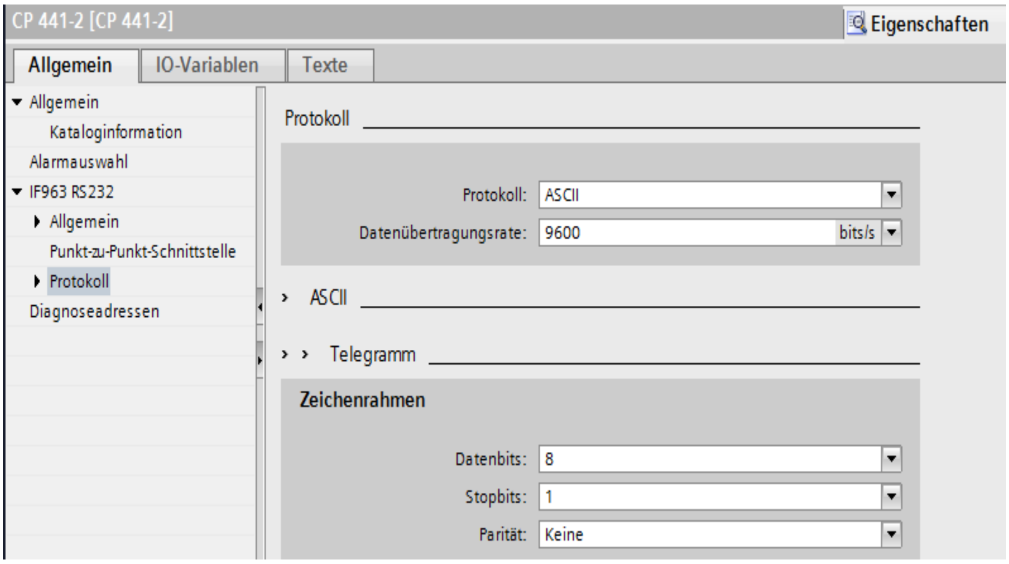
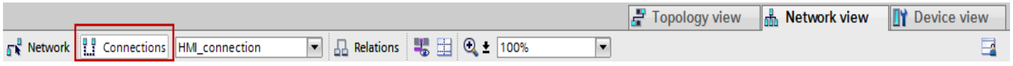
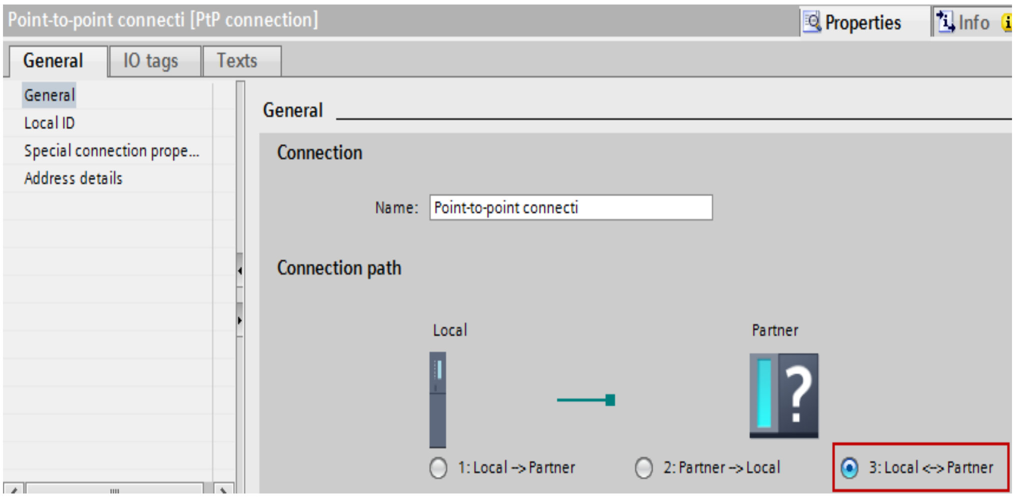
- Protocol: ASCII/Freeport
- Transmission rate: 9600 bps
- Data bits: 8
- Stop bits: 1
- No parity check (none)

Table 5-4

No.	Action
1.	In the device configuration you click on the CP/ the module/ the CM. In the inspector window, open the **Properties** tab. Press on **Protocol**.
2.	Here you can set the parameters listed above.  CP 340-RS232C [CP 340 RS232C] Eigenschaften Allgemein IO-Variablen Texte Allgemein Alarmauswahl Protokoll ASCII Telegramm Datenübertragung Empfangspuffer Schnittstelle E/A-Adressen Protokoll ASCII Telegramm Zeichenrahmen Protokoll: ASCII Datenübertragungsrate: 9600 bits/s Datenbits: 8 Stopbits: 1 Parität: Keine **Note** The configuration user interface of the individual communication modules differs slightly.
3.	Then save your project.

Parameterization and configuration for CP 441-2

Table 5-5

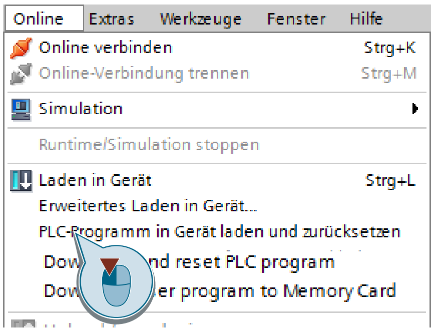
No.	Action
1.	CP_441-2 has the **RS232 module** at **interface 1** and **no module** at **interface 2**. If the interface assignment of your CP differs from the one described above, HW Config must be changed accordingly.
2.	In the device configuration, click on the CP. In the inspector window, open the **Properties** tab. Press on **IF963 RS232 > Protocol**.
3.	Here you can set the parameters listed above.  CP 441-2 [CP 441-2] — Eigenschaften Allgemein | IO-Variablen | Texte Allgemein Kataloginformation Alarmauswahl IF963 RS232 Allgemein Punkt-zu-Punkt-Schnittstelle Protokoll Diagnoseadressen Protokoll Protokoll: ASCII Datenübertragungsrate: 9600 bits/s ASCII Telegramm Zeichenrahmen Datenbits: 8 Stopbits: 1 Parität: Keine
4.	Go to the network view and click on “Connections”.  Topology view Network view Device view Network Connections HMI_connection Relations 100% Right-click on your S7-400 CPU and press the “Add new connection” button.
5.	Add a PtP connection.
6.	Select the connection in “Connections” above the inspection window and then got to the “Properties” in the inspection window.
7.	In “General” you select “3: Local <-> Partner”.  Point-to-point connecti [PtP connection] — Properties | Info General | IO tags | Texts General Local ID Special connection prope... Address details General Connection Name: Point-to-point connecti Connection path Local Partner ○ 1: Local -> Partner ○ 2: Partner -> Local ● 3: Local <-> Partner
8.	Save your project.

Note If you want to configure further S7-400 CPs, you need to select different local IDs respectively (change them via instance DB: con_id)

5.2 Download the STEP 7 project

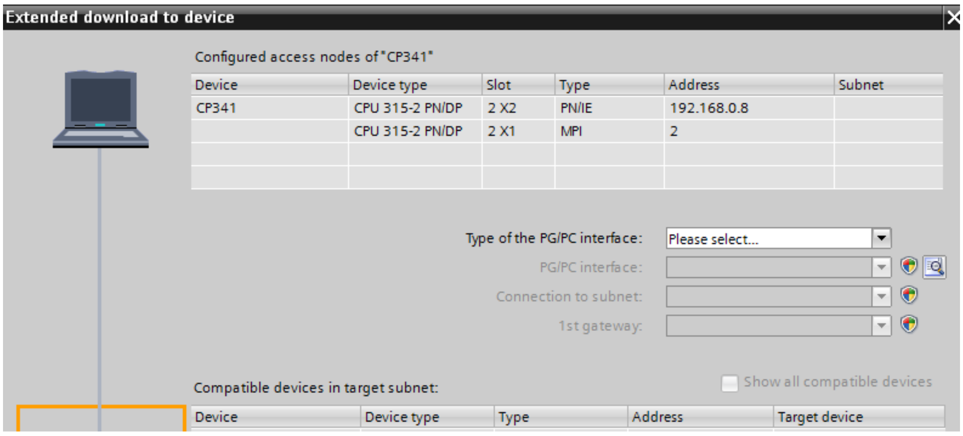
Table 5-6

<table>
<tr><th>No.</th><th>Action</th></tr>
<tr><td>1.</td><td>Depending on your hardware setup, click on the CPU you wish to load and then select “Online > Download and Reset PLC program”.



Online Extras Werkzeuge Fenster Hilfe
Online verbinden Strg+K
Online-Verbindung trennen Strg+M
Simulation
Runtime/Simulation stoppen
Laden in Gerät Strg+L
Erweitertes Laden in Gerät...
PLC-Programm in Gerät laden und zurücksetzen
Dow nd reset PLC program
Dow er program to Memory Card

</td></tr>
<tr><td>2.</td><td>Select the PC interface used by you and, click on your CPU detected by STEP 7 and click on “Load” for loading.



Extended download to device

Configured access nodes of "CP341"

<table>
<tr><th>Device</th><th>Device type</th><th>Slot</th><th>Type</th><th>Address</th><th>Subnet</th></tr>
<tr><td>CP341</td><td>CPU 315-2 PN/DP</td><td>2 X2</td><td>PN/IE</td><td>192.168.0.8</td><td></td></tr>
<tr><td></td><td>CPU 315-2 PN/DP</td><td>2 X1</td><td>MPI</td><td>2</td><td></td></tr>
</table>

Type of the PG/PC interface: Please select...

PG/PC interface:

Connection to subnet:

1st gateway:

Compatible devices in target subnet:

□ Show all compatible devices

<table>
<tr><th>Device</th><th>Device type</th><th>Type</th><th>Address</th><th>Target device</th></tr>
</table></td></tr>
<tr><td>3.</td><td>Load the program into your CPU via the appearing menus.</td></tr>
</table>

6 Operation of the Application

This chapter shows you how to operate the described scenarios of this application.

All necessary variables can be found in the **"Watch_MELD"** variable table.

Note For information on errors that may occur when operating the application, please refer to the library description "25545680_Library_SMS_MD720_DOKU_V2.0_e.pdf". If the support does not lead to the desired result, it may be useful to download the entire project.

6.1 Sending an SMS message (scenario 1/scenario 2)

Sending an SMS message is controlled via the **"sms_send"** flag.

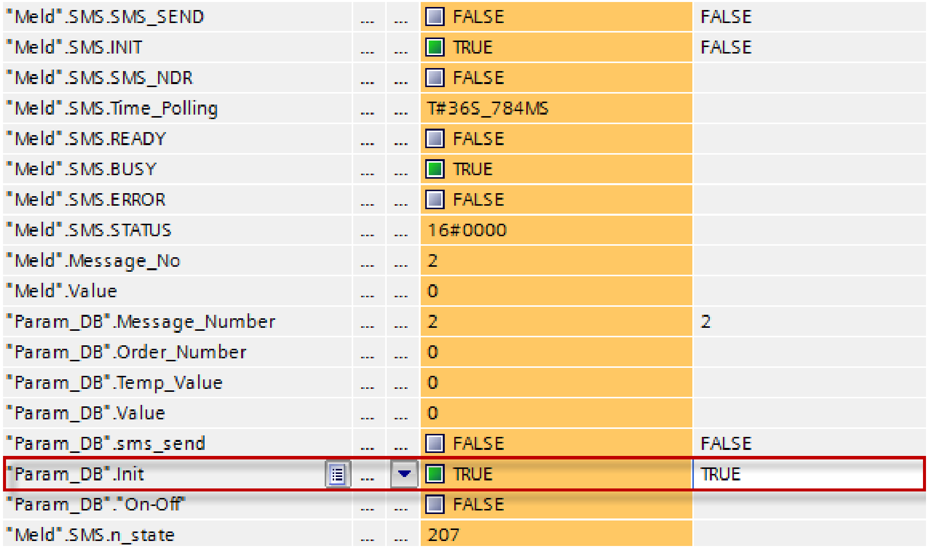
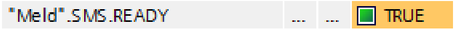
The prerequisites for generating and sending a new SMS message are as follows:

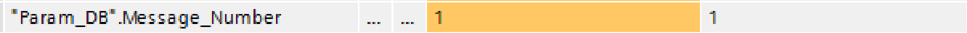
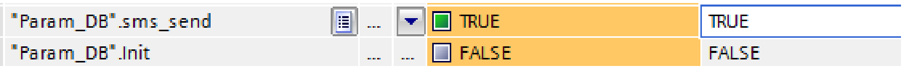
- A value was assigned to the "**Message_No**" variable.
- The **"sms_send"** flag is enabled.
- The modem was initialized.
- No other send operation is being actively processed (**"Meld".SMS.BUSY = true**).

6.1.1 Generating and sending an SMS message

The table below gives instructions for sending an SMS message.

Table 6-1

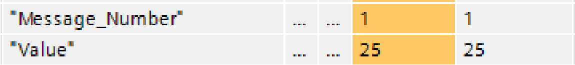
No.	Procedure
1.	Set the desired parameters as described in Table 5-3. Download the entire project as described in chapter 5.2.
2.	Open the "Watch_MELD" watch table and initialize the modem by a positive edge at **"Init"**.  "Meld".SMS.SMS_SEND ... ... □ FALSE FALSE "Meld".SMS.INIT ... ... 🟩 TRUE FALSE "Meld".SMS.SMS_NDR ... ... □ FALSE "Meld".SMS.Time_Polling ... ... T#36S_784MS "Meld".SMS.READY ... ... □ FALSE "Meld".SMS.BUSY ... ... 🟩 TRUE "Meld".SMS.ERROR ... ... □ FALSE "Meld".SMS.STATUS ... ... 16#0000 "Meld".Message_No ... ... 2 "Meld".Value ... ... 0 "Param_DB".Message_Number ... ... 2 2 "Param_DB".Order_Number ... ... 0 "Param_DB".Temp_Value ... ... 0 "Param_DB".Value ... ... 0 "Param_DB".sms_send ... ... □ FALSE FALSE "Param_DB".Init ... 🟩 TRUE TRUE "Param_DB"."On-Off" ... ... □ FALSE "Meld".SMS.n_state 207
3.	In the "Watch_MELD" variable table, check whether the **"Meld.SMS.READY"** variable is set. If this is not the case, enable **"Init"** and reinitialize the modem.  "Meld".SMS.READY ... ... 🟩 TRUE

No.	Procedure
4.	The **“Message_Number”** variable contains the message number that is necessary to generate an SMS message. Use the VAT to modify the value.  "Param_DB".Message_Number 1 1
5.	Enable **“sms_send”** to send the message.  "Param_DB".sms_send ... ■ TRUE TRUE "Param_DB".Init ... ... □ FALSE FALSE
6.	Once the SMS is message has been sent (without errors), reset the send trigger command (“sms_send” flag) since the trigger command reacts only to a positive edge.

6.1.2 Sending an SMS message with attached coefficient

The table below lists instructions for sending an SMS message with an attached coefficient.

Table 6-2

No.	Procedure
1.	Generate a new SMS message as described in steps 1-3 of table 6.1.
2.	In **“Value”**, enter any coefficient as an integer value and enable the value.  "Message_Number" ... ... 1 1 "Value" 25 25
3.	Send the new SMS message as described in steps 4 and 5 of table 6.1.

Note If you want to send an SMS message without an attached coefficient, enter the value “0” in “Value”. If you do not enter a value, the previous value will be applied.

6.2 Receiving an SMS message (scenario 3/scenario 4)

The prerequisites for receiving a new SMS message are as follows:

- The modem was initialized.
- The polling timer has elapsed.

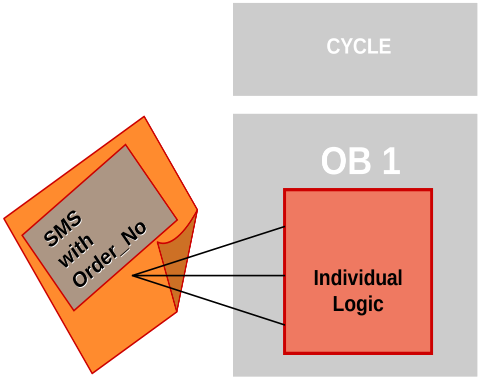
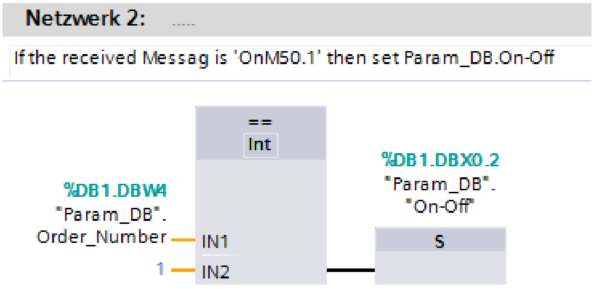
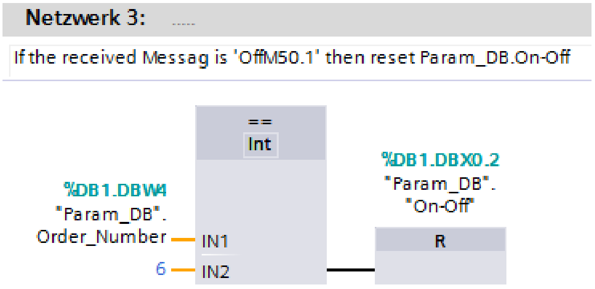
CAUTION **Do not send an SMS message containing the key word ‘ERROR’ or ‘OK’, since MD720-3 evaluates these words when a faulty behavior occurs during sending from MD720-3 to the communication module.**

NOTE For sending SMS to the modem it is recommended to use only alphabetic characters and figures. This is in order to avoid display and recognition errors of the STEP 7 application program.

6.2.1 Receiving and evaluating an SMS message

With the aid of the FB17 **FB “SMS_Meld”** function block, an S7-300/400/ET 200S station can receive and evaluate an order SMS.
The received SMS messages are polled every minute (default setting). You can change the timer as desired.
Once the timer has elapsed, all SMS messages are read out of the memory in the modem and evaluated.

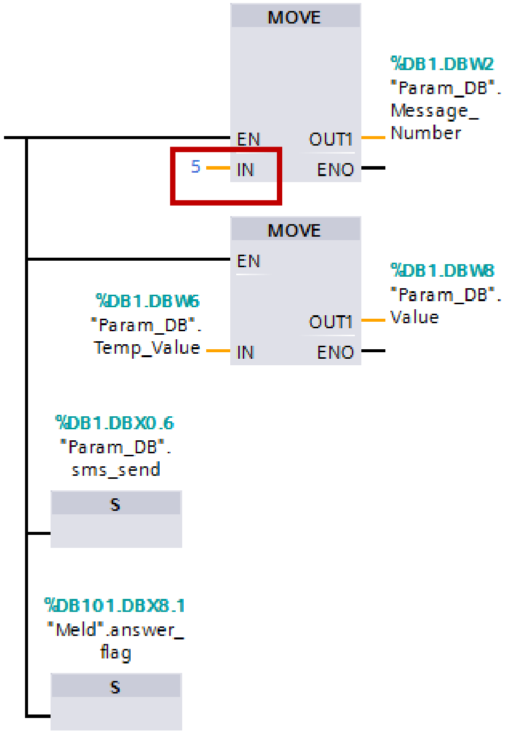
Table 6-3

No.	Action	Note
1.	Write an SMS message that includes one of the “keywords” of Figure 3-12 and send it to the MD720-3. Please ensure correct upper and lower case. The key word must be placed between question marks.	
2.	All SMS messages received by the modem are evaluated. If one of the “keywords” is detected in the receive text, the assigned Order_No (e.g., no. 1 for ‘OnM50.1’) is output on the respective parameter of **FB SMS_Meld** and can then be used for individual further processing.	
3.	An individual logic for the Order_No parameter may look as follows: Set and reset flag. • If the **Order_No** of the received message, is, for example, ‘1’ ‘OnM50.1’, the M50.1 flag will be set. • If the **Order_No** of the received message, is, for example, ‘6’ ‘OffM50.1’, the M50.1 flag will be reset.	Netzwerk 2: ..... If the received Messag is 'OnM50.1' then set Param_DB.On-Off == Int %DB1.DBW4 "Param_DB". Order_Number IN1 1 IN2 %DB1.DBX0.2 "Param_DB". "On-Off" S Netzwerk 3: ..... If the received Messag is 'OffM50.1' then reset Param_DB.On-Off == Int %DB1.DBW4 "Param_DB". Order_Number IN1 6 IN2 %DB1.DBX0.2 "Param_DB". "On-Off" R

6.2.2 Remote query of a value

In this scenario, the **FB “SMS_Meld”** function block and the **FC “SMS_MNG”** function are used to receive and evaluate a query SMS message and to send a feedback message with the desired value back to the user.

Table 6-4

No.	Action
1.	First open the **“Watch_MELD”** variable table and write the current temperature value to the “**Temp_Value”**.
2.	Open the FC50 **“SMS_MNG”** function and specify Message_No: 5.  MOVE %DB1.DBW2 "Param_DB". Message_ Number EN OUT1 5 IN ENO MOVE EN %DB1.DBW8 "Param_DB". Value %DB1.DBW6 "Param_DB". Temp_Value OUT1 IN ENO %DB1.DBX0.6 "Param_DB". sms_send S %DB101.DBX8.1 "Meld".answer_ flag S
3.	Call the **FC “SMS_MNG”** function in OB1. Save and load the organization block.
4.	Make sure that the controller is ready to receive or send an SMS message. See step 4 of table 6.1.
5.	Write an SMS with the keyword “?Temp1” and send it to the S7 station.
6.	Once the message is received, the receive text is evaluated. If the SMS is recognized as an command, the assigned Order_No (e.g., no. 7 for ‘?Temp_1’) will be output on the parameter of **FB “SMS_Meld”** and saved in the data block Param_DB. A feedback message with message text (Message_No: 5) and temperature value will be automatically sent back to the sender of the query.

Limitation

If the modem receives more than one SMS requesting data in a polling cycle, the PLC application program will only answer, at maximum, one SMS.

To answer more than one SMS after one cycle it is necessary to extend the programm with a buffer for received SMS, etc.

7 Further Notes

RS 232 sniffer

Communications processor (CP/CM) and MD720-3 are directly connected to one another via a serial cable (RS232 cable). Communication is full duplex, i.e. data can be simultaneously sent and received.

For this 1:1 communication, it is not possible to track the data traffic between the terminal units using a normal serial cable.

Especially for error analysis and data recording, it is desirable to take a note of the RS232 communication. Therefore, a special cable must be manufactured.

The RS232 cable is extended by two additional ports, with each port tapping one communication direction (send/receive).

RS 232 pin assignment

The following table shows the pin assignment for the RS232 interface:

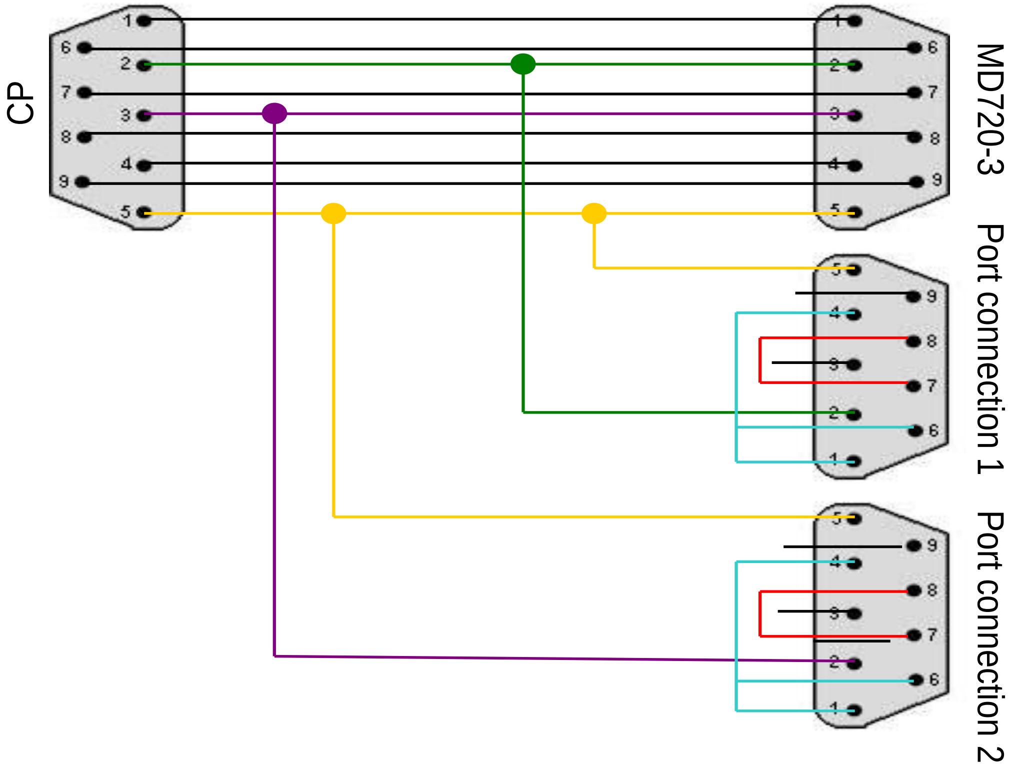
Table 7-1

Pin	Name	Meaning
1	DCD	(Data) Carrier Detect; MD720-3 signals to CP that it detects data on the line.
2	RxD	Receive Data; line for receive data
3	TxD	Transmit Data; line for send data
4	DTR	Data Terminal Ready; via this line, CP signals to MD720-3 that it is ready
5	Gnd	Signal ground
6	DSR	Dataset Ready: Via this line, MD720-3 signals to CP that it is ready
7	RTS	Request to Send; send request: MD720-3 can send data
8	CTS	Clear to Send; MD720-3 can receive data
9	RI	MD720-3 informs CP of an incoming call.

Connection diagram

The figure below shows the connection diagram for a full duplex RS232 sniffer. Port 1 taps the data of the receive line and port 2 taps the data of the send line.

Figure 7-1



If your PC has two serial interfaces, you can display the data traffic in both directions on a screen using a terminal program, e.g. HyperTerminal. A separate terminal must be opened for each data direction.

However, special RS232 sniffer software products are already available, which allow combining the data traffic of several ports on one screen. It is thus no longer necessary to switch between terminal windows.

8 Literature

Table 8-1

	Subject	Link
\1\	Reference to this document	https://support.industry.siemens.com/cs/ww/en/view/25545680
\2\	Siemens Industry Customer Support	https://support.automation.siemens.com
\3\	MD720-3 manual	https://support.industry.siemens.com/cs/ww/en/view/23117745
\4\	CP 340 manual	https://support.industry.siemens.com/cs/ww/en/view/1137332
\5\	First steps with the CP 340	https://support.industry.siemens.com/cs/ww/en/view/12108826
\6\	CP 341 manual	https://support.industry.siemens.com/cs/ww/en/view/1117397
\7\	First steps with the CP 341	https://support.industry.siemens.com/cs/ww/en/view/1188622
\8\	CP 441-2 manual	https://support.industry.siemens.com/cs/ww/en/view/1137419
\9\	First steps with CP 441-2	https://support.industry.siemens.com/cs/ww/en/view/1188835
\10\	SIMATIC ET 200S manual	https://support.industry.siemens.com/cs/ww/en/view/9260793
\11\	PROFINET S7-300 CPU manual	https://support.industry.siemens.com/cs/ww/en/view/48080216
\12\	WAN Access Methods	https://support.industry.siemens.com/cs/ww/en/view/26662448
\13\	STEP 7 V13 system manual	https://support.industry.siemens.com/cs/ww/en/view/77991795
\14\	CM PtP Configurations for Point-to-Point Connections	https://support.industry.siemens.com/cs/ww/en/view/59057093
\15\	SIMATIC S7-1500 CM PtP RS232 BA	https://support.industry.siemens.com/cs/ww/en/view/59057152
\16\	SIMATIC S7-1500 CM PtP RS232 HF	https://support.industry.siemens.com/cs/ww/en/view/59057160
\17\	CM PtP operation with PROFINET	https://support.industry.siemens.com/cs/ww/en/view/68075812
\18\	SIMATIC S7-1500, ET 200MP, ET 200SP communication	https://support.industry.siemens.com/cs/ww/en/view/59192925
\19\	SIMATIC ET 200SP CM Freeport/3964	https://support.industry.siemens.com/cs/ww/en/view/59061378

9 History

Table 9-1

Version	Date	Modifications
V1.0	11/2013	First version for STEP 7 V13
V1.1	05/2015	Screenshots in Table 4-4 and Table 4-6 have been changed