

**Advant Controller 31**  
Intelligent Decentralized  
Automation System

Example program for 907 AC 1131  
07 KT 9x with Siemens M20 / TC35 Terminal

SMS sending and receiving, data exchange via SMS

---

1	General information about the Short Message Service (SMS) .....	2
1.1	SMS with radio data modem SIEMENS M20 or TC35 Terminal.....	2
1.2	Technical support for the radio data modem SIEMENS M20/TC35 Terminal .....	2
2	Functional principle for series 90 (KT95...98).....	4
3	Structure .....	5
3.1	Connecting cables.....	5
4	Configuration of the Siemens M20/TC35 for SMS handling.....	6
4.1	Necessary pre-settings using AT commands.....	6
4.1.1	Determining the available SMS memory.....	6
4.1.2	Checking the receive quality in the radio network .....	7
4.2	AT commands for sending an SMS message.....	8
4.2.1	Sending SMS .....	8
4.2.2	Recognizing incoming SMS .....	8
4.2.3	Reading incoming SMS.....	8
4.2.4	Deleting a read SMS .....	9
4.2.5	Reading an empty memory address .....	9
5	The 907 AC 1131 library SMS_S90_V411.LIB .....	10
5.1	The function block SMS_M20_INIT.....	11
5.2	The function block SMS_M20_SEND .....	13
5.3	The function block SMS_M20_RECV .....	16
5.4	The function block SMS_M20_EXTR.....	20
5.5	The DATA_SPLIT function block – example of a user-specific FB.....	22

# 1 General information about the Short Message Service (SMS)

Short Message Service is a special service offered by various radio network providers. Using this service short messages with a maximum length of 160 characters can be transmitted inside a radio data network.

Precondition is that the sender and the receiver are able to use this service (e.g. mobile phones and radio data modems).

The short messages are transmitted to the service center of the radio network provider together with the receiver's number from where they are forwarded to the receiver.

## 1.1 SMS with radio data modem SIEMENS M20 or TC35 Terminal

The short messages are sent and received by the radio data modems SIEMENS M20/TC35 Terminal. The card contracts were made with D1 network provider DeTeMobil Deutsche Telekom MobilNet GmbH (other providers are possible).

The DeTeMobil service center number is **+49 171 076 000**.

## 1.2 Technical support for the radio data modem SIEMENS M20/TC35 Terminal

### List of distributors and partners for Cellular Engines.

Country	Company	Phone numbers and Web contact
---------	---------	-------------------------------

Germany	<i>KomSa</i> Kommunikation GmbH Sachsen, Untere Hauptstr. 79, D-09232 Hartmannsdorf Germany	<i>Phone:</i> +49 3722 - 713 210 <i>Fax:</i> +49 3722 - 713 305 <i>Website:</i> <a href="http://www.komsa.de/">http://www.komsa.de/</a>
	<i>MÄRTENS Communication GmbH &amp; Co. KG</i> Kabelkamp 2, D-30179 Hannover Germany	<i>Phone:</i> +49 511 - 67495 - 0 <i>Fax:</i> +49 511 - 63 63 41 <i>Website:</i> <a href="http://www.maertens-communication.com/">http://www.maertens-communication.com/</a>
	<i>GAP AG</i> Kolpingring 18 a, D-82041 Oberhaching Germany	<i>Phone:</i> +49 (89) 66 66 95-0 <i>Phone:</i> 089/613 04-229 <i>Fax:</i> +49 (89) 66 66 95-55 <i>Fax:</i> 089/614 04-353 <i>Hotline:</i> +49 (89) 66 66 95-66 <i>E-Mail:</i> <a href="mailto:info@gapag.de">info@gapag.de</a> <i>Website:</i> <a href="http://www.gapag.de/">http://www.gapag.de/</a>
United Kingdom	<i>Hugh Symons Mobile Data</i> Alder Hills Park, 16 Alder Hills, Poole, Dorset BH12 4AR Great Britain	<i>Phone:</i> +44 (0) 1202 718388 <i>Fax:</i> +44 (0) 1202 712017 <i>Website:</i> <a href="http://mobiledata.hughsym.co.uk/">http://mobiledata.hughsym.co.uk/</a>
	<i>Telecom Design Communications Ltd.</i> Stroudley Road, Basingstoke, Hampshire, RG24 8FN Great Britain	<i>Phone:</i> +44 (0) 1256 332800 <i>Fax:</i> +44 (0) 1256 332810 <i>Website:</i> <a href="http://www.tdc.co.uk/">http://www.tdc.co.uk/</a>

**Australia** *Oztrak Group Pty Ltd* *Phone: +613 5330 1184*  
 Greenhill Enterprises Centre, *Fax: +613 5330 1188*  
 University Drive, *E-Mail: [marketing@oztrak.com](mailto:marketing@oztrak.com)*  
 Mt. Helen, *Website: <http://www.oztrak.com>*  
 PO Box 2182 Ballarat,  
 Victoria,  
 3354  
 Australia

**Netherlands** *Allied Data Technologies bv* *Phone: +31 181-611 522*  
 Pascalweg 1, *Fax: +31 181-614 840*  
 Spijkenisse, *Website: <http://www.allieddata.nl/>*  
 Pobox 788,  
 3200 As Spijkenisse  
 Netherlands  
*Telpro plus BV* *Phone: +31 20-4878760*  
 Aletta Jacobsiaan 9, *Fax: +31 20-6170025*  
 106 BP Amsterdam  
 Netherlands

**Belgium** *Telpost* *Phone: +32 16-382716*  
 Geldenaaksebaan 335, *Fax: +32 16-382717*  
 3001 Leuven, *Website: <http://www.telindus.be/telpost/>*  
 Belgium

**Switzerland** *Mobile Solutions AG* *Phone: +41 1 802 44 11*  
 Im Schörli 1, *Fax: +41 1 802 44 99*  
 Ch-8600 Dübendorf, *Email: [engine@mobilesolutions.ch/](mailto:engine@mobilesolutions.ch/)*  
 Switzerland *Website: <http://www.mobilesolutions.ch/>*

**Finland** *Klinkmann* *Phone: +35 8-9- 540 4940*  
 PO Box 38 (Fonseenintie 3), *Fax: +35 8-9- 513 541*  
 Fin-00371 Helsinki, *Website: <http://www.klinkmann.com/>*  
 Finland

## 2 Functional principle for series 90 (KT95...98)

*for sending and receiving SMS and exchanging data via SMS using 907 AC 1131.*

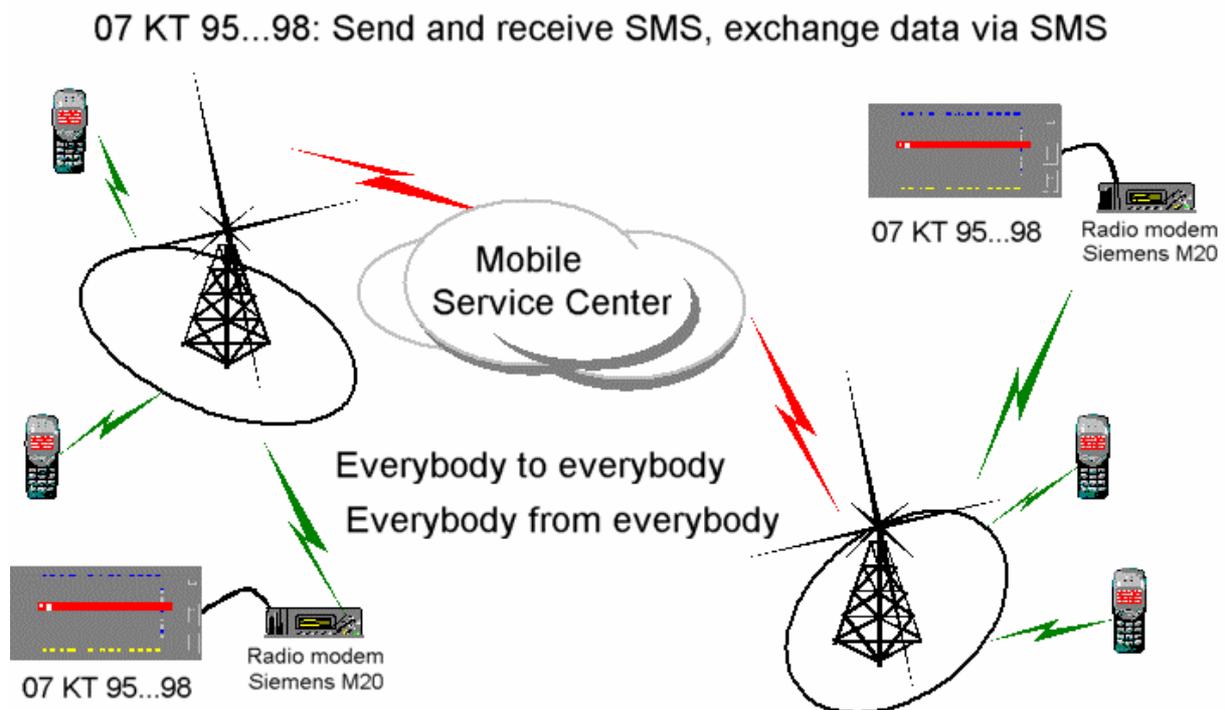
The function blocks (executable under the programming and test software 907 AC 1131 according to IEC1131-3) enable the sending and receiving of SMS messages (Short Message Service, max. 160 characters per message) using a PLC of the type 07KT95...98 to mobile phones, from mobile phones, to corresponding pagers, to other PLCs or from other PLCs.

The required firmware blocks are implemented in the PLCs from version 4.05 on.

The possibility of exchanging SMS between two or more PLCs and/or mobile phones enable the user to exchange data, to automatically output system states and error messages of an installation, to transmit correction values to a basic unit etc.

This is a simple and cheap alternative to all kinds of dial-line connections.

With **one** SMS, for instance, it is possible to transfer 22 six-digit variables (incl. negative sign) or 17 eight-digit variables (incl. negative sign).



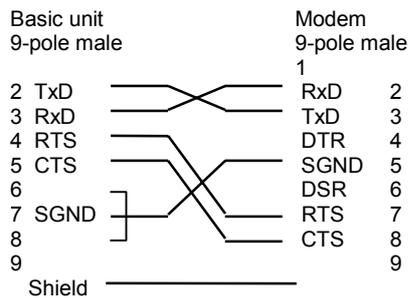
### 3 Structure

The GSM modem used to send and receive short messages has to be connected to one of the serial interfaces COM1/COM2 of a basic unit 07 KT 95...98. The receiver may be a mobile phone, a pager or a radio data modem.

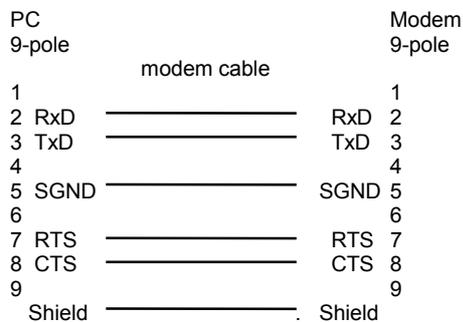
#### 3.1 Connecting cables

The connecting cables between the devices have to be wired as follows:

- **Basic unit 07 KT 95...98 <--> modem (cable 07 SK 92 R1)**



**PC <--> modem**



## 4 Configuration of the Siemens M20/TC35 for SMS handling

### 4.1 Necessary pre-settings using AT commands

ati	Inquiry of the version - has to be 3.00 or higher! If this is not the case, request an update from Siemens service.
at&f	Set modem to factory settings.
atv0	Set format mode for the result code to numerical.
ate0	Turn off the command echo.
at+cmgf? at+cmgf=1	Inquiry of format:       0: PDU mode;       1: TEXT mode. Set format to TEXT mode.
M20 only: at+ipr=19200	Set baud rate to 19200 baud (factory setting).
at+csca? at+csca="+491710760000" at+csca="+491722270333"	Inquiry of the current service center number. Set number for D1 service center. Set number for D2 service center.
ats0? ats0=1	Inquiry of the setting for automatic call acceptance. Automatic call acceptance after 1* ringing.
at+cpin? at+cpin=1234	Inquiry of PIN code. Enter PIN code.
TC35 only: at+cnmi=1,1	Enable "new" message on reception of a new SMS (+CMTI:). Only accepted if a registered SIM card is inserted.
at&w	Save user profile in modem.

#### 4.1.1 Determining the available SMS memory

The amount of SMS memory available in the M20/T35 Terminal strongly depends on the card provider. However, this value is very important for the setting of SMS\_M20\_RECV (MAX\_STORE). Therefore you should request this value from the provider.

The necessary information (card number, network provider) can be found on the card itself or you can poll them using AT commands:

at+cimi                               Number of the network provider; e.g. first numbers 26201 = D1, 26202 = D2.

at^scid                               Card number; you can also find it on the card:  
89490200000127712979

As an alternative, you can also determine this value empirically using the command at+cmgr=xx (refer to section 4.2) and working forward from xx = 01 up to the last value which is echoed with 'ok' or '0'. If the maximum number of available memory locations is exceeded, an error is displayed indicating that this value for xx is inadmissible. The last value which was echoed with 'ok' or '0' must be entered for MAX\_STORE of the SMS\_M20\_RECV.

The D1 SIM cards used in this test provide a maximum of 14 memory locations for short messages in the modem.

## 4.1.2 Checking the receive quality in the radio network

The receive quality of the modem must meet a certain value to guarantee an error-free SMS transmission. This can be tested as follows:

At^moni=5      Now the modem is polled every 5 seconds in order to obtain the current values. Thus, the modem replies the following line every 5 seconds:

```
Chann rs dBm PLMN LAI cell NCC BCC PWR Rxlev C1 I chann TS timAdv PWR ...  
      -67                               -102
```

The important values are dBm and Rxlev. Rxlev is the minimum permitted reception value. If the dBm value is smaller than Rxlev, the modem logs out (no reception). The lower the current dBm value, i.e. the closer the dBm value approximates the minimum reception value (log-off value), the worse is the receive quality. In the test laboratory the dBm value was continuously in the range between -67 and -69 which provided good results.

## 4.2 AT commands for sending an SMS message

In the function blocks SMS\_M20\_SEND/SMS\_TC35\_SEND or SMS\_M20\_RECV/ SMS\_TC35\_RECV the following commands are used to exchange SMS messages.

(standard type face = input to the modem, **bold type face** = modem response)

### 4.2.1 Sending SMS

at+cmgs="01726225107"<CR>

**0d0a**

>\_good morning 12324 bye bye<CTRL Z>

**+CMGS: \_68 0d0a 0 0d**

**0d0a+CMGS: \_68 0d0a 0 0d**

Enter the phone number (do not forget the quotation marks!) and finish with CR.

Modem answers with crlf>blank

the >\_ appears automatically; enter the text and finish with <CTRL Z> which corresponds to \$1A or -> -.

Response of the **M20** modem:

+cmgs:\_xxSMS crlf0cr

which means that the 68<sup>th</sup> SMS has been sent.

Response of the **TC35** modem:

crlf+cmgs:\_xxSMS crlf0cr

which means that the 68<sup>th</sup> SMS has been sent.

### 4.2.2 Recognizing incoming SMS

**+CMTI: \_"SM",\_7 0d0a**

**0d0a+CMTI: \_"SM",\_7 0d0a**

**M20:** SMS was received and stored to memory address 7 (possible: 1...14).

**TC35:** SMS was received and stored to memory address 7 (possible: 1...14).

### 4.2.3 Reading incoming SMS

at+cmgr=7<CR>

Read SMS from memory address 7.

**+CMGR: \_"REC\_UNREAD","+491726225107",,"00/04/22,17:11:33:+04"0d0a**

**bla bla bla 12324 bla bla bla 0d0a**

**0 0d**

**M20** modem answers with the entire SMS string incl. TYPE, SENDER, TIMESTAMP and TEXT, completed with crlf0cr.

Possible TYPES: REC READ, REC UNREAD, STO UNSENT, STO SENT, ALL

**0d0a+CMGR: \_"REC\_UNREAD","+491726225107",,"00/04/22,17:11:33:+04"0d0a**

**bla bla bla 12324 bla bla bla 0d0a**

**0 0d**

**TC35** modem answers with the entire SMS string incl. TYPE, SENDER, TIMESTAMP and TEXT, completed with crlf0cr.

Possible TYPES: REC READ, REC UNREAD, STO UNSENT, STO SENT, ALL

#### 4.2.4 Deleting a read SMS

at+cmgd=7<CR>

0 0d

Delete SMS from memory address 7.  
A maximum of 14 SMS messages can be received until the memory is full. Therefore delete the read SMS messages!  
**M20/TC35** modem answers with 0cr.

#### 4.2.5 Reading an empty memory address

at+cmgr=7<CR>

0 0d

0d0a+CMGR: \_0,,0 0d0a 0 0d

Read memory address 7.

**M20** modem answers with 0cr.

**TC35** modem answers with  
crlf+CMGR: \_0,,0crlf0cr.

#### Only for information:

at^SMGR

Reads SMS messages without setting REC\_UNREAD.

at+CMGL="ALL"

Displays all stored SMS messages.

## 5 The 907 AC 1131 library SMS\_S90\_V411.LIB

The library SMS\_S90\_V411.LIB contains function blocks for initializing an interface (SMS\_M20\_INIT or SMS\_TC35\_INIT) as well as for sending (SMS\_M20\_SEND or SMS\_TC35\_SEND), receiving (SMS\_M20\_RECV or SMS\_TC35\_RECV) and extracting SMS messages (SMS\_M20\_EXTRACT or SMS\_TC35\_EXTRACT).

The blocks were developed and configured exclusively for the GSM modems Siemens M20 Terminal or Siemens TC35 Terminal.

The required firmware blocks (com\_s90\_v411.lib) are implemented in the PLCs from version 4.05 onwards.

**Caution: Not all characters of the simple ASCII character set (0...127dec) can be sent or received!**

The following characters can be used:

- Numbers from 0 to 9
- Small letters from a to z
- Capital letters from A to Z
- Standard characters ! " # % & ( ) \* + - . / , ; < = > ?

The following characters are not allowed:

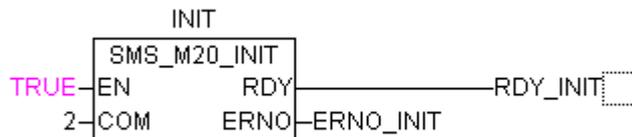
- Umlaut characters (ü, ö, ä, Ü, Ö, Ä, ...) and ß
- Special characters, such as \$, ` ´ @ [ \ ] ^ \_ { | } ~ €

The usage of characters which are not allowed inside the string causes sending or receiving errors!

In the following descriptions only M20 function blocks are used as an example. The configuration and the handling of the TC35 function blocks is identical.

## 5.1 The function block SMS\_M20\_INIT

*INITIALIZE COM1 or COM2 for communication with the GSM modem Siemens M20.*



Parameter	Type	Description
SMS_M20_INIT	Instance	Instance name
EN	BOOL	Enabling of the block processing
COM	INT	Interface number (COM1 or COM2)
RDY	BOOL	Readiness for operation of the selected interface
ERNO	INT	Error number

### Block type      Function block with historical values

The function block SMS\_M20\_INIT serves for the initialization of a serial interface for the usage of the function blocks SMS\_M20\_SEND and SMS\_M20\_RECV.

The function block is internally based on the function block <INIT> which is part of the 907 AC 1131 library COM\_S90\_V41.LIB.

A detailed description of this function block can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

On enabling, this function block initializes the selected interface COM1 or COM2 with 19200bd,8,1,N without direction control, without carrier leading/lagging and with end of telegram via break (mode 4) for establishing the connection to the GSM modem Siemens M20.

### EN      BOOL

If input EN is set to TRUE, the FB initializes the serial interface specified at the COM input.

EN has to remain TRUE until RDY indicates the end of the initialization process.

If EN = FALSE, RDY becomes FALSE and output ERNO becomes 0. This means that the function block is reset.

### COM      INT

At the COM input the number of the serial interface is specified.

COM = 1      COM1

COM = 2      COM2

### RDY      BOOL

RDY = TRUE indicates the end of the function block processing after input EN became TRUE.

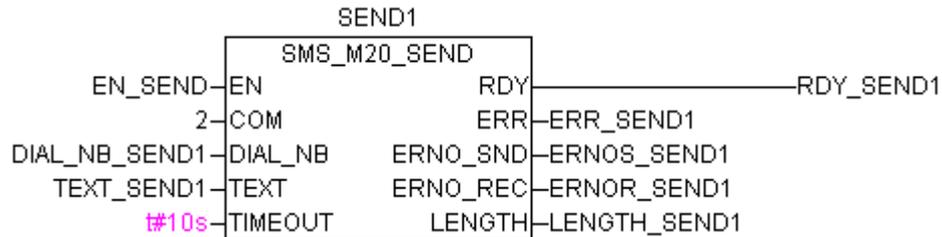
、 **ERNO**                    **INT**

Error code of the contained INIT function.

A detailed description of the INIT's ERNO can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

## 5.2 The function block SMS\_M20\_SEND

### SEND SMS



Parameter	Type	Description
SMS_M20_SEND	Instance	Instance name
EN	BOOL	Enabling of the block processing
COM	INT	Interface number (COM1 or COM2)
DIAL_NB	STRING(15)	Phone number of the SMS recipient
TEXT	STRING(160)	SMS text (160 characters max.)
TIMEOUT	TIME	Transmission time monitoring, approx. 10 seconds
RDY	BOOL	Block was processed. ERR=FALSE -> SMS was transmitted successfully, ERR=TRUE -> Processing failed
ERR	BOOL	Error occurred during processing
ERNO_SND	INT	SEND error code of SMS_M20_SEND
ERNO_REC	INT	RECV error code of SMS_M20_SEND
LENGTH	INT	String length of SMS text (160 characters max.!)>

#### Block type Function block with historical values

The function block SMS\_M20\_SEND sends an SMS message via the selected and previously initialized interface (SMS\_M20\_INIT) using the Siemens M20 GSM modem. The function block is internally based on the function blocks SEND and RECV which are part of the 907 AC 1131 library COM\_S90\_V41.LIB. A detailed description of these function blocks can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

On enabling EN = TRUE the function block sends the string with a maximum length of 160 characters which is provided at TEXT to the phone number specified at DIAL\_NB. For checking the number of characters the string length of the valid TEXT string is applied at output LENGTH.

**The input TEXT as well as the input DIAL\_NB must contain at least one character (LENGTH > 0). Otherwise the function block does not work and the error code 112 is output at both ERNO outputs (ERR+RDY = TRUE).**

If RDY=TRUE and ERR=FALSE, the sending process was performed successfully and the ERROR codes of the contained SEND or RECV are 0. If RDY=TRUE and ERR=TRUE, an error occurred during the sending process and the ERROR codes of the contained SEND or RECV show detailed error messages. In case of a TIMEOUT both ERNO outputs are set to the value 111. In case of missing input parameters (TEXT or DIAL\_NB) the outputs are set to 112. For sending messages longer than 160 characters or for sending the SMS to several recipients, several function blocks SMS\_M20\_SEND can be connected in series. However, a simultaneous processing of the function blocks is not possible! Since the RECV is contained in the function blocks, SMS\_M20\_SEND and SMS\_M20\_RECV have to be mutually interlocked. This means, that SMS\_M20\_RECV can normally always have EN=TRUE, but for sending an SMS this input must be set to FALSE during the sending time of SMS\_M20\_SEND.

Furthermore, a sending procedure should be only possible if SMS\_M20\_RECV indicates emptiness (NOT\_EMPTY = FALSE).

‣ **EN**                    **BOOL**

If input EN is set to TRUE (no edges, EN must not return to FALSE until RDY is TRUE), the FB tries to send the SMS specified on its inputs.

For this purpose, the corresponding COM interface has to be previously initialized with SMS\_M20\_INIT.

EN has to remain TRUE until RDY indicates the end of the sending process.

In case of EN=FALSE, RDY and ERR become FALSE and both ERNO\_SND and ERNO\_REC become 0. That means that the function block is reset.

‣ **COM**                    **INT**

At the COM input the number of the serial interface is specified.

COM = 1            COM1

COM = 2            COM2

‣ **DIAL\_NB**                **STRING(15)**

At input DIAL\_NB, the telephone number of the SMS recipient is applied directly as a STRING. The telephone number can have a max. length of 15 characters, e.g.: '+0491726225107'.

Input DIAL\_NB must contain *at least 1 character*, otherwise the function block does not work (error code 112 on both ERNO\_SND and ERNO\_REC).

‣ **TEXT**                    **STRING(160)**

At input TEXT the content of the SMS is specified. Please note that SMS messages are limited to a max. length of 160 characters.

The current length of the valid TEXT string is indicated at output LENGTH.

Input TEXT must contain *at least 1 character*, otherwise the function block does not work (error code 112 on both ERNO\_SND and ERNO\_REC).

‣ **TIMEOUT**                **TIME**

If the output RDY does not respond within the time specified at input TIMEOUT after EN=TRUE, RDY and ERR become TRUE and both ERNO\_SND and ERNO\_REC indicate error code 111. In this case a TIMEOUT has occurred.

The minimum timeout is 5 seconds because this is the time, the modem needs to return the sending confirmation to the FB. Possibly this time can be longer, e.g. if the network quality is poor, etc.

‣ **RDY**                    **BOOL**

RDY = TRUE signalizes the end of the function block processing after EN was set to TRUE. RDY has always to be considered together with the ERR output.

、 **ERR**      **BOOL**

ERR = FALSE and RDY = TRUE means that the processing was completed successfully. The SMS has been sent.

ERR = TRUE and RDY = TRUE means that an error has occurred during processing. The SMS has not been sent or has been sent incompletely.

According to the contained function blocks SEND and RECV the error code is stored in the associated ERNO (ERNO\_SND (error while sending) or ERNO\_REC (error while receiving)). In case of a timeout both ERNO\_SND and ERNO\_REC are overwritten with the value 111.

、 **ERNO\_SND**      **INT**

Error code of the contained SEND function.

A detailed description of the SEND's ERNO can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

An exception is the timeout. In this case ERNO\_SND is overwritten with the value 111. In case of a missing input parameter (TEXT or DIAL\_NB) 112 is output.

、 **ERNO\_REC**      **INT**

Error code of the contained RECV function.

A detailed description of the RECV's ERNO can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

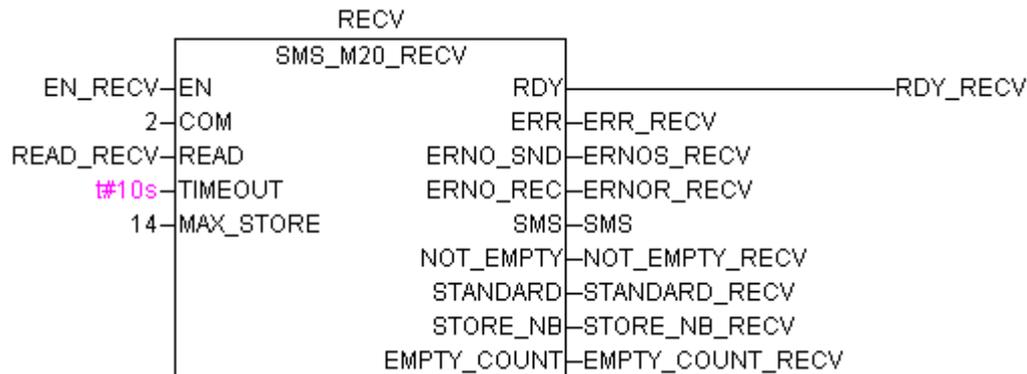
An exception is the timeout. In this case ERNO\_REC is overwritten with the value 111. In case of a missing input parameter (TEXT or DIAL\_NB) 112 is output.

、 **LENGTH**      **INT**

Displays the current length of the valid TEXT string.

### 5.3 The function block SMS\_M20\_RECV

#### READ / DELETE SMS from the modem memory



Parameter	Type	Description
SMS_M20_RECV	Instance	Instance name
EN	BOOL	Enabling of the block processing. Searching for an SMS in the modem memory and reading the first SMS found. This SMS is internally stored and can be output to SMS using READ.
COM	INT	Interface number (COM1 or COM2)
READ	BOOL	Read command and output to SMS
TIMEOUT	TIME	Monitoring of the communication period between modem and FB, approx. 20 seconds
MAX_STORE	INT:=14	Maximum possible number of memory locations for SMSs in the modem (14 memory locations possible for D1)
RDY	BOOL	PULS; the SMS was read and applied to output SMS. RDY is active for only 1 cycle! ERR=FALSE -> SMS was read successfully, ERR=TRUE -> incorrect processing.
ERR	BOOL	Error occurred during processing
ERNO_SND	INT	SEND error code of SMS_M20_RECV
ERNO_REC	INT	RCV error code of SMS_M20_RECV
SMS	STRING(256)	SMS incl. time stamp, sender identifier, etc.
NOT_EMPTY	BOOL	TRUE if an SMS was detected in the memory, FALSE if the SMS memory is empty
STORE_NB	INT	Memory location number, where the internally stored SMS was found in the modem (reading with READ). A maximum of 14 SMSs can be stored in the modem.
EMPTY_COUNT	INT	Counter for empty memory locations. Starts counting at the first free memory location x (no SMS on memory location x (x=1...14)). EMPTY is set to TRUE if the counter outputs 14. If a new SMS was detected or an occupied memory location was found before, EMPTY_COUNT is deleted.

#### Block type Function block with historical values

The function block SMS\_M20\_RECV serves for receiving SMS messages via the selected and previously initialized interface (SMS\_M20\_INIT) using the Siemens M20 GSM modem. The function block is internally based on the function blocks SEND and RCV which are part of the 907 AC 1131 library COM\_S90\_V41.LIB.

A detailed description of these function blocks can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

In principle, the SMS\_M20\_RECV function block works like a FIFO stack memory.

On enabling with EN = TRUE, the function block automatically starts polling the memory location 1 in the modem.

The activity of the function block can be traced at output STORE\_NB.

If memory location 1 is empty: - The FB polls the next memory location (2).  
If it is not empty: - The SMS is read from memory location 1 and stored in the FB.  
Then NOT\_EMPTY = TRUE indicates that there is an SMS which can be polled.

If a read request (READ = TRUE) is applied to the function block, it forwards the SMS string to its SMS output and then deletes the modem memory location which contains the SMS and sets RDY to TRUE for one cycle.

(It is useful to reset READ with RDY = TRUE.

If READ permanently remains TRUE, the function block immediately forwards every message found in the modem to the SMS output and overwrites the previous one).

Then the next memory location is polled. This procedure is repeated as long as the function block has found empty memory locations 14 times in succession and no new SMS was detected during this. The EMPTY\_COUNT output indicates the number of empty memory locations found in succession.

When EMPTY\_COUNT has reached 14, the function block sets the output NOT\_EMPTY = FALSE.

The outputs STORE\_NB and EMPTY\_COUNT are set to 0.

The function block does not start its searching procedure again until a new SMS is detected during EN=TRUE or until EN is again triggered with a FALSE/TRUE edge.

If RDY=TRUE and ERR=FALSE, the procedure of output and deletion was successfully completed and the ERROR codes of the contained SEND or RECV are 0.

If RDY=TRUE and ERR=TRUE, an error occurred during the output and deletion procedure and the ERROR codes of the contained SEND or RECV show detailed error messages.

Only 1 SMS\_M20\_RECV is allowed in a program. Since RECV is contained in the function block, SMS\_M20\_SEND and SMS\_M20\_RECV have to be mutually interlocked.

This means, that SMS\_M20\_RECV can normally always have EN=TRUE, but for sending an SMS the input EN must be set to FALSE during the sending time of SMS\_M20\_SEND.

#### EN BOOL

If EN is set to TRUE, the block processing is enabled. The block starts searching for SMSs in the modem memory and reading the first SMS found.

This SMS is internally stored. The memory location number is displayed at output STORE\_NB. Using READ the SMS can now be applied to the SMS output.

For this purpose, the corresponding COM interface has to be previously initialized using SMS\_M20\_INIT.

EN=FALSE resets the function block.

This means that the outputs RDY, ERR and NOT\_EMPTY are FALSE and that the ERNO outputs and the counters become "0".

#### COM INT

At the COM input the number of the serial interface is specified.

COM = 1 COM1

COM = 2 COM2

#### READ BOOL

At the READ input, the read command has to be applied as a pulse (best until RDY has become TRUE).

If READ is permanently TRUE, incoming SMS are output immediately.

READ can be carried out after the function block has been started with EN = TRUE and an SMS was read and stored in the FB and signaled with EMPTY = FALSE.

By setting READ = TRUE the function block forwards the read message (the modem memory location number of which is indicated at output STORE\_NB) to the SMS output and then deletes the modem's memory location in order to free space for new messages. RDY = TRUE indicates that the READ procedure has been completed.

\ **TIMEOUT            TIME**

Monitors the time period during which the function block is busy with the reception of SMSs from and the deletion of SMSs in the modem memory.  
If the pre-defined time period is exceeded, RDY and ERR become TRUE and the error code 111 is output at ERNO -> A timeout has occurred.  
The minimum timeout period is 15 seconds.

\ **MAX\_STORE        INT:=4**

Number of SMS memory locations to be determined (see also 4.1.1), which are provided by the modem. Since the value strongly depends on the used card, no constant can be used in this case. The pre-defined value is 14 (D1-SIM card).

\ **RDY            BOOL**

RDY = TRUE follows to the execution of the READ command (message forwarded to output SMS and deleted from the modem memory). RDY has a length of only one cycle. (It is useful to reset READ with RDY=TRUE.

If READ remains permanently TRUE, the function block immediately forwards every message found in the modem to the SMS output and overwrites the previous one).

RDY has always to be considered together with ERR.

\ **ERR            BOOL**

ERR=FALSE and RDY=TRUE means that the processing was successful. If ERR=TRUE and RDY=TRUE, an error occurred during the processing.

According to the contained function blocks SEND and RECV the error code is stored in the associated ERNO (ERNO\_SND (error while sending) or ERNO\_REC (error while receiving)).

\ **ERNO\_SND        INT**

Error code of the contained SEND function.

A detailed description of the SEND's ERNO can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

\ **ERNO\_REC        INT**

Error code of the contained RECV function.

A detailed description of the RECV's ERNO can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

\ **SMS            STRING**

Complete SMS string consisting of time stamp, sender identifier, text etc.

˘ **NOT\_EMPTY**      **BOOL**

NOT\_EMPTY = TRUE, if an SMS was read from the modem memory and stored in the function block ready for polling (using READ). NOT\_EMPTY = FALSE, if the modem memory does not contain an SMS message.

˘ **STORE\_NB**      **INT**

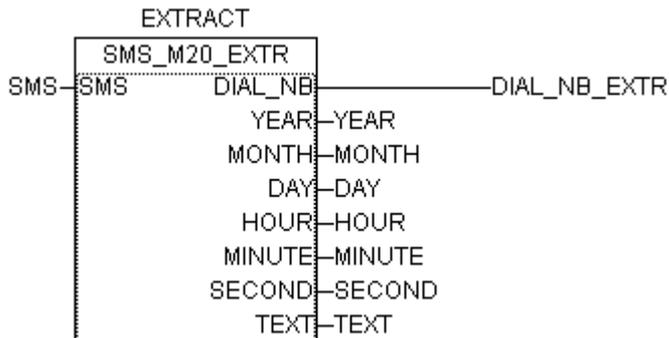
STORE\_NB displays (only for information) the modem memory location number which is currently accessed or where the internally stored SMS (readable by READ) was found in the modem. A maximum of 14 SMSs can be stored in the modem.

˘ **EMPTY\_COUNT**      **INT**

EMPTY\_COUNT counts (only for information) the empty memory locations in the modem. It starts counting at the first free memory location x (no SMS on modem memory location x (x=1...14)). EMPTY is set to TRUE if the counter outputs 14. If a new SMS was detected or an occupied memory location was found before, EMPTY\_COUNT is deleted.

## 5.4 The function block SMS\_M20\_EXTR

*Extract SMSs read with SMS\_M20\_RECV*



Parameter	Type	Description
SMS_M20_EXTR	Instance	Instance name
SMS	STRING(256)	SMS incl. time stamp, sender identifier, etc. from SMS_M20_RECV
DIAL_NB	STRING(20)	Phone number of the SMS sender
YEAR	INT	Year (2 digits)
MONTH	INT	Month (2 digits)
DAY	INT	Day (2 digits)
HOUR	INT	Hour (2 digits)
MINUTE	INT	Minute (2 digits)
SECOND	INT	Second (2 digits)
TEXT	STRING(170)	SMS text (160 characters max.) incl. termination character

**Block type**                      **Function block without historical values**

The SMS\_M20\_EXTR function block extracts an SMS message received by means of SMS\_M20\_RECV.

The function block is internally based on the DATAPOS function which is part of the 907 AC 1131 library COM\_S90\_V41.LIB.

A detailed description of this function can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

An enabling of the function block does not exist. The function block immediately splits an SMS received by SMS\_M20\_RECV into its components SENDER NUMBER, TIME STAMP and SMS TEXT.

Here, the following restrictions have to be considered:

The **SMS\_M20\_RECV** function block is generally able to read every SMS from and to delete it in the modem memory. However, since the SMSs can be of different TYPES, e.g. REC UNREAD or REC READ (the latter, if the message has already been read but not deleted), problems could arise while further processing the SMS by means of the **SMS\_M20\_EXTR** function block. This function block is only intended for the correct extraction of REC UNREAD messages (i.e. unread messages). Otherwise the data areas for the sender's number, the time stamp and the text are shifted due to the different string length, which makes them unusable!

Therefore it is better to completely delete the modem using the terminal and the command "at+cmgd=xx<CR>" (xx = 01...14) before taking it into operation. The same result can be obtained with

a complete run of the **SMS\_M20\_RECV** until the modem is empty. The data read during this procedure should be dismissed.

Furthermore, the **SMS\_M20\_EXTR** function block is only intended for SMSs originating from mobile phones or radio data modems, because their sender numbers **always** consists of 13 characters. In contrast to this, the sender number lengths of SMSs sent via Internet providers (Lycos, Debitel, etc.) are completely different. Such SMSs will cause the same negative effect as the different TYPE identifiers (mentioned above).

For such cases, special function blocks should be created. The adaptation of function blocks is simple.

˘ **SMS**                    **STRING**

Complete SMS string from the SMS output of SMS\_M20\_RECV, consisting of time stamp, sender identifier, text, etc.

˘ **DIAL\_NB**                **STRING**

Contains the extracted phone number of the sender.

˘ **YEAR**                    **INT**

Sending time stamp: YEAR (2 digits, i.e. 00 or 01 or 11, etc).

˘ **MONTH**                  **INT**

Sending time stamp: MONTH.

˘ **DAY**                     **INT**

Sending time stamp: DAY.

˘ **HOUR**                    **INT**

Sending time stamp: HOUR.

˘ **MINUTE**                 **INT**

Sending time stamp: MINUTE.

˘ **SECOND**                 **INT**

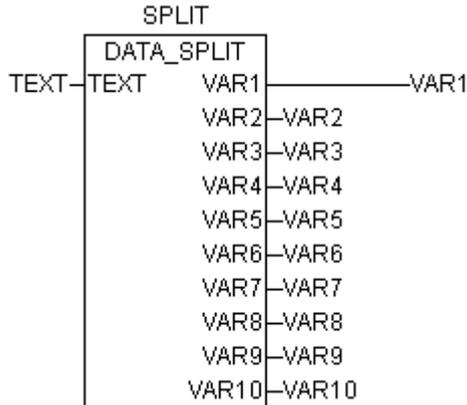
Sending time stamp: SECOND.

˘ **TEXT**                    **STRING**

SMS text (160 characters max.) incl. termination character.

## 5.5 The DATA\_SPLIT function block – example of a user-specific FB

*Split SMSs extracted with SMS\_M20\_EXTR into single variables*



Parameter	Type	Description
DATA_SPLIT	Instance	Instance name
TEXT	STRING(170)	SMS text (160 characters max.) incl. termination character
VAR1	DINT	1 <sup>st</sup> variable in the SMS. Length: 6 characters incl. sign.
VAR2	DINT	2 <sup>nd</sup> variable in the SMS. Length: 6 characters incl. sign.
.		
.		
VAR10	DINT	10 <sup>th</sup> variable in the SMS. Length: 6 characters incl. sign.

Block type      **Function block without historical values**

The DATA\_SPLIT function block splits an SMS message received by means of SMS\_M20\_EXTR into single INTEGER variables.

The function block is internally based on the DATAPOS function which is part of the 907 AC 1131 library COM\_S90\_V41.LIB.

A detailed description of this function can be found in the description of the 907 AC 1131 library COM\_S90\_V41.LIB.

An enabling of the function block does not exist HERE.

DATA\_SPLIT is **only an example**, where an SMS of the form:

```
123456, -00001, 999999, 000999, 777777, 200002, 987654, -99999, 000005, 333666
corresponding to
- VAR1 -, - VAR2 -, - VAR3 -, - VAR4 -, - VAR5 -, - VAR6 -, - VAR7 -, - VAR8 -, - VAR9 -, - VAR10
```

is split into single integer variables of the same length (6 digits incl. sign) and can be applied at the corresponding outputs.

If the SMS contains less than 10 variables, only the existing variables are output. The remaining ones are **not** overwritten and retain their previous values.



