SAIA BURGESS CONTROLS

Manual Modem PCD xx7 Serie

Document-No. 26/793; Edition EN03; 2014-03-25

0.1 Dokument-History

Document-No.	Edition	Modification	Publication	Remarks
26/793	E1	30.06.2003		Published Edition
26/793	E2	30.04.2005		New Chapter 8 SMS TAP function
	EN03	2014-03-25		Change of Logo

0.2 Trademarks

Saia PCD® is a registered trademark of Saia-Burgess Electronics AG

STEP® ,SIMATIC®, S7-300®, S7-400®, and Siemens® are registered trademarks of Siemens AG

Technical modifications and changes depending on state of the art.

1 Introduction

The modem library for the PCD xx7 serie allows a quick integration of telecommunication through phone line inside your S7-project. It has been made flexible to be able of supporting most of the analog modem found on the market.

At the moment the library supports the following communication types:

- Between PCDs with the protocol (ASCII,RK512,DK3964) as Master or Slave.
- Between PCD and Supervisor with the protocol (ASCII,RK512,DK3964) as Master or Slave.
- Between PCD and PG with the MPI protocol as Master or Slave.
- Between PCD and SMS Center with the protocol UCP (send SMS)

The flexibility of the library function gives the possibility to add new communication types without investing too much in development.

1.1 Structure

The modem driver is based on one main function block (FB-Modem DRIVER) which handles the command “sent to the modem” and “the answer received from the modem”. It controls the modem status and allows the other telecommunication function to proceed. This function has to be called cyclically.

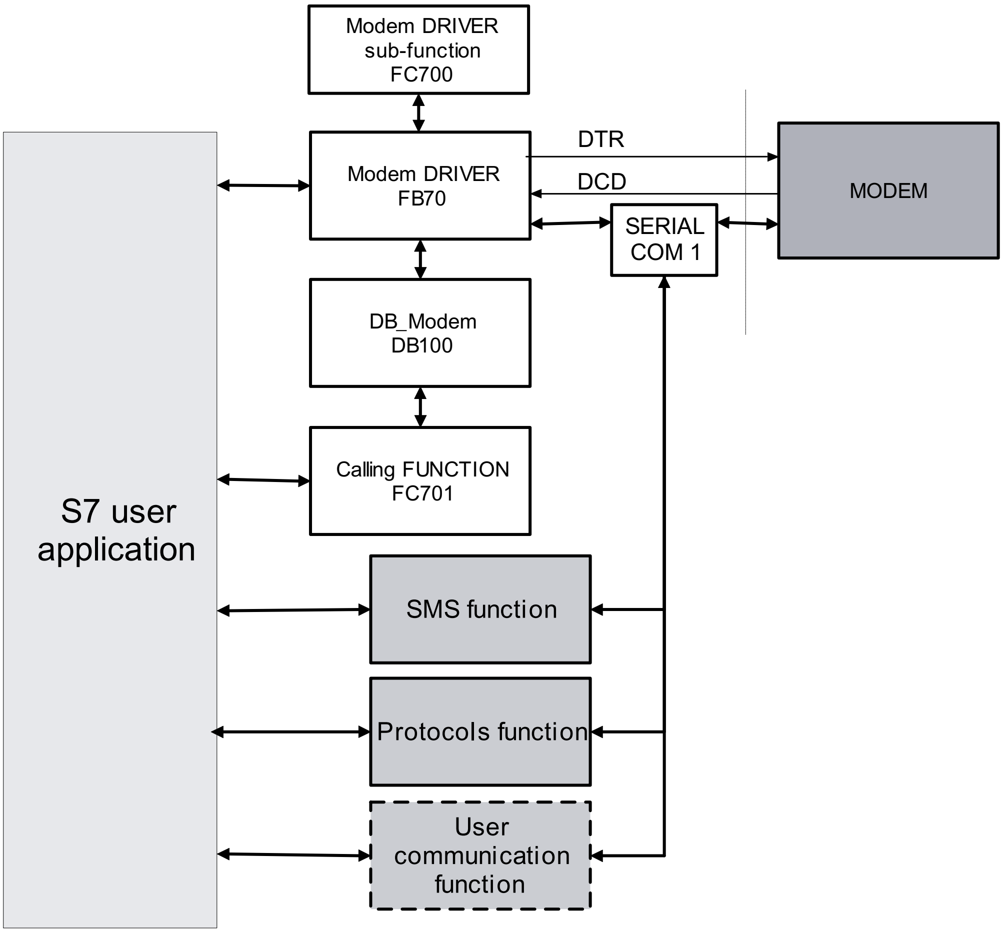
The modem DRIVER is referring to a Data Block (DB_Modem), which contains all the internal parameters of the driver, status and also all the configurable parameters to adapt the driver to your application and modem.

The calling function give the possibility to the system to call and connect with other subscriber. This function doesn’t have direct access to the modem, in fact all the request are transmitted to the DRIVER through the DB_Modem and the DRIVER is then executing them.

When the connection with the other subscriber is established, then the serial COM can be use directly by other function or process. The modem DRIVER stay in back ground task to survey they status of the connection.

The actual function which are also implemented in the library are:

- Switching protocol to communicate with subscriber
- Sending SMS
- Sending Pager messages



All these functions will be seen in details later in this document and examples of using them will be shown.

1.2 Restrictions

There are few restrictions when using the modem library. Those are the following restrictions:

- To handle correctly the modem and the telecommunication, the functions need the modem control signals, as the DCD and DTR. Those have to be wired. Due to this, only the serial interface 1 of the PCD (PCD2.F120) or interface 2 e.g. 4 (PCD7.F522 as one RS232) can be use for the modem communication.
- The number of all the function and data block in the modem library can be changed, except the FC700 (which contains sub-functions for other functions).
- Using old or low cost modem, some restriction can apply.
- If using all possibilities of the modem library, 10 Kbytes of memory are necessary to integrate the whole library in your project.

2 Modem parameters and Main Driver functions

2.1 Configuration of the modem parameters

Most of the parameterization is done in one Data Block.

This Data Block is called “DB_Modem“ and its number in the library is the DB100. The number of the DB can be changed as desired to fit in your project.

Depending on the different modem type in use a different DB has to be copied to DB100, when this is used as DB_Modem

DB102	MODEM_T850	for using the ISDN modem PCD2.T850
DB103	MODEM_T813	for using the analog modem PCD2.T813
DB104	MODEM_T813Z42	for using the analog modem PCD2.T813 or PCD2.T813z42

This “DB_Modem” is divided in two parts.

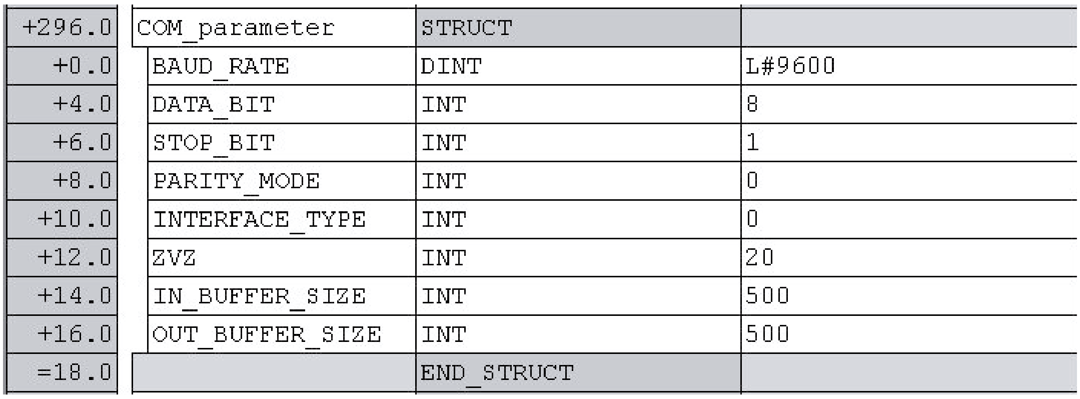
1. The first part is reserved for internal use of the Modem Driver, the field in the DB is called „reserve‟ and it’s the UDT100. **Do not edit and change the values of this field.**
2. The second part is composed of 3 structures, which are:
   - parameterization of the serial com “**COM_parameter**”
   - AT command of the modem for the main function “**AT_COMMAND”**
   - Specific modem answers “**MODEM_MESSAGE**”

Those structure are described in details in the following chapter.

- The “**DB_Modem**“ has to be edited on your S7-editor. After the values have been modified make sure to initialize the value before using it.
- The value of the DB can be changed only by the S7-editor and not by your S7 software during run time.

2.1.1 Structure field “COM_parameter”

In this structure of the “DB_Modem“ you define the serial communication parameters, such as the baud rate, number of data bits, parity mode, etc…



+296.0	COM_parameter	STRUCT	
+0.0	BAUD_RATE	DINT	L#9600
+4.0	DATA_BIT	INT	8
+6.0	STOP_BIT	INT	1
+8.0	PARITY_MODE	INT	0
+10.0	INTERFACE_TYPE	INT	0
+12.0	ZVZ	INT	20
+14.0	IN_BUFFER_SIZE	INT	500
+16.0	OUT_BUFFER_SIZE	INT	500
=18.0		END_STRUCT	

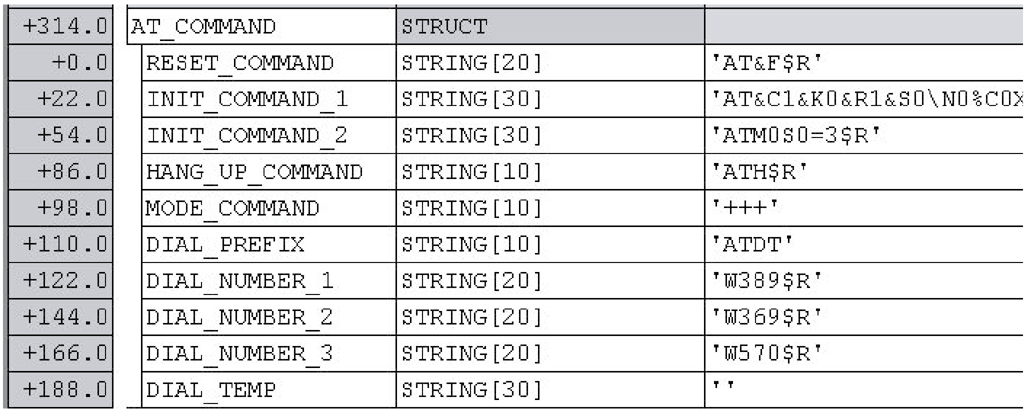
Description of parameters and their possible values (In **Bold**, default values):

Parameter	Possible value	Description
BAUD_RATE	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400	Frequency of the communication between the PCD and the Modem. (In front and behind the connection between the modems).
DATA_BIT	7 or **8**	Number of bit per character.
STOP_BIT	**1** or 2	Number of stop bit at the character end.
PARITY_MODE	**0** none 1 Even 2 Odd 3 Force Low 4 force High	Parity off the character
INTERFACE_TYPE	**0** (RS232)	Only RS232 is possible
ZVZ	1..2000, **20**	Time out in ms. Maximum time between character
IN_BUFFER_SIZE	1..4000, **500**	Size in byte of the receiving buffer. The input buffer is use by the modem DRIVER but also by the other function which will access to the serial port, like the protocol functions. Make sure it's large enough for the amount of data you will need to exchange. This size can be set only here.
OUT_BUFFER_SIZE	1..4000, **500**	Size in byte of the send buffer. The output buffer is use by the modem DRIVER but also by the other function which will access to the serial port, like the protocol functions. Make sure it's large enough for the amount of data you will need to exchange. This size can be set only here.

2.1.2 Structure field "AT_COMMAND"

The structure "AT_COMMAND", contains all the command sent to the modem. Like the DIAL command or the initialization of the modem.

Those command are different from one modem to an other one. So they are specific to the modem you are using. Refer to the operating manual of the modem to find the right configuration. A list of modem and their command already tested on the Saia PCD xx7 is annexed (Annex 1).



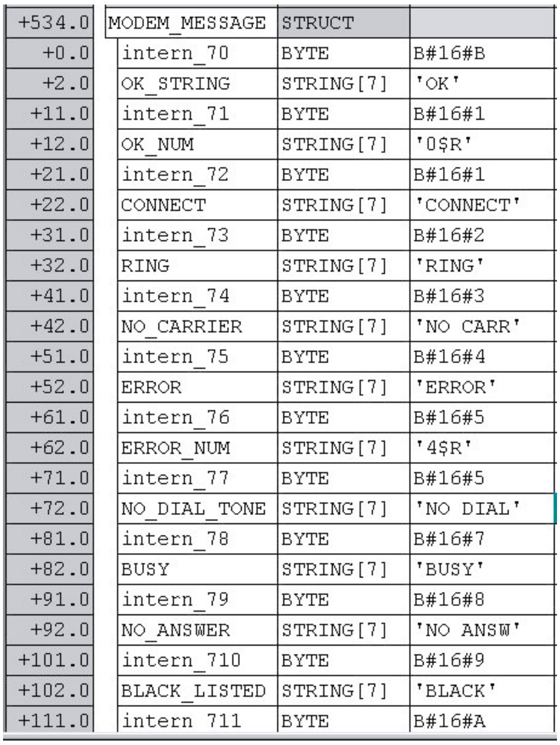
+314.0	AT_COMMAND	STRUCT	
+0.0	RESET_COMMAND	STRING[20]	'AT&F$R'
+22.0	INIT_COMMAND_1	STRING[30]	'AT&C1&K0&R1&S0\N0%C0X
+54.0	INIT_COMMAND_2	STRING[30]	'ATM0S0=3$R'
+86.0	HANG_UP_COMMAND	STRING[10]	'ATH$R'
+98.0	MODE_COMMAND	STRING[10]	'+++'
+110.0	DIAL_PREFIX	STRING[10]	'ATDT'
+122.0	DIAL_NUMBER_1	STRING[20]	'W389$R'
+144.0	DIAL_NUMBER_2	STRING[20]	'W369$R'
+166.0	DIAL_NUMBER_3	STRING[20]	'W570$R'
+188.0	DIAL_TEMP	STRING[30]	''

Description of the parameters and their syntax:

Parameter	Description
RESET_COMMAND	This is the AT command, which reset the modem. The string has to start with **AT** and be finish with **$R .**
INIT_COMMAND_1	This string is the initialization command, to configure the mode of the modem. The string has to start with **AT** and be finish with **$R**. In this command make sure to disable to ECHO, usually the option is “**E0**“.
INIT_COMMAND_2	This is the second initialization string, in case the first was too long. The string has to start with **AT** and be finish with **$R** even if no initialization command are required.
HANG_UP_COMMAND	This is the command to disconnect from the phone line. The string has to start with **AT** and be finish with **$R** .
MODE_COMMAND	String to switch from transparent mode to modem command mode.
DIAL_PREFIX	This is the dialing prefix, depending if you are using a pulse or tone dialling. The string has to start with **AT** .
DIAL_NUMBER_x	Those 3 parameters are the 3 phones number, which the PCD can call to. The string has to be ended with **$R**. The Dial_number will be added to the Dial_prefix, do not duplicate the option.
DIAL_TEMP	This area is not used, leave it free.

2.1.3 Structure field “MODEM_MESSAGE”

When the modem has executed a command or some call is incoming, the modem will signal to the PCD the result of the operation with messages. Those messages could be different between different kind of modem, so they have to be configure corresponding to the modem used in the project. Also, the choice made in the initialization command has to be considered, if you choose the numeric or the message answer.



+534.0	MODEM_MESSAGE	STRUCT	
+0.0	intern_70	BYTE	B#16#B
+2.0	OK_STRING	STRING[7]	'OK'
+11.0	intern_71	BYTE	B#16#1
+12.0	OK_NUM	STRING[7]	'0$R'
+21.0	intern_72	BYTE	B#16#1
+22.0	CONNECT	STRING[7]	'CONNECT'
+31.0	intern_73	BYTE	B#16#2
+32.0	RING	STRING[7]	'RING'
+41.0	intern_74	BYTE	B#16#3
+42.0	NO_CARRIER	STRING[7]	'NO CARR'
+51.0	intern_75	BYTE	B#16#4
+52.0	ERROR	STRING[7]	'ERROR'
+61.0	intern_76	BYTE	B#16#5
+62.0	ERROR_NUM	STRING[7]	'4$R'
+71.0	intern_77	BYTE	B#16#5
+72.0	NO_DIAL_TONE	STRING[7]	'NO DIAL'
+81.0	intern_78	BYTE	B#16#7
+82.0	BUSY	STRING[7]	'BUSY'
+91.0	intern_79	BYTE	B#16#8
+92.0	NO_ANSWER	STRING[7]	'NO ANSW'
+101.0	intern_710	BYTE	B#16#9
+102.0	BLACK_LISTED	STRING[7]	'BLACK'
+111.0	intern 711	BYTE	B#16#A

Numeric answer:

Usually the option for initialization is”**V0**“.*(refer to your modem manual)*

The identification of the answer in the structure need the following syntax, the number of the answer, followed by the **“$R“**.

e.g. For the ERROR answer, the numeric value is 4, so the field ERROR of the structure “Modem_message“ will contains “**4$R**“.

Message answer:

Usually the option for initialization is “**V1**“.*(refer to your modem manual)*

The identification of the answer in the structure need the following syntax. A maximum of 7 characters and minimum of 4 characters of the answer message have to be put in the field. The “$R“ shouldn’t be in. The character case is very important, usually in upper case, be sure to write it correctly.

e.g. For the ERROR answer, the field ERROR of the structure “Modem_message“ will contain “**ERROR**“.

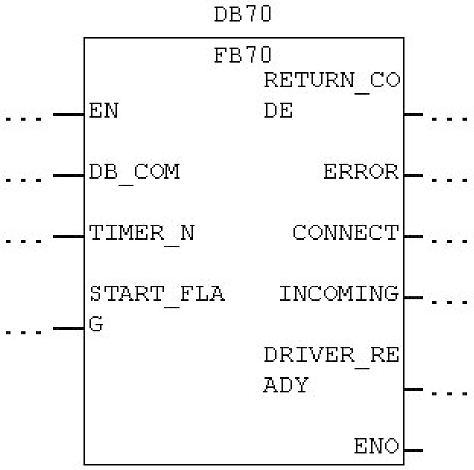
Description of the parameters and their most common values:

Parameters	Common value	Description
intern_7x	-	Those field are reserve for internal use. **!! Do not change the value !!**
OK_STRING	“OK“	Command executed by the modem successfully
OK_NUM	0$R	Same as ”OK“, but the numeric value (needed after reset)
CONNECT	“CONNECT“	Message returned when the two modem have synchronized and the connection is made.
RING	“RING“	Incoming call
NO_CARRIER	“NO CARRIER“	No modem has answered at the receiving side
ERROR	“ERROR“	An error has been made during the last command
ERROR_NUM	4$R	Same as “ERROR“, but the numeric value
NO_DIAL_TONE	“NO DIAL TONE“	No dial tone has been detected
BUSY	“BUSY“	The receiving side is busy
NO_ANSWER	“NO ANSWER“	The receiving side is not answering
BLACK_LISTED	“BLACK LISTED“	This number can’t be dialled anymore, it has failed too many times

2.2 Main Driver function

The Driver of the modem is based on the Function Block “**Modem_DRIVER**“ and its number in the library is the FB70.

This FB is handling the communication between the PCD and the Modem. Also it manages the status of the modem and signals it to the S7 user.



2.2.1 Parameters

Parameter	Type	Description
DB_COM	INT / IN	The number of the “DB_Modem“, see chapter 2.1 .
TIMER_N	TIMER / IN	The number of a TIMER reserved for the modem function. The driver needs internally a TIMER. This TIMER can’t be used by the user anymore.
START_FLAG	BOOL / IN-OUT	Low level on this input will initialize the Modem_DRIVER, when this is done, the Modem_DRIVER set this bit to 1.
RETURN_CODE	INT / OUT	Error code in case of malfunction, see Annex 2
ERROR	BOOL / OUT	1 if a malfunction appends
CONNECT	BOOL / OUT	1, if the modem has connected with an other subscriber for incoming and outgoing calls.
INCOMING	BOOL / OUT	1, if the connect is an incoming call
DRIVER_READY	BOOL / OUT	Modem_DRIVER is READY to start or receive a call.

2.2.2 Function

Starting

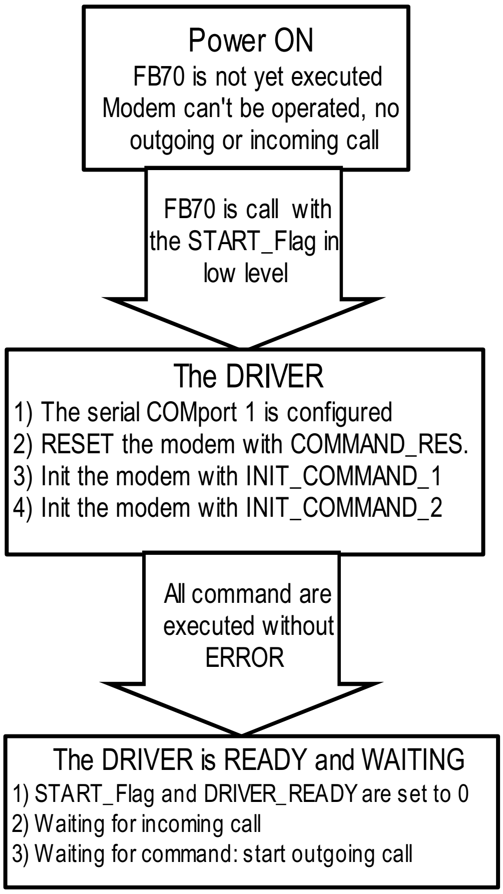
At the Power on, the modem is not ready to handle calls. The FB70 has to run with the input “START_Flag” =0. This will start the initialization procedure of the modem. Concretely the initialization is :

- Configure the serial COM port 1
- Reset the Modem
- Init with the INIT_COMMAND_1
- Init with the INIT_COMMAND_2

If all the init step have been completed without error the Driver is going in a WAITING status.

Otherwise if an error appended, the Driver will not be ready and an error code will be issued.

On this status the Driver is waiting for an incoming call or the command to start an outgoing call.


Power ON
FB70 is not yet executed
Modem can't be operated, no outgoing or incoming call
FB70 is call with the START_Flag in low level
The DRIVER
1) The serial COMport 1 is configured
2) RESET the modem with COMMAND_RES.
3) Init the modem with INIT_COMMAND_1
4) Init the modem with INIT_COMMAND_2
All command are executed without ERROR
The DRIVER is READY and WAITING
1) START_Flag and DRIVER_READY are set to 0
2) Waiting for incoming call
3) Waiting for command: start outgoing call

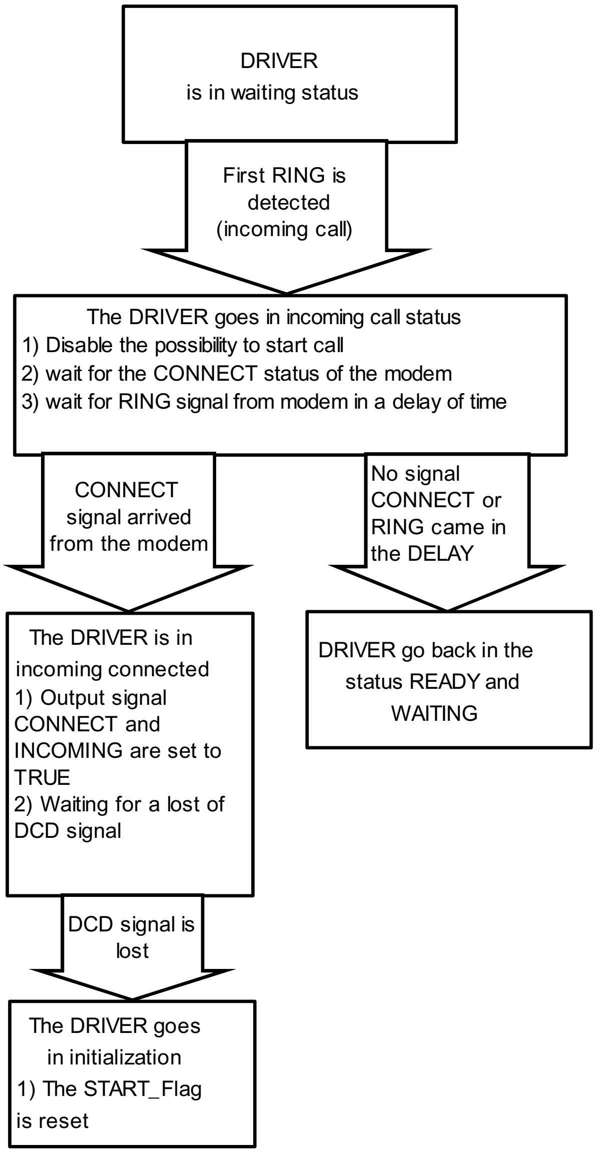
Incoming call

When the Driver is in the WAITING status, an incoming call is detected by the modem message “RING”. At this point the Driver switch to the status “INCOMING CALL” and the Start call are not possible anymore.

The Driver is waiting now for a new “RING” message from the modem or the message “CONNECT”, if this one doesn’t append in a configured time, the Driver is going back to the “WAITING” status.

If the modem get synchronized, the message “CONNECT” is detected by the Driver and this one will change status to “INCOMING CALL CONNECTED”. The output signal of the function “CONNECT” and “INCOMING” are set to 1.

In this status the Driver will not watch anymore what is going or coming through the serial line. For this reason it is possible now to use the serial line with any other kind of protocol. The Driver is just watching the status of the DCD signal.


DRIVER
is in waiting status
First RING is detected (incoming call)
The DRIVER goes in incoming call status
1) Disable the possibility to start call
2) wait for the CONNECT status of the modem
3) wait for RING signal from modem in a delay of time
CONNECT signal arrived from the modem
No signal CONNECT or RING came in the DELAY
The DRIVER is in incoming connected
1) Output signal CONNECT and INCOMING are set to TRUE
2) Waiting for a lost of DCD signal
DRIVER go back in the status READY and WAITING
DCD signal is lost
The DRIVER goes in initialization
1) The START_Flag is reset

When the DCD signal disappears, it means the phone connection is lost or the modem can’t synchronize anymore, this could append any time and the status of the serial line will be unknown.

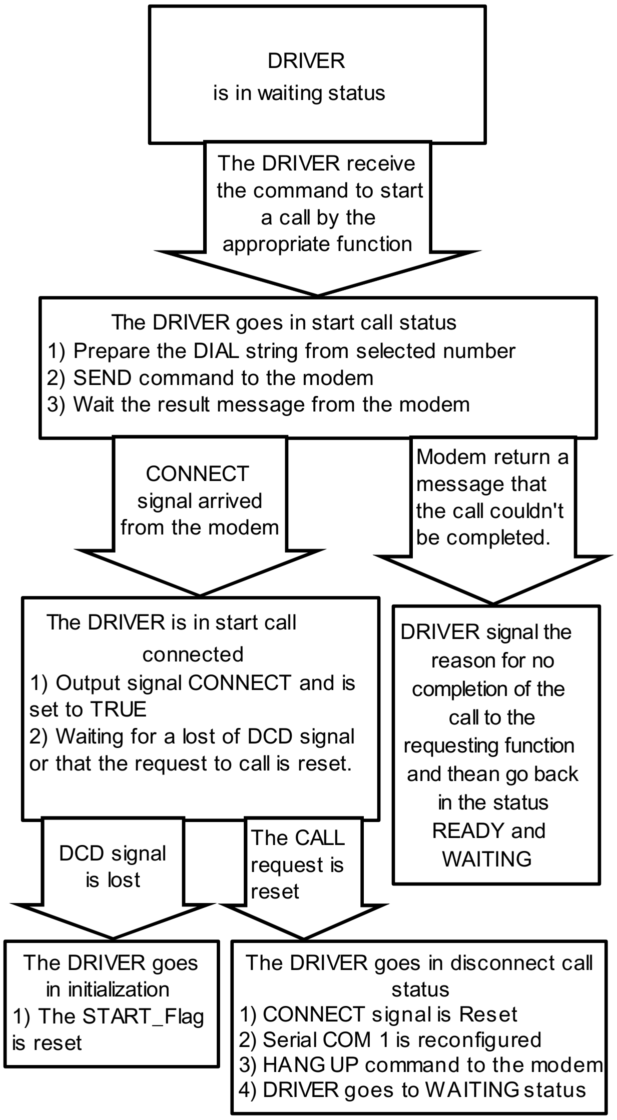
So the modem and the serial line have to be totally reconfigured. The Driver goes in the “STARTING” status.

Start a Call

The user (S7-software) makes a start call request with the appropriate function. The request is transmitted to the Driver through the "DB_Modem".

The Driver goes in "START CALL" Status.and dials depending on the choosen number the prepared dialling string. When an answer message is received, the Driver will:

- if the call couldn't be completed for some reason as Busy, No carrier, etc… , signal this answer to the calling function and then go back to the "WAITING" status.
- if the call succeeded, go in the "Start call connected" status.


DRIVER
is in waiting status
The DRIVER receive the command to start a call by the appropriate function
The DRIVER goes in start call status
1) Prepare the DIAL string from selected number
2) SEND command to the modem
3) Wait the result message from the modem
CONNECT signal arrived from the modem
Modem return a message that the call couldn't be completed.
The DRIVER is in start call connected
1) Output signal CONNECT and is set to TRUE
2) Waiting for a lost of DCD signal or that the request to call is reset.
DRIVER signal the reason for no completion of the call to the requesting function and thean go back in the status READY and WAITING
DCD signal is lost
The CALL request is reset
The DRIVER goes in initialization
1) The START_Flag is reset
The DRIVER goes in disconnect call status
1) CONNECT signal is Reset
2) Serial COM 1 is reconfigured
3) HANG UP command to the modem
4) DRIVER goes to WAITING status

The modem is now connected, the Driver is watching if

- the DCD signal disappear, it mean the phone connection is lost , this could append any time and the status of the serial line will be unknown. So the modem and the serial line have to be totally reconfigured. The Driver goes in the "STARTING" status.
- the Start Call request from the calling function is reset. The Driver goes in the GET DISCONNECTED" status.

The "GET DISCONNECTED" status in the first step, reconfigure the serial COM port 1. Then switch the modem in command mode and send the "HANG UP" command to the modem. When "HANG UP" is done the Driver goes in the "WAITING" status.

3 Receive and make a call

3.1 Receiving a call “Incoming call”

An incoming call is the request of a subscriber to get connected to our PCD through the modem and the serial line.

3.1.1 Enabling the receiving

Depending the application you may need to call and get connected with outside subscriber, but also to be called. For security reasons you may not want to give the possibility for other subscriber to call you and get connect with you. We will seen now how to make this selection.

This is done very easily by enabling or not the auto answer mode of the modem. If the auto answer mode is disabled, the modem will never answer and no connection or communication is possible. The auto answer mode is selected through a Register of the modem (usually this register is the S0, but refer to you modem manual). Setting the value of the register S0 = 0 will disable the auto answer mode, if this register is set to a different value than zero, the modem will answer after the number of RING-signals set in the register.

You can set this value in the initialization string command (chapter 2.1.2). The parameter will be “S0=0“ for no answer or “S0=3“ for answer after 3 RINGs. This parameter can be either in the string INIT_COMMAND_1 or in the string INIT_COMMAND_2.

3.1.2 Receiving process

The Modem Driver handles totally the incoming calls, therefore no specific function is needed for the incoming calls.

Most of the receiving process is transparent to the user, in fact the user will be informed of an incoming call only in 2 situations:

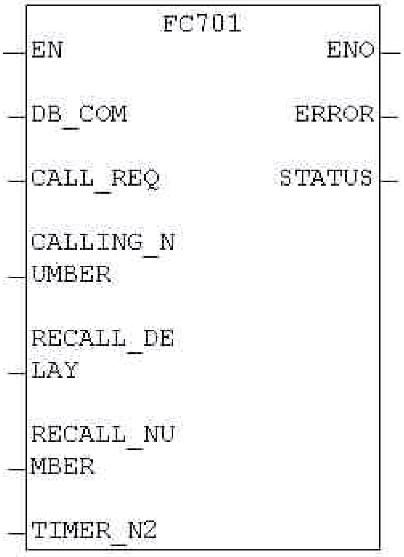
- When the incoming call is accepted and got connected, the Driver function will signal it through the output “CONNECT” and “INCOMING”.
- If you are starting a call during an incoming call, a message error will be returned: “Can’t proceed your request, because actually processing an incoming call“.

3.2 Make an “Outgoing call”

A starting call is the request of the PCD to get connected to a subscriber through the modem and the serial line.

3.2.1 Starting call process

To start a call and get connected to one other subscriber, you will need to use a dedicated function. This function is called “MODEM_CALLING_FC“ and its number in the library is the FC701. It’s possible to renumber this function as you wish.



Parameter	Type	Description
DB_COM	INT / IN	The number of the “DB_Modem“, see chapter 2.1 .
CALL_REQ	BOOL / IN	Rising edge will start the process of making a call, falling edge will cancel the request or disconnect if the subscriber was already connected.
CALLING_NUMBER	INT / IN	The phone number to call from the phone number list, see Chapter 2.1.2 DIAL_Number_x .
RECALL_DELAY	S5TIME	Time between two attempt to connect
RECALL_NUMBER	INT / IN	Number of attempt to connect
TIMER_N2	TIMER / IN	The number of a TIMER reserved for the modem call function. (the call function needs internally this TIMER).This TIMER can’t be use by the S7 program anymore.
ERROR	BOOL / OUT	1 if a malfunction append
STATUS	INT / OUT	If the call could not be completed after the number of specified retries, an error will be outputted, the status will contains the reason. Error code are listed. see Annex 2

3.2.2 Function

When the PCD needs to connect with a subscriber, the function “Modem_Calling_F“ has to be called. A rising edge on **CALL_REQ** will start the process of getting connect. The parameter **CALLING_NUMBER** specifies which is the subscriber phone number taken from the list in the DB_Modem (see chapter 2.1.2).The function then will send the necessary command to the modem Driver.

Unsuccessful:

If for some reason as, the subscriber is busy or didn't answer, the call couldn't be completed, the function will retry to call again by itself. The user specifies the number of time that the function will retry to connect. This is done with the parameter **RECALL_NUMBER**. The function will wait a certain time before to retry to connect, the user also specifies this delay with the parameter **RECALL_DELAY**.

If after the number of retries specified, the connection couldn't be completed, the function will return an error. The output **ERROR** will be set to 1 and the **STATUS** will contain the cause of the unsuccessful call. The **ERROR** output and **STATUS** will be reset when the **CALL_REQ** input is 0.

Successful:

If the call request is completed and the two subscribers get connected, the status connected will be returned by the Modem Driver function (see chapter 2.2.1).The connection will remain as long as the **CALL_REQ** is 1.

If a problem with the phone line happened or the subscriber disconnected first, the function will by itself restart the process of making a call and get connected, but only if the **CALL_REQ** is 1 and the number of retries is not exceeded.

Incoming call:

If a call request is issued when an incoming call is already processed, this one will be memorized and as soon the incoming call is finished, the call will be processed. This only if the **CALL_REQ** still 1.

4 Protocols

Once the modem connection with an other subscriber has been made the serial com port 1 can be freely use with the standard communication function as SFB12, SFB13 and SFB14, **but the SFC240..243 can't be used.**

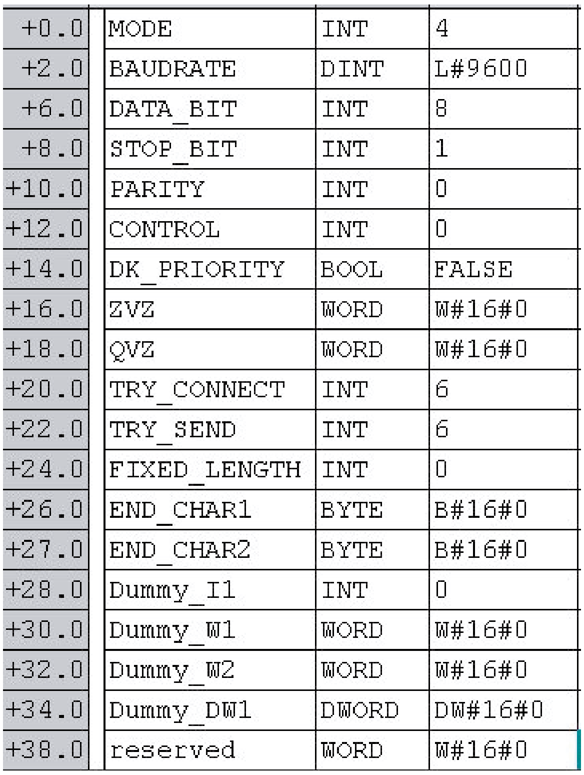
To simplify the enabling of the protocols, a function is available in the modem library and in conjunction with it a DB, which contains the parameters required.

In order to understand clearly the use of the enabling protocol function, you should be in possession of the protocol documentation (manual no. 26/794). We will refer to it several times.

4.1 Configuration DB

The selection of the protocol type and its configuration is made in one DB. This DB is the number 900 in the modem library and also call "PROTOCOL_DB", of course the number of the DB can be freely changed to feed in the application.

The DB has the following structure:



+0.0	MODE	INT	4
+2.0	BAUDRATE	DINT	L#9600
+6.0	DATA_BIT	INT	8
+8.0	STOP_BIT	INT	1
+10.0	PARITY	INT	0
+12.0	CONTROL	INT	0
+14.0	DK_PRIORITY	BOOL	FALSE
+16.0	ZVZ	WORD	W#16#0
+18.0	QVZ	WORD	W#16#0
+20.0	TRY_CONNECT	INT	6
+22.0	TRY_SEND	INT	6
+24.0	FIXED_LENGTH	INT	0
+26.0	END_CHAR1	BYTE	B#16#0
+27.0	END_CHAR2	BYTE	B#16#0
+28.0	Dummy_I1	INT	0
+30.0	Dummy_W1	WORD	W#16#0
+32.0	Dummy_W2	WORD	W#16#0
+34.0	Dummy_DW1	DWORD	DW#16#0
+38.0	reserved	WORD	W#16#0

This DB has to be edited only from the S7-editor and the can't be changed by the software itself during the run time.

Description and possible values of the parameters

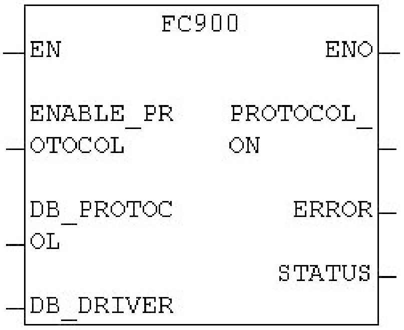
Name	Possible value	Description
MODE	0..9	This select the protocol mode you like to use: 0 = Transparent 1= DK3964 2 = DK3964R 3 = RK512 4 = RK512R 5 = ASCII fixed length 6 = ASCII 1 end char 7 = ASCII 2 end character 8 = ASCII Time Out 9 = MPI Protocol
BAUDRATE	300,600,1200,2400,4800,9600,19200,38400	This parameter defines the Baud Rate use to communicate between the subscriber. (*1)
DATA_BIT	7 or 8	Number of data bits (*1)
STOP_BIT	1 or 2	Number of stop bits (*1)
PARITY	0 None 1 Even 2 Odd 3 Force Low 4 Force High	Parity mode (*1)
CONTROL	0 (RS232)	Only RS232 is possible
DK_PRIORITY	1 or 0	Depending on the chosen protocol, see Man. 26/794.
ZVZ	Depend. on mode	Depending on the chosen protocol, see Man. 26/794.
QVZ	Depend. on mode	Depending on the chosen protocol, see Man. 26/794.
TRY_CONNECT	0..255	Depending on the chosen protocol, see Man. 26/794.
TRY_SEND	0..255	Depending on the chosen protocol, see Man. 26/794.
FIXED_LENGTH	1..1024	Only for the fixed length ASCII driver, the number of character in the frame. see Man. 26/794.
END_CHAR1	0..255	Use only for the one end character ASCII driver, define the end character of the frame. See Man. 26/794.
END_CHAR2	0..255	Use only for the two end characters ASCII driver, define the end characters of the frame. See Man. 26/794.
Dummy_I1	0	Not used actually
Dummy_W1	0	Not used actually
Dummy_W2	0	Not used actually
Dummy_DW1	0	Not used actually
Reserved	0	Not used actually

(*1) Usually this parameter should be set to the same value as the one configured for the modem (DB_Modem, see chapter 2.1.1), but some modem support a different value after being connected, refer to your modem user manual and set the Modem initialization string correctly.

The acknowledge time can reach with modem communication big values. That may cause problems during communication. In this case set a higher value for QVZ. 10 seconds should be also for the worst case enough.

4.2 Enabling protocol

The protocols are enabled by the function “PROTOCOL_FC“, numbered FC900 in the modem library. The FC number can be freely changed.



Description and possible values of the parameters

Parameter	Type	Description
ENABLE_PROTOCOL	BOOL / IN	A rising edge will enable the protocol
DB_PROTOCOL	BLOCK_DB / IN	The DB number containing the protocol parameters
DB_DRIVER	BLOCK_DB / IN	The DB number of the modem Driver “DB_ MODEM“
PROTOCOL_ON	BOOL / OUT	Signal that the serial line has been reconfigured with the desired protocol and it’s now available.
ERROR	BOOL / OUT	Error happened during the initialization of the protocol
STATUS	WORD / OUT	Indication on the error which happened, usually error are due to an incorrect value in the parameter of the “DB_PROTOCOL“.

4.2.1 Function

When a rising edge is detected on the input **ENABLE_PROTOCOL**, the function will initialize the serial COM port 1, with the selected protocol and parameters configured in the PROTOCOL_DB. If the operation was completed successfully the output **PROTOCOL_ON** is set to 1.

When the input **ENABLE_PROTOCOL** is going down to the 0 level, it doesn’t affect the protocol configuration of the serial COM port 1, but the output signal **PROTOCOL_ON** is reset.

The input **ENABLE_PROTOCOL** has to be used in conjunction of the connected signal of the modem DRIVER If the connected signal of the Driver is 1, then you can enable the protocol, if the connected signal is 0 then the signal **ENABLE_PROTOCOL** should also be 0.

In some cases and this depends on the subscriber, you should wait a little bit between the signal connected of the modem Driver and to set the **ENABLE_PROTOCOL**, because sometimes the other subscriber can take more time to get connected.

5 Example: Starting an application

In this example we will show the necessary steps to integrate the modem functionality into your application. We divide this in 4 phases:

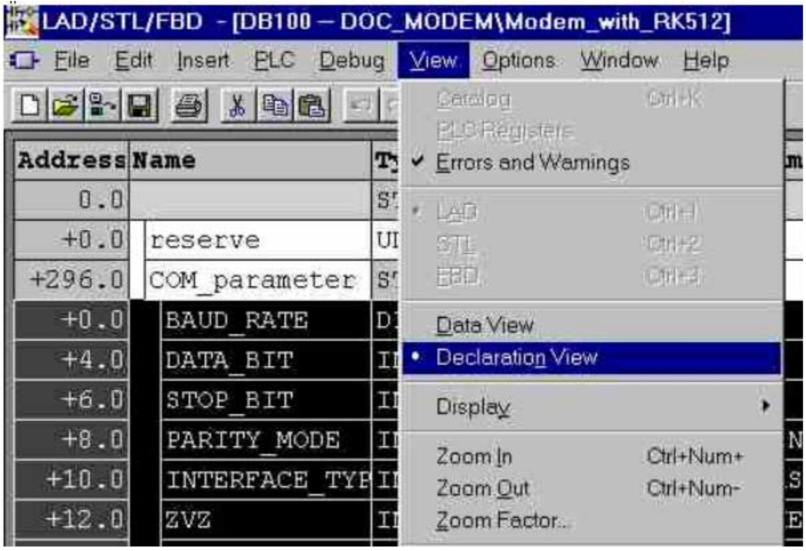
1. Configuring the modem parameters
2. Initialization of the driver and the modem
3. Starting a call
4. Switching to protocol as RK512R

5.1 Configuring the modem parameters

In the first step, you need to integrate the Modem functions in your application. In this case we will need the following ones:

- FB70 and DB70 (Modem Driver) → it's possible to change their number
- FC700 (Modem Driver sub-function) → the number **can't be changed**
- DB100 (DB_Modem) all modem parameters → the number can be changed
- UDT100 (DB_Modem_sub_type) → the number can be changed, but you also need to edit the DB100 and to change the UDT number in it.
- FC701 (Modem_Calling function) → the number can be changed
- FC900 and DB900 (Protocol enable) → the number can be changed

When this is done, you need to edit the DB100 to adapt it for your modem type.
Be sure that you are in "Declaration View"



LAD/STL/FBD - [DB100 – DOC_MODEM\Modem_with_RK512]

Address	Name	Ty
0.0		S
+0.0	reserve	UI
+296.0	COM_parameter	S
+0.0	BAUD_RATE	D
+4.0	DATA_BIT	I
+6.0	STOP_BIT	I
+8.0	PARITY_MODE	I
+10.0	INTERFACE_TYP	I
+12.0	ZVZ	I

5.1.1 Parameter of the serial communication

In the structure `COM_parameter` you adjust the parameter of the serial communication port. This configuration of serial COM has to be adapted to communicate to the modem and probably to communicate with the other subscriber after connection.

In the example we configure the serial COM port as it's used most often:

Baud rate of 9600, 8 data bits, 1 stop bit, no parity bit. The ZVZ is adjusted for the modem reply time to the commands given. This value (of 20 ms) works for the most modems.The buffer size of 500 bytes, means that the serial COM will not be able to receive or send more than 500 byte per time.

BAUD_RATE	DINT	L#9600
DATA_BIT	INT	8
STOP_BIT	INT	1
PARITY_MODE	INT	0
INTERFACE_TYPE	INT	0
ZVZ	INT	20
IN_BUFFER_SIZE	INT	500
OUT_BUFFER_SIZE	INT	500

5.1.2 AT command for the modem

In the structure `AT_COMMAND` you define which are the correct AT commands to operate your modem and also how you like it to be configured.

Those commands are found in the annex 1 for the modems tested on the PCD xx7.

It’s better for the modem Driver that the answers of the modem are in strings and not in numeric messages (but both are possible). This is done with the parameter **“V1”** in the **INIT_COMMAND**. Also it’s betterif no echo are issued from the modem. This is the parameter **“E0”** in the **INIT_COMMAND**.

The application specific parameter are the DIAL_NUMBER, it’s were you define the list of the phone numbers you will be able to make calls. There are 3 possible numbers.

RESET_COMMAND	STRING[20]	‚ATZ$R‘
INIT_COMMAND_1	STRING[30]	‚ATE0V1X3$R‘
INIT_COMMAND_2	STRING[30]	‚AT%C0S0=3$R‘
HANG_UP_COMMAND	STRING[10]	‚ATH0$R‘
MODE_COMMAND	STRING[10]	‚+++‘
DIAL_PREFIX	STRING[10]	‚ATDT‘
DIAL_NUMBER_1	STRING[20]	‚W389$R‘
DIAL_NUMBER_2	STRING[20]	‚W369$R‘
DIAL_NUMBER_3	STRING[20]	‚W570$R‘

5.1.3 Answer message from the modem

When a command or an event is happening, the modem sends a message to the Modem Driver. Those messages could be various between different modems. To be able to identify the right event, the Driver needs a reference with the possible message. This is done in the structure `MODEM_MESSAGE`. Those messages can be found in the Annex 2 for the modems tested on the PCD xx7.

In our case we are taking the string message for the selected modem.

intern_70	BYTE	B#16#B	**! DO NOT EDIT !**
OK_STRING	STRING[7]	‚OK‘	Configurable
intern_71	BYTE	B#16#1	**! DO NOT EDIT !**
OK_NUM	STRING[7]	‚0$R‘	**! DO NOT EDIT !**
intern_72	BYTE	B#16#1	**! DO NOT EDIT !**
CONNECT	STRING[7]	‚CONNECT‘	Configurable
intern_73	BYTE	B#16#2	**! DO NOT EDIT !**
RING	STRING[7]	‚RING‘	Configurable
intern_74	BYTE	B#16#3	**! DO NOT EDIT !**
NO_CARRIER	STRING[7]	‚NO CARR‘	Configurable
intern_75	BYTE	B#16#4	**! DO NOT EDIT !**
ERROR	STRING[7]	‚ERROR‘	Configurable
intern_76	BYTE	B#16#5	**! DO NOT EDIT !**
ERROR_NUM	STRING[7]	‚4$R‘	**! DO NOT EDIT !**
intern_77	BYTE	B#16#5	**! DO NOT EDIT !**
NO_DIAL_TONE	STRING[7]	‚NO DIAL‘	Configurable
intern_78	BYTE	B#16#7	**! DO NOT EDIT !**
BUSY	STRING[7]	‚BUSY‘	Configurable
intern_79	BYTE	B#16#8	**! DO NOT EDIT !**
NO_ANSWER	STRING[7]	‚NO ANSW‘	Configurable
intern_710	BYTE	B#16#9	**! DO NOT EDIT !**
BLACK_LISTED	STRING[7]	‚BLACK‘	Configurable
Intern_711	BYTE	B#16#A	**! DO NOT EDIT !**

▨ DO NOT EDIT, system parameters

5.1.4 Initialize DB

When all the parameters of the DB100 have been set, you need now to initialize the DB, **this is very important.**

1. In the menu View of the editor, select the option “Data view“.

2. In the menu Edit, select the function “Initialize Data Block“.

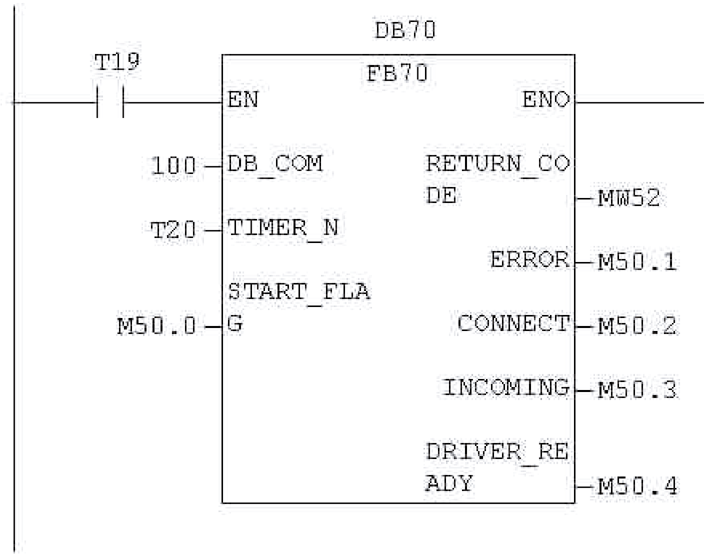
Is this done, you can save your DB and close it.

5.2 Initialization of the Driver

In our example, all the modem functions including the modem driver are used in the FC1. To insure a proper use of the modem, it’s necessary that the modem Driver function is called every cycle after the initialization of it.

Usually we initialize the modem and its Driver after the POWER ON. If the POWER ON of the PCD and the modem is done in the same time, then the initialization of the modem should be delayed a little bit, to be sure that the modem is in order to answer.

In the example we do it with the timer T19:



On the POWER ON or on the STOP → RUN phase, all the non-retentive flags are resetted, so we are sure that, when we call the FB70, the input START_FLAG (M50.0) is 0. This will start the Driver and initialize the modem. After the first execution of the FB70 the flag M50.0 is set to 1. As soon the modem and Driver are ready the flag M50.4 is 1.

A low level (0) on the input START_FLAG done anytime, will reset and initialize again the Driver and modem.

5.3. Calling

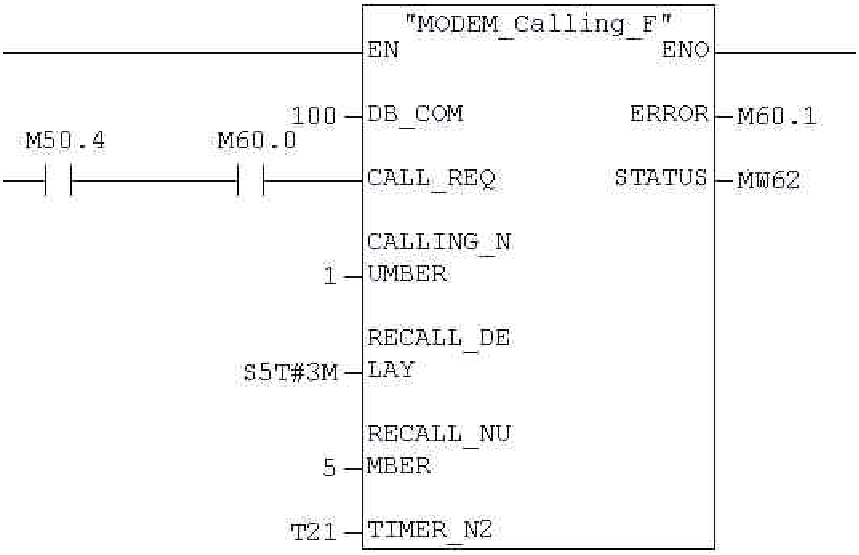
To give the possibility of calling and getting connected with other subscriber, we need to integrate the calling function into the project. This is done very easily with the FC701.

The FC can be called every cycle. It does not affect a possible incoming call and will not start a call until the required condition are fulfilled.

In this example the calling condition is: the flag M50.4 (DRIVER_READY) is 1 and the M60.0 also 1. When a rising edge is detected on the CALL_REQ, the calling process starts.

It will call the phone number 1 in the list (see 5.1.2). If the call can’t be completed the first time, it will retry it after 3 minutes and 5 attempts in maximum. If it still can’t connect, M60.1 (ERROR) will be set and MW62 will contain the error status.

The function needs internally a TIMER and we allocate the T21, but this TIMER can’t be used somewhere else in the S7-program anymore.

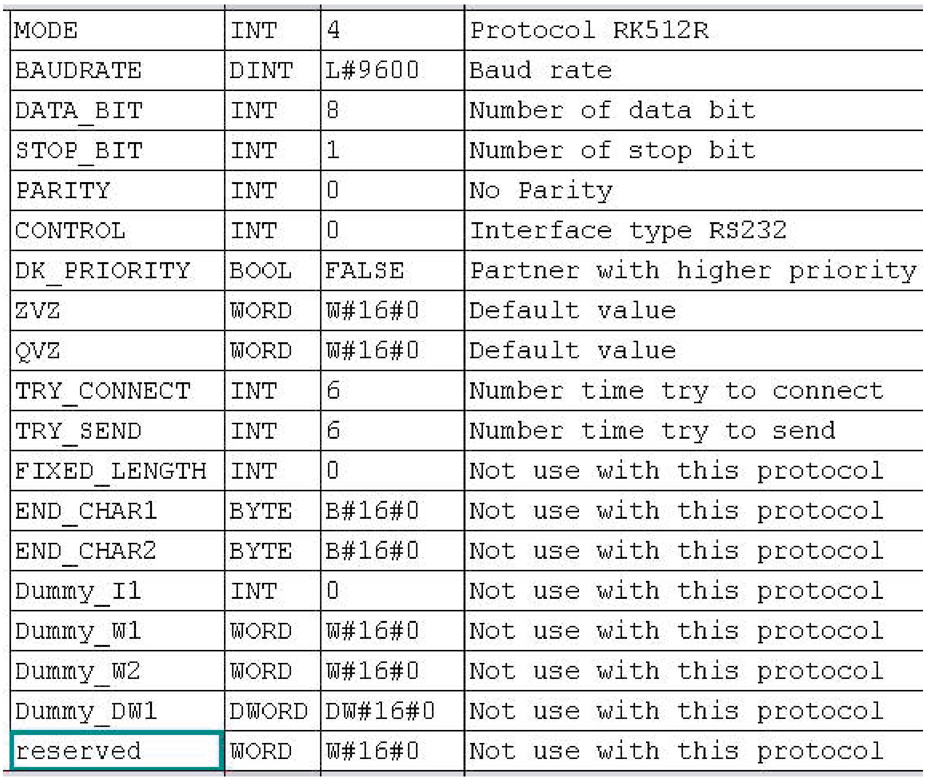


As soon the connection is completed, the Driver signals it by setting the bit M50.2 to 1 and as it is an outgoing call the bit M50.3 stay at 0.

5.4 Switching the protocol

5.4.1 Configuration

To use a protocol through the modem in your application, you need first to configure it. In the example the configuration of the protocol is in the DB900.



MODE	INT	4	Protocol RK512R
BAUDRATE	DINT	L#9600	Baud rate
DATA_BIT	INT	8	Number of data bit
STOP_BIT	INT	1	Number of stop bit
PARITY	INT	0	No Parity
CONTROL	INT	0	Interface type RS232
DK_PRIORITY	BOOL	FALSE	Partner with higher priority
ZVZ	WORD	W#16#0	Default value
QVZ	WORD	W#16#0	Default value
TRY_CONNECT	INT	6	Number time try to connect
TRY_SEND	INT	6	Number time try to send
FIXED_LENGTH	INT	0	Not use with this protocol
END_CHAR1	BYTE	B#16#0	Not use with this protocol
END_CHAR2	BYTE	B#16#0	Not use with this protocol
Dummy_I1	INT	0	Not use with this protocol
Dummy_W1	WORD	W#16#0	Not use with this protocol
Dummy_W2	WORD	W#16#0	Not use with this protocol
Dummy_DW1	DWORD	DW#16#0	Not use with this protocol
reserved	WORD	W#16#0	Not use with this protocol

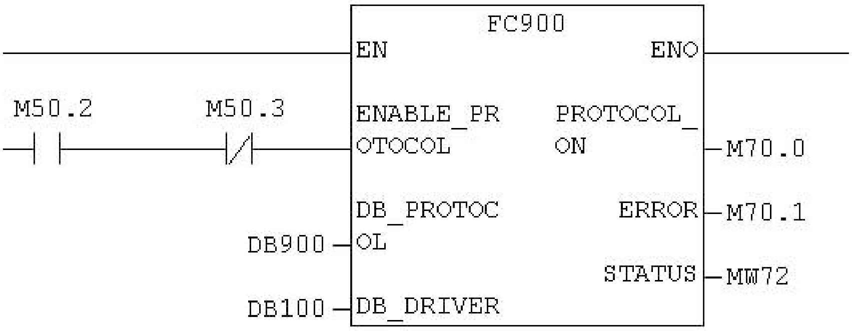
The protocol is configured as follows:

Protocol RK512R, 9600 Baud, 8 data bits, 1 stop bit, no parity. All other parameters are the default one. More details about it can be found in the manual 26/794.

5.4.2 Protocol enabling

To enable a protocol, the function of the library needed is the FC900. This FC needs the inputs:

- the **ENABLE_PROTOCOL** signal which is edge sensitive. Here it is created with **CONNECT** (M50.2) and **INCOMING** (M50.3), this has the effect to enable the protocol after connection, only for outgoing calls. Then it’s possible to enable a different protocol for incoming calls.
- **PROTOCOL_DB**, which contains the protocol configuration parameters.
- **DB_DRIVER**, which is the DB_MODEM containing the information of the buffer size.



When the function is executed once, the protocol is enabled if no error was done in the configuration parameters, then the output **PROTOCOL_ON** is set to 1. If an ERROR happened during enabling the protocol, the output **ERROR** is set, the **STATUS** includes the error code.

You do not have to execute this function every cycle after enabling the protocol, only if you want the output **PROTOCOL_ON** updated.

At this point, the example is finished, the project contains the minimum required for the communication through a modem. Of course, in the final application the error bit and status have to be considered.

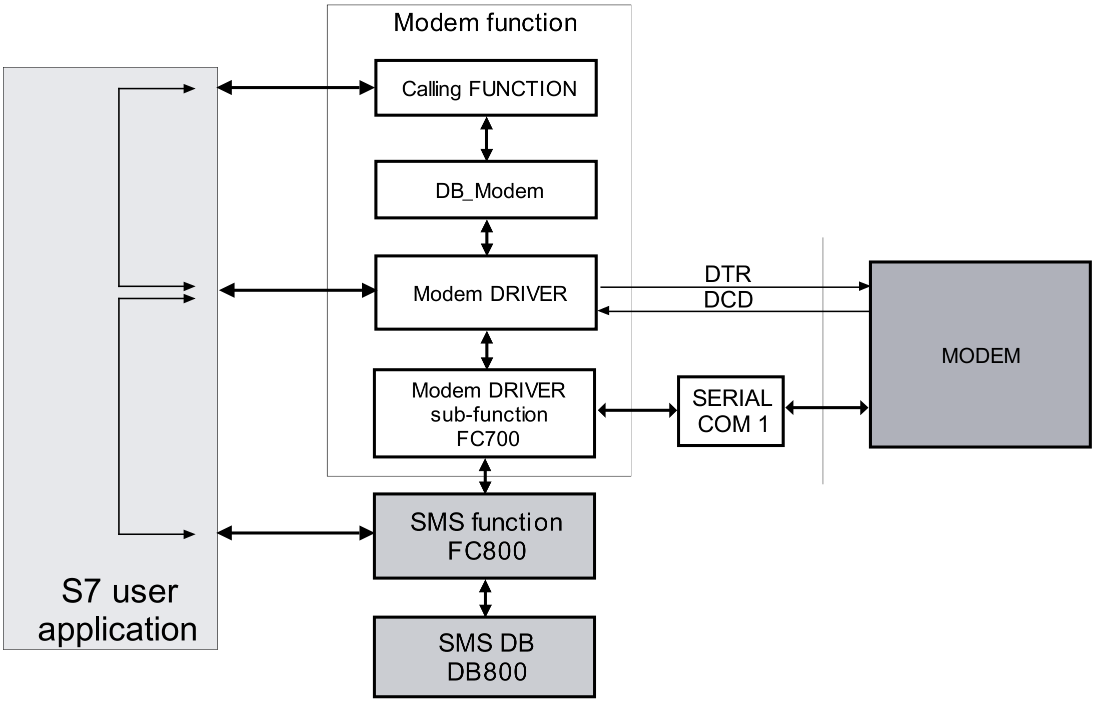
6 Sending SMS

The SMS function of the library gives the possibility to the system to send Short Messages. This can be very useful for the system to signal an event or alarm to the maintenance staff.

The protocol used for this SMS function is the UCP and the function allows sending Messages with a maximum of 140 characters.

6.1 Structure

The SMS function has to be used in conjunction with the Modem function and Driver. In fact you need the Modem function to establish the connection with the SMS center and then the SMS function can be called.



The SMS function is associated with an SMS Data Block. The Data Block is divided in two areas. One is reserved for internal use of the function and the other area contains the information about the application. The SMS function also need to access the sub-function of the FC700 (Modem Driver sub-function). The synchronization between the Modem Driver and the SMS function is made by the S7-application.

6.2 Parameterization of the SMS system

Most of the SMS system parameterization is done in the SMS Data Block. In the library its number is the DB800 **“SMS_DB”**. But there is also one parameter that have to be set in the modem Data Block **“DB_Modem”**.

6.2.1 Parameter in the “DB_Modem”

As the Modem Driver and the Calling function are establishing and handling the connection with the SMS Center, one parameter has to be adjusted in the DB_Modem. The parameter to adjust is the **DIAL_Number**. In fact the Calling function will call the SMS Center and then when the connection is established, the SMS function will be called. So the DIAL_Number of the SMS Center has to be inserted in the DB_Modem (for details see Chapter 2.1.2). Here are some numbers of SMS Centers and their providers. For other countries, you can obtain the phone number of the SMS Center by your local provider.

Phone numbers for Analogue modems:

Country	Provider	Phone number
Switzerland	Swisscom	079 / 4998990
Germany	Mannesmann D2	01722 / 278020

Make sure that the protocol is UCP.

Phone numbers for ISDN modem:

Country	Provider	Phone number
Switzerland	Swisscom	0900 900 941

Make sure that the ISDN protocol is V.120

6.2.2 SMS system parameters “DB_SMS”

This DB is divided in two parts:

- The first part is created by the UDT800 and the purpose of this area is a working space for the SMS function. **Its content doesn’t have to be edited or modified!**
- The second part contains all the parameter of the SMS system.

Communication

Parameter	Type	Description
BAUD_RATE	DINT	(*1) BAUD RATE to communicate with the SMS center.
DATA_BIT	INT	(*1) Number of data bits
STOP_BIT	INT	(*1) Number of stop bits
PARITY_MODE	INT	(*1) Parity bit mode
INTERFACE_TYPE	INT	“0“, only possible with RS232
ZVZ	INT	Time out between characters , normally → 100
IN_BUFFER_SIZE	INT	(*2) Size of the receive buffer
OUT_BUFFER_SIZE	INT	(*2) Size of the send Buffer

(*1) Those parameters should be the same as the ones configured in the DB_Modem (see chapter 2.1.1), only in special case those parameters can be different.
(*2) Those parameters should have exactly the same values as configured in the DB_Modem, see chapter 2.1.1

SMS

Parameter	Type	Description
SMS_Recipient_N1	STRING[16]	Phone number 1, only numeric characters.
SMS_Recipient_N2	STRING[16]	Phone number 2, only numeric characters.
SMS_Recipient_N3	STRING[16]	Phone number 3, only numeric characters.
SMS_Recipient_N4	STRING[16]	Phone number 4, only numeric characters.
SMS_Sender_ID	STRING[16]	Identification number sender, only numeric characters.
Wait_after_connection	S5TIME	Time to wait between the request to SEND SMS is made and the moment that the PCD sends the SMS to the center. A certain time is needed, because both modem do not get connected in the same time.

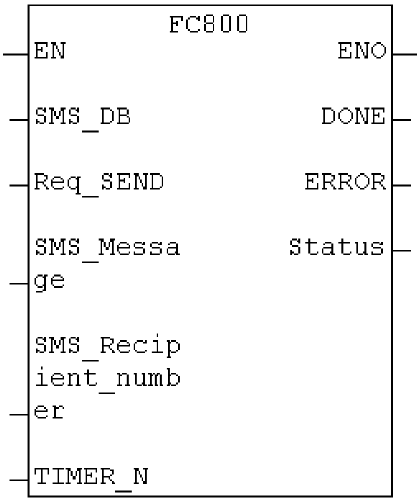
Those are the only parameters to configure for the SMS system. Do not forget to initialize the Data Block after any modification.

6.3 SMS function

This function will interface the PCD to the SMS Center, sending the Message and receive back the answer for acknowledge or error.

In order to work correctly, the PCD needs the connection to be established with the SMS Center. The connection is done with the Calling Function (see chapter 3.2). When then connection is established the function SMS can be called.

Once the function is called with the **Req_SEND** =1, the function will wait a delay configured “Wait_for_Connection“ (see 6.2.2) to be sure that the SMS Center is also connected. Then the function sends the SMS message to the SMS Center, the answer will be return on the output **DONE** for a successful transmission or on the output **ERROR** and Status if an error has been detected in the SMS configuration or message. If after a delay of 30 seconds the SMS Center didn’t answer to our request the **ERROR** output will be set.



- The function affects the Data Block Register 1.
- Internally the function is working with the address register, to avoid any problem with some alarm interrupts, those one are disabled.

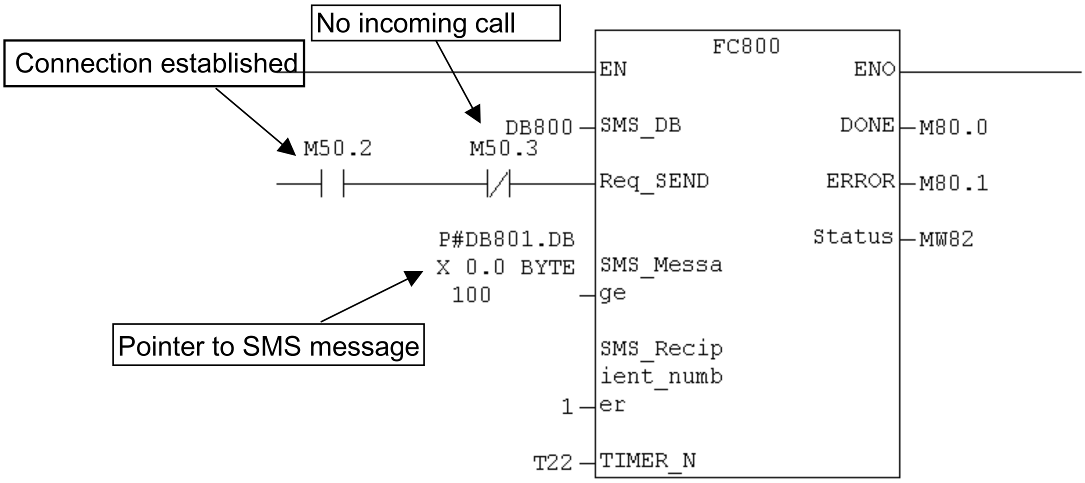
Parameter	Type	Description
SMS_DB	BLOCK_DB / IN	Number of the SMS_DB
Req_SEND	BOOL / IN	Request to send the SMS, 1 will process the request, 0 will cancel.
SMS_Message	ANY / IN	Pointer to the string “Message“ (maximum 140 characters)
SMS_Recipient_num.	INT / IN	Number in the list of the SMS recipient
TIMER_N	TIMER / IN	Number of a TIMER reserved for the SMS function (needed internally). TIMER can’t be used anymore.
DONE	BOOL / OUT	The SMS has been sent with success, stay at 1 until the Req_SEND is reset.
ERROR	BOOL / OUT	An error occurred during the transmission of the SMS, stay at 1 until the Req_SEND is reset.
Status	INT / OUT	Error code of the operation, output the code until the Req_SEND is reset.

6.3.1 Integration to the application

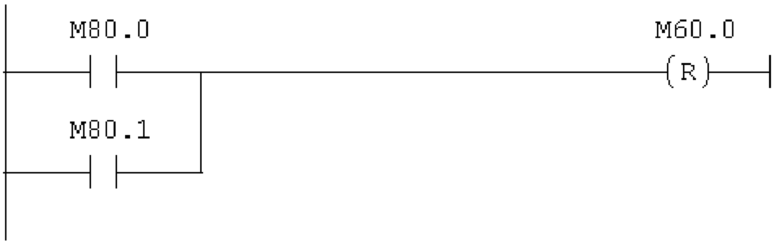
Integrate in your project the modem driver library (see chapter 5 to 5.3). But when doing the configuration of the **DB_Modem** you need to consider which are the value for the parameters to use in conjunction to the SMS Center. It means, what is the phone number of the SMS Center (with necessary prefix), which is the baud rate supported by the SMSC.

1. Integrate in your project the SMS function FC800,DB800 and UDT800.
2. Configure the **DB_SMS** (see 6.2.2 and 6.2.3).
3. In Data Blocks, prepare the SMS message in a string format and not longer than 140 characters.
4. Call the function FC800 in such a way that:
   - Make the request to send SMS, when the connection has been established. For this use the signal **CONNECT** from the modem Driver.
   - Cancel the Calling request when the SMS function return an signal **DONE** or **ERROR**.

Call of the function



Use the return signals to disconnect



You can find a similar examples in the modem library.

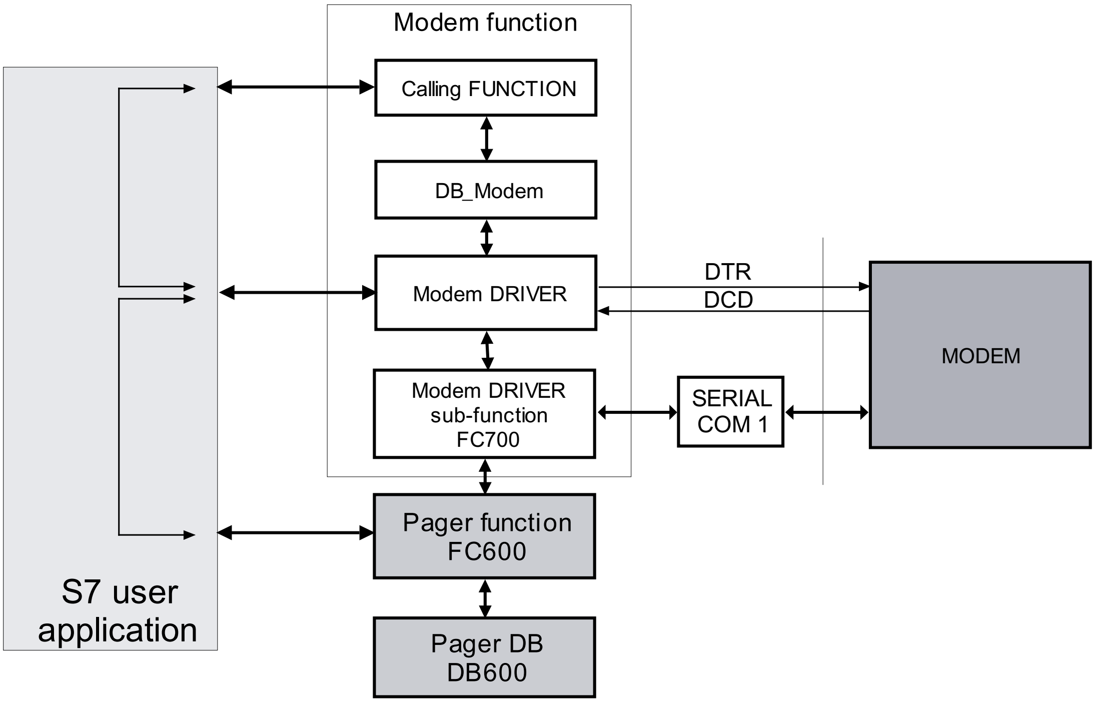
7 Sending Pager Messages

The Pager function of the library gives the possibility to the system to send Pager Messages. This can be very useful for the system to signal an event or alarm to the maintenance staff.

The protocol used for this Pager function is the UCP and the function allows sending Messages with a maximum of 140 characters. But this is also depending of the **maximum length allowed by the Pager provider.**

7.1 Structure

The Pager function has to be used in conjunction with the Modem function and Driver. In fact you need the Modem function to establish the connection with the Page Center and then the Pager function can be called.



The Pager function is associated with a Pager Data Block. The Data Block is divided in two areas. One is reserved for internal use of the function and the other area contains the information about the application. The Pager function also needs to access the sub-function of the FC700 (Modem Driver sub-function).

The synchronization between the Modem Driver and the Pager function is made by the S7-application.

7.2 Parameterization of the Pager system

Most of the Pager system parameterization is done in the Pager Data Block. In the library its number is the DB600 **“Pager_DB”**. But there is also one parameter that have to be set in the modem Data Block **“DB_Modem”**.

7.2.1 Parameter in the “DB_Modem”

As the Modem Driver and the Calling function are establishing and handling the connection with the Page Center, one parameter has to be adjusted in the DB_Modem. The parameter to adjust is the **DIAL_Number**. In fact the Calling function will call the Page Center and then when the connection is established, the Pager function will be called. So the DIAL_Number of the Page Center has to be inserted in the DB_Modem (for details see Chapter 2.1.2). Here are some numbers of Page Centers. For other countries, you can obtain the phone number of the Page Center by your local provider.

Phone numbers for Analogue modems

Country	Provider	Phone number	Note	Type
Switzerland	Swisscom	074 090 0003	V.22,V.21	Analog
Switzerland	Swisscom	074 090 0103	→V.34	Analog

Make sure that the protocol is UCP.

Phone numbers for ISDN modems

Country	Provider	Phone number	Note	Type
Switzerland	Swisscom	074 090 0103	→V.110	ISDN
Switzerland	Swisscom	074 090 0108	→X.75	ISDN

7.2.2 Pager system parameters “Pager_DB”

This DB is divided in two parts:

- The first part is created by the UDT600 and the purpose of this area is a working space for the Pager function. **Its content doesn’t have to be edited or modified !**
- The second part contains all the parameter of the Pager system.

Communication

Parameter	Type	Description
BAUD_RATE	DINT	**9600** (*1) BAUD RATE to communicate with the Page Center.
DATA_BIT	INT	**8** (*1) Number of data bits
STOP_BIT	INT	**1** (*1) Number of stop bits
PARITY_MODE	INT	**0** (*1) Parity bit mode
INTERFACE_TYPE	INT	“**0**“, only possible with RS232
ZVZ	INT	Time out between characters, normally → 100
IN_BUFFER_SIZE	INT	(*2) Size of the receive buffer
OUT_BUFFER_SIZE	INT	(*2) Size of the send Buffer

(*1) Those parameters should be the same as the ones configured in the DB_Modem (see chapter 2.1.1), only in special case those parameters can be different.

(*2) Those parameters should have exactly the same values as configured in the DB_Modem, see chapter 2.1.1

7.2.3 Pager

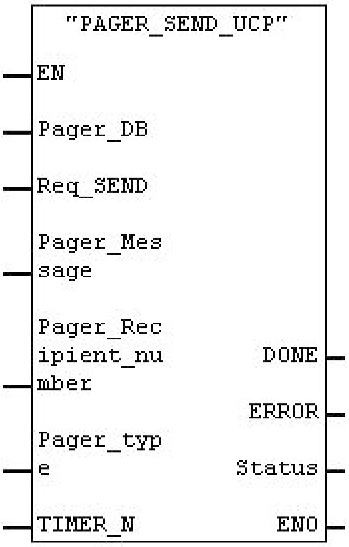
Parameter	Type	Description
Pager_Recipient_N1	STRING[16]	Phone number 1, only numeric characters.
Pager_Recipient_N2	STRING[16]	Phone number 2, only numeric characters.
Pager_Recipient_N3	STRING[16]	Phone number 3, only numeric characters.
Pager_Recipient_N4	STRING[16]	Phone number 4, only numeric characters.
Pager_Sender_ID	STRING[16]	Identification number of sender, only numeric
Wait_after_connection	S5TIME	Time to wait between the request to SEND Pager is made and the moment that the PCD sends the Message to the center. A certain time is needed, because both modem do not get connected in the same time.

Those are the only parameters to configure for the Pager system. Do not forget to initialize the Data Block after any modification.

7.3 Pager function

This function will interface the PCD to the Page Center. Sending the Message and receive back the answer for acknowledge or error.

In order to work correctly, the PCD need the connection to be established with the Page Center. The connection is done with the Calling Function (see chapter 3.2). When then connection is established the function Pager can be called. Once the function is called with the Req_SEND to 1. The function will wait a delay configured in “Wait_for_Connection“ (see 7.2.3) to be sure that the Page Center is also connected. Then The function send the Message to the Provider, the answer form the Provider will be returned on the output DONE for a successful transmission or on the output ERROR and Status if an error has been detected in the Pager configuration or message. If after a delay of 30 seconds the Page Center didn’t answer to our request the ERROR output will be set.



- The function affects the Data Block Register 1.
- Internally the function is working with the address register, to avoid any problem with some alarm interrupts, those one are disabled.

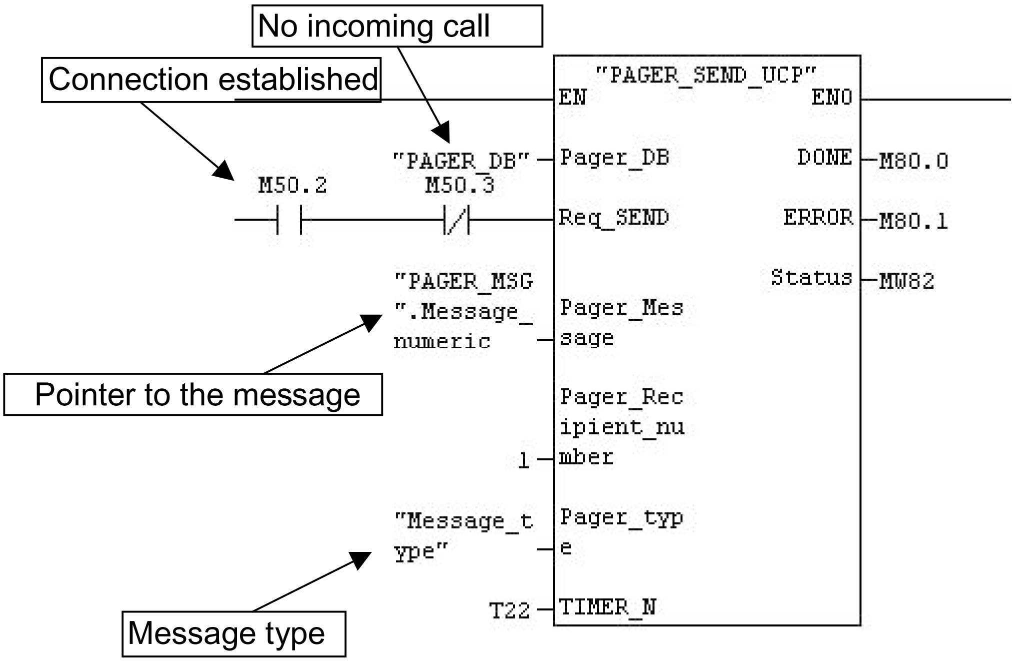
Parameter	Type	Description
Pager_DB	BLOCK_DB / IN	Number of the Pager_DB
Req_SEND	BOOL / IN	Request to send the Pager message, 1 will process the request, 0 will cancel.
Pager_Message	ANY / IN	Pointer to the string “Message“ (max. 140 char).
Pager_Recipient_num.	INT / IN	Number in the list of the Pager recipient
Pager_Type	BOOL / IN	0: for numeric message, 1: for Alphanumeric message
TIMER_N	TIMER / IN	Number of a TIMER reserved for the Pager function (needed internally). This TIMER can’t be used anymore.
DONE	BOOL / OUT	The message has been sent with success, stay at 1 until the Req_SEND is reset.
ERROR	BOOL / OUT	An error occurred during the transmission of the message, stay at 1 until the Req_SEND is reset.
Status	INT / OUT	Error code of the operation, output the code until the Req_SEND is reset.

7.3.1 Integration to the application

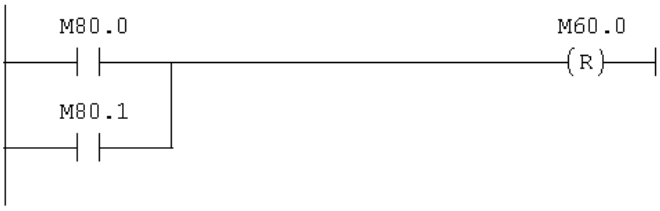
Integrate in your project the modem driver library (see chapter 5 to 5.3). But when doing the configuration of the **DB_Modem** you need to consider, which are the values for the parameters to use in conjunction to the Page Center. It means, what is the phone number of the Page Center (with necessary prefix), which is the baud rate supported by the Page Center.

1. Integrate in your project the Pager function FC600,DB600 and UDT600.
2. Configure the **Pager_DB** (see 7.2.2 and 7.2.3).
3. In Data Blocks, prepare the message in a string format and not longer than 140 characters (maximum is usually less, depends on the provider).
4. Call the function FC600 in a such way that:
    - Make the request to send a message, when the connection has been established, for this use the signal CONNECT from the modem Driver.
    - Cancel the Calling request when the SMS function return an signal **DONE** or **ERROR**.

Call of the function



Use the return signals to disconnect



You can find a similar examples in the modem library.

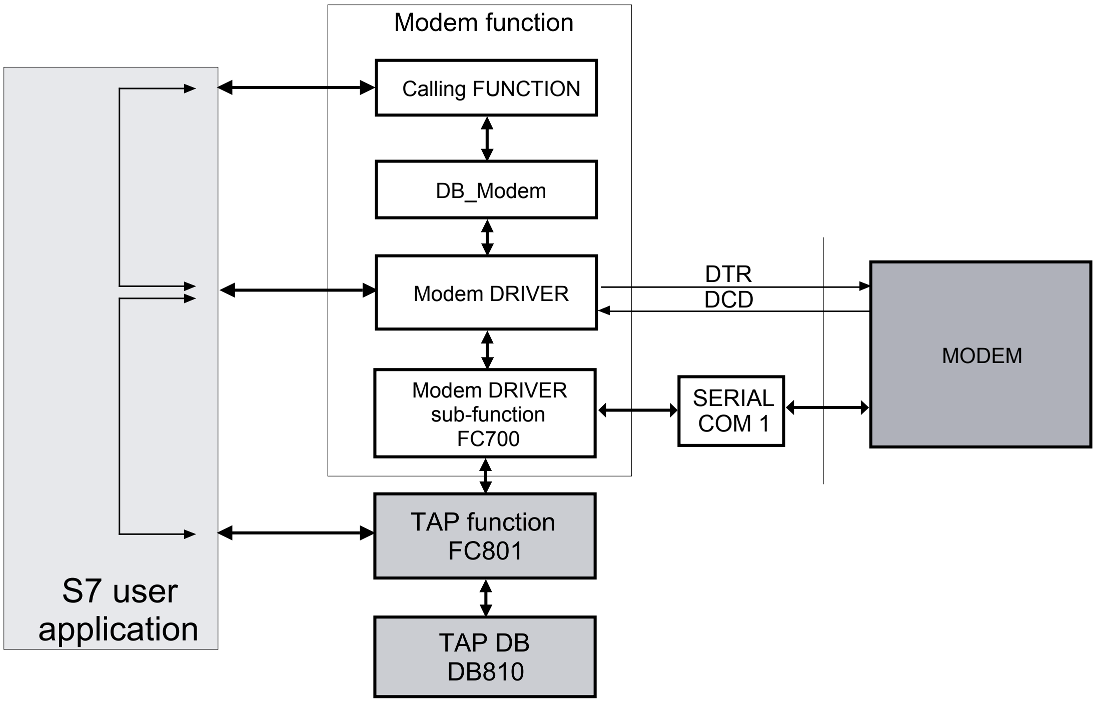
8 Sending SMS-TAP

The TAP function of the library gives the possibility to the system to send Short Messages. This can be very useful for the system to signal an event or alarm to the maintenance staff.

The protocol used for this SMS function is the TAP and the function allows sending Messages with a maximum of 250 characters. Some TAP-Service provider allows a smaller maximum of characters to send.

8.1 Structure

The TAP function has to be used in conjunction with the Modem function and Driver. In fact you need the Modem function to establish the connection with the TAP center and then the TAP function can be called.



The TAP function is associated with an TAP Data Block. The Data Block is divided in two areas. One is reserved for internal use of the function and the other area contains the information about the application. The TAP function also need to access the sub-function of the FC700 (Modem Driver sub-function). The synchronization between the Modem Driver and the TAP function is made by the S7-application.

8.2 Parameterization of the TAP system

Most of the TAP system parameterization is done in the TAP Data Block. In the library its number is the DB810 **“TAP_DB”**. But there is also one parameter that have to be set in the modem Data Block **“DB_Modem”**. The number of the TAP-center has to be entered in the modem Data Block.

8.2.1 Parameter in the “DB_Modem”

As the Modem Driver and the Calling function are establishing and handling the connection with the TAP Center, one parameter has to be adjusted in the DB_Modem. The parameter to adjust is the **DIAL_Number**. In fact the Calling function will call the TAP Center and then when the connection is established, the TAP function will be called. So the DIAL_Number of the TAP Center has to be inserted in the DB_Modem (for details see Chapter 2.1.2). Here are some numbers of TAP SMS Centers and their providers. For other countries, you can obtain the phone number of the TAP Center by your local provider.

Phone numbers for Analogue modems:

Country	Provider	Phone number	Baudrate	Comm settings
Germany	DeTeMobil	0171 25 21 002	1200	8N1
Germany	e-plus	0117 11 67	9600	8N1
Germany	VIAG-Interkom	0179 76 73 425	9600	8N1
Austria	Telekom Austria	0900 664 914	9600	7E1

Make sure that the protocol is TAP.

Phone numbers for ISDN modem:

Country	Provider	Phone numbe	Protocol	Baudrate	Comm settings
Germany	DeTeMobil	0171 25 21 001	X.75	1200	8N1

8.2.2 TAP SMS system parameters “DB_TAP”

This DB is divided in two parts:

- The first part is created by the UDT801 and the purpose of this area is a working space for the TAP function. **Its content doesn’t have to be edited or modified!**
- The second part contains all the parameter of the SMS system.

TAP

Parameter	Type	Description
SMS_Recipient_N1	STRING[16]	Phone number 1, only numeric characters.
SMS_Recipient_N2	STRING[16]	Phone number 2, only numeric characters.
SMS_Recipient_N3	STRING[16]	Phone number 3, only numeric characters.
SMS_Recipient_N4	STRING[16]	Phone number 4, only numeric characters.
SMS_Recipient_N5	STRING[16]	Phone number 5, only numeric characters.
Wait_after_connection	S5TIME	Time to wait between the request to SEND SMS is made and the moment that the PCD sends the SMS to the TAP center. A certain time is needed, because both modem do not get connected in the same time.
Timeout_1 Timeout_2 Timeout_3 Timeout_4 Timeout_5	S5TIME	This times are different for certain TAP Service centers.

Those are the only parameters to configure for the TAP system. Do not forget to initialize the Data Block after any modification.

The following table lists the different timeout value for the different TAP centre.

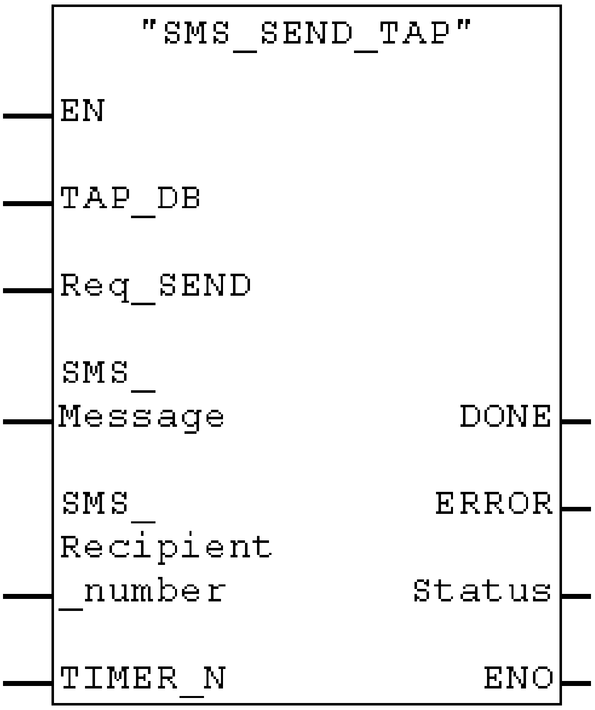
Parameter	Values für VIAG-Interkomm	Values for other TAP centre
Wait_after_connection	10s	10s
Timeout_1	2s	2s
Timeout_2	50s	10s
Timeout_3	10s	10s
Timeout_4	50s	2s
Timeout_5	2s	2s

8.3 TAP function

This function will interface the PCD to the TAP Center, sending the Message and receive back the answer for acknowledge or error.

In order to work correctly, the PCD needs the connection to be established with the TAP Center. The connection is done with the Calling Function (see chapter 3.2). When then connection is established the function TAP can be called.

Once the function is called with the **Req_SEND** =1, the function will wait a delay configured “Wait_for_Connection“ (see 8.2.2) to be sure that the TAP Center is also connected. Then the function sends the SMS message to the TAP Center, the answer will be return on the output **DONE** for a successful transmission or on the output **ERROR** and Status if an error has been detected in the SMS configuration or message. If after a delay of 30 seconds the TAP Center didn’t answer to our request the **ERROR** output will be set.



- The function affects the Data Block Register 1.
- Internally the function is working with the address register, to avoid any problem with some alarm interrupts, those one are disabled.

Parameter	Type	Description
TAP_DB	BLOCK_DB / IN	Number of the TAP_DB
Req_SEND	BOOL / IN	Request to send the SMS, 1 will process the request, 0 will cancel.
SMS_Message	ANY / IN	Pointer to the string “Message“[1)]
SMS_Recipient_num.	INT / IN	Number in the list of the SMS recipient
TIMER_N	TIMER / IN	Number of a TIMER reserved for the SMS function (needed internally). TIMER can’t be used anymore.
DONE	BOOL / OUT	The SMS has been sent with success, stay at 1 until the Req_SEND is reset.
ERROR	BOOL / OUT	An error occurred during the transmission of the SMS, stay at 1 until the Req_SEND is reset.
Status	INT / OUT	Error code of the operation, output the code until the Req_SEND is reset.

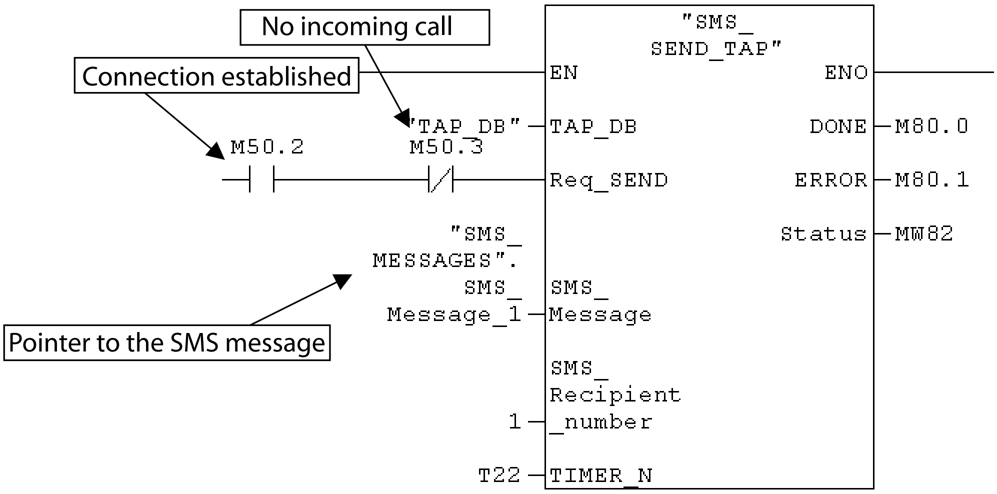
[1)] The total length of SMS_Message and SMS_Recipient_num can be 250 characters. Some TAP centre allows a maximum length of 160 characters.

8.3.1 Integration to the application

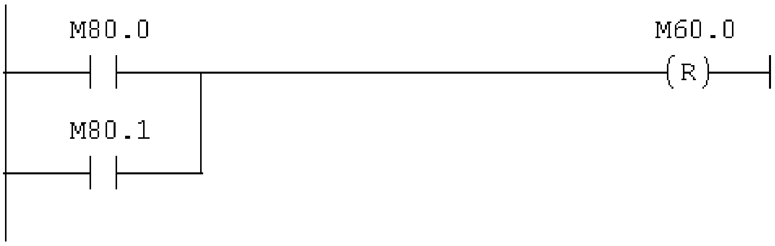
Integrate in your project the modem driver library (see chapter 5 to 5.3). But when doing the configuration of the **DB_Modem** you need to consider which are the value for the parameters to use in conjunction to the TAP Center. It means, what is the phone number of the TAP Center (with necessary prefix), which is the baud rate supported by the TAP- Center.

1. Integrate in your project the TAP function FC801,DB810 and UDT801.
2. Configure the **DB_TAP** (see 8.2.2 and 8.2.3).
3. In Data Blocks, prepare the SMS message in a string format and not longer than 250 characters (incl. SMS_Reciipient_num).
4. Call the function FC801 in such a way that:
   - Make the request to send SMS, when the connection has been established. For this use the signal **CONNECT** from the modem Driver.
   - Cancel the Calling request when the TAP function return an signal **DONE** or **ERROR**.

Call of the function



Use the return signals to disconnect



You can find a similar examples in the modem library.

9 Appendix

9.1 Appendix 1, Initialization strings

In this annex you find the initialization string for the modem tested and most often use on the PCD2 xx7 serie.

Modem PCD2.T813(z42), analog (Can be find in DB104)

Function	Field in DB	String
Reset modem	RESET_COMMAND	‚ATZ$R‘
First init command	INIT_COMMAND_1	‚ATE0V1X3+A8E=,,,0$R‘
2th init command	INIT_COMMAND_2	‚AT%C0S0=3$R‘
Hangup command		‚ATH0$R‘
Break mode	MODE_COMMAND	‚+++‘
ZVZ		20

The initialization parameter “+A8E=,,,0“ (Switch off the V8bis) is only necessary for the PCD2.T813z42 but doesn’t harm the functionality of the PCD2.T813.

Modem PCD2.T850, ISDN (Can be find in DB102)

Function	Field in DB	String
Reset modem	RESET_COMMAND	‚ATZ$R‘
First init command	INIT_COMMAND_1	‚AT&FE0V1S0=1&D2B10#Z=*$R‘
2th init command	INIT_COMMAND_2	‚AT&W$R‘
Hangup command		‚ATH$R‘
Break mode	MODE_COMMAND	‚+++‘
ZVZ		20

Parameter Description First init command:

S0=x Set Auto answer Mode
x=0: Auto answer Mode off
x=1: Immediate call acceptance by the terminal adapter (default)
x=2..n: Call acceptance through the terminal adapter after *n* “RING” messages.
Bxx B channel protocol

Transmission protocol for data communication in the B channel:

ATB0: V.110 asynchronous ex. for Pager CH (Phone Number: 074 090 01 03)
ATB3: HDLC async to sync conversion (PPP asynchronous)
ATB4: HDLC transparent (octets are packed into HDLC frames)
ATB5: Byte transparent (raw B channel data)
ATB10: X.75-NL ex. Pager CH (Phone Number 074 090 01 08)
ATB13: V.120 ex. for SMS CH
ATB20: X.31 B channel (X.25 B channel)
ATB21: X.31 D channel
ATB22: T.70-NL-CEPT
ATB23: T.90-NL
ATB31: MLPPP

Each protocol could be used, but use on both modem the same.
#Z=nn Define own msn
If the number is set to “*” (default), all incoming calls are acceptable.
AT#Z=*nn* set msn to *nn*

9.2 Appendix 2, Error Codes

9.2.1 Error returned by the Modem Driver function

Error during initialization

Error number	**Description**
-101	Not valid COM port number
-102	Not enough S7 memory to create the Buffer, but will be possible if you Compress the S7 memory.
-103	Not enough S7 memory to create the Buffer, even if you compress the Compress the S7 memory.
-105	Not valid interface parameters (BaudRate, DataBit, StopBit or Parity)
-108	Not valid value in ZVZ or QVZ Parameter
-111	Not valid value in SENDBuffer or RCVBuffer Parameter
-112	The total memory of the RCVBuffer and the SENDBuffer is bigger than the 64k Bytes allowed
-113	The SFC was called with the sum of the parameter RCVBuffer and the SENDBuffer different from the first call of the SFC.

Error in Receiving

Error number	**Description**
-396	Length too long compare to the buffer size
-401	Invalid com port
-402	COM port not initialized
-403	Invalid DB number
-404	DB not loaded

Error in Sending

Error number	Description
-304	DB is not loaded
-305	Invalid length
-303	Invalid DB number
-289	Can’t execute function, because already being executed

Error during Init modem or Reset modem

Error number	**Description**
-270	Modem didn’t answer during the preset time
-275	Modem returned an answer which could not be identified
-303	Invalid DB number
-289	Can’t execute function, because already being executed

In all this case, the ERROR bit will be set and the error code will be returned through the RET_VAL parameter. The driver will not be initialized correctly and can’t be operated. Parameters have to be corrected and a restart of the driver will be necessary.
To restart the driver the flag START_FLAG has to be reset.

9.2.2 Error return by the calling function

Status	Error	Description
3	0	An incoming call is actually processed
8	1	BUSY
16	1	No carrier
32	1	No DIAL TONE
10	1	Black listed (can’t process this number)

When a telephone number. is black listed this number automatically will be set free again after 2 hours. A command for switch off black listing is not possible due to telecommunication restrictions.

9.2.3 Error return by the SMS function

Status	Error	Description
-1 (FFFFh)	1	TIME out, the SMS center didn’t answer to the SMS request in the 30 seconds.
1	1	Check sum error (internal)
2	1	Syntax Error
3	1	Operation not supported by the system
4	1	Operation not allowed at this moment
5	1	Call barring active
6	1	Recipient number invalid
7	1	Authentication failure
8	1	Legitimization code for all calls, failure
21	1	Standard text not valid
24	1	Message too long

9.2.4 Error return by the PAGER function

Status	Error	Description
-1 (FFFFh)	1	TIME out, the SMS center didn’t answer to the SMS request in the 30 seconds.
1	1	Check sum error (internal)
2	1	Syntax Error
3	1	Operation not supported by the system
4	1	Operation not allowed at this moment
5	1	Call barring active
6	1	Recipient number invalid
7	1	Authentication failure
8	1	Legitimization code for all calls, failure
21	1	Standard text not valid
24	1	Message too long
701	1	Message type numeric has unsupported characters

9.2.5 Error return by the TAP SMS function

Status	Error	Description
-1 (FFFFh)	1	TIME out, the SMS center didn't answer to the SMS request in the 30 seconds.
1	1	Check sum error (internal)
4	1	Operation not allowed at this moment
6	1	Recipient number invalid
24	1	Message too long

9.3 Appendix 3, Pager character set

For numeric messages some special characters are allowed, the following table shows the allowed characters and their correspondence in the S7 string.

This table is valid for the provider SWISSCOM, service TELEPAGE® swiss and private.

Character in the S7 string	Character received on the Pager
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
B	U
C	'SPACE'
D	-
E	]
F	[

Example:

If the S7 string is: "**079D8008001**", the pager will receive
the message "**079-8008001**".

9.4 Appendix 4, General

9.4.1 Icons

i	In manuals, this symbol refers the reader to further information in this manual or other manuals or technical information documents. As a rule there is no direct link to such documents.
⚡	This symbol warns the reader of the risk to components from electrostatic discharges caused by touch. **Recommendation:** at least touch the Minus of the system (cabinet of PGU connector) before coming in contact with the electronic parts. Better is to use a grounding wrist strap with its cable attached to the Minus of the system.
!	This sign accompanies instructions that must always be followed.
Classic	Explanations beside this sign are valid only for the Saia PCD Classic serie.
xx7	Explanations beside this sign are valid only for the Saia PCD xx7 serie.

9.4.2 Address

Saia-Burgess Controls AG

Bahnhofstrasse 18
CH-3280 Murten / Switzerland

Telephone	+41 26 672 72 72
Telefax	+41 26 672 74 99
E-mail:	support@saia-pcd.com
Homepage:	www.saia-pcd.com
Support:	www.sbc-support.com