Library for GSM

TXV 003 40.01
second edition
June 2008
subject to alterations

Modifications history

Date	Publication	Modification description
February 2008	1	First version
June 2008	2	Supplementation of modifications for library versions 1.1 to 1.4

Content

1 Library for GSM

GSMLIB.mlb library contains the support for usage of the GSM gate GSM2-01. In the version 1.0 it enables in particular the reception and sending of short text messages SMS.

1.1 Types

GSMLIB.mlb library defines the following types of variables:

Type	*Description*	*Basic type*
NUMBER_STRING	Text string of the 20 characters lenght for phone number entry	STRING[20]
PIN_STRING	Text string of the 4 characters lenght for SIM card PIN entry	STRING[4]
SMS_STRING	Text string of the 160 characters lenght for SMS message text	STRING[160]
TARRAY255	The array for transmission of UNI channel communication zones	ARRAY [1..255] OF USINT
TGSM_UNI	The structure for connection of the block to the communication channel in the UNI mode	STRUCT
TGSMGateStateOut	Communication status with the GSM gate	ENUM
TGSMGateError	Enumeration of possible errors occuring during communication with the GSM gate	ENUM

The relevance of enumeration values:

TGSMGateStateOut – Communication status with the modem		
0	ggso_Init	Echo off
1	ggso_Pin	PIN status detection
2	ggso_PinValue	PIN value emission
3	ggso_Conf1	Error format settings
4	ggso_Conf2	SMS format settings
5	ggso_Conf3	New SMS notification settings
6	ggso_Conf4	SMS storage settings – step 1
7	ggso_Conf5	SMS storage settings – step 2
8	ggso_Center	SMS centre settings
9	ggso_EraseQuery	Request on deleting
10	ggso_Erase	SMS deleting

TGSMGateStateOut – Communication status with the modem		
11	ggso_EraseOne	One SMS deleting
12	ggso_ReadQuery	Request on reading
13	ggso_Read	SMS reading
14	ggso_ReadOk	Waiting for a read receipt
15	ggso_SendText	SMS text sending
16	ggso_Send	SMS sending initialisation
17	ggso_Ussd	Sending of a command in the form of the phone number
18	ggso_SignalQ	Signal quality determination
19	ggso_RingNumber	Incoming call number determination
20	ggso_Ringing	Number dialing (from the version 1.4)
21	ggso_Disconnected	Dialing termination (from the version 1.4)
22	ggso_Error	Communication error

TGSMGateError – Error specification within communication with the modem		
0	gger_None	Errorless
1	gger_No_Pin	PIN not entered
2	gger_Pin_Error	PIN entry error
3	gger_Puk_Required	PUK entry required (from the version 1.1)
4	gger_Cfg_Error	Configuration error
5	gger_Erase_Failed	Deleting failed
6	gger_SMS_Center_Error	Setting of SMS centre number error
7	gger_No_Center_Number	SMS centre number missing
8	gger_No_Recipient_Number	The recipient number of outgoing SMS not set
9	gger_Sending_failed	SMS sending failed
10	gger_Receiving_failed	SMS reception failed
11	gger_Receiving_OK_Missing	SMS reception not confirmed
12	gger_Channel_error	Communication channel indicates an error

1.2 Function blocks

1.2.1 SMS_Handler

SMS_Handler is a function block that performs communication with the GSM gate. The function block works above the structure of the communication PLC channel which must be in the UNI mode with the following parameters:

- Receiving zone lenght 360 bytes
- Sending zone lenght 360 bytes
- Communication rate 9600 bauds
- Data format 8 bits without a parity
- Maximum message lenght 360
- Minimum idle period on the line between received messages 5 bytes
- Minimum idle period on the line between sent messages 40 bytes

To ensure the correct operation, it is necessary that there is for each communication channel only one instance of the function block which will be called just once during the PLC cycle.

The interlacing of the function block with the communication channel is realized by the pair of variables *CH_IN* and *CH_OUT* of the class VAR_IN_OUT. There must be into the variable *CH_IN* assigned a channel zone *UNI_CHx_IN* and into the *CH_OUT* the zone *UNI_CHx_OUT* where *x* represents the number of the correspondent channel in the UNI mode. With regard to the fact that variables are not of an identical type, it is vital during the assignment to use the construction with the key word *VOID,* see the example bellow.

During the initialisation of the function block (in progress after the PLC restart, communication error with the gate or within the entering edge on the output *Reset*) the initialisation of the modem is performed whereat the SIM card PIN is sent (if required) that was loaded from the variable on the input *Pin* and the number of the SMS messages centre from the variable on the input *SMSCentre* is entered. If the input *Delete* is during the initialisation set to the value *true*, then when necessary all SMS saved on the SIM card are deleted subsequently. After the initialisation the block indicates the readiness for reception and sending of messages by setting the output *Ready* to the value *true.* The number of the message centre must be presented in the international format.

Operator	Value *SMSCentre*
O2	'+420602909909'
T-mobile	'+420603052000'
Vodafone	'+420608005681'

SMS sending is undertaken by the entering edge induction onto the input *Send*. If the block output *Ready* is not set to the value *true*, the sending is delayed till this input is set. When more requests is received during this period, only the last one is settled. During the request processing the input *SendPending* is set to *true.* Within the sending itself the output *Ready* is set to the value *false*.

SMS message is sent to the number from the variable on the input *Recipient* with the text from the variable on the input *MessToSend*.

SMS message reception is indicated by the setting of the output *NewMess* to the value *true* for the duration of one cycle. The text of the received message is entered into the variable on the input *RecvMess*, the number of the sender into the variable *Sender* and time of the message delivery to the SMS centre into the variable *RecvTime*.

The block also supports the sending of Unstructured Supplemetary Services Data (USSD) where the command is sent as the phone number. This service can be used for reading of the credit amount of prepaid SIM card or for its possible top-ups.

Operator	Value *Recipient* for credit amount reading
O2	'*104*#'
T-mobile	'*101#'
Vodafone	'*22#'

The command sending is undertaken by the entering edge induction onto the input *Ussd*, the string from the variable on the input *Recipient* is used as a command. During the request processing the output *UssdPending* is set to *true.* Rules for the output *Ready* are similar to the SMS sending.

The answer to the USSD is indicated by the setting of the output *NewMess* for the duration of one cycle. The received answer is saved in the variable on the input *RecvMess.* The variable on the input *Sender* is set according to the command used.

If the dialling of the GSM gate number occurs, the incoming call is indicated by the setting of the output *Ring* to *true* and the number of the caller is saved in the variable on the input *Caller.*

The output *Signal* states the strenght of the GSM signal in per cent units. The value -1 signalizes that the level of the signal was not yet determined or that it is not possible to detect it.

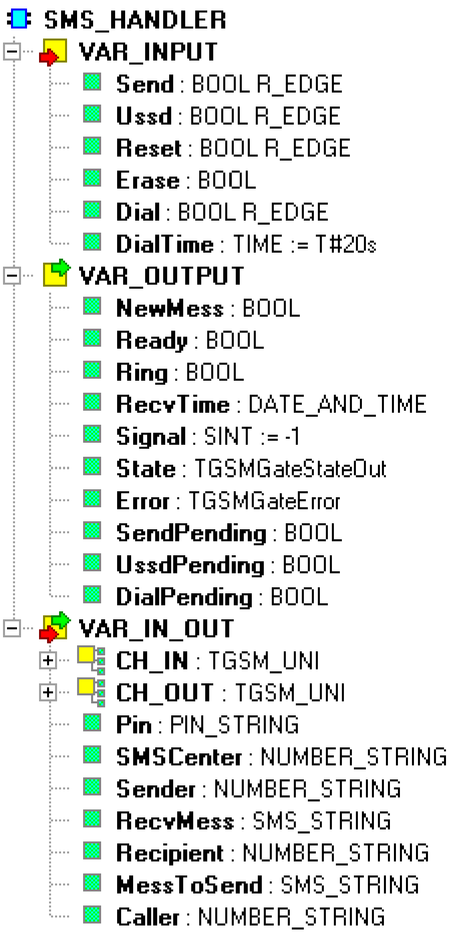
The output *State* indicates the communication status with the modem.

The output *Error* indicates difficulties within communication with the modem. Since the version of the library 1.1, there is, for the further identification of the problem, to the variable on the input *RecvMess* entered the error message from the GSM gate, if available.

Since the version of the library 1.1, the function block does not automatically undertake further attempts on initialisation of the GSM gate when the gate refuses the entered PIN (*gger_Pin_Error*) or it notifies the request on the PUK entry (*gger_Puk_Required*). The initialisation is, in such case, accomplished again only onto the entering edge on the input *Reset*.

The version 1.2 approachs the GSM gate slower which has the positive impact on the longterm stability. Moreover, the modem text report, that specifies the failure cause, is returned when an error within the variable on the input *RecvMess* occurs.

Since the version of the library 1.4, a new function „SMS recipient dialling“ is available. The function is activated by the edge on the input *Dial*. The dialling is activated only when the output *Ready* has the value *true.* The modem dials the number and after lapse of time *DialTime* or if someone accept the call, the modem itself hangs up. During the dialling itself the output *Ready* is set to the value *false* and the flag *DialPending* has the value *true*.



- **SMS_HANDLER**
  - **VAR_INPUT**
    - **Send** : BOOL R_EDGE
    - **Ussd** : BOOL R_EDGE
    - **Reset** : BOOL R_EDGE
    - **Erase** : BOOL
    - **Dial** : BOOL R_EDGE
    - **DialTime** : TIME := T#20s
  - **VAR_OUTPUT**
    - **NewMess** : BOOL
    - **Ready** : BOOL
    - **Ring** : BOOL
    - **RecvTime** : DATE_AND_TIME
    - **Signal** : SINT := -1
    - **State** : TGSMGateStateOut
    - **Error** : TGSMGateError
    - **SendPending** : BOOL
    - **UssdPending** : BOOL
    - **DialPending** : BOOL
  - **VAR_IN_OUT**
    - **CH_IN** : TGSM_UNI
    - **CH_OUT** : TGSM_UNI
    - **Pin** : PIN_STRING
    - **SMSCenter** : NUMBER_STRING
    - **Sender** : NUMBER_STRING
    - **RecvMess** : SMS_STRING
    - **Recipient** : NUMBER_STRING
    - **MessToSend** : SMS_STRING
    - **Caller** : NUMBER_STRING

Fig. 1 The structure of the function block SMS_Handler

Variable description:

	Variable	***Type***	***Signification***
VAR_INPUT			
➡	*Send*	BOOL R_EDGE	Send SMS onto the entering edge
➡	*Ussd*	BOOL R_EDGE	Send the cammant as a phone number onto the entering edge
➡	*Reset*	BOOL R_EDGE	Initialize the GSM gate onto the entering edge
➡	*Erase*	BOOL	Delete SMS saved on the SIM card during the modem initialisation
➡	*Dial*	BOOL	SMS recipient dialling
➡	*DialTime*	BOOL	Dialling period
VAR_OUTPUT			
↗	*NewMess*	BOOL	New SMS received
↗	*Ready*	BOOL	GSM gate is ready for reception and sending
↗	*Ring*	BOOL	Incoming call indication
↗	*RecvTime*	DATE_AND_TIME	Time of SMS reception
↗	*Signal*	SINT	The signal strenght in per cents. Value -1 indicates the unknown level of the signal.
↗	*State*	TGSMGateStateOut	Communication status with the modem
↗	*Error*	TGSMGateError	Error specification within communication with the modem
↗	*SendPending*	BOOL	SMS sending in process
↗	*UssdPending*	BOOL	The sending of the command as the phone number in process
↗	*DialPending*	BOOL	Phone number dialling in process
VAR_IN_OUT			
⇄	*CH_IN*	TGSM_UNI	Input communication zone of the UNI channel
⇄	*CH_OUT*	TGSM_UNI	Output communication zone of the UNI channel

	Variable	***Type***	***Signification***
⇄	*Pin*	PIN_STRING	SIM card PIN
⇄	*SMSCenter*	NUMBER_STRING	SMS message centre number
⇄	*Sender*	NUMBER_STRING	The phone number of the sender of the received SMS
⇄	*RecvMess*	SMS_STRING	SMS message text
⇄	*Recipient*	NUMBER_STRING	The phone number of the recipient of the message to be sent
⇄	*MessToSend*	SMS_STRING	Text of SMS to be sent
⇄	*Caller*	NUMBER_STRING	The phone number of the caller

Example of calling:

```
VAR_GLOBAL
 g_SendText  : SMS_STRING := 'Hello world!';
 g_RecvText  : SMS_STRING;
 g_Recipient : NUMBER_STRING := '+420608511845';
 g_Sender    : NUMBER_STRING;
 g_Caller    : NUMBER_STRING;
 g_Center    : NUMBER_STRING := '+420602909909';
 g_Pin       : PIN_STRING := '1234';
END_VAR

PROGRAM prgMain
 VAR
  iSMS : SMS_HANDLER;
 END_VAR

 iSMS(CH_IN := void(UNI_CH1_IN), CH_OUT := void(UNI_CH1_OUT),
    Pin := g_Pin, SMSCenter := g_Center, Sender := g_Sender,
    RecvMess := g_RecvText, Recipient := g_Recipient,
    MessToSend := g_SendText, Caller := g_Caller);

END_PROGRAM
```