How to handle SMS messages

This section gives examples of the AT commands for sending and receiving SMS (Small Message Service) messages with the GM12. First we will discuss what an SMS is, and some details on how an SMS message is sent in the GSM network.

The Short Message Service (SMS) allows a user to send up to 140 bytes of information from one mobile station (MS) to another, using the receiving MS phone number as the destination address. What happens with this information when it arrives at the destination is dependent on the receiving station. For instance, if an SMS is sent from one mobile telephone to another, an SMS usually takes the form of a simple personal text message. If the receiving mobile station is a GSM module like the GM12, then maybe the received information is some kind of Data which will be processed by an application.

Whatever the destination of the SMS message, the message is ALWAYS sent via a Service Centre (SC). One of the SC’s jobs, among others, is to forward SMS messages to the destination, or store the SMS until the receiving MS becomes available in the GSM network. The message can be stored at the SC up to a maximum time period decided by the sender. Also the message is “time stamped” at the SC and its contents are interpreted according to choices made by the sender.

The conclusion of this is that an SMS message, along with other information decided upon by the sender, always contains 2 addresses. First the Service Centre Address (SCA), and then the destination address. Also, other useful information is added at the SC.

Setting Service Centre Address

The service centre address is usually stored on the SIM card, but it can be set manually using the AT command AT+CSCA (see the chapter using AT commands later in the manual). Note that the SCA must be given in the international phone number format, e.g. for the Telia service centre address in Sweden:

```
AT+CSCA=”+46705008999”, 145
```

When you press enter you should get OK.

If you are unsure of the SCA you should use then contact your GSM operator.

Format of an SMS message

It is important to realise that SMS messages sent with the GM12 are not sent in the same manner as with a normal mobile phone. The GM12 uses the PDU (Packet Data Unit) mode message format to send SMS, where as a normal mobile telephone uses text input, and menu choices to send SMS. The PDU format is described in detail in the GSM specifications GSM 03.40 and GSM 03.38.

What is PDU mode? First a PDU is a Service Centre Address (SCA) followed by a Transport Protocol Data Unit (TPDU). There are several formats of TPDU, but the GM12 only uses two forms:

SMS-SUBMIT TPDU	Outgoing SMS messages "submitted" to the SC from the originating MS.
SMS-DELIVER TPDU	Incoming SMS messages "delivered" by the SC to final destination MS.

Therefore we have the following general format of an SMS message:

PDU = SCA+TPDU

The TPDU format is a hexadecimal encoded binary format i.e. 2 hexadecimal digits represents a byte or an octet. In general, there is a header part of the TPDU, which contains the control information, and then follows user data, which can be any type of information (default 7-bit GSM alphabet, 8-bit ISO8859-1 alphabet, or even 8-bit data).

Let's start with some examples of PDU SMS messages using an SMS-SUBMIT type TPDU. For simplicity's sake we will send a text message which reads "TEST", using the default GSM 7-bit alphabet.

Here is the message written out in full:

```
0011000B916407861582F50000A704D4E2940A
```

Equivalently, we could write this as

```
07916407058099F911000B916407861582F50000A704D4
E2940A
```

These two have one difference and that is how we have chosen to describe the SCA. In the first SMS, we use:

00

This is a valid SCA with a length equal to zero. Therefore, we are using the default SCA given by the SIM card, or as given by the AT command, AT+CSCA, described above.

07916407058099F9

Here is the SCA for Telia, Sweden again, but now represented in the PDU format. Remember, this is now hex encoded binary. The first octet (07) is a length indicator, in this case there are 7 octets to follow. The second octet (91) tells us about the numbering plan and type of number of the coming address. In this case the 91 says that the address is in international phone number format. Then follows the actual service centre address (6407058099F9). It doesn’t look like a phone number does it?….until you realise that every pair of digits is swapped, and, also, that there is an odd number of digits in the phone number and, therefore, a half byte of fill digits is required at the end (hence the hexadecimal F9). The exact meaning of all the octets and fields in the SCA address is described in detail in GSM 03.40.

Lets analyse the message by breaking it down into its component parts:

11

This octet contains many 1-bi tand 2-bit fields which are described in detail in GSM 03.40. These include:

- TP-MTI – message type indicator
- TP-RD – More messages waiting indicator
- TP-VPF – Validity period format
- TP-SRR – Status report request flag

- TP-UDHI – User data header indicator flag
- TP-RP – Replay path setting flag

Please refer to GSM 03.40 for the exact meaning of these flags.

00	This octet is only associated with the 8-bit field TP-MR, and is basically a message reference number which can be chosen by the user. Here we have set the message reference number to zero (00HEX).
0B916407861582F5	This is the destination address (TP-DA). The format of the address field is given in detail in GSM 03.40. Again it is similar to the format of the SCA but now it refers to the phone number of the receiving MS. The first octet (0B) is a phone number length indicator, in this case the phone number of the destination MS consists of 11 digits. The second octet (91) is the numbering plan and type of number of the coming address. Again the address is in international phone number format. Then follows the mobile number, with every pair of digits in reverse order, and, because we have an odd number of digits a half octet of fill bits are required (hence the F5HEX).
00	This is the Protocol ID (TP-PID) which refers to a possible higher level protocol being used or indicates interworking with certain types of telematic devices. This can usually be left set to zero.

00	This is the Data coding scheme field (TP-DCS) which is described in detail in GSM 03.38. Basically this octet tells us what kind of user data is being sent. For instance, if we are using a 7-bit or 8-bit alphabet, or we are sending 8bit data. It also informs receiving MS what the class of the SMS message is, i.e. how to handle and store the incoming SMS message.
A7	This is where the validity period (TP-VP). The validity period is how long the SC will store the SMS, waiting for the receiving MS to come in contact with the GSM network. If the SMS is not received by the destination MS within this time then the message will be discarded. Again GSM 03.40 describes this field in detail.
04	TP-UDL. This is the length of the user data. It is given in septets, if TP-DCS describes the message as using the default 7-bit GSM alphabet, or octets otherwise.

D4E2940A	Finally we come to the user data (TP-UD) This is 4-septets long, defined above in TP-DCS and TP-UDL) The message reads “TEST” Note that one character is NOT represented by one octet, but one character is represented by a septet (7-bits) and, therefore, it can be quite complicated to decipher a GSM text SMS by looking at the PDU. GSM specification 03.38 describes in detail how the 7-bit GSM alphabet works and how to convert from the hexadecimal to the characters. You may know that one can send 160 characters in an SMS message. This only possible in 140 bytes using a 7-bit alphabet.

Before we finish we should also take a look at the TPDU of SMS-DELIVER type:

04	TP-MTI, TP-MMS, TP-SRI, TP-UDHI, TP-RP
0B916407861582F5	This is the originating address (TP-OA).
00	TP-PID
00	TP-DCS
99107021230400	This is the Service Centre time stamp. Again every pair of digits are reversed. This time stamp reads, in octets from left to right: the date 990107 (year, month, day), the time 123240 (hours, minutes, seconds). Finally the last octet is a reference for the difference in time between local time and GMT.
04	TP-UDL
D4E2940A	TP-UD

Sending SMS messages

Now we know how to construct an SMS message lets try to send a message using AT commands.

The AT command to use is AT+CMGS. The total number of octets in the TPDU is given as a parameter for the command. Notice that this is the length of the TPDU and not the PDU so it doesn’t include the SCA.

Example 1 to default SCA and to destination address 46706851285. User data “TEST” (underlined)

```
AT+CMGS=18 <CR>

>0011000B916407861582F50000A704D4E2940A
          <CTRL-Z>

+CMGS: 122

OK
```

The response string contains a message reference number (MR), in this case 122.

Here pressing Ctrl-Z terminates the message, or alternatively the process can be aborted by pressing Esc.

Notice also that we used the default SCA in the PDU (00Hex) entered using the AT command AT+CSCA instead of entering it manually in the PDU.

Example 2 to SCA 4670508999 and to destination address 0706876902 7060786620. User data “This as a PDU message” (underlined)

```
at+cmgs=32

> 07916407058099F911000A8170607866200000A7
1354747A0E4ACF416110945805B5CBF379F85C06
<CTRL-Z>

+CMGS: 123

OK
```

Receiving SMS messages

SMS messages can be stored in the modules (ME, mobile equipment) memory, or on the SIM card (SM, SIM module). The GM12 can hold ten SMS messages in ME. The number that can be stored in the SIM depends on the operator and SIM card but it is usually higher that in ME.

In general, incoming SMS messages are normally stored automatically in ME, but this depends on other factors such as the Class of the incoming SMS message, and if the receiving module has been initialised using any AT commands which affect storage.

Setting the SMS message storage

Of the two physical memory storage places, there are three logical types of memory <mem1>, <mem2>, <mem3>:

<mem1> indicates which physical storage messages are read into or deleted from. Affected AT commands are +CMGR, +CMGL, and +CMGD.

<mem2> indicates which physical storage outgoing messages (written to memory by the user) are stored into. Affected AT commands are +CMGS, and +CMG

<mem3> indicates which physical storage incoming messages are stored into. For GM12 this is always set “ME”.

We can start by setting up the memory for incoming messages using the AT commands AT+CPMS

First we can query the module with this command

```
AT+CPMS?
```

+CPMS: "ME", 0,10	;this relates to <mem1> meaning we have zero messages of a possible ten.
+CPMS: "ME", 0,10	;this relates to <mem2> meaning we have zero messages of a possible ten.

```
+CPMS: "ME", 0,10                        ;this relates to
                                         <mem3> meaning
                                         we have zero
                                         messages of a
                                         possible  ten.


OK
```

Perhaps we wish to change <mem2> to the SIM card “SM”

```
AT+CPMS=”ME”,”SM”

+CPMS: 0,10,0,10,0,10

OK
```

Again the response shows that all the memory places are empty.

Now we send three SMS messages to the module and query the module again and we get the response:

```
AT+CPMS?

+CPMS: "ME", 3,10                        ;this relates to
                                         <mem1> meaning
                                         we have zero
                                         messages of a
                                         possible  ten.

+CPMS: "SM", 0,10                        ;this relates to
<mem2>

+CPMS: "ME", 3,10                        ;this relates to
<mem3>

OK
```

We see that “ME” now contains three messages.

Reading SMS messages from storage

To read SMS messages from storage, the AT commands AT+CMGL and AT+CMGR should be used. Both commands read messages from selected <mem1> storage.

So lets try and read the first message that we have sent to “ME”:

```
AT+CMGR=1

+CMGR: 1, ,23
```

```
07916407058099F9040B916407950303F1000089212221
40140004D4E2940A

OK
```

Here we see a SCA followed by a TPDU of the SMS-DELIVER type (try and decipher it yourselves). The status code is 1, indicating that it is a received and read message. The length of the message is 23 octets (the length of the TPDU and NOT the PDU). Note that the two commas, which reserve an empty field for a non-implemented parameter.

Alternatively to the AT+CMGR command we could list all the messages:

```
AT+CMGL=4                  ;the 4 here means "all
                           messages". See the AT
                           command list later in this
                           chapter for more details.

+CMGL: 1,1,23

07916407058099F9040B916407950303F1000089212221
40140004D4E2940A

+CMGL: 2,1,26

07916407058099F9040B916407950303F1000089212221
6000000841E19058341E91

+CMGL: 3,1,27

07916407058099F9040B916407950303F1000089212221
7064000941E19058341E9149

+CMGL: 4,1,23

07916407080007F8040B916407588735F3000089212221
5404400441E19008

OK
```

Deleting SMS messages

To delete an SMS message from the selected <mem1> use the AT+CMGD command.

```
AT+CMGD=1

OK
```

This will erase the SMS message at position 1. There is no function for erasing more than one message at a time.