



SMS and the PDU format

Introduction

The SMS message, as specified by the [Etsi](#) organization (documents GSM 03.40 and GSM 03.38), can be up to 160 characters long, where each character is 7 bits according to the [7-bit default alphabet](#). Eight-bit messages (max 140 characters) are usually not viewable by the phones as text messages; instead they are used for data in e.g. smart messaging (images and ringing tones) and [OTA provisioning of WAP settings](#). 16-bit messages (max 70 characters) are used for Unicode (UCS2) text messages, viewable by most phones. A 16-bit text message of class 0 will on some phones appear as a [Flash SMS](#) (aka blinking SMS or alert SMS).

The PDU format

There are two ways of sending and receiving SMS messages: by text mode and by PDU (protocol description unit) mode. The text mode (unavailable on some phones) is just an encoding of the bit stream represented by the PDU mode. Alphabets may differ and there are several encoding alternatives when displaying an SMS message. The most common options are "PCCP437", "PCDN", "8859-1", "IRA" and "GSM". These are all set by the [at-command](#) AT+CSCS, when you read the message in a computer application. If you read the message on your phone, the phone will choose a proper encoding. An application capable of reading incoming SMS messages, can thus use text mode or PDU mode. If text mode is used, the application is bound to (or limited by) the set of preset encoding options. In some cases, that's just not good enough. If PDU mode is used, any encoding can be implemented.

Receiving a message in the PDU mode

The PDU string contains not only the message, but also a lot of meta-information about the sender, his SMS service center, the time stamp etc. It is all in the form of hexa-decimal *octets* or decimal *semi-octets*. The following string is what I received on a [Nokia 6110](#) when sending the message containing "hellohello" from [www.mtn.co.za](#).

```
07 917283010010F5 040BC87238880900F10000993092516195800AE8329BFD4697D9EC37
```

This octet sequence consists of three parts: An initial octet indicating the length of the SMSC information ("07"), the SMSC information itself ("917283010010F5"), and the SMS_DELIVER part (specified by ETSI in GSM 03.40).

Note: on some phones (e.g. Ericsson 888?) the first three (colored) parts are omitted when showing the message in PDU mode!

| Octet(s) | Description |
|----------------------|---|
| 07 | Length of the SMSC information (in this case 7 octets) |
| 91 | Type-of-address of the SMSC. (91 means international format of the phone number) |
| 72 83 01 00 10 F5 | Service center number(in decimal semi-octets). The length of the phone number is odd (11), so a trailing F has been added to form proper octets. The phone number of this service center is "+27381000015". See below. |
| 04 | First octet of this SMS-DELIVER message. |
| 0B | Address-Length. Length of the sender number (0B hex = 11 dec) |
| C8 | Type-of-address of the sender number |
| 72 38 88 09 00 F1 | Sender number (decimal semi-octets), with a trailing F |
| 00 | TP-PID. Protocol identifier . |
| 00 | TP-DCS Data coding scheme |
| 99 30 92 51 61 95 80 | TP-SCTS. Time stamp (semi-octets) |
| 0A | TP-UDL. User data length, length of message. The TP-DCS field indicated 7-bit data, so the length here is the number of septets (10). If the TP-DCS field were set to indicate 8-bit data or Unicode, the length would be the number of octets (9). |
| E8329BFD4697D9EC37 | TP-UD. Message "hellohello" , 8-bit octets representing 7-bit data. |

All the octets above are hexa-decimal 8-bit octets, except the Service center number, the sender number and the timestamp; they are decimal semi-octets. The message part in the end of the PDU string consists of hexa-decimal 8-bit octets, but these octets represent 7-bit data (see below).

The semi-octets are decimal, and e.g. the sender number is obtained by performing internal swapping within the semi-octets from "72 38 88 09 00 F1" to "27 83 88 90 00 1F". The length of the phone number is odd, so a proper octet sequence cannot be formed by this number. This is the reason why the trailing F has been added. The time stamp, when parsed, equals "99 03 29 15 16 59 08", where the 6 first characters represent date, the following 6 represents time, and the last two represents time-zone related to GMT.

Interpreting 8-bit octets as 7-bit messages

This transformation is described in detail in GSM 03.38, and an [example of the "hellohello" transformation is shown here](#). The transformation is based on the [7 bit default alphabet](#) , but an application built on the PDU mode can use any character encoding.

Sending a message in the PDU mode

The following example shows how to send the message "hellohello" in the PDU mode from a Nokia 6110.

```
AT+CMGF=0      //Set PDU mode
AT+CSMS=0      //Check if modem supports SMS commands
AT+CMGS=23     //Send message, 23 octets (excluding the two initial zeros)
>0011000B916407281553F80000AA0AE8329BFD4697D9EC37<ctrl-z>
```

There are 23 octets in this message (46 'characters'). The first octet ("00") doesn't count, it is only an indicator of the length of the SMSC information supplied (0). The PDU string consists of the following:

| Octet(s) | Description |
|--------------------|--|
| 00 | Length of SMSC information. Here the length is 0, which means that the SMSC stored in the phone should be used. <i>Note: This octet is optional. On some phones this octet should be omitted! (Using the SMSC stored in phone is thus implicit)</i> |
| 11 | First octet of the SMS-SUBMIT message. |
| 00 | TP-Message-Reference. The "00" value here lets the phone set the message reference number itself. |
| 0B | Address-Length. Length of phone number (11) |
| 91 | Type-of-Address. (91 indicates international format of the phone number). |
| 6407281553F8 | The phone number in semi octets (46708251358). The length of the phone number is odd (11), therefore a trailing F has been added, as if the phone number were "46708251358F". Using the unknown format (i.e. the Type-of-Address 81 instead of 91) would yield the phone number octet sequence 7080523185 (0708251358). Note that this has the length 10 (A), which is even. |
| 00 | TP-PID. Protocol identifier |
| 00 | TP-DCS. Data coding scheme. This message is coded according to the 7bit default alphabet. Having "04" instead of "00" here, would indicate that the TP-User-Data field of this message should be interpreted as 8bit rather than 7bit (used in e.g. smart messaging, OTA provisioning etc). |
| AA | TP-Validity-Period. "AA" means 4 days. <i>Note: This octet is optional, see bits 4 and 3 of the first octet</i> |
| 0A | TP-User-Data-Length. Length of message. The TP-DCS field indicated 7-bit data, so the length here is the number of septets (10). If the TP-DCS field were set to 8-bit data or Unicode, the length would be the number of octets. |
| E8329BFD4697D9EC37 | TP-User-Data. These octets represent the message "hellohello". How to do the transformation from 7bit septets into octets is shown here |

Links

- [CMS error codes](#)
- [Versenden von Kurznachrichten mittels PDU](#)
- [Etsi](#)

This page has been visited [an error occurred while processing this directive] times and is written and maintained by Lars Pettersson (lars.pettersson@email.nu).