Enabling Future Computer Applications Using GSM Phones

Graeme J. Proudler
Network Systems Department
HP Laboratories Bristol
HPL-97-68
May, 1997

GSM,
At commands,
GSM 07.05,
GSM 07.07

This paper describes simple functions that are enabled by recent ETSI command interfaces for GSM mobile phones.

Enabling Future Computer Applications Using GSM Phones

G.J.Proudler, Hewlett Packard Laboratories (Bristol), Filton Road, Stoke Gifford, Bristol BS12 6QZ, UK.
Switchboard: +44 117 9799 910, Phone Direct: +44 117 9228 753, Fax: +44 117 9228924, email: gjp@hplb.hpl.hp.com

Abstract
This paper describes simple functions that are enabled by recent ETSI command interfaces for GSM mobile phones.

1 Introduction

Digital cellular radio systems can provide communication links for computers, but until recently there was no comprehensive way to control a mobile phone from a computer.

In 1994 a working group from Ericsson, Nokia and Hewlett Packard defined a set of commands for interaction between computers and digital cellular phones, specifically GSM phones. The commands were adapted in 1995 and adopted in 1996 by the European Telecommunications Standards Institute[1].

The commands are written in the 'AT' style, and allow the computer to find out what is happening at the phone, to control the phone functions normally controlled by a human user, and to control the sending and receiving of messages. This paper introduces the types of applications that are enabled by some commands.

Precise details and examples of all the commands are in the ETSI standards: GSM 07.05[2] deals with messaging while GSM 07.07[3] deals with handset identification, call control, network services, handset status, and control of a handset. Those standards are the definitive reference for any and all of the commands mentioned in this paper, and always take precedence over any statements in this paper. They should always be consulted to determine the precise definition of commands.

2 Applications using mobile phones

Applications can make it easier for the user to operate the radio or can themselves use the radio for mobile communications. An application needs to find out what is happening at the radio, control the functions normally controlled by a human user, send and receive data, send and receive fax, and send and receive short messages.

2.1 Phonebooks

Computer manufacturers want to encourage the use of laptop or palmtop computers, or Personal Digital Assistants, for storage of phonebook information. These devices generally have a friendly user interface for storing phone numbers, and can store other information asociated with a phone number (such as address and the context of the relationship to that individual). In the past a user had to read the desired phone number from the computer screen and type it into the phone. The usefulness of the computer would obviously be enhanced if the user could select a destination on the computer and an application automatically dialled the number on the phone.

Only a few commands are required to allow the phonebook in the computer to communicate with the phonebook in the phone: listing, reading, writing and deleting entries. However, the phone contains several different memories, and a user might want to store different parts of his telephone numbers in different locations. Personal numbers might be stored in the SIM, since that identifies the user, while business numbers might be stored in the phone *per se* if the user has a particular phone (and phone number) that he uses for work. All of these need to be taken into account.

2.2 Expanded user interface

Ergonomic details and cost usually restrict the size of screen and keypad on the phone. There are often too many functions to allow single-key-press access, so multiple key presses are used for less common functions, and the information on the display is often terse.

A computer, even a small one, usually has a bigger display and larger number of keys, and offers the potential of an easier phone interface if the phone controls and display can be emulated on the computer. This requires access to phone controls, to the phone display, and to other phone indicators not on the display. Then the computer can track events at the phone, learn keystrokes for less common functions, and even provide full control of the phone from the computer. In particular, an expanded user interface would make it easier to send and receive short textual messages over the phone.

2.3 Fax and data

If the phone supports fax or data communications, obviously the computer can use the phone as a modem.

2.4 Voice messaging

For most users, a phone is primarily a mechanism for voice communication. This functionality can be improved by the use of a computer. The radio transmitter, radio

receiver, phone microphone, phone loudspeaker, and computer memory can be interconnected in several ways. The computer can be used to record a conversation and to act as an answer phone, using the phone as both the communication means and the audio input and output.

3 Supporting Commands

3.1 Information about the phone

When an application starts, it must decide whether the phone can support the application. It must get the details of the phone, and possibly match those details to a database to determine the capabilities of the phone.

The first question is whether the phone supports a suitable command set. This can be done using the command **+GCAP complete capabilities** which is part of V.25ter. If the response includes **+CGSM** then the phone supports the ETSI command set. Of course, the phone may not support V.25ter, in which case +GCAP will not be recognised. Then the only option is to try a mandatory command from some command set and check for the correct response. Since control of phonebooks is likely to be a standard feature, a good choice would be a phonebook command. In the ETSI set, **+CPBS phonebook memory storage** is mandatory if phonebook read or write is supported, and would be a suitable choice.

Possibly the most important attribute of a phone is its phone number. This might be different according to the information class currently in use at the phone, because a phone might have a different number for voice calls, data calls, or fax calls. Also, the number might be in a local format (country specific, for example) or in full international format (ITU-T E.163). The ETSI command set uses **+CNUM subscriber number** to make the phone return its current phone number.

It may be important to know the phone manufacturer, or the model type, the revision of that model and its serial number. The ETSI set does this with a series of commands: **+CGMI manufacturer identification** returns a text string that a manufacturer can use to represent its name in whatever way it wishes. Similarly **+CGMM model identification** returns a text string that a manufacturer can use to represent the model type, **+CGMR revision identification** returns a text string for representing the revision of the model, and **+CGSN serial number** returns a text string for representing the serial number. All of these can be chosen without reference to an external authority.

3.2 Network Interaction

An application probably needs information about the cellular operators available to the phone, and may want to select certain options available in the network. It may choose to use a certain operator based on a knowledge of tarifs, or based on the degree of service or extent of service associated with particular operators. It may attempt to provide the lowest cost communications for the application, or select the best available data service, whatever. All of these depend on the exact nature of the application and the database incorporated into the application.

The ETSI command set uses **+COPS? operator selection** to provide information about network operators and their availabilities. The operator name can be written in long format alphnumeric, short format alphanumeric, or as a number. The status of available operators is returned by COPS=?, listing available operators and showing whether the operator is unknown, available, current or forbidden. The current operator is returned by +COPS?.

+CREG network registration shows whether the phone is 'connected' to the current operator. The state of the connection could be: not-registered and not attempting to register, registered at the home network, not registered but attempting to register, registration-denied, unknown, registered outside the home network. Obviously communications are possible only if the phone is registered with the operator. If this is not the case, the application must select another operator or present an error message to a user, and ask for the problem to be resolved. In that case the application might prompt with the telephone numbers of the service agents for a particular operator.

3.3 Configuring the channel

The network provided by the chosen operator needs to be configured.

First of all, the application may wish to select the *qos* characteristics of the protocol used over the radio channel. A reliable service provides a low error rate but indeterminate throughput and latency. An unreliable service provides an unknown error rate but fixed throughput and latency. The choice depends on the particular application. A real-time application that is tolerant of errors would use the unreliable service. A conventional data application would probably be intolerant of errors but latency would be of little concern, and hence the reliable service could be used. In the ETSI set, **+CRLP radio link protocol** allows the application to select either a reliable service or a non-reliable service.

If the phone supports data communications, the network must be notified of the required land-based service. This involves selecting a particular land-based service and the characteristics of that land-based service. In the ETSI set, **+CBST bearer service type** selects between asynchronous modem, synchronous modem, PAD, Packet Access, and the speed of the chosen service.

GSM in particular allows identification information to be presented to a user, and the ETSI set provides four commands to use that information. They allow each end of a communication link to enable or disable presentation of information about either end of the link. **+CLIP** controls whether 'calling number' information is presented to the phone when the phone accepts an incoming call, provided the caller has used **+CLIR** to allow such information to be presented. Similarly, **+COLP** controls whether 'called identity' information is presented to the phone when the phone makes an outgoing call.

Other commands can tell the network how to treat incoming calls. Arguably the most powerful are control of forwarding, depending on the current mode (data, voice, fax, etc.) of the phone. ETSI uses **+CCFC call forwarding number and conditions** to allow a distinct forwarding number for each combination of mode and situation. To use this command, the phone is put into the relevant information mode with +CSIC, and then +CCFC is repeatedly used with different values of reason and individual forwarding number. Calls can be forwarded because of various reasons, including mobile busy, no reply, not reachable. Alternatively **+CTFR call transfer** tells the network to transfer an existing call to a phone number included in the command syntax.

3.4 Configuring the phone

Probably the most important configuration decision is to dictate how the phone deals with unsolicited reporting of events. Should the phone pass such indications immediately or buffer them ? Which events should be reported to the application ? An application may decide to request immediate indications when the computer is switched on, but need to request buffering if the computer is about to be switched off (and is unable to wake up in response to external signals).

The ETSI command is **+CMER mobile event report**, and is complex. The command determines what happens to result codes caused by events at the phone keypad, the phone display, or the other phone indicators[1]. All result codes are treated in the same way, but reporting of individual classes of events can be enabled or not.

- The parameter 'buffer' determines what happens when +CMER is executed. Should any existing results (stored in a buffer) be cleared or should they be flushed to the application ? If the command does allow result codes to be passed to the application, the choice is:
  - clear existing result codes from the buffer.
  - flush existing result codes to the application.

[1] See the section 'Accessing phone resources' for more information about the keypad, display and indicators, and the commands to use them.

- The 'mode' parameter determines what happens to the result codes. The choice is:
  - buffer the result codes, so that they are stored for later access by the application. These buffered codes can be flushed to the application by a subsequent use of +CMER that sets the 'buffer' parameter to flush the buffer **and** does not select this 'mode' option.
  - pass result codes directly to the application, but throw away result codes that occur when the link between computer and phone is busy.
  - pass result codes directly to the application, but buffer result codes that occur when the link between computer and phone is busy and flush them when the link becomes free.
  - pass result codes directly to the application, but use the standard in-band embedded technique to pass result codes that occur when the link between computer and phone is busy.
- The 'keypad' parameter determines whether the pressing of buttons on the phone keypad should be reported. The names of the buttons are the same as used in the command +CKPD. The choice is:
  - don't report events at the keypad.
  - report only those keypad events that were not caused by the application itself. If any buttons are being pressed at the instant the +CMER command is activated, report those events.
  - report all keypad events, irrespective of their cause. If any buttons are being pressed at the instant the +CMER command is activated, report those events.
- The 'display' parameter determines whether changes on the phone display should be reported. The position of the change on the display is shown by a number which is that of the appropriate display element in a list starting from top left of the display through to the bottom right. This is the convention as used by the command +CDIS. The choice is:
  - don't report changes at the display.
  - report only those display changes that were not caused by the application itself.
  - report all display changes, irrespective of their cause.
- The 'indicator' parameter determines whether changes on the phone indicators should be reported. The particular indicator is identified by a number which is the order of the indicator in the list produced by the command +CIND=?, which returns a list of all indicators and their possible states. The choice is:
  - don't report indicator changes.

- report only those indicator changes that were not caused by the application itself.
- report all indicator changes, irrespective of their cause.

A similar requirement is to control whether the phone returns intermediate result codes to the application during connection negotiation for data calls. These confirm the type of call which has been setup and indicate which type of data call has been negotiated. ETSI provides **+CR service report** which indicates async transparent, sync transparent, async non-transparent, or sync non-transparent.

3.5 Making a call

This is more complex than might be expected because of differences between traditional networks and modern networks. In the traditional PSTN, the call was established and then the information type (voice, data, fax) was declared. In newer networks such as GSM, the network needs to know the information type before the call can be established, because different network resources are required for different information types.

The peculiar method of call control used in the ETSI command set is the result of reconciling these contradictory requirments, so that traditional applications will still operate.

The procedure is as follows: before starting or accepting a connection, the nature of the connection must be declared. This is necessary so that appropriate network resources can be allocated. The declaration is whether the connection will use a single type of information, or will be voice followed by data, or will alternate between voice and fax or between voice and data. The originator of a call will simply declare the type of call that is required. The terminator of the call will know the type of the proposed call from the unsolicited result code +CRING. Both originator and terminator must then ensure that current settings are correct (no action is needed if current settings are already as required). Before originating or accepting a call, the originator/terminator must have declared any non-voice type using a conventional **+FCLASS** command and declared the call mode using **+CMOD call mode**. Note that +CMOD must be changed back to *single mode* at the end of any alternating call.

After ensuring that its current settings are correct, the originator dials the destination number. If the dial string finishes with a semicolon (;) the call is initiated as a voice call. Otherwise it is initiated as a data or fax call, depending on the previous settings.

When the first 'call' is finished, the connection can be terminated or (if relevant) the other mode of call can be started using the same connection. When that second call is finished the call must be terminated, unless the call mode is alternating voice/data, when the mode can continue to switch between voice and data.

A new 'hangup' command **+CHUP hangup call** is defined by ETSI to assure connection hangup. Other methods of hangup are not guaranteed to take down the connection, because of the historical context of those commands and the complications of alternating call modes.

Dialling is done using the standard **D dial** command. If the connection is a single voice call, or is a dual type call and the first call over the connection is voice, the dial command must end in a semicolon (ATD...;). If the connection is a single non-voice call, or is a dual type and the first call is not voice, the dial command must not end in a semicolon (ATD...).

3.5.1 Dialling from a phonebook

An application can request dialling of a phone number in a particular location in phone memory or the phone number corresponding to a name.

The commands **D>....** cause the phone to dial a number stored in phone memory. Since a phone is logically composed of three separate parts, it is possible to specify the memory in either the **M**obile **E**quipment (the phone *per se*), the **S**ubscriber **I**dentification **M**odule (the SIM card), or the **T**erminal **A**daptor (an adaptor connected to the phone). It is also possible to dial a number matched to a particular alphanumeric string.

The command must end with a semicolon (;) if the connection is a single voice call, or is a dual type call and the first call over the connection is voice. Otherwise it should not end with a semicolon.

3.5.2 Extended error reports

Some phones may support extended error reporting.

If the phone has extra information about the last unsuccessful attempt to initiate or receive a call, the command **+CEER extended error report** causes it to pass a textual string with that information to the application.

3.6 Accessing phone resources

The most basic control function is to switch the phone on or off, and ETSI does this with **+CFUN functionality**. The phone is switched on by selecting 'full functionality' and switched off by selecting 'minimum functionality', instead of just 'on' or 'off', because of difficulties of definition - a phone may have several degrees of 'on' and 'off'. Other degrees of functionality are also provided - various rf circuits may be disabled.

Before attempting to take control of a phone, the application may wish to check that the phone is not in use. This can be done in ETSI with **+CPAS phone activity status**. This replies with

ready (commands will be accepted by the phone)
unavailable (commands will not be accepted by the phone)
unknown (commands may or may not be accepted by the phone)
ringing (the phone will accept commands, but there is an incoming call)
call in progress (the phone will accept commands, but a call is in progress)
asleep (the phone will not accept commands because it is 'switched off')

So if a call is in progress or the phone is ringing, the application may choose to wait and try again in a few moments.

Normally (of course) a phone is operated by a user with its normal switches and keypad. However, an application providing an expanded user interface to the phone may wish to deactivate certain controls actually on the phone, to prevent accidental interference. ETSI command **+CMEC mobile equipment control mode** in conjunction with others does this by allowing the application to take exclusive control of the phone keypad, the phone display, or other phone indicators (such as independent lights or audio buzzers). However, dual control (both the application and the phone) is an option. This may be useful for control of a display or indicators.

- The ETSI command **+CKPD keypad control** allows the application to simulate key presses on the phone itself. This important command is useful when an application is simulating the phone keypad, or when the application needs to use some special sequence of key strokes to activate a special phone feature not covered by an individual 'AT' command. +CKPD allows the application to specify the key which is pressed and the time for which it is pressed. This is because some keys change function depending on whether they are merely touched or held down. A common set of keys is defined within the command. Colon (:) is used as an escape character so that other keys may be defined - a colon followed by a byte is the IA5 58 character defined by that byte. A semicolon (;) is the escape character for entering a string.

- The phone textual display may be read and written using **+CDIS display**. +CDIS passes characters which are part of a character set (defined by **+CSCS select character set**) and addresses characters from the top left corner of the display through to the bottom right hand corner of the display. +CDIS=? returns the number of elements in the display. +CDIS? returns the current text on display and +CDIS= writes text to the display. The character sets (obviously) define the visual symbol appropriate to a particular character. The sets supported by the command set are:

the GSM character set
international alphabet number 5 (IA5)
PC character sets
some ISO 8859 character sets
hexadecimal representation of characters

- The other indicators on a phone may be read and written with **+CIND indicator**. This is a 'catch all' command intended to address all indicators, whether visual or audio, other than the main display. Some common indicators are defined, but it is certain that not all possibilities are covered ! Naturally, the values associated with any particular indicator depend on the precise nature of that indicator. Generally, however, the value 0 is taken to mean off, and higher values indicate progressively more and more on. +CIND=? returns a list of the indicators and the values supported by each indicator. +CIND? returns a list of the current values in the same indicator order as '+CIND=?'. +CIND= allows a list of values to be written to the indicators. The list must be in the same indicator order as '+CIND=?'. A null value in the '+CIND=' list implies that the indicator is to retain its present value. The indicators and values defined so far are:

indicator	**range of values**
battery charge level	0-5
signal quality level	0-5
service available	0-1
audio sounder	0-1
message received	0-1
call in progress	0-1
voice operated transmit in progress	0-1
roaming in operation	0-1

It is also necessary, at times, to disable certain features of the phone. In the ETSI command set, **+CLCK lock** allows certain types of call to be barred or enabled, can disable the phone keypad, disable the SIM, and can lock the phone to the SIM. +CLCK uses a password(s), to ensure that the locking or unlocking is authorised. Each lockable facility can have its own independent password, which is stored somewhere in the phone.

- If the phone keypad is locked, many of the normal controls will not operate. If the phone is locked to the SIM, it will not operate with any other SIM card.
- If a particular type of call is barred, the phone will refuse to either initiate or receive that type of call. A few of the types of barred call are:

all outgoing calls
outgoing international calls
outgoing international calls except to home country
all incoming calls
incoming calls when roaming

Phone password(s) can be changed through the normal phone menu or in ETSI by the use of **+CPWD password**. +PWD requires use of both the old password and the new password. Note that if the command executes correctly it *will* change the password ! It is the responsibility of the application to ensure that a new password is the actual password intended by a user.

- +CPWD=? returns a list of the facilities that may be locked, and the maximum length of each password.

3.6.1 Using the phonebook

Accessing the phonebook is likely to be a popular application, requiring the means to read, write and delete entries which may be stored in memory in various places.

The ETSI command **+CPBS phonebook storage** allows the application to deal with phonebooks in either the phone *per se* (the **M**obile **E**quipment), the SIM, or the **T**erminal **A**daptor.

Once the memory has been selected, **+CPBR phonebook read** allows reading of all entries from one selected index through to a second selected index. Each entry consists of some text and a number in the format previously selected. +CPBR=? returns the maximum lengths of numbers and text fields that are supported by the phone.

+CPBW phonebook write allows a number (in the selected format) and associated text to be stored in a stated index in the selected memory. This command is also used to delete entries from the phonebook: if the command parameters are empty, all data at the index are deleted.

3.7 Dealing with mobile messages

Actual message packets (Protocol Data Units) contain protocol information such as addresses, as well as a message. Phones normally allow a user to enter these elements (including, of course, the actual characters of the message) using the phone keypad and then assemble the packet within the phone (invisibly to the user). An application could also work this way and cause the phone to assemble the PDUs, but alternatively might

communicate with the phone using entire PDUs, and do assembly and disassembly in the application. The main differences between the methods are:

- mimicing a human user dictates that messages be comprised of characters. It also uses extra commands to specify other PDU fields to the phone, but these are required in any case to allow applications to configure a phone so that a user can do messaging with the phone keypad.
- using entire PDUs allows messages to contain arbitary data and uses generic commands only.

In both cases the application must 'understand' the contents of a PDU and provide the necessary information.

Most other messaging commands are also generic: sending messages immediately, sending messages previously stored in the phone, receiving messages, deleting messages, specifying where messages are to be stored, and so on. So while the following ETSI commands were developed for GSM, many of them are not GSM specific.

First of all, the actual messaging service must be selected. In the ETSI set, **+CSMS select message service** selects a particular messaging service (GSM, CDMA, etc.) and also determines whether messages may originate at a mobile (**mo**), terminate at a mobile (**mt**), or support broadcast messages (**bm**).

+CPMS preferred message storage specifies which of the phone memories are to be used for received messages.

+CMGF message format specifies either PDU mode or text mode. In PDU mode, messaging commands operate on a data structure that includes all the fields associated with a message, and a phone does not need to parse the message. In text mode, messaging commands operate only on the message part of the PDU (the field that is the purpose for the message). If text mode has been selected, the character set that defines the symbols represented by individual bytes is dictated by **+CSCS select character set** (described previously).

+CNMI new message indications is a powerful command that determines how newly received messages should be indicated. Should an indication be passed to the computer, or should the message itself be passed ? If indications are requested, should they be passed immediately or buffered ? An application may decide to request immediate indication of a message (or the message itself) when the computer is switched on, but need to request buffering if the computer is about to be switched off (and is unable to wake up in response to external signals).

Separate fields determine what happens to unsolicited result codes, point-to-point messages, broadcast messages, delivery reports. Another field (*buffer*) determines whether existing unsolicited result codes are flushed or discarded when the command executes.

- *bfr* determines what happens to existing unsolicted result codes when the command executes **and** the *mode* parameter has selected an option where data is passed to the application. The choice is:
  - flush to the application
  - delete
- *Mode* determines what happens to unsolicited result codes. The choice is:
  - buffer in the TA. If the TA is full, buffer elsewhere or discard. These buffered codes can be flushed to the application by a later use of +CNMI that sets the 'buffer' parameter to flush the buffer and does **not** select this mode option.
  - discard when the computer-phone link is in use, otherwise pass them directly to the application.
  - buffer when the computer-phone link is in use, and flush them to the application when the link becomes free. Otherwise pass them directly to the application.
  - pass directly to the application using specified inband techniques.
- *mt* determines what happens to new point-to-point messages. *bm* determines what happens to new broadcast messages. In both cases, the choice is:
  - do nothing
  - indicate new messages to the application using unsolicited result codes
  - pass new messages to the application,
- *ds* determines what happens to delivery status reports. The choice is:
  - do nothing
  - indicate new status reports to the application using unsolicited result codes

There are two commands for sending messages. One (**+CMGS send message**) passes a message to the phone and requests immediate transmission. The other (**+CMSS message store send**) requests the transmission of a message that has previously been stored at some *index* in a memory in the phone. These messages are written to memory using the command **+CMGW message write**, and the return value is the index (position) of the message in that memory. The type of memory is that previously selected with +CPMS.

The messages currently stored in a phone are listed by **+CMGL message list**. This allows the user to select the type of message to be returned:

unread rx messages
read rx messages
unsent tx messages
sent tx messages
all messages

Individual stored messages may be read using **+CMGR message read** from location *index* in the memory type selected previously with +CPMS. The command returns the status (read, unread, sent, unsent) of the message as well as the message itself. Messages are deleted using the command **+CMGD message delete** which specifies a particular index.

Most messaging services are likely to include a service centre, so the command **+CSCA service centre address** allows its telephone number to be written.

Finally, the commands **+CSAS save settings** and **+CRES** allow different profiles of settings related to short messages to be stored and recalled. This is intended for use in phones (or pagers) which deal with different message delivery services, each of which may require different options depending on the service. For example, one messaging service may relate to financial information, another to personal information, etc.. A user may want personal information to be presented immediately, but wants only indications of financial messages. The personal messages may be stored in the ME, and the financial messages stored in the TA.

3.8 Commands from existing standards

Some other command sets have individual commands which are useful in this context. In particular **+CBC battery charge** and **+CSQ signal quality**, both of which are self explanatory.

Another set of commands from TIA is useful for directing the flow of audio in the computer and the phone. This particular command set allows various combinations of audio paths, so that transmitted audio may come from the phone microphone or the computer, received audio may go to the phone loudspeaker or the computer, and various combinations thereof.

4 References

[1] *ETSI*: (postal address) F-06921 Sophia Antipolis CEDEX - FRANCE, (office address) 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE, (email) secretariat@etsi.fr, (telephone) +33 92 94 42 00, (fax) +33 93 65 47 16

[2] *GSM 07.05*: Short Message Service (SMS) and Cell Broadcast Service (CBS)

[3] *GSM07.07*: AT command set for GSM Mobile Equipment (ME)