CERTIFICATION OF APPROVAL

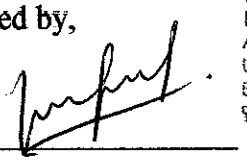
SCHOOL ATTENDANCE AND SECURITY SMS SYSTEM

by

MOHD SILMI BARIED BIN MAJID

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,



Dr Mohamad Naufal B. Mohamad **Saad**
Lecturer
Electrical & Electronic engineering
Academic Block No. 22 (22-03-04)
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh,
Perak Darul Ridzuan, MALAYSIA

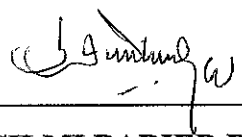
DR. MOHAMAD NAUFAL MOHAMAD SAAD,
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

MAY 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



MOHD SILMI BARIED BIN MAJID

ABSTRACT

The objectives of this final year project are to design and implement the real world problem so that they can prepare themselves for future employment. Besides, this project is a requirement for the undergraduate students in order to complete their studies.

The topic chosen for this project is ***School Attendance SMS System.*** The system will be able to do certain tasks. This is discussed in introduction part. The project objectives and the scope of study are highlighted at the end of the chapter.

Information is a very important and powerful item in delivering this project successfully. Brief descriptions of the theory that will be used are covered in the literature review or theory section.

This report also states the methodology used and tools required during this project in the methodology and project work chapter. The last part is the conclusion which describe the student's expectation from this project

ACKNOWLEDGEMENTS

First and foremost, I would like to praise God Almighty and thank Him for this countless blessing and guidance throughout the hard times I have endured through and through. Without Him, I am sure I would not have made it this far.

Hereby, I would like to extend my utmost gratitude and heartfelt appreciation to the individuals who made this project an enriching and invaluable experience.

I would like to thank my supervisor, Mr. Mohamad Naufal bin Mohamad Saad for his guidance, support and trust. Being under his supervision, has indeed allowed me to uncover my strength in exploring the elements of project management and independent decision-making throughout the semester. His passing-down of knowledge and expertise has been very valuable in allowing me to understand the requirements of the project and also increase my know-how on the project. His insights and guidance helped me look at problems in a different view and approach them in a positive manner.

Lastly, a special thanks to the UTP Electrical and Electronics department FYP committee and technicians, with special mention of Ms. Azrina, Ms. Noorashikin, Ms. Siti Hawa Tahir, Mr Azhar and Mr. Isnani. They have been very warm, hospitable and helpful.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Background

The rapid growing economy has made it possible for almost every individual to have their own mobile phone whether for personal or business purposes. The increasing popularity of SMS system leads to production and design of many various and interesting SMS system ranging from simple to complex systems. Now, there are a lot of cases of students missing from school, absentees, kidnapping and other that worry the parents. One popular growing system is the School Attendance SMS system. This system will be able to monitor and to inform the parents whether their children enter the school safely or not. This system also will inform the parents the exact time their children enter the school via SMS (Short Message System). In this wireless era, most of the wealthy parents provide their kids with mobile phone. Lots of excuse that been given by parents such as easy to connect their kids and others. Unfortunately, they are few students misuse the phone by playing games, sharing the x-rated porn stuff and using the phone camera for inappropriate purposes.

1.2 Problem Statement

Communication and safety has been a vital aspect of our daily life since the beginning of the time. The project is about developing a school attendance system which integrates both aspects. The system must be efficient in scanning the smart card, keep record of student's attendance and sending the SMS to parents. The project will use the basic principals in communication system. This system can also be used by offices and library to inform the data directly to users via SMS.

1.3 Objectives

This Final Year Project course plays a vital role in achieving UTP's vision which is to produce a well rounded graduate. It is also a very great opportunity for students to relate the theoretical knowledge from class and applying it in project. Despite that, students will develop skills in work ethics, communication, management, interpersonal skills and etc. The objectives of the Final Year Project are:

- To develop a framework, this will enhance student's skills in the process of applying knowledge, expanding thoughts, solving problem independently and presenting findings.
- Develop a system that can store and send student's attendance to their parent using communication system preferably wireless communication.
- To produce a system that is reliable and can be easily handled by other people and also low in cost if possible.
- To integrate the hardware and the software part of the system to make it easier to handled and managed.
- Develop skill on handling programming chip such as 16F84 and 16F877 and gain knowledge on developing the hardware and software.
- Develop a security system for student to reduce the number of kidnapping or students missing from school.

1.4 Scope of Study

There are several topics and issues that must be considered before proceeding any further in the design of the system. The scope of study depends mainly on these few areas:

- Handling with Barcode as input and produce the output.
- Interface at the PC for the exchange of inputs and outputs from the barcode scanner.
- Database application to record and store the data of student attendance.
- Sending the data to parents using modem via SMS ***(Short Message system)***

The applications and devices that are considered must be reliable, cost-effective and practical for a feasible implementation. This will ensure that the design produced is economical and marketable.

Barcode scanner is the easiest and cheapest way in designing or recording attendance compare to smart card. For this project, perhaps using the fix barcode is more practical but for testing we just use the portable barcode scanner.

Graphical User Interface or GUI is known to be the best form of interfacing the user to the system. With this note in mind, the Microsoft Visual Basic is selected to accomplish the task. Furthermore, Visual Basic also has serial communication ability.

The last area to be considered is the database application. Since Microsoft Access has Visual Basic commands embedded in it, it is efficient to record and store the necessary surveillance information. To add to that, Microsoft Access is also a very popular choice for database applications.

CHAPTER 2

LITERATURE REVIEW

2.1 Barcode Application

2.1.1 Barcode symbologies

Bar codes are like a printed version of the Morse code. Different bar and space patterns are used to represent different characters. Sets of these patterns are grouped together to form a "symbology". There are many types of bar code symbologies each having their own special characteristics and features. Most symbologies were designed to meet the needs of a specific application or industry. For example the UPC symbology was designed for identifying retail and grocery items and PostNET was designed to encode Zip Codes for the US Postal Service. For this project, Code 39 is used because it is easier to handle and widely used.

The Normal CODE 39 is a variable length symbology that can encode the following 44 characters: 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *$/+%. Code 39 is the most popular symbology in the non-retail world and is used extensively in manufacturing, military, and health applications. Each Code 39 bar code is framed by a start/stop character represented by an asterisk (*). The Asterisk is reserved for this purpose and may not be used in the body of a message. B-Coder automatically adds the start and stop character to each bar code therefore you should not include them as part of your bar code message. If you select the NORMAL version of CODE 39 and your bar code text contains lower case characters, B-Coder will convert them to upper case. If your bar code message contains any invalid characters, B-Coder will prompt you with a warning message (if the Enable Invalid Warning Messages option is selected in the Preferences menu).

Below are the figures of barcode that usually use in scanning;

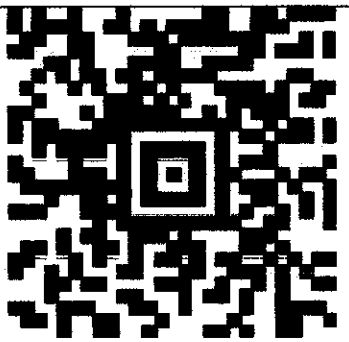
Figure	Type of barcode
	Code 39
	UPC (Universal Product Code)
	EAN-8/ EAN-13 (ISBN Version)
	RSS 14
	Codabar
	INTERLEAVED 2 OF 5
	Aztec Code

Table 1: image of the barcodes

2.1.2 Barcode scanner

There are currently four different types of bar code readers available. Each uses a slightly different technology for reading and decoding a bar code. There are pen type readers (e.g. bar code wands), laser scanners, CCD readers and camera-based readers.

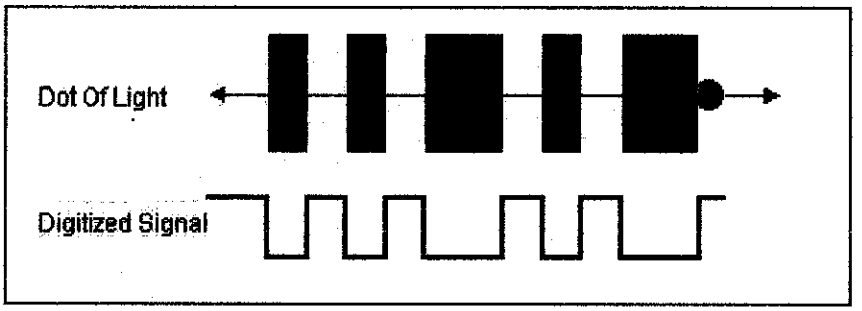


Figure 1: how a barcode scanner scans

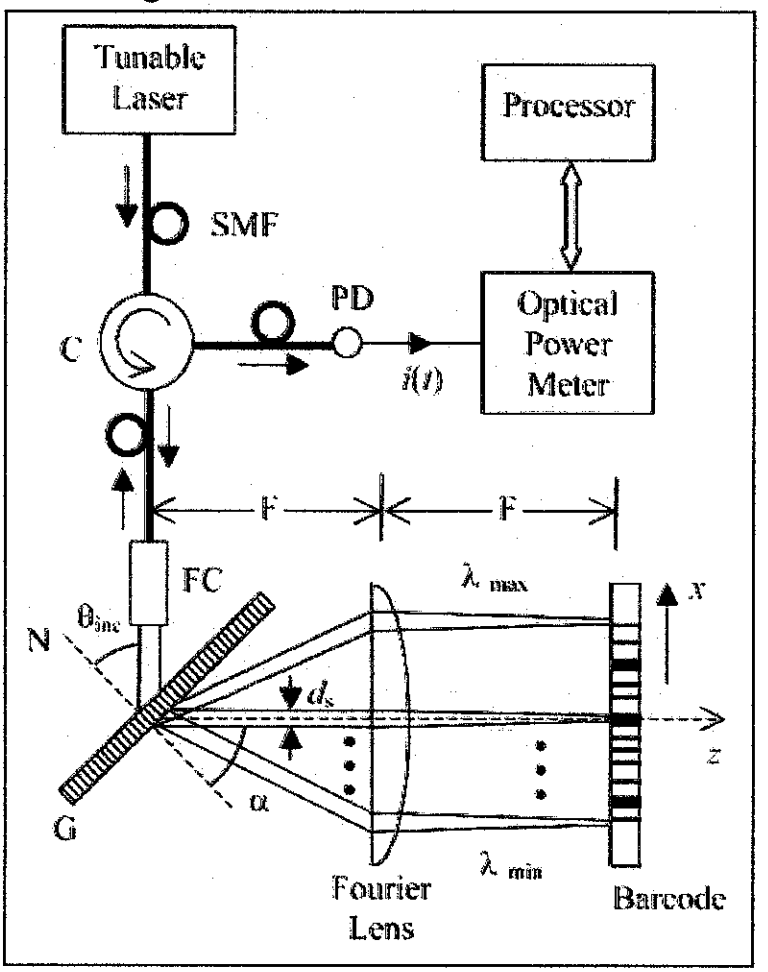


Figure 2: high security all fiber wavelength-multiplexed barcode scanner to detect barcode

2.1.3 Creating a barcode

For creating or printing a barcode, software is needed to decode the data. BARCODE 39 SAMPLE software is used to create barcodes. This is actually a sample software that can create barcode easily just by entering the number and after run the software, we can get the results immediately.

Figure 3: creating barcode software

2.2 Database information

The usage of database has become the norm when it comes to presenting information in most system and application. When the record keeping needs outgrow a filing cabinet or bundle of index cards, a computerized system can help to manage the growing quantity of data, and enable more effective usage of data. Spreadsheet programs can manage small, simple databases. Desktop programs such as Microsoft Access and FileMaker can handle databases too big or too complicated for a spreadsheet. Larger databases, databases accessed by many people at once, and databases that feed client/server and web applications need real database servers.

The relational model, invented by IBM researcher Ted Codd in 1970, was not turned into a commercial product until almost 1980. Since then database systems based on the relational model, called relational database management systems or RDBMSs, have come to dominate the database software market. Today few people know about any other kind of database management system.

The benefit of a good RDBMS and a well-designed relational database:

- Data integrity and consistency maintained and/or enforced by the RDBMS
- Redundant data eliminated or kept to a practical minimum
- Data retrieved by unique keys
- Relationships expressed through matching keys
- Physical organization of data managed by RDBMS
- Optimization of storage and database operation execution times
- Concurrency: database users don't corrupt each other's work
- Scalable: can spread load across multiple CPUs or servers

Examples of well-known industrial-strength relational RDBMSs include:

- Oracle
- Microsoft SQL Server
- IBM DB2
- Informix

Well-known PC-based (desktop) relational RDBMSs include:

- Microsoft Access
- Microsoft FoxPro
- Borland dBase

2.2.1 Microsoft Access

Microsoft Access is a relational database management system (RDBMS). At the most basic level, a RDBMS is a program that facilitates the storage and retrieval of structured information on a computer's hard drive.

The Microsoft Access package contains the following elements:

- A relational database system that supports two industry standard query languages: Structured Query Language (SQL) and Query By Example (QBE)
- A full-featured procedural programming language - essentially a subset of Visual Basic
- a simplified procedural macro language unique to Access
- A rapid application development environment complete with visual form and report development tools
- a sprinkling of objected-oriented extensions
- various wizards and builders to make development easier

When designing databases in Microsoft Access, it is important to understand the nature of database designing. The three different view of computing in Microsoft Access, better known as 'personalities' are:

- The relational database personality; viewing the application as sets of data
- The procedural programming personality; viewing the application as commands to be executed sequentially
- The object-oriented personality; viewing the application as objects which encapsulate state and behavior information

Since there are often several vastly different ways to implement a particular feature in Access, recognizing the different personalities and exploiting the best features and avoiding the pitfalls of each are important skills for Access developers.

The advantage of these multiple personalities is that it is possible to use Access to learn about an enormous range of information systems concepts without having to interact with a large number of 'single-personality' tools, for example:

- Oracle for relational databases
- PowerBuilder for rapid applications development,
- SmallTalk for object-oriented programming.

Microsoft Access has a particular hierarchy system for databases, which consists of:

- Database File: the main file that encompasses the entire database and that is saved to hard-drive or floppy disk
- Table: a collection of data about a specific topic. There can be multiple tables in a database
- Field: different categories within a Table. Tables usually contain multiple fields
- Datatypes: properties of each field. A field only has 1 datatype.

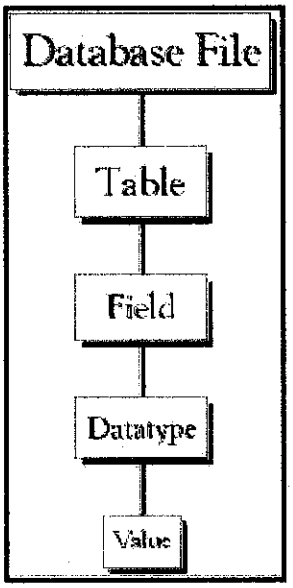


Figure 4: Database Hierarchy in Microsoft Access

2.3 Graphical User Interface (GUI)

A GUI is a graphical, (rather than purely textual) user interface to a computer. The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands that we had to remember and computer responses that were infamously brief. The command interface of the DOS operating system is an example of the typical user-computer interface before GUIs arrived. An intermediate step in user interfaces between the command line interface and the GUI was the non-graphical menu-based interface, which allows interaction by using a mouse rather than by having to type in keyboard commands.

Nowadays, major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own graphical user interface elements and ideas. A GUI sometimes uses one or more metaphors for objects familiar in real life, such as the desktop, the view through a window, or the physical layout in a building. Elements of a GUI include such things as:

- windows
- pull-down menus
- buttons
- scroll bars
- iconic images
- wizards

With the increasing use of multimedia as part of the GUI, sound, voice, motion video, and virtual reality interfaces seem likely to become part of the GUI for many applications. A system's graphical user interface along with its input devices is sometimes referred to as its "look-and-feel."

The GUI familiar to most of us today in either the Mac or the Windows operating systems and their applications originated at the Xerox Palo Alto Research Laboratory in the late 1970s. Apple used it in their first Macintosh computers. Later, Microsoft used many of the same ideas in their first version of the Windows operating system for IBM-compatible PCs.

2.3.1 Microsoft Visual Basic Environment

The Microsoft Visual Basic is known to be among the most popular choice to create Windows GUI. In Visual Basic, new windows created are called forms. Elements (such as text boxes and buttons) that are placed inside a form are called controls. The Visual Basic allows event-driven programming, where the user's actions cause events, and each event in turn triggers a procedure that is associated with it.

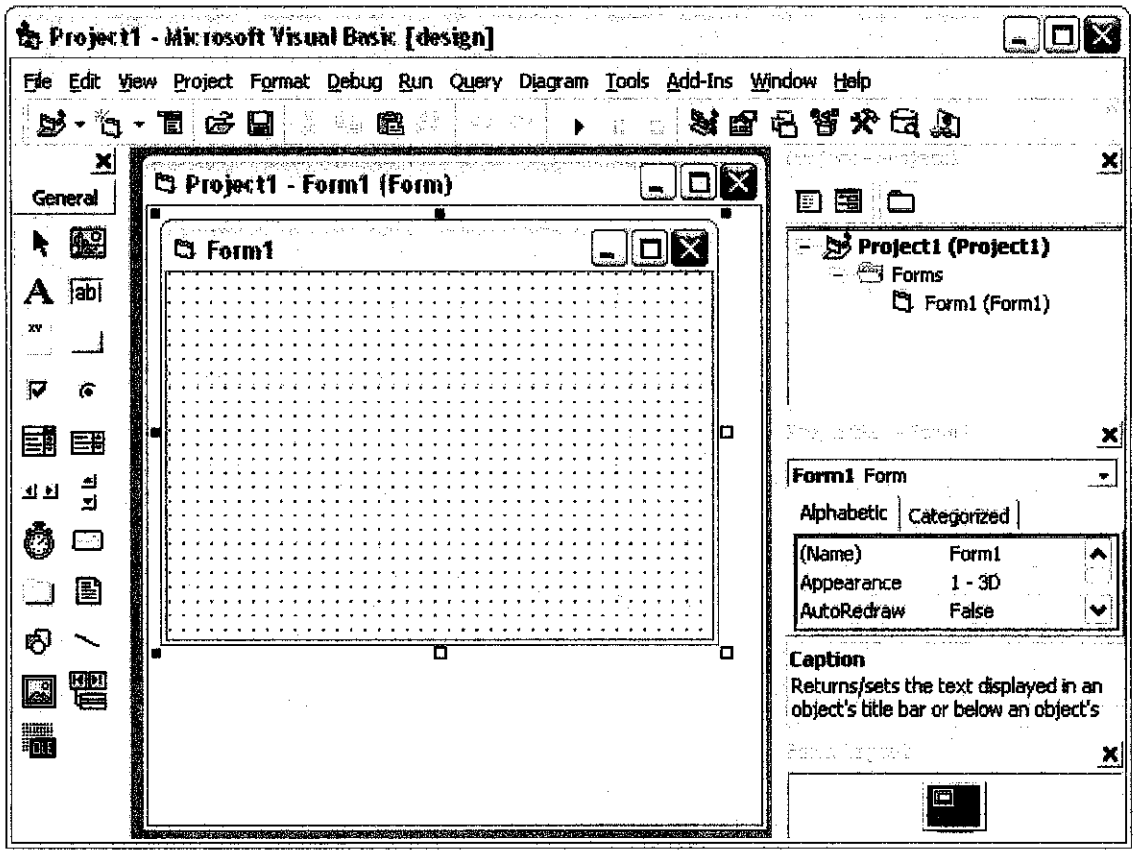


Figure 5: Visual Basic Editor

The properties of creating an object model in Visual Basic are:

- Objects have properties and methods
- Forms and controls are examples of objects
- Properties describe an object. Examples: name, color, size, or how they will behave
- Methods are actions associated with an object. Examples: move, clear and print

2.4 Sending SMS through PC

In general, there are two ways to send SMS messages from a computer / PC to a mobile phone:

- Connect a mobile phone or GSM/GPRS modem to a computer / PC. Then use the computer / PC and AT commands to instruct the mobile phone or GSM/GPRS modem to send SMS messages.
- Connect the computer / PC to the SMS center (SMSC) or SMS gateway of a wireless carrier or SMS service provider. Then send SMS messages using a protocol / interface supported by the SMSC or SMS gateway.

2.4.1 Sending SMS from PC Using GSM Modem

The SMS specification has defined a way for a computer to send SMS messages through a mobile phone or GSM/GPRS modem. A GSM/GPRS modem is a wireless modem that works with GSM/GPRS wireless networks. A wireless modem is similar to a dial-up modem. The main difference is that a wireless modem transmits data through a wireless network whereas a dial-up modem transmits data through a copper telephone line. Most mobile phones can be used as a wireless modem. However, some mobile phones have certain limitations comparing to GSM/GPRS modems. To send SMS messages, first place a valid SIM card from a wireless carrier into a mobile phone or GSM/GPRS modem, which is then connected to a computer. There are several ways to connect a mobile phone or GSM/GPRS modem to a computer. For example, they can be connected through a serial cable, a USB cable, a Bluetooth link or an infrared link. The actual way to use depends on the capability of the mobile phone or GSM/GPRS modem. For example, if a mobile phone does not support Bluetooth, it cannot connect to the computer through a Bluetooth link.

After connecting a mobile phone or GSM/GPRS modem to a computer, you can control the mobile phone or GSM/GPRS modem by sending instructions to it. The instructions used for controlling the mobile phone or GSM/GPRS

modem are called AT commands. (AT commands are also used to control dial-up modems for wired telephone system.) Dial-up modems, mobile phones and GSM/GPRS modems support a common set of standard AT commands. In addition to this common set of standard AT commands, mobile phones and GSM/GPRS modems support an extended set of AT commands. One use of the extended AT commands is to control the sending and receiving of SMS messages.

The following table lists the AT commands that are related to the writing and sending of SMS messages:

AT command	Meaning
+CMGS	Send message
+CMSS	Send message from storage
+CMGW	Write message to memory
+CMGD	Delete message
+CMGC	Send command
+CMMS	More messages to send

Table 2: List of AT command

One way to send AT commands to a mobile phone or GSM/GPRS modem is to use a terminal program. A terminal program's function is like this: It sends the characters you typed to the mobile phone or GSM/GPRS modem. It then displays the response it receives from the mobile phone or GSM/GPRS modem on the screen. The terminal program on Microsoft

Windows is called HyperTerminal. To send SMS messages from an application, you have to write the source code for connecting to and sending AT commands to the mobile phone or GSM/GPRS modem, just like what a terminal program does. You can write the source code in C, C++, Java, Visual Basic, Delphi or other programming languages you like. However, writing your own code has a few disadvantages:

- You have to learn how to use AT commands.
- You have to learn how to compose the bits and bytes of an SMS message. For example, to specify the character encoding (e.g. 7-bit encoding and 16-bit Unicode encoding) of an SMS message, you need to know which bits in the message header should be modified and what value should be assigned.
- Sending SMS messages with a mobile phone or GSM/GPRS modem has a drawback -- the SMS transmission speed is low. As your SMS messaging application becomes more popular, it has to handle a larger amount of SMS traffic and finally the mobile phone or GSM/GPRS modem will not be able to take the load. To obtain a high SMS transmission speed, a direct connection to an SMSC or SMS gateway of a wireless carrier or SMS service provider is needed. However, AT commands are not used for communicating with an SMS center or SMS gateway. This means your have to make a big change to your SMS messaging application in order to move from a wireless-modem-based solution to a SMSC-based solution.

In most cases, instead of writing your own code for interacting with the mobile phone or GSM/GPRS modem via AT commands, a better solution is to use a high-level SMS messaging API (Application programming interface) / SDK (Software development kit) / library. The API / SDK / library encapsulate the low-level details. So, an SMS application developer does not need to know AT commands and the composition of SMS messages in the bit-level. Some SMS messaging APIs / SDKs / libraries support SMSC protocols in addition to AT commands. To move from a

wireless-modem-based SMS solution to a SMSC-based SMS solution, usually you just need to modify a configuration file / property file or make a few changes to your SMS messaging application's source code.

2.4.2 GSM Modem

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer.

Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

As mentioned in earlier sections of this SMS tutorial, computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. You can use a GSM modem just like a dial-up modem.

In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, you can do things like:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.

- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low -- only about six to ten SMS messages per minute.

2.4.3 GPRS Modem

A GPRS modem is a GSM modem that additionally supports the GPRS technology for data transmission. GPRS stands for General Packet Radio Service. It is a packet-switched technology that is an extension of GSM. (GSM is a circuit-switched technology.) A key advantage of GPRS over GSM is that GPRS has a higher data transmission speed.

GPRS can be used as the bearer of SMS. If SMS over GPRS is used, an SMS transmission speed of about 30 SMS messages per minute may be achieved. This is much faster than using the ordinary SMS over GSM, whose SMS transmission speed is about 6 to 10 SMS messages per minute. A GPRS modem is needed to send and receive SMS over GPRS. Note that some wireless carriers do not support the sending and receiving of SMS over GPRS.

If you need to send or receive MMS messages, a GPRS modem is typically needed.

2.4.4 Which is Better: Mobile Phone or GSM / GPRS Modem?

In general, a GSM/GPRS modem is recommended for use with a computer to send and receive messages. This is because some mobile phones have certain limitations comparing to GSM/GPRS modems. Some of the limitations are described below:

- Some mobile phone models (example: Ericsson R380) cannot be used with a computer to receive concatenated SMS messages.

What is a concatenated SMS message?
A concatenated SMS message is a message that contains more than 140 bytes. (A normal SMS message can only contain at most 140 bytes.) Concatenated SMS works like this: the sender's mobile device breaks a message longer than 140 bytes into smaller parts. Each of these parts are then fitted in a single SMS message and sent to the recipient. When these SMS messages reach the destination, the recipient's mobile device will combine them back to one message.

What is the cause of the problem?
When the mobile phone receives the SMS messages that are parts of a concatenated SMS message, it combines them to one message automatically. The correct behavior should be: when the mobile phone receives the SMS messages that are parts of a concatenated SMS message, it forwards them to the computer without combining them.

- Many mobile phone models cannot be used with a computer to receive MMS messages. Because when they receive a MMS notification, they handle it automatically instead of forwarding it to the computer.
- A mobile phone may not support some AT commands, command parameters and parameter values. For example, some mobile phones do not support the sending and receiving of SMS messages in text mode. So, the AT command "AT+CMGF=1" (it instructs the mobile phone to use text mode) will cause an error message to be returned. Usually GSM/GPRS modems support a more complete set of AT commands than mobile phones.
- Most SMS messaging applications have to be available 24 hours a day. (For example, an SMS messaging application that provides ringtone downloading service should be running all the time so that a user can download ringtones any time he/she wants.) If such SMS messaging

applications use mobile phones to send and receive SMS messages, the mobile phones have to be switched on all the time. However, some mobile phone models cannot operate with the battery removed even when an AC adaptor is connected, which means the battery will be charged 24 hours a day.

Besides the above issues, mobile phones and GSM/GPRS modems are more or less the same. Actually, you can consider an AT-command-enabled mobile phone as "GSM/GPRS modem + keypad + display".

There is not much difference between mobile phones and GSM/GPRS modems in terms of SMS transmission rate, since the determining factor for the SMS transmission rate is the wireless network.

2.4.5 Microsoft HyperTerminal

Microsoft HyperTerminal is a small program that comes with Microsoft Windows. You can use it to send AT commands to your mobile phone or GSM/GPRS modem. It can be found at *Start -> Programs -> Accessories -> Communications -> HyperTerminal*. If you cannot find it and you are using Windows 98, then probably you have not installed it. You can go to *Control Panel -> Add/Remove Programs -> Windows Setup* tab -> *Communications* list box item -> *Details* button to install MS HyperTerminal.

Before you start programming your SMS application, you may want to check if your mobile phone, GSM/GPRS modem and SIM card are working properly first. The MS HyperTerminal is a handy tool when it comes to testing your GSM devices. It is a good idea to test your GSM devices beforehand. When a problem occurs, sometimes it is difficult to tell what causes the problem. The cause can be your program, the GSM device or the

SIM card. If you test your GSM device and SIM card with MS HyperTerminal and they operate properly, then it is very likely that the problem is caused by your program.

For Linux users, minicom can be used instead of HyperTerminal.

2.4.6 Procedure for Sending AT Commands to a Mobile Phone or GSM/GPRS Modem Using MS HyperTerminal

To use MS HyperTerminal to send AT commands to your mobile phone or GSM/GPRS modem, you can follow the procedure below:

- Put a valid SIM card into the mobile phone or GSM/GPRS modem. You can obtain a SIM card by subscribing to the GSM service of a wireless network operator.
- Connect your mobile phone or GSM/GPRS modem to a computer and set up the corresponding wireless modem driver. You should find the wireless modem driver in the CD or disk that was provided by the manufacturer. If the manufacturer does not provide such CD or disk with your mobile phone or GSM/GPRS modem, you can go to the manufacturer's web site and see whether the wireless modem driver can be downloaded there. If the wireless modem driver cannot be found on the web site, you can still use Windows' standard modem driver.
- Run MS HyperTerminal by selecting *Start -> Programs -> Accessories -> Communications -> HyperTerminal.*
- In the *Connection Description* dialog box, enter a name and choose an icon you like for the connection. Then click the *OK* button.

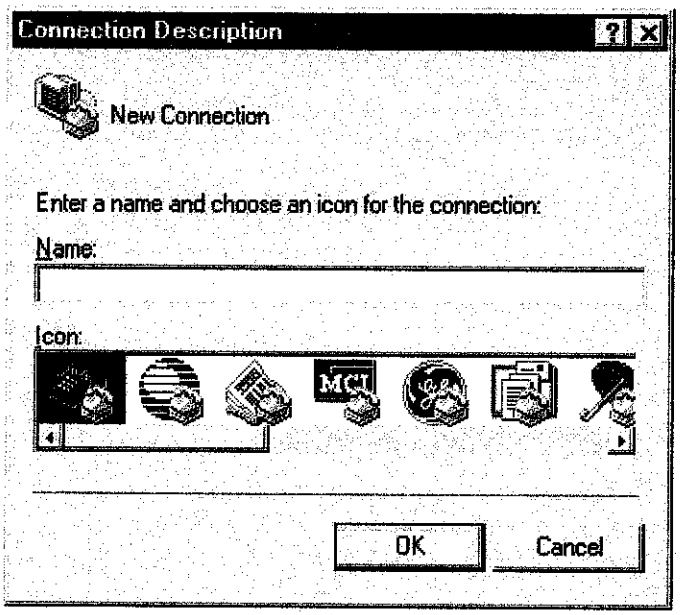


Figure 6: The screenshot of MS HyperTerminal's Connection Description dialog box

- In the *Connect To* dialog box, choose the COM port that your mobile phone or GSM/GPRS modem is connecting to in the *Connect using* combo box. For example, choose COM1 if your mobile phone or GSM/GPRS modem is connecting to the COM1 port. Then click the *OK* button.

  (Sometimes there will have more than one COM port in the *Connect using* combo box. To know which COM port is used by your mobile phone or GSM/GPRS modem, follow the procedure below:

  In Windows 98:
  Go to *Control Panel -> Modem*. Then click the *Diagnostics* tab. In the list box, you can see which COM port the mobile phone or GSM/GPRS modem is connected to.

  In Windows 2000 and Windows XP:
  Go to *Control Panel -> Phone and Modem Options*. Then click the

Modems tab. In the list box, you can see which COM port the mobile phone or GSM/GPRS modem is connected to.)

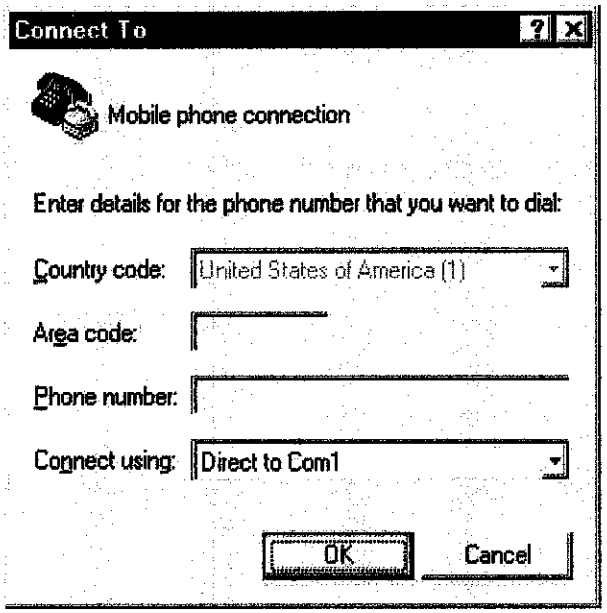


Figure 7: The screenshot of MS HyperTerminal's Connect to dialog box

- The *Properties* dialog box comes out. Enter the correct port settings for your mobile phone or GSM/GPRS modem. Then click the *OK* button.

  (To find the correct port settings that should be used with your mobile phone or GSM/GPRS modem, one way is to consult the manual of your mobile phone or GSM/GPRS modem. Another way is to check the port settings used by the wireless modem driver that you installed earlier.

  To check the port settings used by the wireless modem driver on Windows 98, follow these steps:
  a. Go to *Control Panel* -> *Modem*.
  b. Select your mobile phone or GSM/GPRS modem in the list box.
  c. Click the *Properties* button.

d. The *Properties* dialog box appears. The *Maximum speeds* field on the *General* tab corresponds to HyperTerminal's *Bits per second* field. Click the *Connection* tab and you can find the settings for data bits, parity and stop bits. Click the *Advanced* button and you can find the setting for flow control.

To check the port settings used by the wireless modem driver on Windows 2000 and Windows XP, follow these steps:
a. Go to *Control Panel -> Phone and Modem Options -> Modems* tab.
b. Select your mobile phone or GSM/GPRS modem in the list box.
c. Click the *Properties* button.
d. The *Properties* dialog box appears. Click the *Advanced* tab and then click the *Change Default Preferences* button.
e. The *Change Default Preferences* dialog box appears. The *Port speed* field on the *General* tab corresponds to HyperTerminal's *Bits per second* field. You can also find the setting for flow control on the *General* tab. On the *Advanced* tab, you can find the settings for data bits, parity and stop bits.)

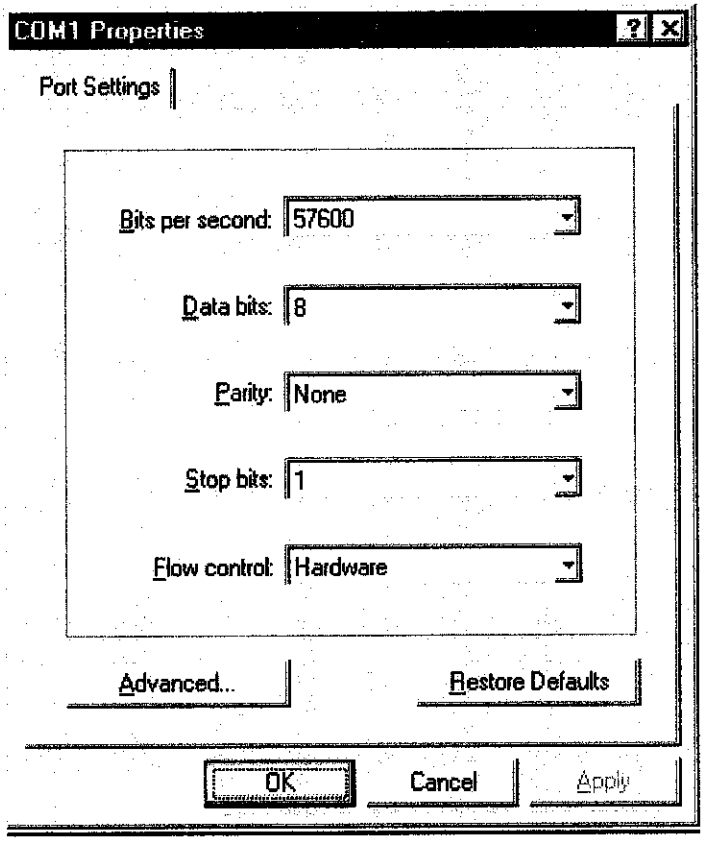


Figure 8: The screenshot of MS HyperTerminal's Properties dialog box

- Type "AT" in the main window. A response "OK" should be returned from the mobile phone or GSM/GPRS modem. Type "AT+CPIN?" in the main window. The AT command "AT+CPIN?" is used to query whether the mobile phone or GSM/GPRS modem is waiting for a PIN (personal identification number, i.e. password). If the response is "+CPIN: READY", it means the SIM card does not require a PIN and it is ready for use. If your SIM card requires a PIN, you need to set the PIN with the AT command "AT+CPIN=<PIN>".

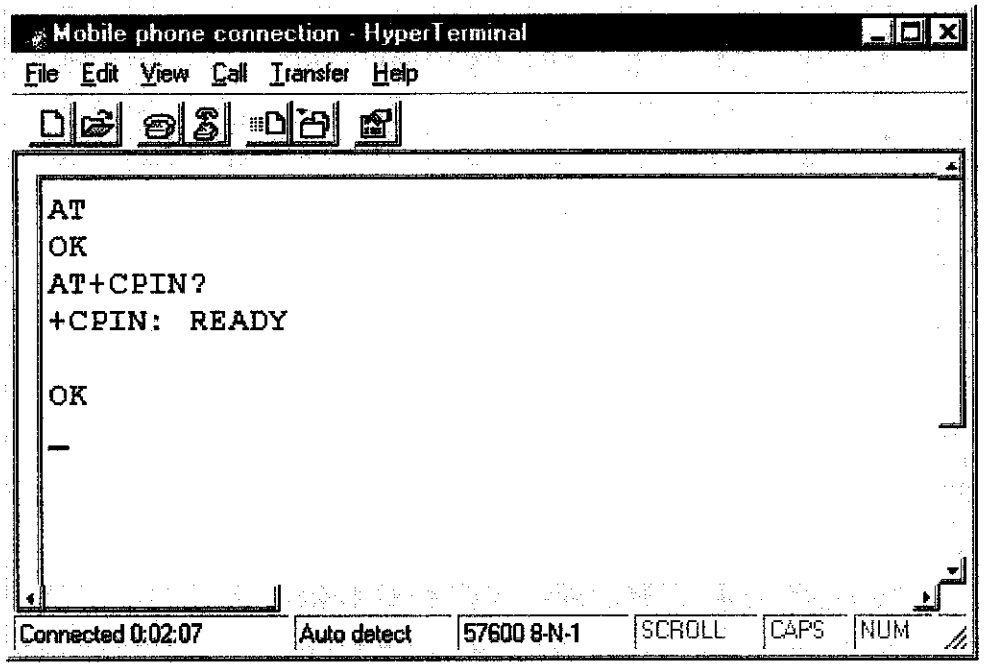


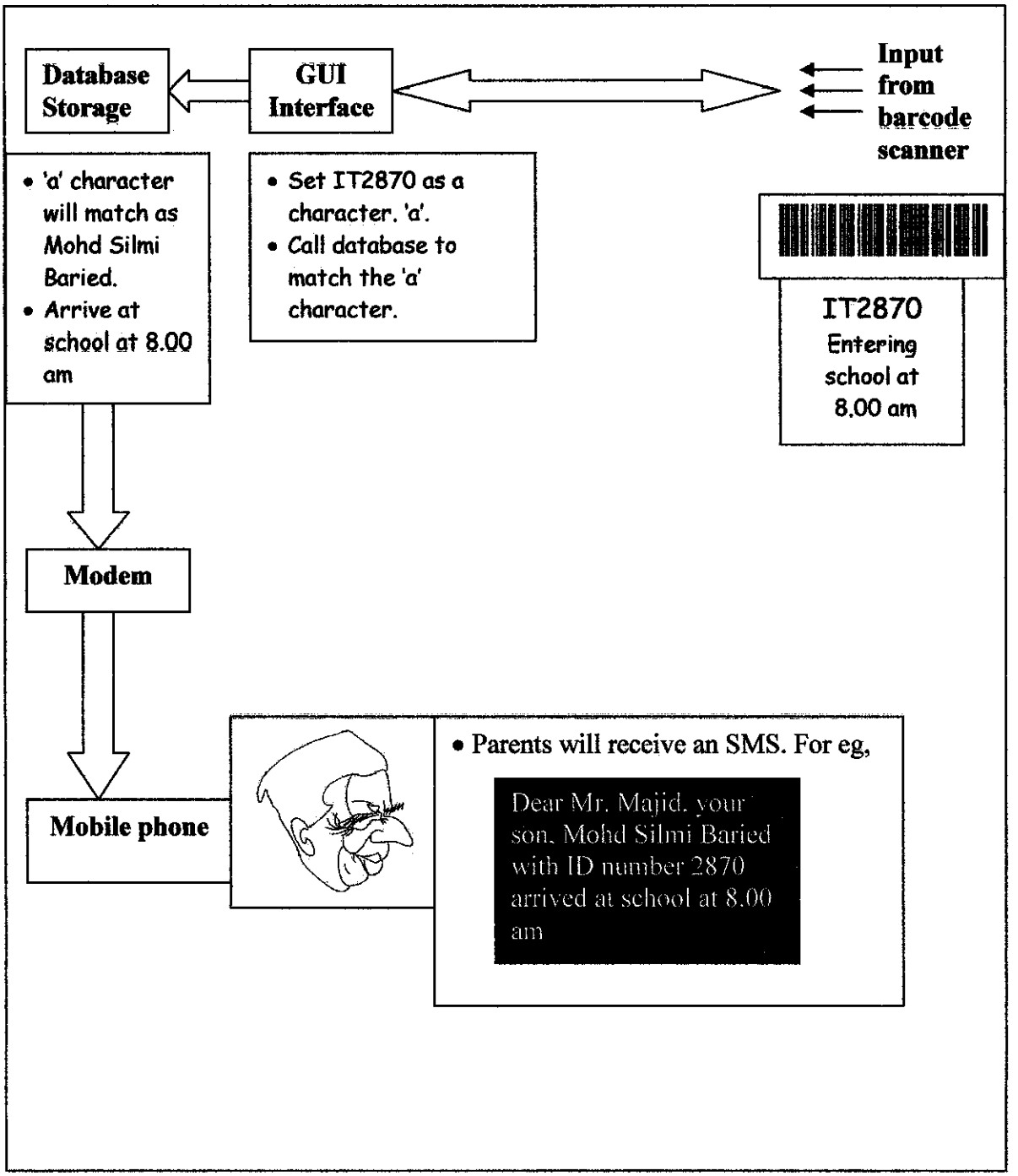
Figure 9: The screenshot of MS HyperTerminal's main window

CHAPTER 3

METHODOLOGY AND PROJECT WORK

3.1 Procedure identification

Figure 10: project diagram



The basic structure of this system consists of:

1. An input interface which consists of the microcontroller circuitry that connects the barcode scanner
2. A USB interface to gather the feedback from the microcontroller to transmit to the USB of a PC.
3. A GUI interface to extract the input from the serial port of the PC while allowing the user to monitor and track down the location details.
4. A database interface to provide the necessary storage for the attendance records.

3.1.1 GUI Interface

Microsoft Visual Basic is chosen to design the GUI interface for the system. Therefore, it must accomplish several important characteristics of the interface. The details are as in Figure 11 below.

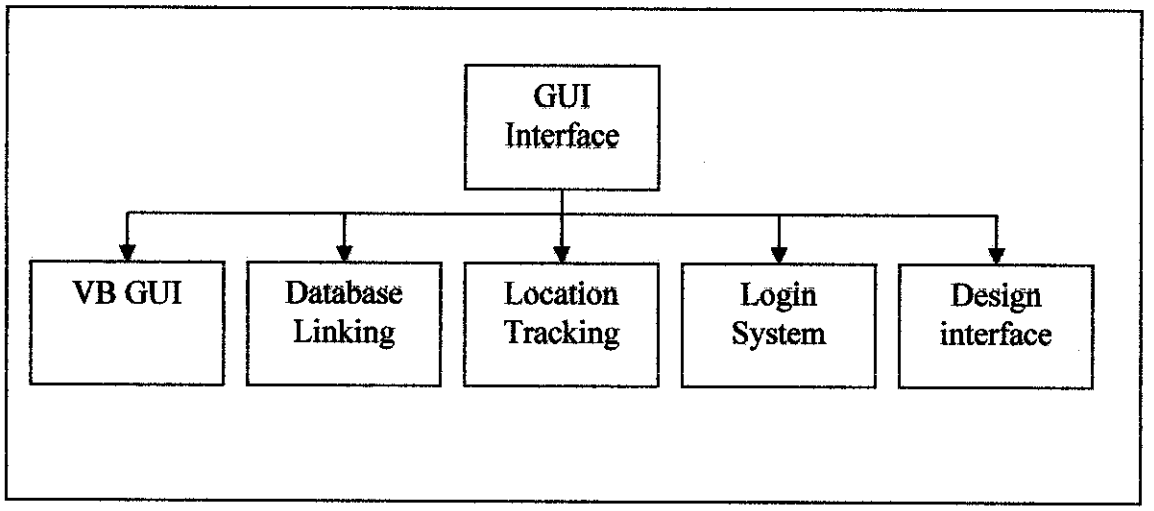


Figure11: GUI interface details

3.1.2 Database Storage

The last part of the system at the end of the day is to be able to store the attendance history for future references. The details are presented in Figure 8 below.

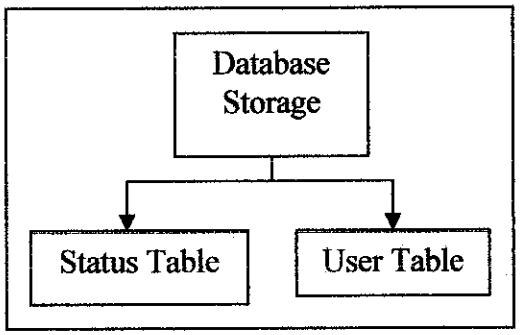


Figure12: Database storage details

As presented in the figure above, besides storing the attendance records, there is also a need for storage of the user identity for the login system. It will include the username and password for the verification of the user's identity.

3.2 Project work

3.2.1 Microsoft Visual Basic

Microsoft Visual Basic is Windows GUI programming software. Therefore, it is the appropriate software option to extract the input from the serial port and alert the user of any activity via graphical interface.

A component known as the Microsoft Comm Control 6.0 is applied to allow the connection between Visual Basic and the serial port. Among the properties of MSComm that need to be considered are listed in Table 2 below.

Table 3: Microsoft Comm Control 6.0 Properties

Properties	Description
CommPort	Sets and returns the communications port number
Settings	Sets and returns the baud rate, parity, data bits and stop bits as a string
PortOpen	Sets and returns the state of a communication port. Also opens and closes a port
Input	Returns and remove character from the receive buffer
Output	Writes a string of characters to the transmit buffer
InputLen	Sets the maximum number of characters that will be returned when the input property is accessed

This project consists of two interfaces. The first interface is log in form where the user should key in the input which had been performed by barcode scanner. The second form is to display result from interface 1. Below are the screenshot of the form.

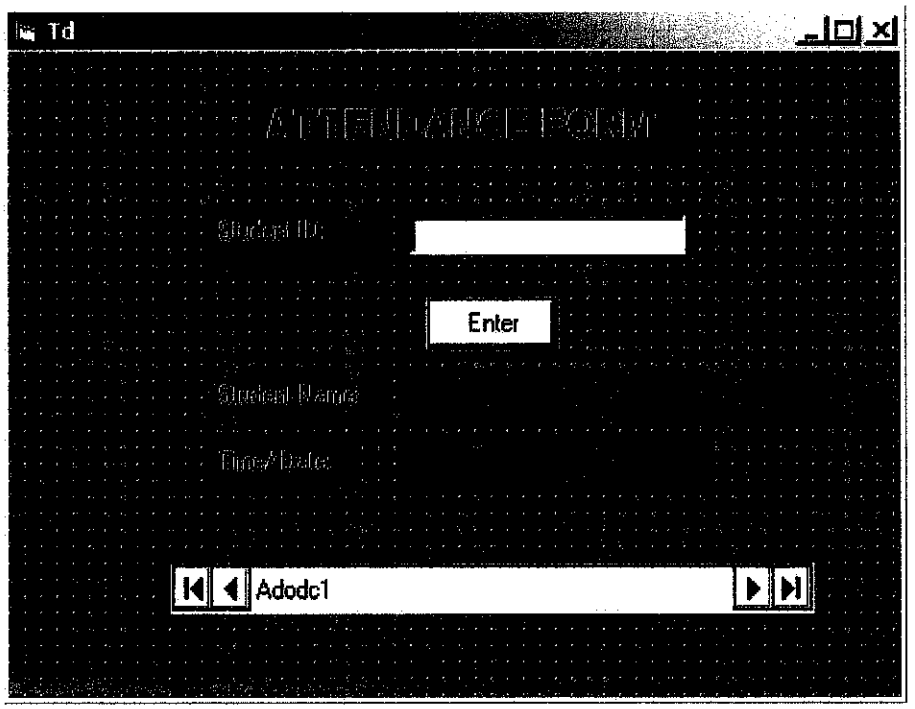


Figure 13: screenshot of form 1

The idea is when the barcode scanner scan barcode label, it will display the product number in the student ID column. The ENTER button will automatically triggered. Student name and time/date will be appearing on this form which tells us this form has been linked to the database. The ADODC1 below is the function to link the form to the database

Figure 14: screenshot of the ADODC function

The function of second interface is to display the results which consist:

a) The receiver of the SMS (parents name)

b) Receiver's phone number (parent's mobile number)

c) Information about attendance (In text)

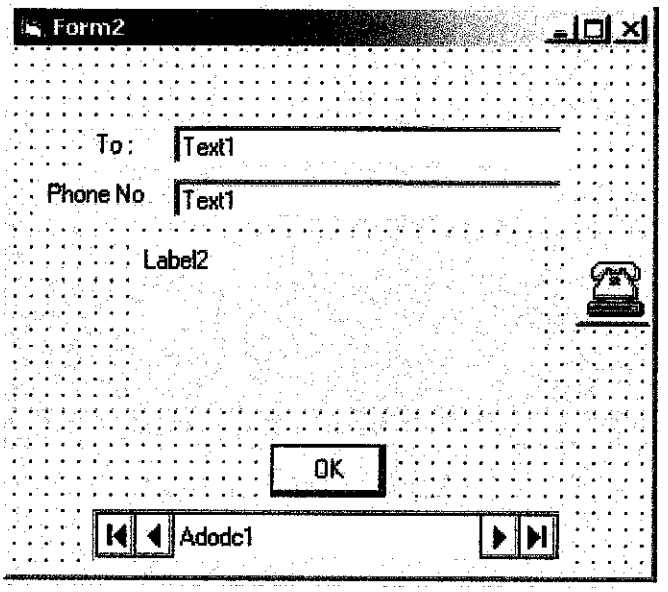


Figure 15: Screenshot of form 2

The information about attendance will be located at label 2 consists of text that will appear on the receiver's phone in SMS. The OK button will automatically triggered and this form will communicate with GSM modem through serial port.

3.2.2 Microsoft Access

The design of the database to store surveillance records are created by using the design view. Details of classes and types of data to be stored are defined to a complete database. A view from the design window is presented in Figure 13 below.

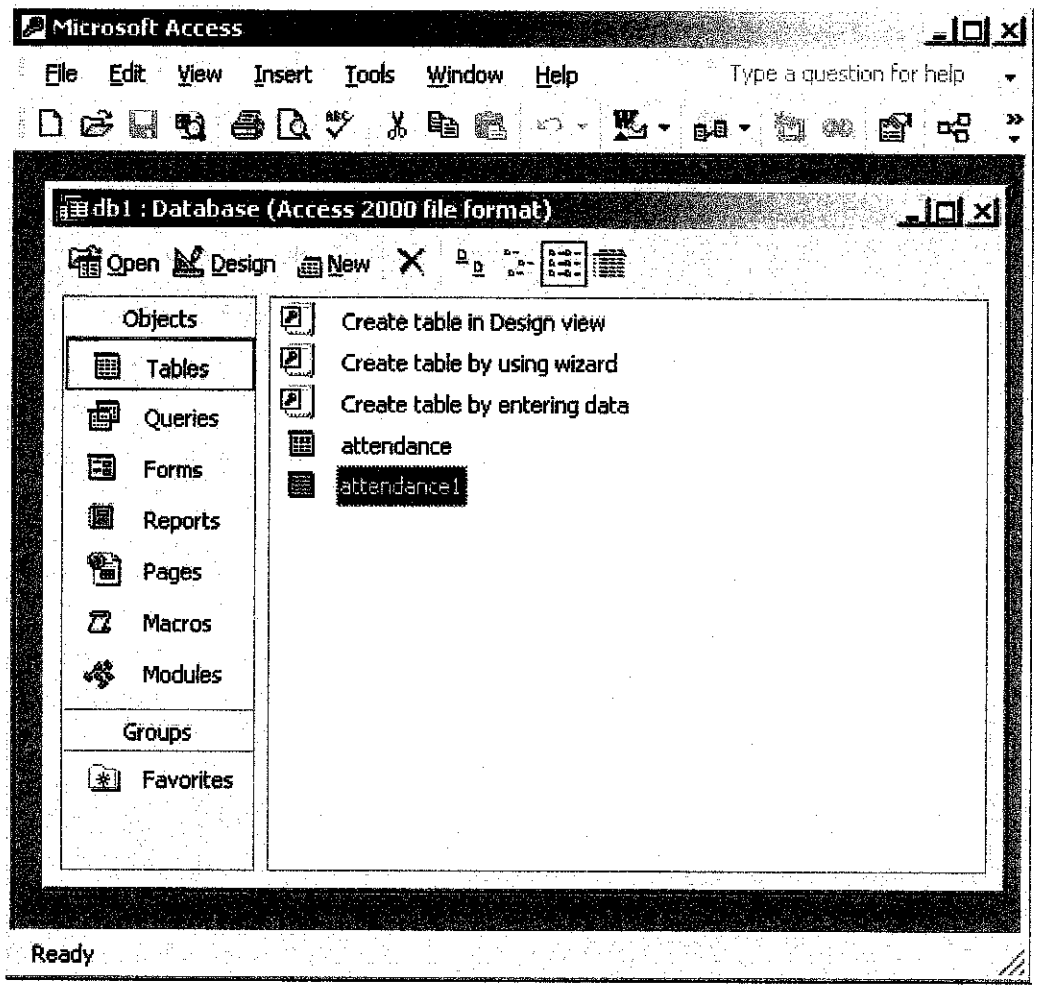


Figure16: Microsoft Access Database Design Window

3.2.3 GSM Modem

GSM Modem is connected to PC through Serial cable. GSM modem is connected to link the output from visual basic in order to send SMS to receiver. This GSM modem uses the HyperTerminal instruction that has been reprogram by visual basic in order to trigger the sequence all automatically. Below is the picture of the GSM modem that been used for this project.

Figure 17: GSM Modem

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Creating a barcode

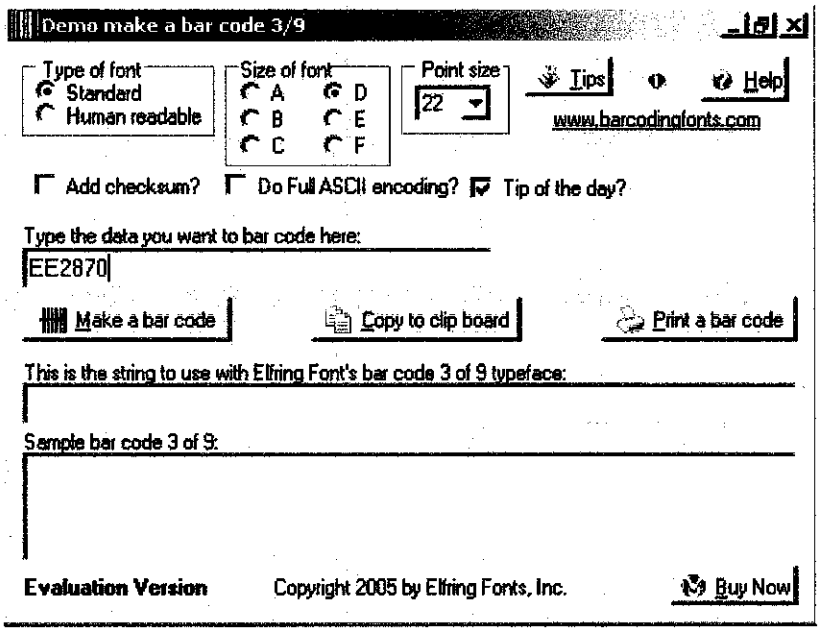


Figure 18: entering data

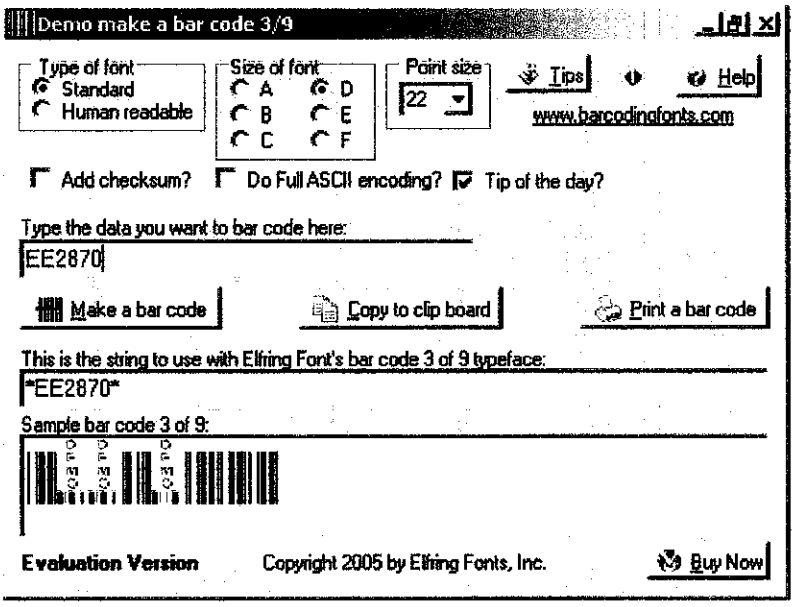


Figure19: Sample barcode is performed

4.2 Microsoft ADO Data Control 6.0 (OLEDB)

The component which allows Visual Basic to communicate with Microsoft Access is the Microsoft ADO Data Control 6.0 (OLEDB). This feature can be implemented by activating the Adodc1 component in Visual Basic. The properties of this component are presented Figure 12 below.

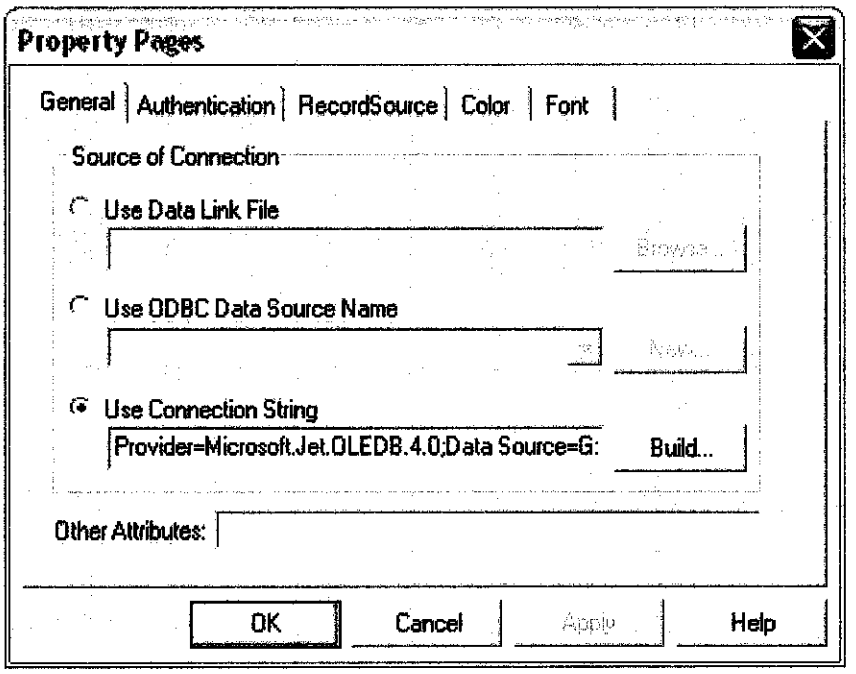


Figure 20: ADO Data Control Property

Initialization of the ADO Data Control is completed to allow Visual Basic to access any designated database from Microsoft Access. The basic operations on that database are represented by the ADO object model. The three main operations on a database are:

- Connection establishment
- Commands execution
- Information retrieval

The three important settings that must be initialized under the ADO Data Control properties are:

- Connection String; using the Microsoft Jet 4.0 OLEDB Provider, which is the normal choice for real-time application
- Connection Destination; declaring the path of the designated database
- RecordSource; setting the command type and table or stored procedure name to compliment the RecordSet

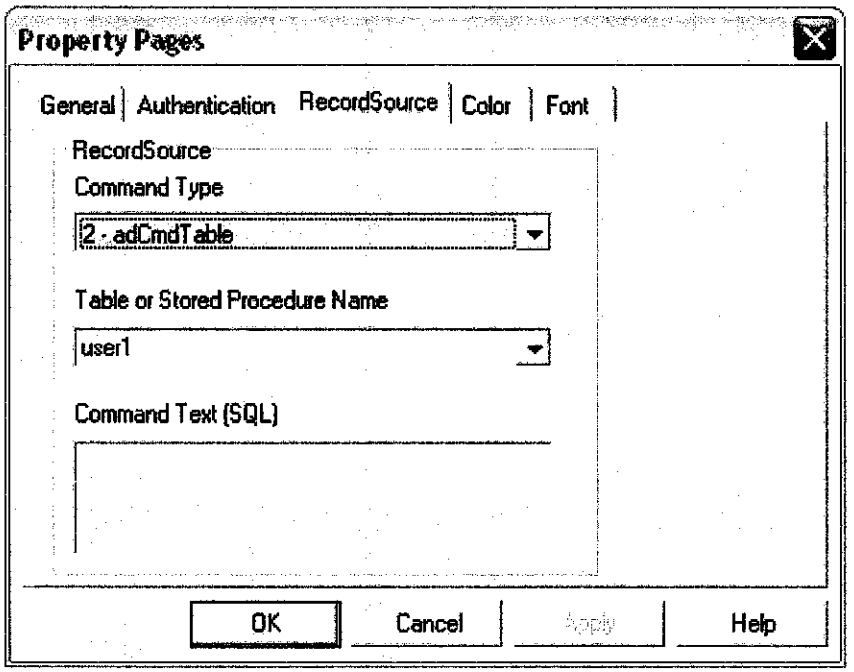


Figure 21: RecordSource Properties

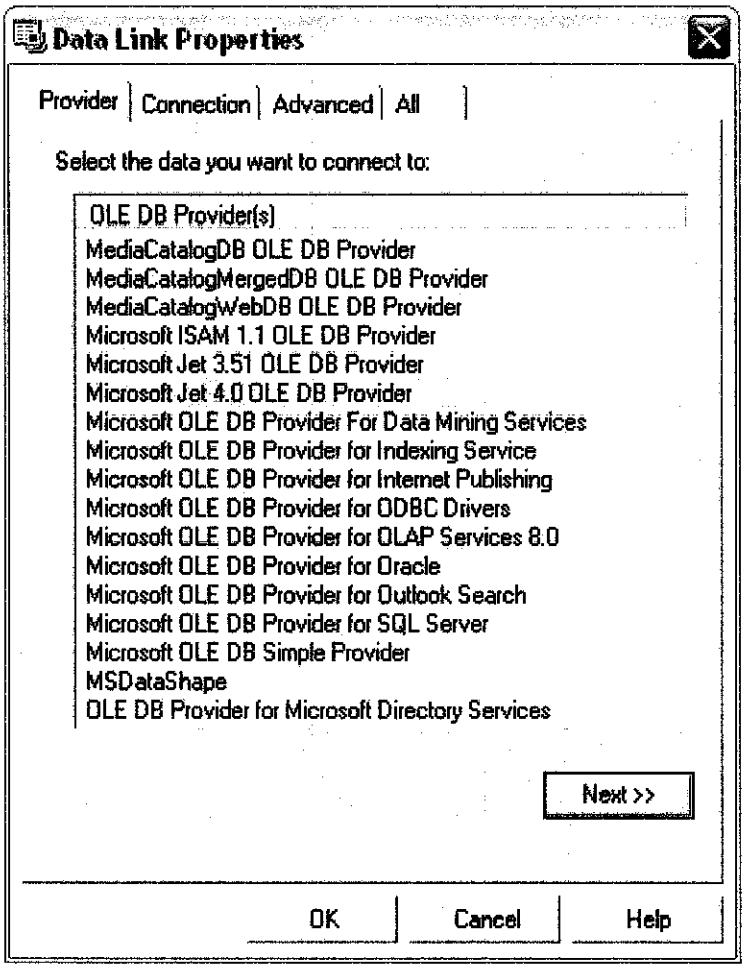


Figure 22: Data Link Properties (Provider)

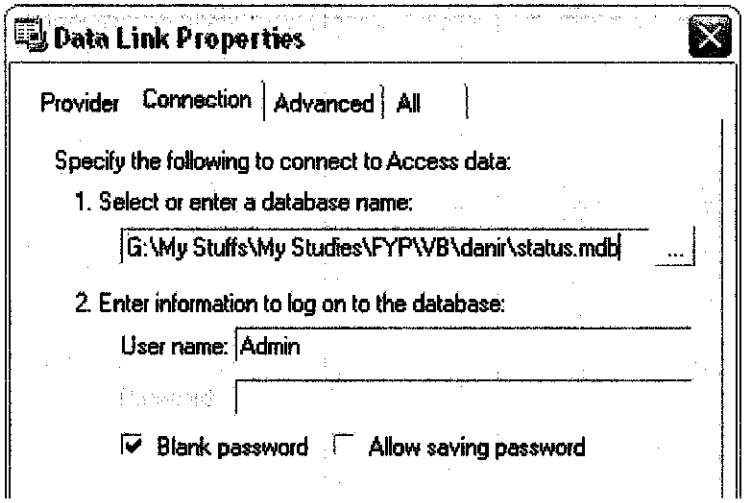


Figure 23: Data Link Properties (Database Path Declaration)

4.3 Microsoft Access for database

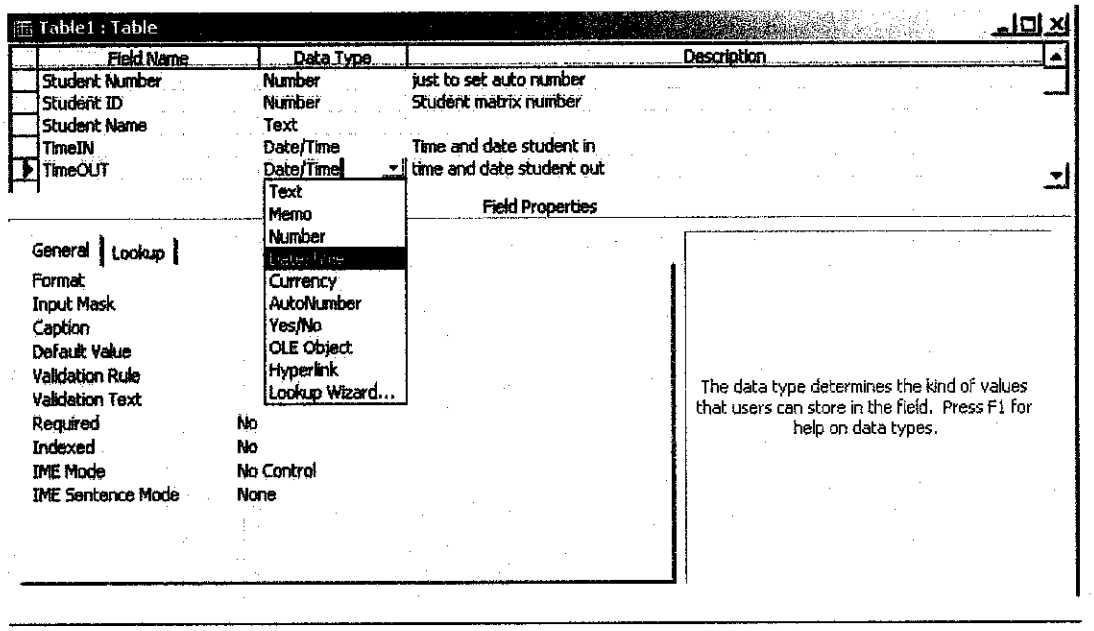


Figure24: table for database

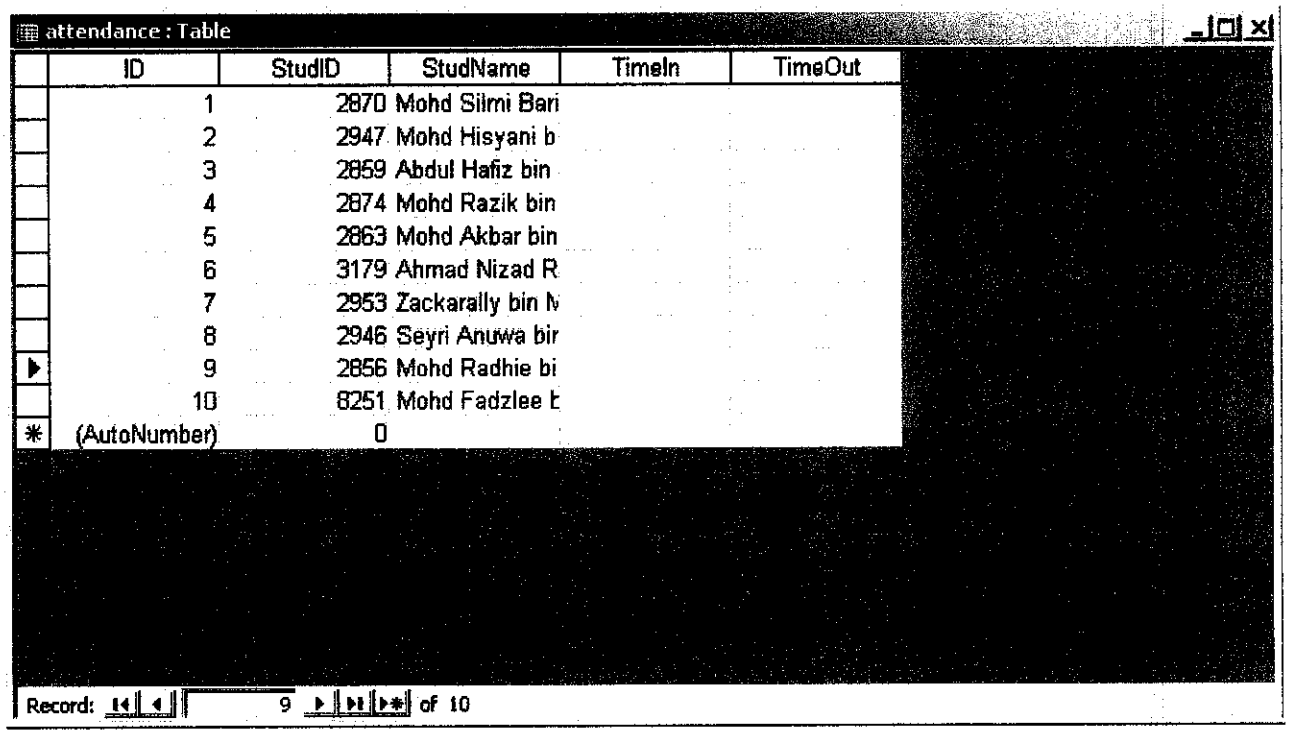


Figure 25: database stored in table 1

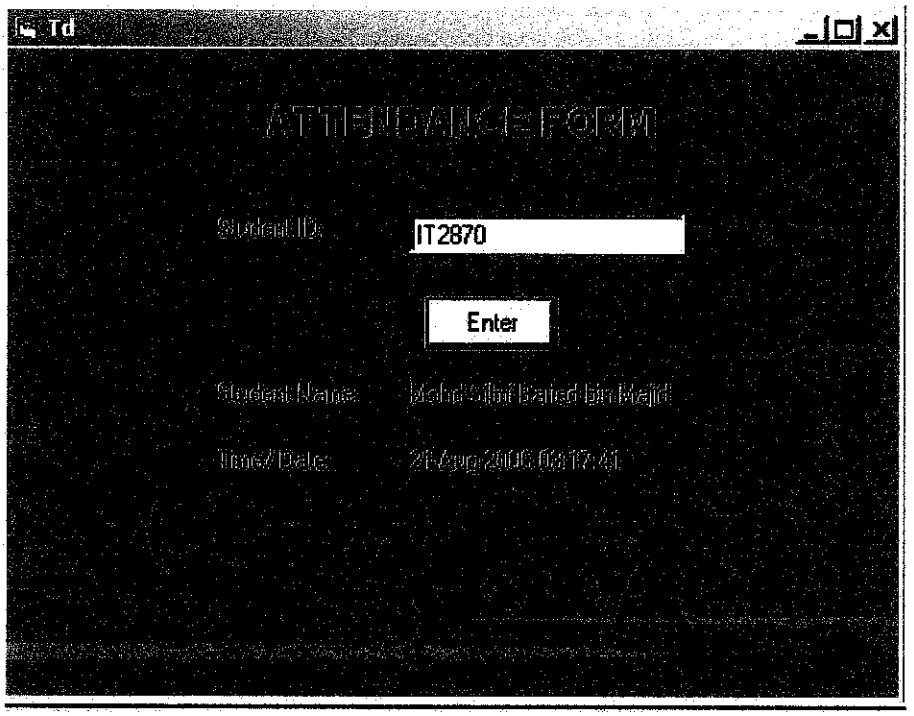
attendance1 : Table

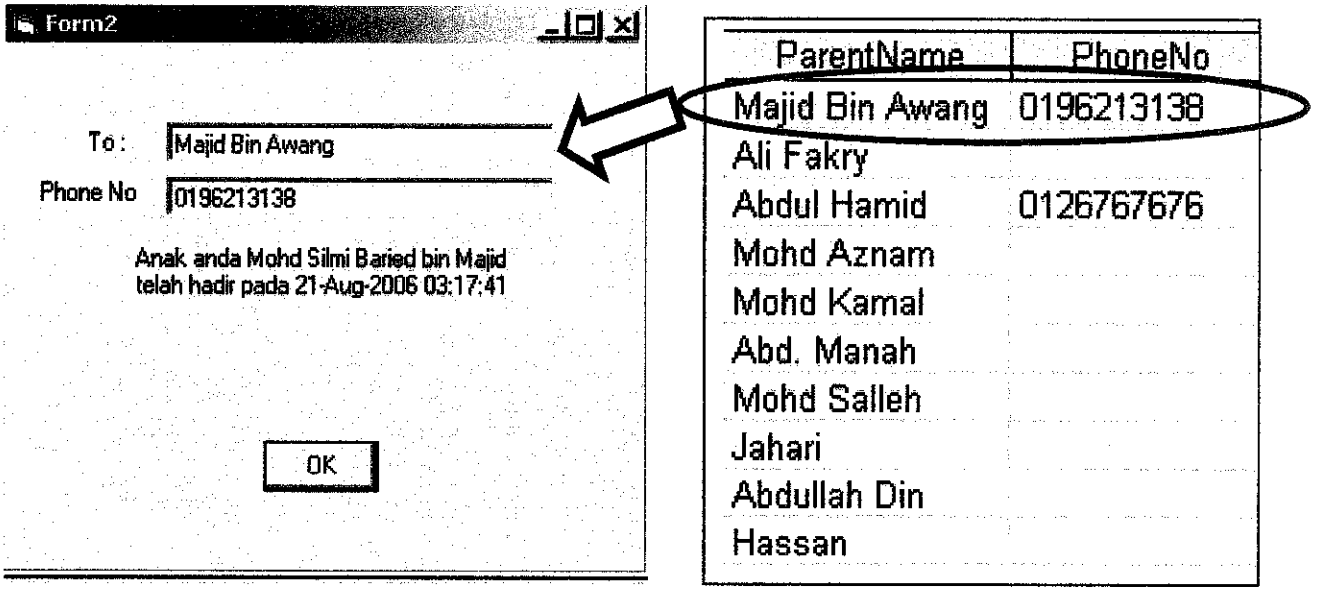
	ID1	StudID	StudName	TimeIn	TimeOut	ParentName	PhoneNo
	1	IT2870	Mohd Silmi Bari			Majid Bin Awang	0196213138
	2	IT2947	Mohd Hisyani b			Ali Fakry	
	3	IT2859	Abdul Hafiz bin			Abdul Hamid	0126767676
	4	IT2874	Mohd Razik bin			Mohd Aznam	
	5	IT2863	Mohd Akbar bin			Mohd Kamal	
	6	IT3179	Ahmad Nizad R			Abd. Manah	
	7	IT2953	Zackarally bin M			Mohd Salleh	
	8	IT2946	Seyri Anuwa bir			Jahari	
	9	IT2856	Mohd Radhie bi			Abdullah Din	
	10	IT8251	Mohd Fadzlee b			Hassan	
	11	IT3070	Mohd Shuhaiza			Hassan	
	12	IT2973	Mohd Iskandar I			Yaakob	
	13	IT2856	Mohd Radhie bi			Abdullah Din	
	14	IT2949	Mohd Syazwan			Ahmad Shafei	
	15	00025770	Mohd Afham Na			Majid Awang	
*	(AutoNumber)						

ParentName	PhoneNo
Majid Bin Awang	0196213138
Ali Fakry	
Abdul Hamid	0126767676
Mohd Aznam	
Mohd Kamal	
Abd. Manah	
Mohd Salleh	
Jahari	
Abdullah Din	
Hassan	
Hassan	
Yaakob	
Abdullah Din	
Ahmad Shafei	
Majid Awang	

Figure 26: New Column Added for Parent's Name and phone number

4.4 Visual Basic Interface





Taken from database

Figure 27: form results

4.5 Communication from Visual Basic to Serial Port for SMS

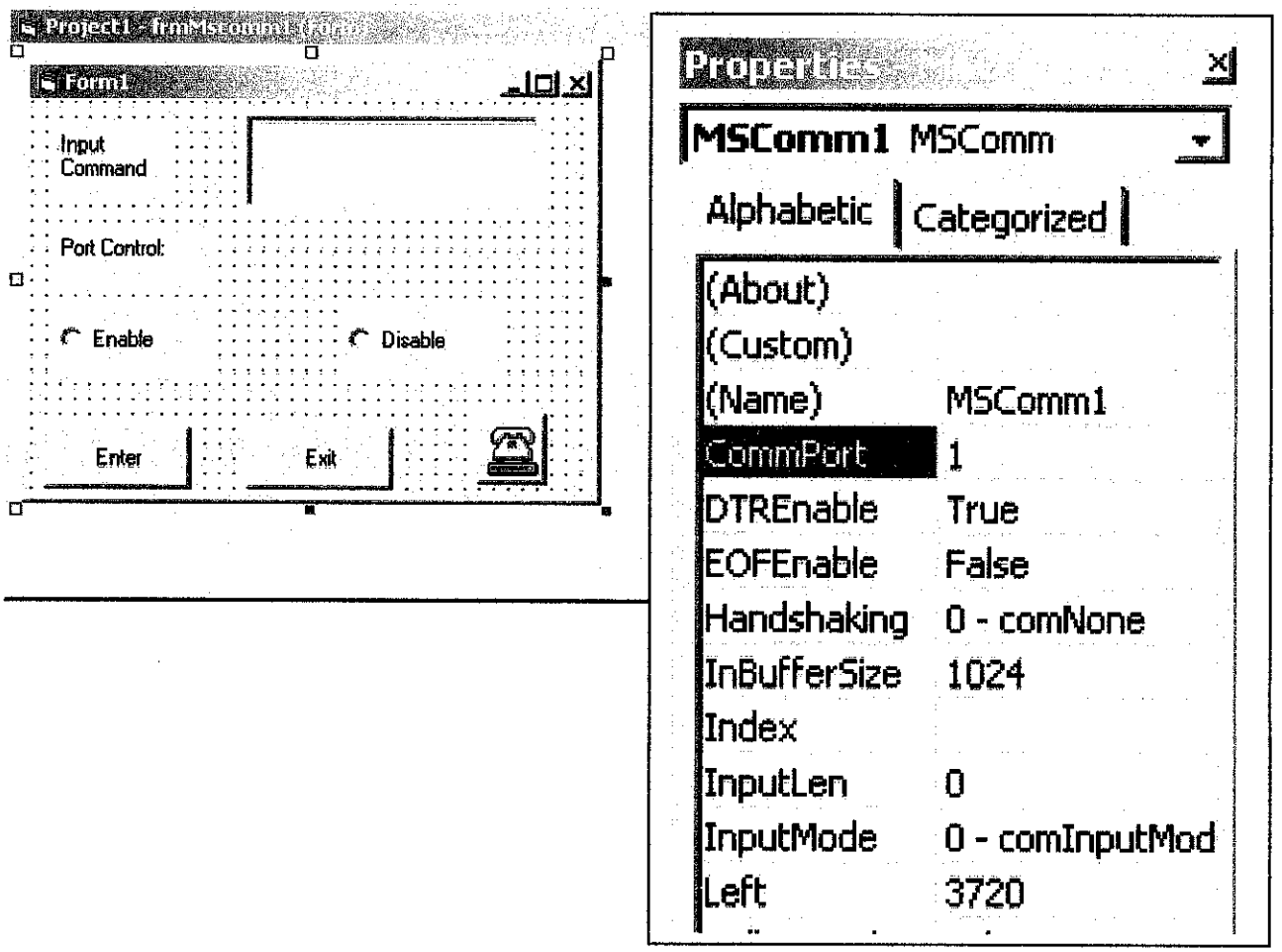


Figure 28: How to communicate between VB 6.0 to Serial Port

4.6 GSM modem

The GSM network offers a wireless infrastructure which extends your reach to anywhere in the world. There are several means to tap onto this infrastructure as a communication medium. One way is to use a direct data call to connect a point-to-point data link from one place to another. Or, to log into the World-Wide-Web via GPRS. A simple method is also via Short Message System (SMS).

Which ever means is used, a user wants to exchange or send data or information from one point to another. We will describe here of how to use an AT command on mobile phones to be as a GSM communication information node. This will be the cheapest means to create your own DIY remote messaging system.

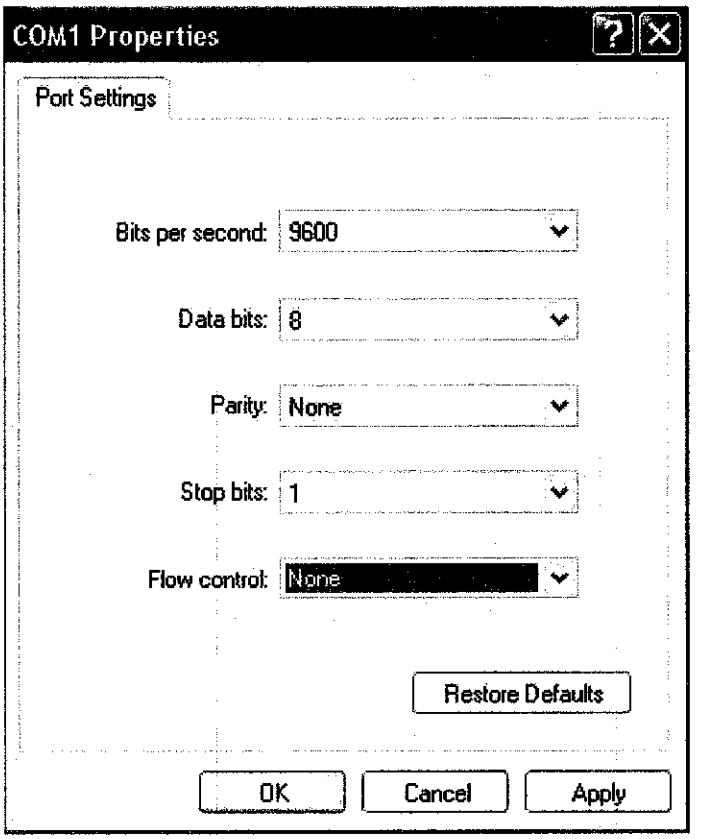


figure 29: Setting the properties at Hyper Terminal

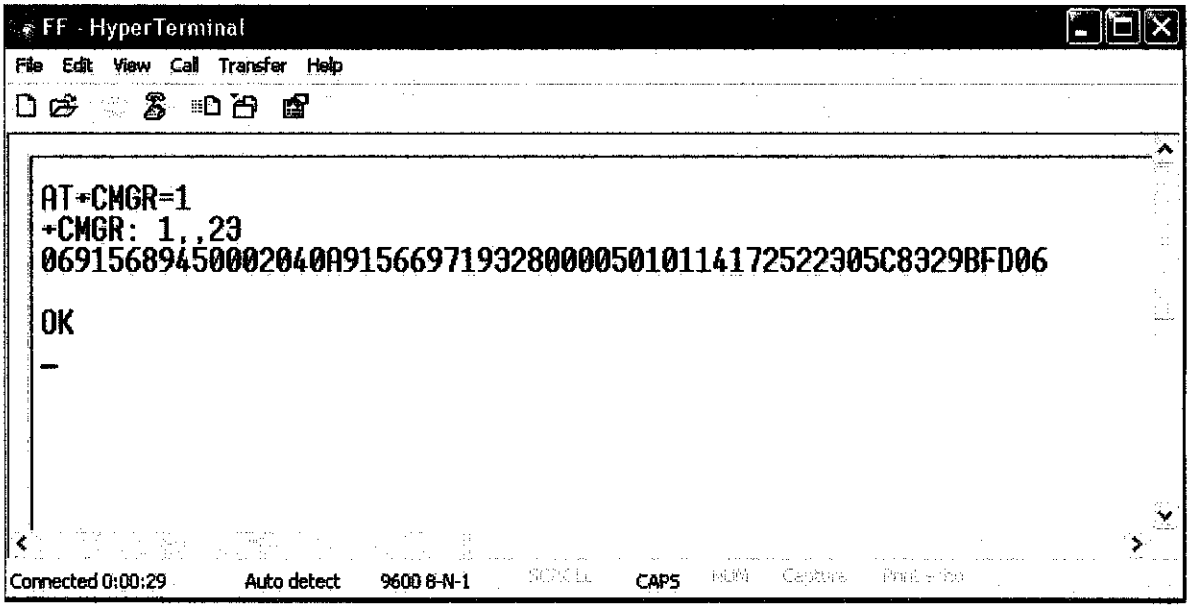


figure 30: ALT+CMGR for received messages

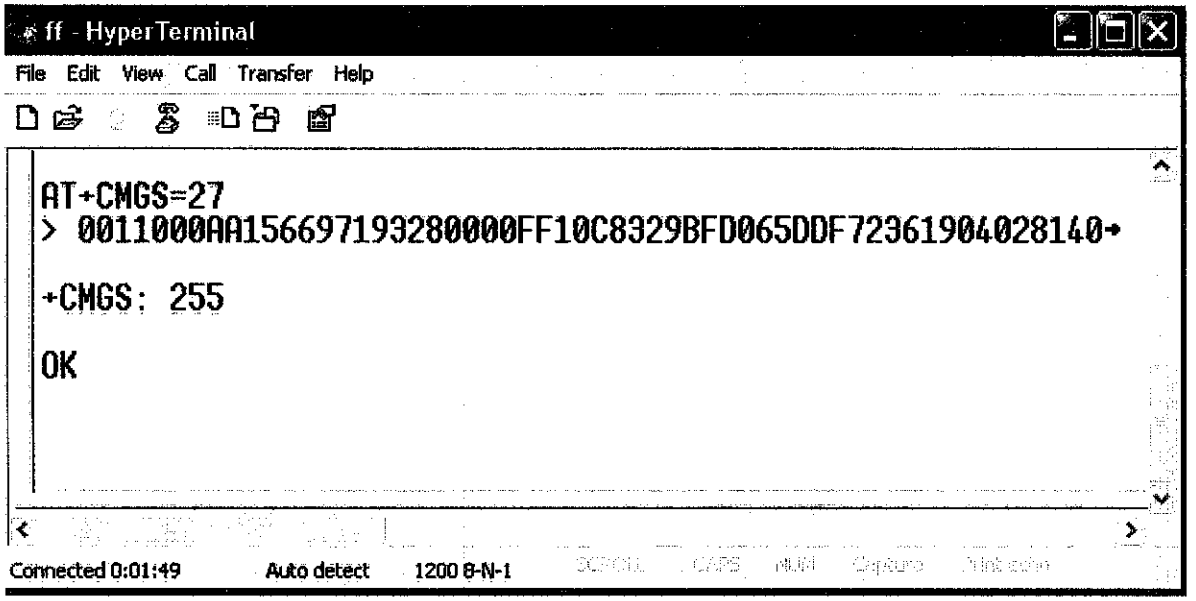


figure 31: ALT+CMGS for sending messages

CHAPTER 5
CONCLUSION AND RECOMMENDATION

Communication is very important in our everyday lives. It is now evolving from the traditional communication using radio frequency to the latest wireless communication. This project is hoped to improve school attendance system by applying the latest technology available so that we will be in the same wireless communication evolution era.

The project is a preparation for the final year student to enter the working environment. All the theories learn throughout the five years in UTP can be applied and implemented in this project. Student will also learn how to manage their time for study for other courses and completing this project. All of this experience will be very helpful in producing a well rounded and quality engineer.

REFERENCES

1. Wallace Wang, *Visual Basic 6 for Dummies*, IDG Books Worldwide, 1998
2. Evangelos Petroutsos, *Database Programming with Visual Basic 6*, SYBEX Inc., USA, 2000
3. Mike Tooley, *PC-based Instrumentation and Control*; 2nd Edition, Newnes, Great Britain, 1991
4. Myke Predko, *PC Ph.D. Inside PC Interfacing*. McGraw Hill, Washington DC, 2000.

APPENDIX A

GANTT CHART

No	Detail/Work	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
2	Preliminary Research Work														
3	Logbook preparation/ submission														
4	Submission of Preliminary report														
5	Project Work														
	•1 Reference/ Literature														
	•2 Research														
	•3 Practical/ Laboratory Work														
6	Submission of Progress Report														
7	Project Work Continue														
	•1 Practical/ Laboratory Work														
	•2 Research														
8	Submission of Interim Report Draft														
9	Submission of Interim Report														

Semester 1 Gantt chart

No	Detail/Work	1	2	3	4	5	6	7	8	9	10	11	12	13	14	20
1	Research Work															
2	Logbook preparation/ submission															
3	Submission of Progress Report 1															
4	Project Work															
	•4 Reference/ Literature															
	•5 Research															
	•6 Practical/ Laboratory Work															
5	Submission of Progress Report 2															
6	Project Work Continue															
	•3 Practical/ Laboratory Work															
7	Pre EDX Presentation															
8	Submission of Report Draft															
9	Submission of Soft Cover Dissertation															
10	Final Presentation															

Semester 2 Gantt chart

APPENDIX B

DATASHEETS

Coding for Form 1 (Visual Basic)

```
Private Sub Form_Load()
Database.Refresh
End Sub

Public Function MyTime() As String
MyTime = Format(Now, "dd-MMM-yyyy HH:nn:ss")
'& "." & Right(Format(Timer, "#0.00"), 2)
End Function

Private Sub cmdEnter_Click()
Database.Recordset.MoveFirst
Database.Recordset.Find "StudID='" & txtID.Text & "'"
If Database.Recordset.EOF Then
MsgBox "There are no such student", vbCritical
Else
lblName.Caption = Database.Recordset("StudName")
'Me.Hide


Form2.Show
'call fnction send sms
'Call sms_send(nophone, Database.Recordset("StudName") & mytime

End If
lblTime = MyTime
End Sub
```

Coding for Form 2 (Visual Basic)

```
Public Function MyTime() As String
MyTime = Format(Now, "dd-MMM-yyyy HH:mm:ss")

'& "." & Right(Format(Timer, "#0.00"), 2)
End Function

Private Sub cmdOK_Click()

form1.txtID = ""
form1.lblName = ""
form1.lblTime = ""
Form2.Text1 = ""
Form2.Text2 = ""
Form2.Label2 = ""

Unload Form2

End Sub

Private Sub Form_Load()

Dim strDate As String

Database.Refresh

Database.Recordset.MoveFirst
Database.Recordset.Find "StudID='" & form1.txtID.Text & "'"

If Database.Recordset.EOF Then
   MsgBox "There are no such student", vbCritical
Else
   Text1.Text = Database.Recordset("ParentName")
   Text2.Text = Database.Recordset("PhoneNo")

   strDate = form1.lblTime
   strDate = MyTime

   Label2.Caption = "Anak anda " & form1.lblName.Caption & " telah hadir pada " &
strDate


End If

End Sub
```

5 AT commands originating from GSM 07.05 for SMS

The SMS related AT Commands are according to the GSM 07.05 specification issued by ETSI (European Telecommunications Standards Institute).

5.1 AT+CMGC Send an SMS command

Test command
AT+CMGC=?

Response
OK

Write command
if text mode (AT+CMGF=1):
AT+CMGC=<fo>,<ct>[,<pid>[,<mn>[,<da>[,<toda>]]]]<CR>
text is entered <ctrl-Z/ESC>

Response
if text mode (+CMGF=1) and sending successful:
+CMGC: <mr>[,<scts>]
if sending fails:
+CMS ERROR: <err>

Write command
if PDU mode (AT+CMGF=0):
AT+CMGC=<length><CR>
PDU is given <ctrl-Z/ESC>

Response
if PDU mode (+CMGF=0) and sending successful:
+CMGC: <mr>[,<ackpdu>]
if sending fails:
+CMS ERROR: <err>

Test command
+CMGC=?

Parameter
<length> Length of PDU
<pdu> See "AT+CMGL"
<mr> Message reference
<fo> depending on the command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), SMS- STATUS-REPORT, or SMS -COMMAND (default 2) in integer format
<ct> GSM 03.40 TP-Command-Type in integer format (default 0)
<pid> GSM 03.40 TP-Protocol-Identifier in integer format (default 0)
<toda> GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)
<da> GSM 03.40 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <toda>
<scts> GSM 03.40 TP-Service-Centre-Time-Stamp in time-string format (refer to <dt>)

Reference
GSM 07.05

Note
- After invoking the commands CMGW, CMGS, CMGC wait for the prompt "> " before entering text or PDU.
- At baudrates below 19200 it is recommended to use the line termination character only (refer to ATS3, default <CR>, e.g. 28) before entering the text/pdu. Use of the line termination character followed by line feed character (refer to ATS4, default <LF>, e.g. 29) can cause problems.

5.2 AT+CMGD Delete SMS message

Test command AT+CMGD=?	Response **OK** Parameter
Execute command AT+CMGD= <index>	Response TA deletes message from preferred message storage <mem1> location <index>. **OK** If error is related to ME functionality: **+CMS ERROR <err>** Parameter <index> integer type; value in the range of location numbers supported by the associated memory
Reference GSM 07.05	Note If there is no SMS stored at the selected index, the response is OK too.

5.3 AT+CMGF Select SMS message format

Test command AT+CMGF=?	Response **+CMGF:** (list of supported <mode>s) **OK** Parameter See write command
Read command AT+CMGF?	Response **+CMGF: <mode> OK** Parameter See write command
Write command AT+CMGF = [<mode>]	Response TA sets parameter which specifies the input and output format of messages to be used. **OK** Parameter **<mode>** 0 PDU mode 1 text mode
Reference GSM 07.05	Note

8.4 AT+CMGL List SMS messages from preferred store

Test command

AT+CMGL=?

Response

+CMGL: (list of supported <stat>s) OK

Parameter

See execute command

Execute command

AT+CMGL[=<stat>]

Parameter

1) If text mode:

<stat>	"REC UNREAD"	Received unread messages (default)
	"REC READ"	Received read messages
	"STO UNSENT"	Stored unsent messages
	"STO SENT"	Stored sent messages
	"ALL"	All messages

2) If PDU mode:

<stat>	0	Received unread messages (default)
	1	Received read messages
	2	Stored unsent messages
	3	Stored sent messages
	4	All messages

Response

TA returns messages with status value <stat> from message storage <mem1> to the TE. If status of the message is 'received unread', status in the storage changes to 'received read'

Note: If the selected <mem1> can contain different types of SMs (e.g. SMS-DELIVERs, SMS-SUBMITs, SMS-STATUS-REPORTs and SMS-COMMANDs), the response may be a mix of the responses of different SM types. The application can recognize the response format by examining the third response parameter.

Response

1) If text mode (+CMGF=1) and command successful:

for SMS-SUBMITs and/or SMS-DELIVERs:

+CMGL: <index>,<stat>,<oa/da>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<CR><LF><data>[<CR><LF>

+CMGL: <index>,<stat>,<da/oa>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<CR><LF><data>[...]] OK

for SMS-STATUS-REPORTs:

+CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>
[<CR><LF>
+CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>
[...]] OK

for SMS-COMMANDs:
+CMGL: <index>,<stat>,<fo>,<ct>[<CR><LF>
+CMGL: <index>,<stat>,<fo>,<ct>[...]] OK

for CBM storage:
+CMGL:
<index>,<stat>,<sn>,<mid>,<page>,<pages><CR><LF><data>[<CR><LF>
+CMGL: <index>,<stat>,<sn>,<mid>,<page>,<pages>
<CR><LF><data>[...]]OK

2) If PDU mode (+CMGF=0) and command successful:
+CMGL: <index>,<stat>,[<alpha>],<length><CR><LF><pdu>
[<CR><LF>+CMGL: <index>,<stat>,[alpha],<length><CR><LF><pdu>
[...]] OK

for CBM storage:
+CMGL: <index>,<length><CR><LF><pdu>

3) If error is related to ME functionality:
+CMS ERROR: <err>

Parameter

<alpha> string type alphanumeric representation of <da> or <oa> corresponding to the entry found in phonebook; implementation of this feature is manufacturer- specific

<ct> GSM 03.40 TP-Command-Type in integer format (default 0)

<da> GSM 03.40 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <toda>

<data>

In case of SMS: GSM 03.40 TP-User-Data in text mode responses; format:

- if <dcs> indicates that GSM 03.38 default alphabet is used and <fo> indicates that GSM 03.40 TP-User-Data-Header-Indication is not set: ME/TA converts GSM alphabet into current TE character set according to rules of Annex A
- if <dcs> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that GSM 03.40 TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65))

In the case of CBS: GSM 03.41 CBM Content of Message in text mode responses, format:

- if <dcs> indicates that GSM 03.38 default alphabet is used: ME/TA converts GSM alphabet into current TE character set according to rules of Annex A
- if <dcs> indicates that 8-bit or UCS2 data coding scheme is used: ME/TA converts each 8-bit octet into hexadecimal numbers containing two IRA characters

Parameter	
<dt>	GSM 03.40 TP-Discharge-Time in time-string format: "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone. For example, 6th of May 1994, 22:10:00 GMT+2 hours equals "94/05/06,22:10:00+08"
<fo>	depending on the command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), SMS- STATUS-REPORT, or SMS -COMMAND (default 2) in integer format
<length>	integer type value indicating in text mode (+CMGF=1) the length of the message body <data> (or <cdata>) in characters; or in PDU mode (+CMGF=0), the length of the actual TP data unit in octets (i.e. the RP layer SMSC address octets are not counted in the length)
<index>	integer type; value in the range of location numbers supported by the associated memory
<mid>	GSM 03.41 CBM Message Identifier in integer format
<mr>	GSM 03.40 TP-Message-Reference in integer format
<oa>	GSM 03.40 TP-Originating-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <tooa>
<pages>	GSM 03.41 CBM Page Parameter bits 0-3 in integer format
<pdu>	In the case of SMS: GSM 04.11 SC address followed by GSM 03.40 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: GSM 03.41 TPDU in hexadecimal format.
<page>	GSM 03.41 CBM Page Parameter bits 4-7 in integer format
<ra>	GSM 03.40 TP-Recipient-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <tora>
<scts>	GSM 03.40 TP- Service Centre Time-Stamp in time-string format (refer <dt>)
<sn>	GSM 03.41 CBM Serial Number in integer format
<st>	GSM 03.40 TP-Status in integer format
<toda>	GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)
<tooa>	GSM 04.11 TP-Originating-Address Type-of-Address octet in integer format (default refer<toda>)
<tora>	GSM 04.11 TP-Recipient-Address Type-of-Address octet in integer format (default refer<toda>)

Reference	Note
GSM 07.05	The parameters <ra> and <tora> will only be displayed if AT^SSCONF=1 has been set before. See Chapter 8.37 for details on AT^SSCONF.

6.5.7 AT+CMGR Read SMS message

Test command AT+CMGR=?	Response OK
Execute command AT+CMGR= <index>	Parameter <index> integer type; value in the range of location numbers supported by the associated memory Response TA returns SMS message with location value <index> from message storage <mem1> to the TE. If status of the message is 'received unread', status in the storage changes to 'received read'. 1) If text mode (+CMGF=1) and command successful: for SMS-DELIVER: +CMGR: <stat>,<oa>,[<alpha>],<scts> [,<toda>,<fo>,<pid>,<dcs>, <sca>,<tosca>,<length>]<CR><LF><data> for SMS-SUBMIT: +CMGR: <stat>,<da>,[<alpha>] [,<toda>,<fo>,<pid>,<dcs>,[<vp>], <sca>,<tosca>,<length>]<CR><LF><data> for SMS-STATUS-REPORT: +CMGR: <stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> for SMS- COMMAND: +CMGR: <stat>,<fo>,<ct> [,<pid>,[<mn>],[<da>],[<toda>],<length> <CR><LF><cdata>] for CBM storage: +CMGR: <stat>,<sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data> 2) If PDU mode (+CMGF=0) and command successful: +CMGR: <stat>,[<alpha>],<length><CR><LF><pdu> OK for CBM storage: +CMGR: <length><CR><LF><pdu> 3)If error is related to ME functionality: +CMS ERROR: <err> Parameter <alpha> string type alphanumeric representation of <da> or <oa> corresponding to the entry found in phonebook; implementation of this feature is manufacturer specific <stat> integer type in PDU mode (default 0), or string type in text mode (default "REC UNREAD"); indicates the status of message in memory: defined values:

0 "REC UNREAD" received unread message (i.e. new message)
1 "REC READ" received read message
2 "STO UNSENT" stored unsent message (only applicable to SMs)
3 "STO SENT" stored sent message (only applicable to SMs)

<ct> GSM 03.40 TP-Command-Type in integer format (default 0)

<da> GSM 03.40 TP- Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <toda>

<data>
In case of SMS: GSM 03.40 TP-User-Data in text mode responses; format:
-if <dcs> indicates that GSM 03.38 default alphabet is used and <fo> indicates that GSM 03.40 TP-User-Data-Header-Indication is not set: ME/TA converts GSM alphabet into current TE character set according to rules covered in Annex A
-if <dcs> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that GSM 03.40 TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)

In case of CBS: GSM 03.41 CBM Content of Message in text mode responses; format:
- if <dcs> indicates that GSM 03.38 default alphabet is used: ME/TA converts GSM alphabet into current TE character set according to rules covered in Annex A
-if <dcs> indicates that 8-bit or UCS2 data coding scheme is used: ME/TA converts each 8-bit octet into hexadecimal numbers containing two IRA characters

<dcs> depending on the command or result code: GSM 03.38 SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format

<cdata> GSM 03.40 TP-Command-Data in text mode responses; ME/TA converts each 8-bit octet into two IRA character long hexadecimal numbers (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65))

<dt> GSM 03.40 TP-Discharge-Time in time-string format: "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone. For example, 6th of May 1994, 22:10:00 GMT+2 hours equals "94/05/06,22:10:00+08"

<fo> depending on the command or result code: first octet of GSM 03.40 SMS- DELIVER, SMS-SUBMIT (default 17), SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format

<length> integer type value indicating in text mode (+CMGF=1) the length of the message body <data> (or <cdata>) in characters; or in PDU mode (+CMGF=0), the length of the actual TP data unit in octets (i.e. the RP layer SMSC address octets are not counted in the length).
In text mode, the maximum length of an SMS depends on the used coding scheme. It is **160** characters if the 7 bit GSM coding scheme is used, and **140** characters according to the 8 bit GSM coding scheme.

<index>	integer type; value in the range of location numbers supported by the associated memory
<mid>	GSM 03.41 CBM Message Identifier in integer format
<mr>	GSM 03.40 TP-Message-Reference in integer format
<oa>	GSM 03.40 TP-Originating-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <tooa>
<page>	GSM 03.41 CBM Page Parameter bits 4-7 in integer format
<pages>	GSM 03.41 CBM Page Parameter bits 0-3 in integer format
<pdu>	In the case of SMS: GSM 04.11 SC address followed by GSM 03.40 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: <ra> GSM 03.40 TP-Recipient-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <tora>
<pid>	GSM 03.40 TP-Protocol-Identifier in integer format (default 0)
<ra>	GSM 03.40 TP-Recipient-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted to characters of the currently selected TE character set (refer command AT+CSCS Select TE character set.); type of address given by <tora>
<sca>	GSM 04.11 RP SC address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted to characters of the currently selected TE character set (refer command AT+CSCS Select TE character set); type of address given by <tosca>
<scts>	GSM 03.40 TP-Service-Centre-Time-Stamp in time-string format (refer <dt>)
<sn>	GSM 03.41 CBM Serial Number in integer format
<st>	GSM 03.40 TP-Status in integer format
<toda>	GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)
<tooa>	GSM 04.11 TP-Originating-Address Type-of-Address octet in integer format (default refer<toda>)
<tora>	GSM 04.11 TP-Recipient-Address Type-of-Address octet in integer format (default refer<toda>)
<tosca>	GSM 04.11 RP SC address Type-of-Address octet in integer format (default refer <toda>)
<vp>	depending on SMS-SUBMIT <fo> setting: GSM 03.40 TP-Validity-Period either in integer format (default 167) or in time-string format (refer <dt>)

Reference

GSM 07.05

Note

- Response to a CMGR to an empty record index: +CMGR: 0,,0
- Response to a CMGR to a not existing record index: +CMS ERROR: invalid memory index
- The parameters <ra> and <tora> will only be displayed if AT^SSCONF=1 has been set before. See Chapter 6.37 for details on AT^SSCONF.

5.6 AT+CMGS: Send SMS message

Test command	Response
AT+CMGS=?	OK Parameter

Execute command	Response
1) If text mode (+CMGF=1): +CMGS=<da>[,<toda>]<CR> text is entered <ctrl-Z/ESC>	TA transmits SMS message from TE to network (SMS-SUBMIT). Message reference value <mr> is returned to TE on successful message delivery. Value can be used to identify message upon unsolicited delivery status report result code. 1) If text mode (+CMGF=1) and sending successful: +CMGS: <mr>[,<scts>] OK
2) If PDU mode (+CMGF=0): +CMGS=<length> <CR> PDU is given <ctrl-Z/ESC> ESC aborts message	2) If PDU mode (+CMGF=0) and sending successful: +CMGS: <mr>[,<ackpdu>] OK 3) If error is related to ME functionality: +CMS ERROR: <err> For example, if a message was too long <err> code 305 ("Invalid text mode parameter") is returned.

Parameter

<da>	GSM 03.40 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <toda>
<toda>	GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)
<length>	integer type value indicating in PDU mode (+CMGF=0), the length of the actual TP data unit in octets (i.e. the RP layer SMSC address octets are not counted in the length).
<mr>	GSM 03.40 TP-Message-Reference in integer format
<scts>	GSM 03.40 TP-Service-Centre-Time-Stamp in time-string format (refer <dt>)
<dt>	GSM 03.40 TP-Discharge-Time in time-string format: "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone. For example, 6th of May 1994, 22:10:00 GMT+2 hours equals "94/05/06,22:10:00+08"
<ackpdu>	GSM 03.40 RP-User-Data element of RP-ACK PDU; format is same as for <pdu> in case of SMS, but without GSM 04.11 SC address field and parameter shall be enclosed in double quote characters like a normal string type parameter
<pdu>	For SMS: GSM 04.11 SC address followed by GSM 03.40 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: GSM 03.41 TPDU in hexadecimal format.

Reference	Note
GSM 07.05	• After invoking the commands CMGW, CMGS, CMGC wait for the prompt "> " and then start to send text to the module. • To send the message simply enter <CTRL-Z>. See Execute command for possible responses. • Sending can be aborted by entering <ESC>. Of course, the message will not be sent, though the operation is acknowledged with OK. • Sending e-mails via SMS: Note that some providers do not recognize @ symbol. Possible alternative "!" for "@" • At baudrates lower than 19200 it is recommended to use the line termination character only (refer to +ATS3, default <CR>, pg. 28) before entering the text/pdu. Use of the line termination character followed by the response formating character (see +ATS4, default <LF>, pg. 28) can cause problems. • All characters entered behind the ">" prompt will be recognized as GSM characters. For example, "Backspace" (ASCII character 8) does not delete a character, but will be inserted into the SMS as an additional physical character. As a result, the character you wanted to delete still appears in the text, plus the GSM code equivalent of the Backspace key. See also Chapter 7.5 which provides the supported alphabet tables. • In text mode, the maximum length of an SMS depends on the used coding scheme. It is **160** characters if the 7 bit GSM coding scheme is used, and **140** characters according to the 8 bit GSM coding scheme.

3.7 AT+CMGW Write SMS message to memory

Test command	Response
AT+CMGW=?	**OK**

Execute command	Response
1) If text mode (+CMGF=1): +CMGW[=<oa/da> [,tooa/toda>[,stat>]]] <CR> text is entered ctrl-Z/ESC><ESC> quits without sending	TA transmits SMS (either SMS-DELIVER or SMS-SUBMIT) from TE to memory storage <mem2>. Memory location <index> of the stored message is returned. Message status will be set to 'stored unsent' unless otherwise given in parameter <stat>. Note: SMS-COMMANDs and SMS-STATUS-REPORTs cannot be stored in text mode.
2) If PDU mode (+CMGF=0): +CMGW=<length> [,stat]<CR> PDU is given <ctrl-Z/ESC>	If writing is successful: **+CMGW: <index> OK** If writing is not successful: **OK** If writing fails, for example if a message was too long or writing was aborted, ME simply returns **OK** instead of an ERROR code. Users should be aware that, in this case, the message will not be written to the SIM card. This behaviour has been implemented for compatibility to M20 (Siemens GSM Terminal). To verify whether or not a message was stored check for **+CMGW: <index> OK** as described above. If error is related to ME functionality: **+CMS ERROR: <err>**

Parameter

<oa>	GSM 03.40 TP-Originating-Address Address value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by **<tooa>**
<da>	GSM 03.40 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by **<toda>**
<tooa>	GSM 04.11 TP-Originating-Address Type-of-Address octet in integer format (default refer **<toda>**)
<toda>	GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of **<da>** is + (IRA 43) default is 145, otherwise default is 129)
<length>	integer type value indicating in PDU mode (+CMGF=0), the length of the actual TP data unit in octets (i.e. the RP layer SMSC address octets are not counted in the length).
<stat>	integer type in PDU mode (default 0), or string type in text mode (defauld "REC UNREAD"); indicates the status of message in memory; defined values: 0 "REC UNREAD" Received unread messages (default) 1 "REC READ" Received read messages 2 "STO UNSENT" Stored unsent messages 3 "STO SENT" Stored sent messages

<pdu> In the case of SMS: GSM 04.11 SC address followed by GSM 03.40 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into hexadecimal numbers containing two IRA characters (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: GSM 03.41 TPDU in hexadecimal format.

<index> Index of message in selected storage <mem2>

Reference

GSM 07.05

Note

- After invoking the commands CMGW, CMGS, CMGC wait for the prompt ">" and then start to send text to the module.
- To store the message simply enter <CTRL-Z>. See Execute command for possible responses.
- Writing can be aborted by entering <ESC>. Of course, the message will not be stored, though the operation is acknowledged with OK.
- When sending e-mails via SMS the @ character may be replaced with "*" as defined in GSM 03.40 (3GPP TS 23.040).
- At baudrates lower than 19200 it is recommended to use the line termination character only (refer to +ATS3, default <CR>, pg. 28) before entering the text/pdu. Use of the line termination character followed by the response formating character (refer to +ATS4, default <LF>, pg. 28) can cause problems.
- All characters entered behind the ">" prompt will be recognized as GSM characters. For example, "Backspace" (ASCII character 8) does not delete a character, but will be inserted into the SMS as an additional physical character. As a result, the character you wanted to delete still appears in the text, plus the GSM code equivalent of the Backspace key. See Chapter 7.5 which provides the supported alphabet tables. Also refer to Chapter 1.5 for general remarks on character sets.
- In text mode, the maximum length of an SMS depends on the used coding scheme: It is **160** characters if the 7 bit GSM coding scheme is used, and **140** characters according to the 8 bit GSM coding scheme.

5.8 AT+CMSS: Send SMS message from storage

Test command	Response
AT+CMSS=?	OK Parameter
Execute command	**Response**
+CMSS= <index>[,<da>[,<toda>]]	TA sends message with location value <index> from message storage <mem2> to the network (SMS-SUBMIT or SMS-COMMAND). If new recipient address <da> is given for SMS-SUBMIT, it shall be used instead of the one stored with the message. Reference value <mr> is returned to the TE on successful message delivery. Values can be used to identify message upon unsolicited delivery status report result code. 1) If text mode (+CMGF=1) and send successful: +CMSS: <mr>[,<scts>] OK 2) If PDU mode (+CMGF=0) and send successful: +CMSS: <mr>[,<ackpdu>] OK 3) If error is related to ME functionality: +CMS ERROR: <err> Parameter
Reference	Note
GSM 07.05	

<ackpdu>	GSM 03.40 RP-User-Data element of RP-ACK PDU; format is same as for <pdu> in case of SMS, but without GSM 04.11 SC address field and parameter shall be bounded by double quote characters like a normal string type parameter.
<index>	integer type; value in the range of location numbers supported by the associated memory
<da>	GSM 03.40 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <toda>
<scts>	GSM 03.40 TP-Service-Centre-Time-Stamp in time-string format.
<toda>	GSM 04.11 TP-Destination-Address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)
<mr>	GSM 03.40 TP-Message-Reference in integer format

9.9 AT+CNMA New SMS message acknowledge to ME/TE, only phase 2+

Test command	Response
AT+CNMA=?	1) If text mode (+CMGF=1): **OK** 2) If PDU mode (+CMGF=0): **+CNMA:** (list of supported <n>s) **OK** Parameters See execute command
Execute command	**Response**
1) If text mode: AT+CNMA 2) If PDU mode: AT+CNMA[=<n>]	TA confirms successful receipt of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE. TA shall not send another +CMT or +CDS result code to TE until previous one is acknowledged. If ME does not receive acknowledgment within required time (network timeout), ME sends RP-ERROR to the network. TA shall automatically disable routing to TE by setting both <mt> and <ds> values of +CNMI to zero. Note: The command shall o n l y be used when +CSMS parameter <service> equals 1 (= phase 2+). 1) If text mode: **OK** 2) If PDU mode: **OK** 3) If error is related to ME functionality: **+CMS ERROR:** <err> Parameters <n> 0 command operates similarly as defined for the text mode
Reference	**Note**
GSM 07.05	If multiplex mode is activated (+CMUX=0) the +CNMI parameter will be set to zero on all channels, if one channel fails to acknowledge an incoming message within the required time.

5.10 AT+CNMI New SMS message indications

Test command	Response
AT+CNMI=?	**+CNMI:** (list of supported **<mode>**s), (list of supported **<mt>**s), (list of supported **<bm>**s), (list of supported **<ds>**s), (list of supported **<bfr>**s) **OK**
	Parameter
	See set command
Read command	Response
AT+CNMI?	**+CNMI: <mode>,<mt>,<bm>,<ds>,<bfr> OK**
	Parameter
	See set command
Write command	Response
AT+CNMI = [<mode>] [,<mt>][,<bm>] [,<ds>][,<bfr>]	TA selects the procedure how the receipt of new SMS messages from the network is indicated to the TE when TE is active, e.g. DTR signal is ON. If TE is inactive (e.g. DTR signal is OFF), the reception of messages shall be performed as specified in GSM 03.38. Note1: If the DTR signal is not available or the state of the signal is ignored (V.25ter command &D0), reliable message transfer can be assured by using +CNMA acknowledgment procedure. Note2: The rules **<mt>**=2 and **<mt>**=3 for storing received SM are possible only if phase 2+ compatibility is activated with **+CSMS=1** Note3: The parameter **<ds>**=1 is only available in phase 2+ **OK** If error is related to ME functionality: **+CMS ERROR: <err>**

Parameter

<mode>	0	Buffer unsolicited result codes in the TA. If TA result code buffer is full, indications can be buffered in some other place or the oldest indications may be discarded and replaced with the new received indications.
	1	Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved (e.g. in on-line data mode). Otherwise forward them directly to the TE.
	2	Buffer unsolicited result codes in the TA when TA-TE link is reserved (e.g. in on-line data mode) and flush them to the TE after reservation. Otherwise forward them directly to the TE.
	3	Forward unsolicited result codes directly to the TE. TA-TE link specific inband technique used to embed result codes and data when TA is in on-line data mode.
<mt>		Rules for storing received SMS depend on the relevant data coding method (refer to GSM 03.38 [2]), preferred memory storage (+CPMS) setting and this value Note: If AT command interface is acting as the only display device, the ME must support storage of class 0 messages and messages in the message waiting indication group (discard message)
	0	No SMS-DELIVER indications are routed to the TE.
	1	If SMS-DELIVER is stored in ME/TA, indication of the memory location is routed to the TE using unsolicited result code: **+CMTI: <mem>,<index>**

		2	SMS-DELIVERs, except class 2 messages and messages in the message waiting indication group (store message) are routed directly to the TE using unsolicited result code: +CMT: ,<length><CR><LF><pdu> (PDU mode enabled) +CMT: <oa>,, <scts> [,<tooa>, <fo>, <pid>, <dcs>, <sca>, <tosca>, <length>] <CR> <LF> <data> (text mode enabled)
		3	Class 3 SMS-DELIVERs are routed directly to the TE using unsolicited result codes defined in <mt>=2. Messages of other data coding schemes result in indication as defined in <mt>=1.
	<bm>		Rules for storing received CBMs depend on the relevant data coding method (refer to GSM 03.38 [2]), the setting of Select CBM Types (+CSCB) and this value:
		0	No CBM indications are routed to the TE.
		2	New CBMs are routed directly to the TE using unsolicited result code: +CBM: <length><CR><LF><pdu> (PDU mode enabled) or +CBM: <sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data> (text mode enabled).
		3	Class 3 CBMs are routed directly to TE using unsolicited result codes defined in <bm>=2.
	<ds>	0	No SMS-STATUS-REPORTs are routed to the TE.
		1	SMS-STATUS-REPORTs are routed to the TE using unsolicited result code: +CDS: <length><CR><LF><pdu> (PDU mode enabled) or +CDS: <fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> (text mode enabled)
		2	If SMS-STATUS-REPORT is routed into ME/TA, indication of the memory location is routed to the TE using unsolicited result code: +CDSI: <mem>,<index>
	<bfr>	1	TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1...3 is entered.
Unsolicited result code	Syntax of responses output when SMS is received: +CMTI: <mem>,<index> Indicates that new message has been received +CBMI: <mem>,<index> Indicates that new CB message has been received +CMT: ,<length><CR><LF><pdu> Short message is output directly +CBM: <length><CR><LF><pdu> Cell broadcast message is output directly During each SMS or Cell Broadcast Messages the Ring Line goes Logic "1" for one second.		

Reference
GSM 07.05

General remarks:

- Parameters <mt>=2,3 and <ds>=1 are only available with GSM phase 2+ (see +CSMS=1). Incoming SMs or Status Reports have to be acknowledged with AT+CNMA=0 when using these phase 2+ parameters.
- The parameters <ra> and <tora> will only be displayed if AT^SSCONF=1 has been set before. See Chapter 6.37 for details on AT^SSCONF.

Handling of Class 0 short messages:

- If the host application is provided with a display and AT^SSDA=1 has been set Class 0 short messages can be displayed immediately. Refer to Chapter 6.38 for details.
- If the host application does not include a display, ME handles Class 0 short messages as though there was no message class, i.e. it will ignore bits 0 and 1 in the TP-DCS and normal rules for exceeded memory capacity shall apply. This approach is compliant with GSM 03.38.

Requirements specific to Multiplex mode:

- In multiplex mode (AT+CMUX=0) only one channel can use a phase 2+ parameter. The parameter for <mt> and <ds> on the other channels have to be set to zero.
- If either a SM or a Status Report is not acknowledged, all +CNMI parameters will be set to zero on all channels.

5.11 AT+CPMS: Preferred SMS message storage

Test command	Response
AT+CPMS=?	+CPMS: (list of supported <mem1>s), (list of supported <mem2>s), (list of supported <mem3>s)
	Parameter
	See write command
Read command	Response
AT+CPMS?	+CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2>,<total2>, <mem3>,<used3>,<total3> OK
	If error is related to ME functionality: +CMS ERROR
	Parameter
	See write command
Write command	Response
AT+CPMS= <mem1>	TA selects memory storages <mem1>, <mem2> and <mem3> to be used for reading, writing, etc.
[,<mem2>	+CPMS: <used1>,<total1>,<used2>,<total2>,<used3>,<total3> OK
[,<mem3>]]	If error is related to ME functionality: +CMS ERROR:<err>
	Parameter
	<mem1> Memory to be used when listing, reading and deleting messages: "SM" SIM message storage
	<mem2> Memory to be used when writing and sending messages: "SM" SIM message storage
	<mem3> Received messages will be placed to this storage if routing to TE is not set. See AT+CNMI command with parameter <mt>=2 (Chapter 5.10). "SM" SIM message storage
	<usedx> Number of messages currently in <memx>
	<totalx> Number of messages storable in <memx>
	<totalx> Number of messages storable in <memx>
Reference	Note
GSM 07.05	

5.12 AT+CSCA SMS service centre address

Test command AT+CSCA=?	Response **OK**
Read command AT+CSCA?	Response +CSCA: <sca>,<tosca> **OK** Parameter See write command
Write command AT+CSCA=<sca> [,<tosca>]	TA updates the SMSC address, through which mobile originated SMs are transmitted. In text mode, setting is used by send and write commands. In PDU mode, setting is used by the same commands, but only when the length of the SMSC address coded into <pdu> parameter equals zero. Note: this command writes the service centre address to non-volatile memory. Response **OK** Parameter <sca> GSM 04.11 RP SC address Address value field in string format; BCD numbers (or GSM default alphabet characters) are converted into characters; type of address given by <tosca> Maximum length of address: 20 characters <tosca> Service centre address format GSM 04.11 RP SC address Type-of-Address octet in integer format (default refer <toda>)
Reference GSM 07.05	Note In case of using no parameter after AT+CSCA= the content of <sca> will be deleted.

3.13 AT+CSCB Select cell broadcast messages

Test command AT+CSCB=?	Response +CSCB: (list of supported <mode>s) Parameter See write command
Read command AT+CSCB?	Response +CSCB: <mode>,<mids>,<dcss> Parameter See write command
Write command AT+CSCB=[<mode>[,<mids>[,<dcss>]]]	Parameter <mode> 0 Accepts messages that are defined in <mids> and <dcss> 1 Does not accept messages that are defined in <mids> and <dcss> <mids> String type; combinations of CBM message IDs (e.g. "0,1,5,320-478,922"). The number of ranges in <mids> parameter string is limited to 6 <dcss> String type; combinations of CBM data coding schemes (e.g. "0-3,5") Note If <mode>=1 is selected the parameter <mids> has to be given as only one area (e.g. "0-99")
Reference GSM 07.05	Note

5.14 AT+CSDH Show SMS text mode parameters

Test command AT+CSDH=?	Response +CSDH: (list of supported <show>s) OK Parameter See write command		
Read command AT+CSDH?	Response +CSDH:<show> OK Parameter See write command		
Write command AT+CSDH=<show>	Response TA sets whether or not detailed header information is shown in text mode result codes. OK Parameter		
	<show>	0	do not show header values defined in commands +CSCA and +CSMP (<sca>, <tosca>, <fo>, <vp>, <pid> and <dcs>) nor <length>, <toda> or <tooa> in +CMT, +CMGL, +CMGR result codes for SMS-DELIVERs and SMS-SUBMITs in text mode; for SMS-COMMANDs in +CMGR result code, do not show <pid>, <mn>, <da>, <toda>, <length> or <cdata>
		1	show the values in result codes
Reference GSM 07.05	Note		

9.15 AT+CSMP: Set SMS text mode parameters

Test command AT+CSMP=?	Response OK
Read command AT+CSMP?	Response +CSMP:<fo>,<vp/scts>,<pid>,<dcs> OK Parameter See set command
Set command AT+CSMP=<fo>[,<vp/scts>[,<pid>[,<dcs>]]]	Response TA selects values for additional parameters needed when SM is sent to the network or placed in a storage when text format message mode is selected. It is possible to set the validity period starting from when the SM is received by the SMSC (<vp> is in range 0... 255) or define the absolute time of the validity period termination (<vp> is a string). The format of <vp> is given by <fo>. If TA supports the enhanced validity period format, see GSM 03.40), it shall be given as a hexadezimal coded string (refer e.g. <pdu>) with quotes. Note When storing a SMS_DELIVER from the TE to the preferred memory storage in text mode (refer write command to Message Memory +CMGW), <vp> field can be used for <scts> Parameter <fo> depending on the command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), or SMS-COMMAND (default 2) in integer format <scts> GSM 03.40 TP-Service-Centre-Time-Stamp in time-string format (refer <dt>) <vp> depending on SMS-SUBMIT <fo> setting: GSM 03.40 TP-Validity-Period either in integer format (default 167)), in time-string format (refer <dt>), or if is supported, in enhanced format (hexadecimal coded string with quotes) <pid> Protocol-Identifier in integer format (default 0), refer GSM 03.40 <dcs> SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format depending on the command or result code: GSM 03.38
Reference GSM 07.05	Note The command writes the parameters to the non-volatile memory.

5.16 AT+CSMS Select Message Service

Test command	Response
AT+CSMS=?	+CSMS: (list of supported <service>s) OK
	Parameter
	See write command
Read command	Response
AT+CSMS?	+CSMS: <service>,<mt>,<mo>,<bm> OK
	Parameter
	See write command
Write command	Response
AT+CSMS= <service>	+CSMS: <mt>,<mo>,<bm> OK If error is related to ME functionality: +CMS ERROR: <err>
	Parameter
	<service> 0 GSM 03.40 and 03.41 (the syntax of SMS AT commands is compatible with GSM 07.05 Phase 2 version 4.7.0; Phase 2+ features which do not require new command syntax may be supported, e.g. correct routing of messages with new Phase 2+ data coding schemes)
	1 GSM 03.40 and 03.41 (the syntax of SMS AT commands is compatible with GSM 07.05 Phase 2+ version; the requirement of <service> setting 1 is mentioned under corresponding command descriptions).
	<mt> Mobile Terminated Messages:
	0 Type not supported
	1 Type supported
	<mo> Mobile Originated Messages:
	0 Type not supported
	1 Type supported
	<bm> Broadcast Type Messages:
	0 Type not supported
	1 Type supported
Reference	Note
GSM 07.05	If CSMS Mode is switched from Phase 2+ to Phase 2 and one or more CNMI Parameter are Phase 2+ specific a '+CMS ERROR: unknown error' will apear. It is recommended to switch the CNMI Parameters to Phase 2 specific values before entering Phase 2.

6 Siemens defined AT commands for enhanced functions

Self-defined commands do not have to be implemented in accordance with the official syntax. The "+C" string can therefore be replaced by "^S" ("^" = 0x5E). If a self-defined command with the same syntax will be included in future in the GSM recommendations, the command can be addressed with both strings.

6.1 AT+CXXCID Display card ID (identical to AT^SCID)

Test command AT+CXXCID=?	Response **OK** If error is related to ME functionality: **+CME ERROR:** <err> Parameter
Execute command AT+CXXCID	Response TA returns the card identification number in SIM (SIM file EF ICCID, see GSM 11.11 Chap.10.1.1) as string type. See ^SCID Parameter See ^SCID
Reference Siemens	Note See also Chapter 6.6 AT^SCID.

6.2 AT^MONI Monitor idle mode and dedicated mode

Test command AT^MONI=?	Response ^MONI: (list of supported < **period** >s) OK
Write command AT^MONI[=<period>]	This command can be used to retrieve, *automatically* every *n* seconds, information on the serving/dedicated cell. The display can be terminated by any character sent to serial port except if autobauding is enabled (+IPR=0). Then type character 'a' to abort. Note: The two header lines (see below) are output after every ten data lines. Response See execute command Parameter **<period>** 1 – 254 Display period in seconds
Execute command AT^MONI	This command can be used to retrieve, *on request*, the cell parameters of the serving/dedicated cell. Note: The length of following output lines exceeds 80 characters. Therefore a terminal program may draw a carriage return on a screen. However, this is not part of the response.

Response (Examples)

ME is not connected:

a) ME is camping on a cell

```
Serving Cell                                          I Dedicated channel
chann rs  dBm  PLMN  LAC cell NCC BCC PWR RXLev  C1 I chann TS timAdv PWR  dBm Q ChMod
1013  21  -71 00101 1001 0103   7   7  33  -105  33 I    No connection
```

b) ME camping on a cell, but searching for a better cell (cell reselection)

```
Serving Cell                                          I Dedicated channel
chann rs  dBm  PLMN  LAC cell NCC BCC PWR RXLev  C1 I chann TS timAdv PWR  dBm Q ChMod
1013   4 -106 00101 1001 0103   7   7  33  -105  -1 I    in Reselecting
```

c) ME is not camping on a cell and could not (yet) find a suitable cell

```
Serving Cell                                          I Dedicated channel
chann rs  dBm  PLMN  LAC cell NCC BCC PWR RXLev  C1 I chann TS timAdv PWR  dBm Q ChMod
```

ME is connected:

```
Serving Cell                                          I Dedicated channel
chann rs  dBm  PLMN  LAC cell NCC BCC PWR RXLev  C1 I chann TS timAdv PWR  dBm Q ChMod
1013  19  -76 00101 1001 0103   7   7  33  -105  33 I 1015   1      0   5  -76 0 S_HR
```

Parameters

Serving Cell:

chann	ARFCN (Absolute Frequency Channel Number) of the BCCH carrier
rs	RSSI (Received signal strength) of the BCCH carrier from 0 to 63. The indicated value is composed of the measured value in dBm plus an offset. This is in accordance with a formula specified in 3GPP TS 05.08.
dBm	receiving level of the BCCH carrier in dBm
PLMN	PLMN ID code
LAC	location area code, see note below.
cell	cell ID, see note below.
NCC	PLMN colour code
BCC	base station colour code
PWR	maximal power level used on RACH channel in dBm.
RXLev	minimal receiving level (in dBm) to allow registration
C1	coefficient for base station selection

Dedicated channel:

chann	ARFCN (Absolute Frequency Channel Number) of the TCH carrier Note: <**chann**> = **h** indicates frequency hopping.
TS	timeslot number
timAdv	timing advance in bits
PWR	current power level, coded according to 3GPP TS 05.05.
dBm	receiving level of the traffic channel carrier in dBm
Q	receiving quality (0...7)
ChMod	channel mode (S_HR: Half rate, S_FR: Full rate, S_EFR: Enhanced Full Rate)

Reference

Siemens

Note

- The parameters **LAC** and **cell** are presented as hexadecimal digits, the remaining parameters are composed of decimal digits.
- If the radio cell changes during a connection, the parameters **PWR** and **RXLev** of the 'Serving Cell' part cannot be updated under certain conditions and, therefore, are left blank (see also **+CREG**, pg 107). This is because the MS does not update the cell selection and reselection parameters since, in this mode, they are not relevant for operation.
- If the BS supports frequency hopping during a connection, the dedicated channel (parameter **chann**) is not stable. This mode is indicated by **chann** = '**h**'.
- The cell information can be issued in the form of unsolicited result codes (related to <**period**>), or it can be queried directly using the Execute command AT^MONI. In the first case, the ME activates its RING line (Logic "1") for one second to send the URC to the connected application. In the second case, the RING line does not change.
- To some extent, the cell monitoring commands AT^MONI, AT^MONP and AT^SMONC cover the same parameters. The receiving level, for example, can be queried with all three commands. Yet the resulting values may be slightly different, even though obtained over a time period of a few seconds. This is quite normal and nothing to worry about, as the cell information is permanently updated.

6.3 AT^MONP Monitor neighbour cells

Test command
AT^MONP=?

Response
^MONP: (list of supported < period >s) OK

Write command
AT^MONP=[<period>]

This command can be used to retrieve, *automatically* every *n* seconds, information of up to six neighbour cells. The display can be terminated by any character sent to the serial port except if autobauding is enabled (+IPR=0). In this case, type character 'a' to abort.

Response
See execute command

Parameter

<period>	1 – 254	Display period in seconds

Execute command
AT^MONP

This command can be used to obtain, *on request*, information of up to six neighbour cells.

Response (Example)

```
chann rs  dBm  PLMN BCC  C1  C2
  504 18  -78 26203   1  27  27
  476 15  -83 26203   3  22  22
  421 13  -88 26203   1  17  17
  440 10  -93 26203   7  12  12
  446  9  -95 26203   7  10  10
  417  8  -97 26203   4   8   8

OK
```

Parameter:

Chann	ARFCN (Absolute Frequency Channel Number) of the BCCH carrier
rs	RSSI (Received signal strength) of the BCCH carrier, decimal value from 0 to 63. The indicated value is composed of the measured value in dBm plus an offset. This is in accordance with a formula specified in 3GPP TS 05.08.
dBm	receiving level in dBm
PLMN	PLMN ID code
BCC	base station colour code
C1	coefficient for base station selection
C2	coefficient for base station reselection

Reference
Siemens

Note

- Cell information can be issued in the form of unsolicited result codes (related to <period>), or it can be queried directly using the Execute command AT^MONI. In the first case, the ME activates its RING line (Logic "1") for one second to send the URC to the connected application. In the second case, the RING line does not change.
- Due to the fact that not all necessary information of the neighbour cells can be decoded during a connection, there are several constraints to be considered:
  - Only neighbour cells that have already been visible in IDLE mode will be further updated, as long as they are still included in the list.
  - Though new neighbour cells can be added to the list (e.g. due to handover), their C1 and C2 parameters cannot be displayed until the connection is released. In this case "-" is presented for C1 and C2.
- The list does not include the serving cell.
- Further cell information can be obtained with AT^SMONC (see Chapter 6.17).

6.4 AT^SACM Advice of charge and query of ACM and ACMmax

Test command AT^SACM=?	Response ^SACM: (list of supported <n>s) OK Parameter See write command
Execute command AT^SACM	The execute command can be used to query the current mode of the Advice of Charge supplementary service, the SIM values of the accumulated call meter (ACM) and accumulated call meter maximum (ACMmax). Response ^SACM: <n>,<acm>,<acm_max> OK If error is related to ME functionality: +CME ERROR: <err> Parameter <n> See write command <acm> ACM, string type; three bytes of the current ACM value in hexadecimal format (e.g. "00001E" indicates decimal value 30) 000000-FFFFFF <acm_max> ACMmax, string type; three bytes of the max. ACM value in hexadecimal format (e.g. "00001E" indicates decimal value 30) 000000 disable ACMmax feature 000001-FFFFFF <ccm> string type; three bytes of the current CCM value in hexadecimal format (e.g. "00001E" indicates decimal value 30); bytes are coded in the same way as ACMmax value in the SIM 000000-FFFFFF
Write command AT^SACM=<n>	The write command enables or disables the presentation of unsolicited result to report the call charges. Response OK or if error is related to ME functionality: +CME ERROR: <err> Parameter <n> 0 suppress unsolicited result code 1 display unsolicited result code When you power down or reset the ME with AT+CFUN=1,1 the URC presentation mode will be reset to its default. To benefit from the URC it is recommended to have the setting included in the user profile saved with AT&W, or to select <n>=1 every time you reboot the ME. Unsolicited result code When activated, an unsolicited result code is sent when the CCM value changes, but not more often than every 10 seconds +CCCM: <ccm>
Reference Siemens	Note See also GSM07.07: AT+CACM, AT+CAMM, AT+CAOC

6.5 AT^SBC Battery charging / discharging and charge control

This chapter is only applicable to TC35 and TC37, it is not intended for the TC35 Terminal.

Responses returned by the AT^SBC command vary with the operating mode of the ME:

Normal mode:	ME is switched on by Ignition pin and running the SLEEP, IDLE, TALK or DATA mode. Charger is not connected. AT^SBC can be used to query the battery capacity and the power consumption of ME and application (if value of application was specified before as <current>).
Normal mode + charging:	Allows charging while ME is switched on by Ignition pin and running the SLEEP, IDLE, TALK or DATA mode. AT^SBC returns charger status and power consumption of ME / application. Battery capacity is not available.
Charge-only mode:	Allows charging while ME is detached from GSM network. When started, the mode is indicated by the URC "SYSSTART CHARGE-ONLY MODE". AT^SBC returns charger status and power consumption of ME / application. Percentage of battery capacity is not available. In Charge-only mode a limited number of AT commands is accessible (see Table 9). There are several ways to activate the Charge-only mode: a) from Power Down mode: Connect charger while ME was powered down with AT^SMSO. b) from Normal mode: Connect charger, then enter AT^SMSO.
Alarm mode:	No charging functionality, i.e. charging does not start even though the charger connects to the POWER lines. Battery parameters are not available.

Charging begins once the charger connects to the POWER pins of the ZIF connector (except for the Alarm mode). Please refer to the [1] ("Hardware Interface Description") supplied with your GSM engine and the Application Note "Charging the Battery Pack" for details on the charging process.

Test command	Response
AT^SBC=?	^SBC: (list of supported <bcs>s),(list of supported <bcl>s),<mpc> module power consumption

Defined values

<bcs>	0	No charging adapter is connected
	1	Charging adapter is connected
	2	Charging adapter is connected, charging in progress
	3	Charging adapter is connected, charging has finished
	4	Charging error, charging is interrupted
	5	False charging temperature, charging is interrupted while temperature is beyond allowed range

<bcl> Battery capacity
0, 20, 40, 60, 80, 100 percent of remaining capacity (6 steps)
0 indicates that either the battery is exhausted or the capacity value is not available

<mpc> Average power consumption.
Value (0...5000) of average power consumption (mean value over a couple of seconds) in mA. See read and write command for details.

Read command AT^SBC?	Response ^SBC: <bcs>,<bcl>,<mpc> <bcs> Connection status of battery pack <bcl> Battery charge level While charging is in progress (charging adapter connected) the battery capacity is not available. Consequently, parameter <bcl>=0. To query the battery capacity disconnect the charger. <mpc> Average power consumption <mpc> is obtained from the ME's power consumption, plus the value you have specified for the application by using the write command AT^SBC=<current>. Remember that the ME's power consumption varies with its operating mode (IDLE, TALK, DATA) and the power level. If <current> was not yet specified and no battery pack NTC is detected <mpc> returns only the module's present power consumption. If <current> was not yet specified, but the NTC of the connected battery pack is detected, an offset value of 200mA will, by default, be added. 200mA is an estimated value which represents the power consumption of a typical external application. Drawn from practical experience it serves as a precaution to ensure proper charging in case you have not entered <current>. It is strongly recommended that you enter the correct power consumption of your application as described below. Note: If the battery does not incorporate an NTC, or the battery and the NTC are not compliant with the requirements specified in [1], the battery cannot be detected by the ME.
Write command AT^SBC=<current>	Use the write command to specify the power consumption of your external application. This information enables the ME to calculate the average power consumption <mpc> and to properly control the charging process. If the value is not correct the entire charging process may be affected. Resulting problems may be wrong responses to the AT^SBC read command, overcharging, or the battery does not reach full capacity. The write command registers the serial port as the output channel for unsolicited result codes related to charging. When the ME is powered down or reset, the value of <current> is restored to its default. This affects the charging control and disables the presentation of unsolicited result codes. Therefore, the parameter should be set every time when needed after rebooting the ME. Response OK If error is related to ME functionality: +CME ERROR: <err> Parameter <current> Enter the current consumption of your application in mA (0...5000). If used, the current provided over the by 2.9V VDD pin of the ZIF interface (maximum 70mA) must be added, too.

	Unsolicited result code **^SBC: Undervoltage** The message will be reported, for example, when you attempt to set up a call while the voltage is close to the critical limit and further power loss is caused during the transmit burst. To remind you that the battery needs to be charged soon, the URC appears several times before the module switches off. In this case, the battery capacity is still sufficient to set up a short call. When the module is in IDLE mode it takes typically one minute to deregister from the network and to switch off.
Reference Siemens	Note • If Multiplex mode is active, any virtual channel can be used to enter the write command and to specify <current>. The undervoltage URC, however, appears simultaneously on all three channels. • The URC "SYSSTART CHARGE-ONLY MODE" is indicated automatically when the engine enters this mode (except when autobauding is active). Unlike the undervoltage URC, it cannot be disabled or enabled by the user.

Table 9: Summary of AT commands available in Charge-only and Alarm mode

AT command	Use
AT+CALA	Set alarm time
AT+CCLK	Set date and time of RTC
AT^SBC	Monitor charging process Note: While charging is in progress, no battery parameters are available. To query the battery capacity disconnect the charger. If the charger connects *externally* to the host device no charging parameters are transferred to the module. In this case, the command cannot be used.
AT^SCTM	Query temperature of GSM engine, enable or disable URCs
AT^SMSO	Power down GSM engine

6.6 AT^SCID Display SIM card identification number

Test command	Response
AT^SCID=?	**OK** If error is related to ME functionality: **+CME ERROR: <err>** Parameter
Execute command	Response
AT^SCID	TA returns the identification number of the SIM card (see GSM 11.11 Chapter 10.1.1). **^SCID: <cid> OK** If error is related to ME functionality: **+CME ERROR: <err>** Parameter **<cid>** string type: card identification number of SIM card
Reference	Note
Siemens	

6.7 AT^SCKS Set SIM connection presentation mode and query SIM connection status

Test command AT^SCKS=?	Response ^SCKS: (list of supported <n>s) OK Parameter See write command
Read command AT^SCKS?	Response TA returns the URC presentation mode and the status of the SIM card connection. ^SCKS: <n>, <m> OK Parameter See write command
Write command AT^SCKS=<n>	Response TA enables or disables the presentation of URCs to report whether or not the SIM card is connected. When the ME is powered down or reset with AT+CFUN=1,1 the presentation mode <n> will be restored to its default. To benefit from the URCs, it is recommended to have the setting <n>=1 included in the user profile saved with AT&W, or to activate the setting every time you reboot the ME. OK Parameter <n> 0 Suppress unsolicited result codes 1 Output unsolicited result codes <m> 0 No card 1 Card in card reader Unsolicited result code When the status "SIM connected" has changed, an unsolicited result code is sent to the TE. ^SCKS: <m> Parameter See write command
Reference Siemens	Note Note that the connection status of <m> reflects only the status of the card holder tray. If an empty SIM card tray is inserted, two URCs will be output, indicating the status 1 and 0 (= SIM card connected and not connected).

6.3 AT^SCNI List Call Number Information

Test command	Response
AT^SCNI=?	OK

Execute command	Response
AT^SCNI	TA returns a list of current calls of ME. [^SCNI: <id1>[,<cs>[,<number>,<type>]]] [^SCNI: <id2>[,<cs>[,<number>,<type>]]] [...] OK If error is related to ME functionality: +CME ERROR: <err>

Parameter

<idx>	1–7	integer type; call identification number as described in GSM 02.30[19] subclause 4.5.5.1; this number can be used in +CHLD command operations
<cs>	Call status of respective call number (first parameter)	
	0	call hold
	1	call in progress
	2	Waiting call
<number>	string type phone number in format specified by <type>	
<type>	type of address octet in integer format; 145 when dialling string includes international access code character "+", otherwise 129	

Reference	Note
Siemens	See also GSM 07.07: AT+CLCC

6.3 AT^SCTM Set critical operating temperature presentation mode or query temperature

Use this command to monitor the temperature range of the module and the battery. The write command enables or disables the presentation of URCs to report critical temperature limits.

Test command
AT^SCTM=?

Response
^SCTM: (list of supported **<n>**s) **OK**

Parameter
See write command

Read command
AT^SCTM?

Response
TA returns the URC presentation mode and information about the current temperature range of the module (not of the battery).
^SCTM: <n>, <m> OK

Parameters
<n> 0 Presentation of URCs is disabled.
1 Presentation of URCs is enabled.

<m> -2 Below lowest temperature limit (causes immediate switch-off)
-1 Below low temperature alert limit
0 Normal operating temperature
1 Above upper temperature alert limit
2 Above uppermost temperature limit (causes immediate switch-off)

Write command
AT^SCTM=<n>

Select **<n>** to enable or disable the presentation of the URCs. Please note that the setting will not be stored upon Power Down, i.e. after restart or reset, the default level 0 will be restored. To benefit from the URCs **<n>**=1 needs to be selected every time you reboot the GSM engine.

Response
OK

Parameters
<n> 0 Suppress URCs.
1 Output URCs.

Unsolicited result code
If enabled, URCs will be automatically sent to the TA when the temperature reaches or exceeds the critical level, or when it is back to normal.
^SCTM_A: <m> for battery (accumulator) temperature
^SCTM_B: <m> for module (board) temperature

Reference	Note
Siemens	**Important:** • Please refer to the "Hardware Interface Description" supplied with your GSM engine for specifications on critical temperature ranges. • To avoid damage the module will shut down once the critical temperature is exceeded. The procedure is equivalent to the power-down initiated with **AT^SMSO**. • The shutdown takes effect no matter whether URCs are enabled or disabled: URCs indicating the alert level "2" or "-2" are followed by immediate shutdown. If <n> is 0 the user is not informed before the module shuts down. • URCs indicating the alert level "1" or "-1" are intended to enable the user to take appropriate precautions, such as protect the module or battery from exposure to extreme conditions, or save or back up data etc.

Examples

URCs issued when the operating temperature is out of range:

^SCTM_A: 1	Caution: Battery close to overtemperature limit.
^SCTM_A: 2	Alert: Battery above overtemperature limit. Engine switches off.
^SCTM_B: 1	Caution: Engine close to overtemperature limit.
^SCTM_B: 2	Alert: Engine is above overtemperature limit and switches off.
^SCTM_A: -1	Caution: Battery close to undertemperature limit.
^SCTM_A: -2	Alert: Battery below undertemperature limit. Engine switches off.
^SCTM_B: -1	Caution: Engine close to undertemperature limit.
^SCTM_B: -2	Alert: Engine is below undertemperature limit and switches off.

Example

URCs issued when the temperature is back to normal (URC is output once):

^SCTM_A: 0	Battery temperature back to normal temperature.
^SCTM_B: 0	Engine back to normal temperature

6.10 AT^SDLD Delete the "last number redial" memory

Test command	Response
AT^SDLD=?	OK
Execute command	The execute command deletes all numbers stored in the LD memory.
AT^SDLD	Response
	OK/ERROR/+CME ERROR
Reference	Note
Siemens	

6.11 AT^SHOM Display Homezone

Test command	Response		
AT^SHOM=?	OK		
	Parameter		
	See execute command		
Execute command	Response		
AT^SHOM	TA returns homezone state		
	^SHOM: <homezonestate> OK		
	Parameters		
	<homezonestate>	0	ME is out of Homezone
		1	ME is within the Homezone
Reference	Note		
Siemens			

6.12 AT^SLCD Display Last Call Duration

Test command	Response
AT^SLCD=?	OK
	Parameter
	See execute command
Execute command	Response
AT^SLCD	TA returns last call duration or current call duration
	^SLCD: <time> OK
	Parameter
	<time> string type value; format is "hh:mm:ss", where characters indicate hours, minutes, seconds; e.g. 22:10:00 "22:10:00", max values are 9999:59:59
Reference	Note
Siemens	

6.13 AT^SLCK Facility lock

Test command AT^SLCK=?	Response ^SLCK: (list of supported <fac>s) OK Parameter See write command
Write command AT^SLCK= <fac>,<mode> [,<passwd> [,<class>]]	Response This command is used to lock, unlock or interrogate a ME or a network facility <fac>. The command can be aborted while network facilities are being set or interrogated. If <mode><>2 and command is successful OK If <mode>=2 and command successful ^SLCK: <status>[,<class1>[<CR><LF> ^SLCK: <status>, class2....[] OK If error is related to ME functionality: +CME ERROR: <err> Parameter <fac> Phone security locks set by user / provider "PS" Phone locked to SIM card (phone code). ME requests password when other than current SIM card inserted; ME may remember certain number of previously used cards thus not requiring password when they are inserted. "SC" SIM (lock SIM cards). SIM requests password upon ME power-up and when this lock command issued. "FD" SIM fixed dialling memory. If the mobile is locked to FD, only the numbers stored to the FD memory can be dialled (up to 7 numbers). If PIN2 authentication has not been performed during the current session, PIN2 is required as <passwd>. "CS" Keypad lock (not supported since keypad cannot be connected) Supplementary Service: Call barring "AO" BAOC (Bar All Outgoing Calls) "OI" BOIC (Bar Outgoing International Calls) "OX" BOIC-exHC (Bar Outgoing International Calls except to Home Country) "AI" BAIC (Bar All Incoming Calls) "IR" BIC-Roam (Bar Incoming Calls when Roaming outside the home country) "AB" All Barring services (applicable only for <mode>=0) "AG" All outGoing barring services (applicable only for <mode>=0) "AC" All inComing barring services (applicable only for <mode>=0) Factory set facility locks: "PF" lock Phone to the very First SIM card "PN" Network Personalisation "PU" Network subset Personalisation "PP" Service Provider Personalisation "PC" Corporate Personalisation

	<mode> 0 unlock 1 lock 2 query status <passwd> password <class> integer or sum of integers each representing a <class> of information: 1 voice 2 data 4 fax 8 short message service 16 data circuit sync 32 data circuit async 64 dedicated packet access 128 dedicated PAD access x combination of some of the above classes. For example, the default setting 7 represents the sum of the integers 1, 2 and 4 (call barring for voice, data and fax). The value 255 covers all classes. If the <class> parameter is omitted, the default value 7 is used. See examples in 4.18.3 for the correct handling of class numbers. <status> 0 off 1 on
Reference	Note
Siemens	See also specification of AT+CLCK in GSM 07.07 and further details in Chapter 4.18.

6.14 AT^SMGL List SMS messages from preferred storage

Test command	Response
AT^SMGL=?	See write command + CMGL Parameters See command +CMGL
Execute/Write command AT^SMGL [=<stat>]	Response TA returns messages with status value <stat> from message storage <mem1> to the TE. The status of the messages is u n c h a n g e d (unread remains unread). Otherwise: See command +CMGL Parameters See command +CMGL
Reference	Note
Siemens	See also GSM 07.05: +CMGL

6.15 AT^SMGO Set or query SMS overflow presentation mode or query SMS overflow

Test command AT^SMGO=?	Response ^SGMO: (list of supported <n>s) OK Parameter See write command
Read command AT^SMGO?	Response TA returns overflow presentation mode and SMS overflow status ^SGMO: <n>,<mode> OK If error is related to ME functionality: +CME ERROR: <err> Parameter See write command
Write command AT^SMGO=<n>	Response TA sets overflow presentation mode OK Parameter <n> SMS overflow presentation mode 0 disable (default) 1 enable <mode> SMS overflow status 0 space available 1 SMS buffer full (SIM card) 2 Buffer full and new message waiting in SC for delivery to phone Unsolicited result code When the status SIM overflow changes, an unsolicited result code is sent to TE ^SMGO: <mode> Parameter See write command
Reference Siemens	Note Indication during data transfer via break (100ms). Data transmission will only be interrupted by a break and for only 100ms.

6.16 AT^SMGR Read SMS message without set to REC READ

Test command	Response
AT^SMGR=?	OK
Execute command	**Parameter**
AT^SMGR=<index>	See AT+CMGR
Reference	**Note**
GSM 07.05	The AT^SMGR command is a specific Siemens command with the same syntax as "AT+CMGR Read SMS message". The only difference is that the SMS Message, which has REC_UNREAD status, is not overwritten to REC_READ.

6.17 AT^SMONC: Cell Monitoring

Test command AT^SMONC=?	Response **OK**
Execute command AT^SMONC	Response **^SMONC:** **232,03,3010,4EAF,32,82,38,30,30,232,03,3010,0000,36,88,26,18,18,232,03,3010,4EC3,32,112,23,15,15,232,03,3010,4BDA,34,90,17,9,9,232,03,3010,0000,32,99,15,7,7,232,03,2010,00C0,35,113,9,1,1,232,03,3520,0000,32,85,8,0,0** **OK** If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

The output contains 9 values from a maximum of 7 base stations. The first base station is the serving cell.

Values for one base station in output order:

MCC	Mobile country code, 3 decimal digits, e.g. 232 Value 000: not decoded
MNC	Mobile network code, 2 decimal digits, e.g. 03 Value 00: not decoded
LAC	Location area code, 4 hexadecimal digits, e.g. 3010 Value 0000: not decoded
cell	Cell ID, 4 hexadecimal digits, e.g. 4EAF Value 0000: not decoded
BSIC	Base station identity code, 2 decimal digits, e.g. 32
chann	ARFCN (Absolute Frequency Channel Number) of the BCCH carrier, decimal, e.g. 82. Value 0: not decoded. In this case, all remaining parameters related to the same channel are neither decoded. For example, a non-existing cell appears as follows: **000,00,0000,0000,00,0,0,0,0**
RSSI	Received signal level of the BCCH carrier, decimal value from 0 to 63. The indicated value is composed of the measured value in dBm plus an offset. This is in accordance with a formula specified in 3GPP TS 05.08.
C1	Coefficient for base station reselection, decimal, e.g. 30
C2	Coefficient for base station reselection, decimal, e.g. 30

Reference

Siemens

Note

- In dedicated mode, the parameters C1 and C2 cannot be updated, and therefore, should be ignored.
- To some extent, the cell monitoring commands AT^MONI, AT^MONP and AT^SMONC cover the same parameters. The receiving level, for example, can be queried with all three commands. Yet the resulting values may be slightly different, even though obtained over a time period of a few seconds. This is quite normal and nothing to worry about, as the cell information is permanently updated.

See also Chapters 6.2 and 6.3