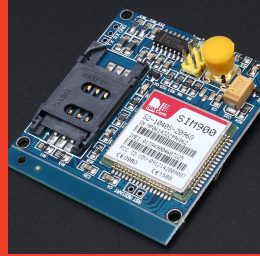




SOS SMS USING - 8051

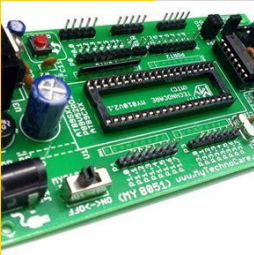
KVSSK PRADEEP
180060010

#3 things to know



the gsm module

8051 MICR-CONTROLLER



MOBILES

SITUATION: faced by many of us

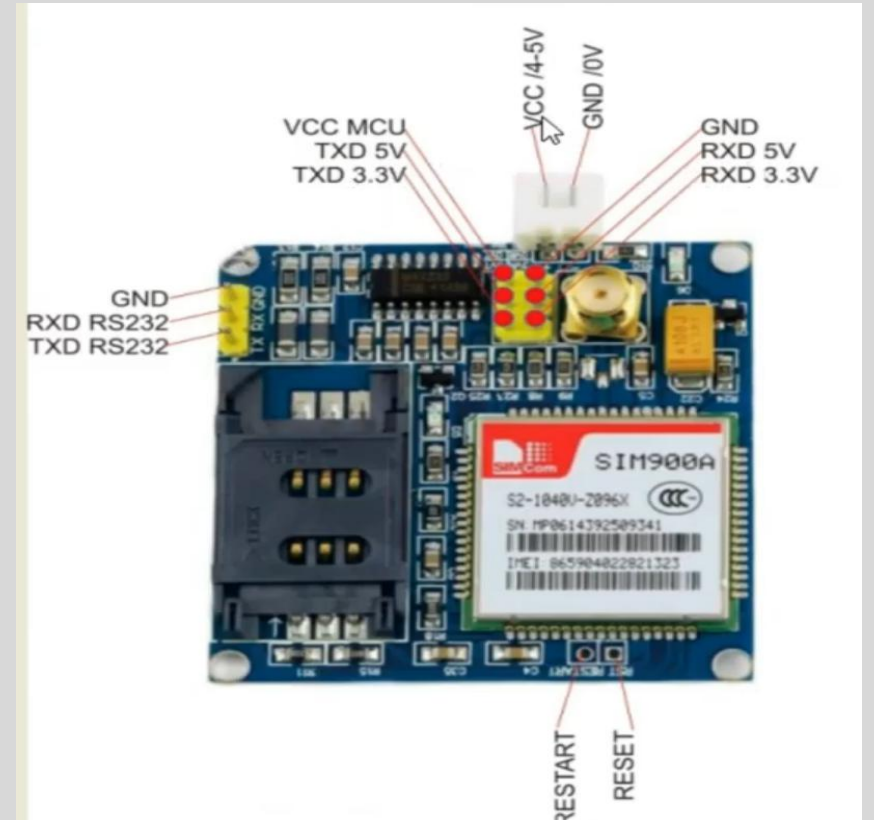


Idea comes here very simple and sweet the remote simple mobile in handy when in need.



GSM as part of this project involves because,

- GSM is a mobile communication modem.
- global system for mobile communication
- **850MHz, 900MHz, 1800MHz and 1900MHz** frequency bands.
- TDMA technique relies on assigning **different time slots** to each user on the same frequency
- data transmission and voice communication and can carry **64kbps to 120Mbps** of data rate.



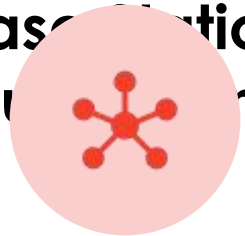
Many Features of GSM:



A Mobile



**Base Station
Subsystem**



**Network
Subsystem**

Single supply voltage: 3.4V – 4.5V

Power saving mode: Typical power consumption in SLEEP mode is 1.5mA

Frequency bands: SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the two frequency bands automatically. The frequency bands also can be set by AT command.

GSM class: Small MS

GPRS connectivity: GPRS multi-slot class 10 (default) , GPRS multi-slot class 8 (option)

Transmitting power: Class 4 (2W) at EGSM 900, Class 1 (1W) at DCS 1800

Operating Temperature: -30°C to +80°C

Storage Temperature: -5°C to +90°C

DATA GPRS: download transfer max is 85.6KBps, Upload transfer max 42.8KBps

Supports CSD, USSD, SMS, FAX

Supports MIC and Audio Input

Speaker Input

Features keypad interface

Features display interface

Features Real Time Clock

Supports UART interface

Supports single SIM card

Firmware upgrade by debug port

Communication by using AT commands

ARCHITECTURE OF GSM

Applications

Cellular Communication
Robotics
Mobile Phone Accessories
Servers
Computer Peripherals
Automobile
USB Dongles

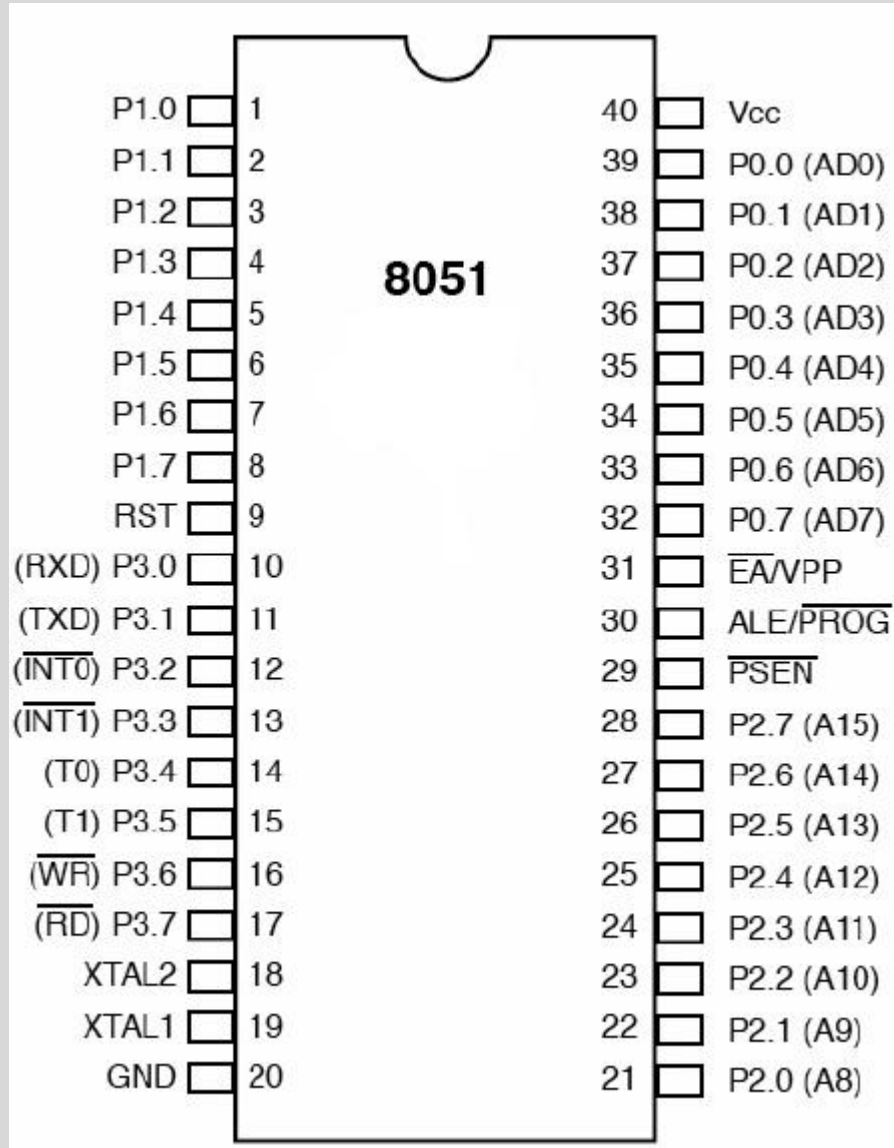


Table 5.4 Modes of SCON

SM0	SM1	Serial Mode	Explanation Baud Rate
0	0	0	0 8-bit Shift Register Oscillator / 12
0	1	1	8-bit UART Set by Timer 1 (*)
1	0	2	9-bit UART Oscillator / 32 (*)
1	1	3	9-bit UART Set by Timer 1 (*)

Pins used in 8051 are:

- **GND**
- **XTAL1**
- **XTAL2**
- **Pin 1**
- **SM0, SM1:** Serial Mode control Bits
- **TI:** It is known as Transmit Interrupt flag(Pin3.1)
- **RI:** It is known as Receive Interrupt flag(Pin3.0)

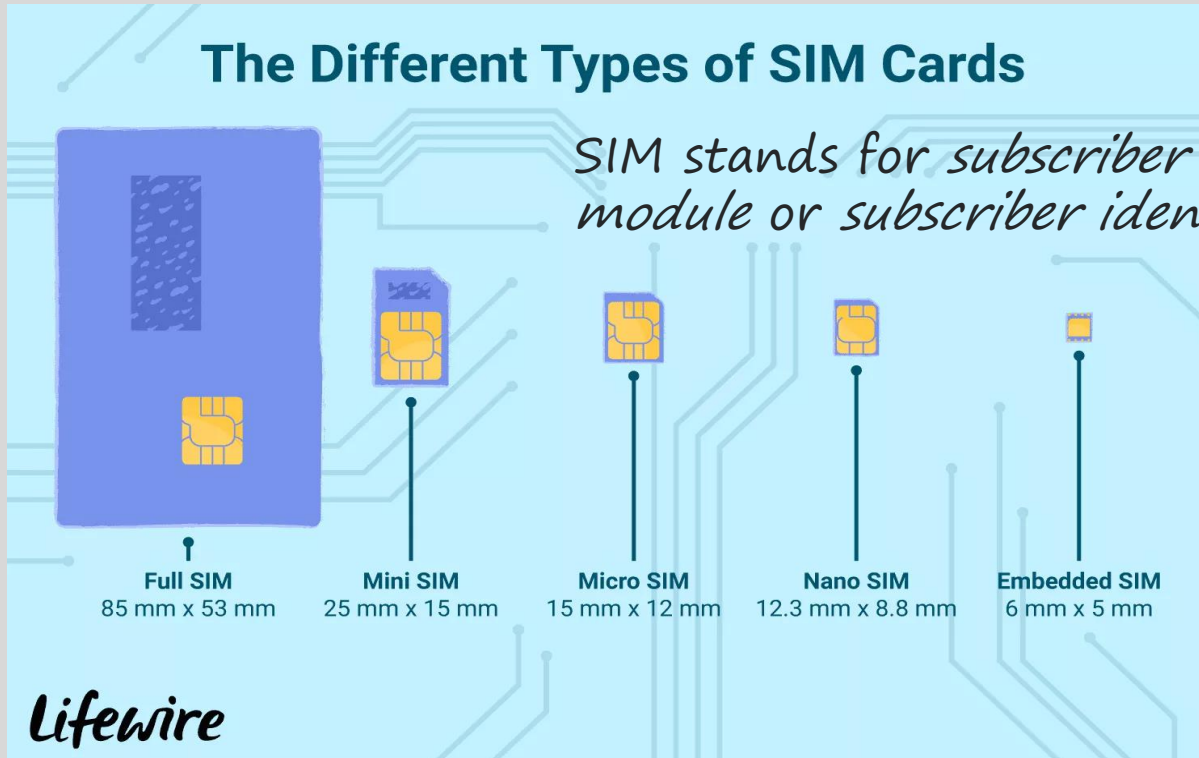


Google - Pixel 3a with 64GB

A common thing
that combines
these too apart
from the
technology you
change too



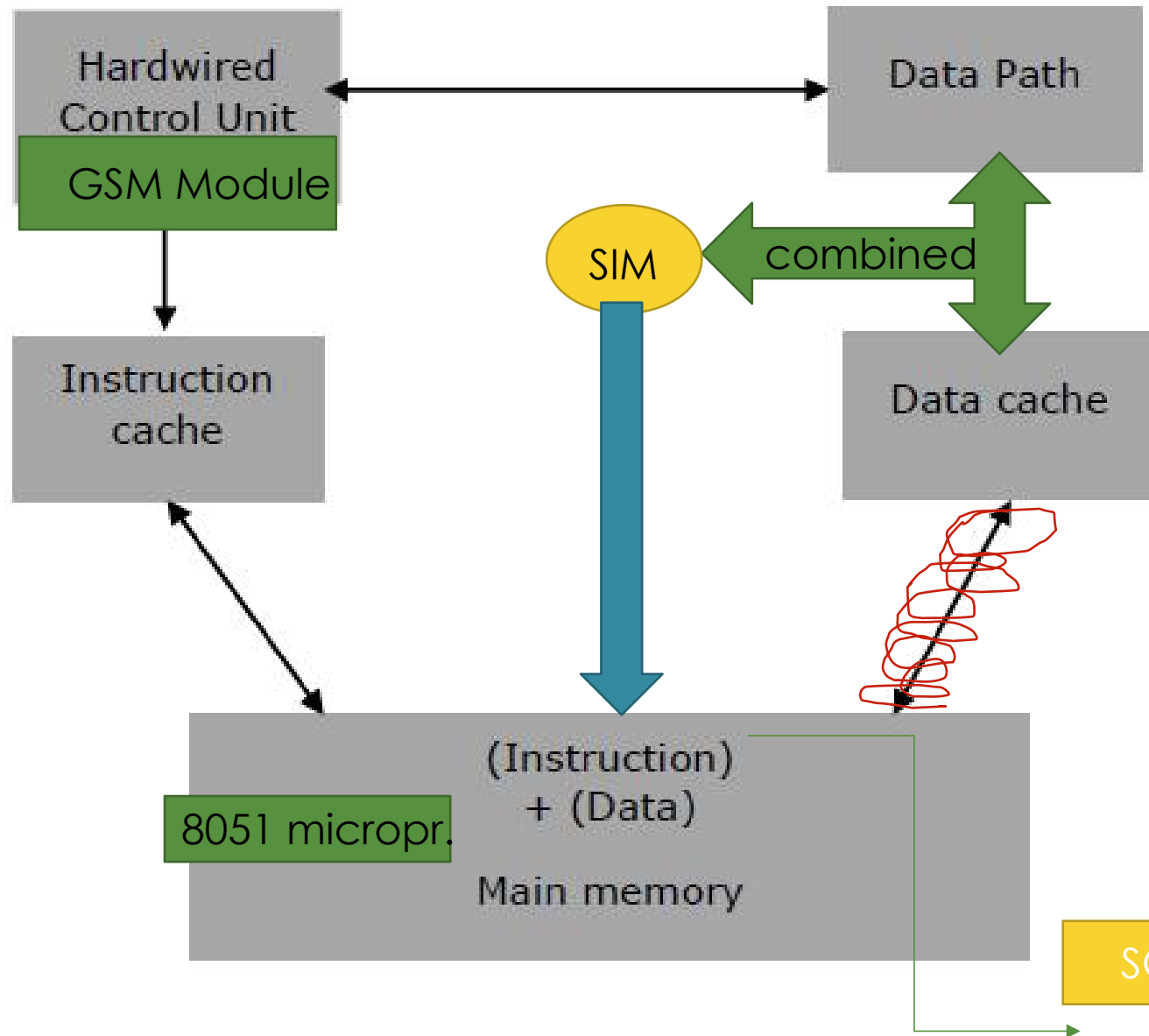
Nokia 3310 New
16 MB + 32 GB Expandable



in order to identify the owner and communicate with the mobile network

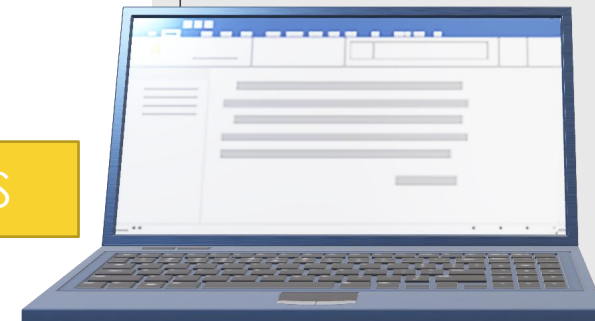
if your phone runs out of battery and you desperately need to make a phone call, and you have a spare around, you can just put the SIM card into the other phone and immediately use it.

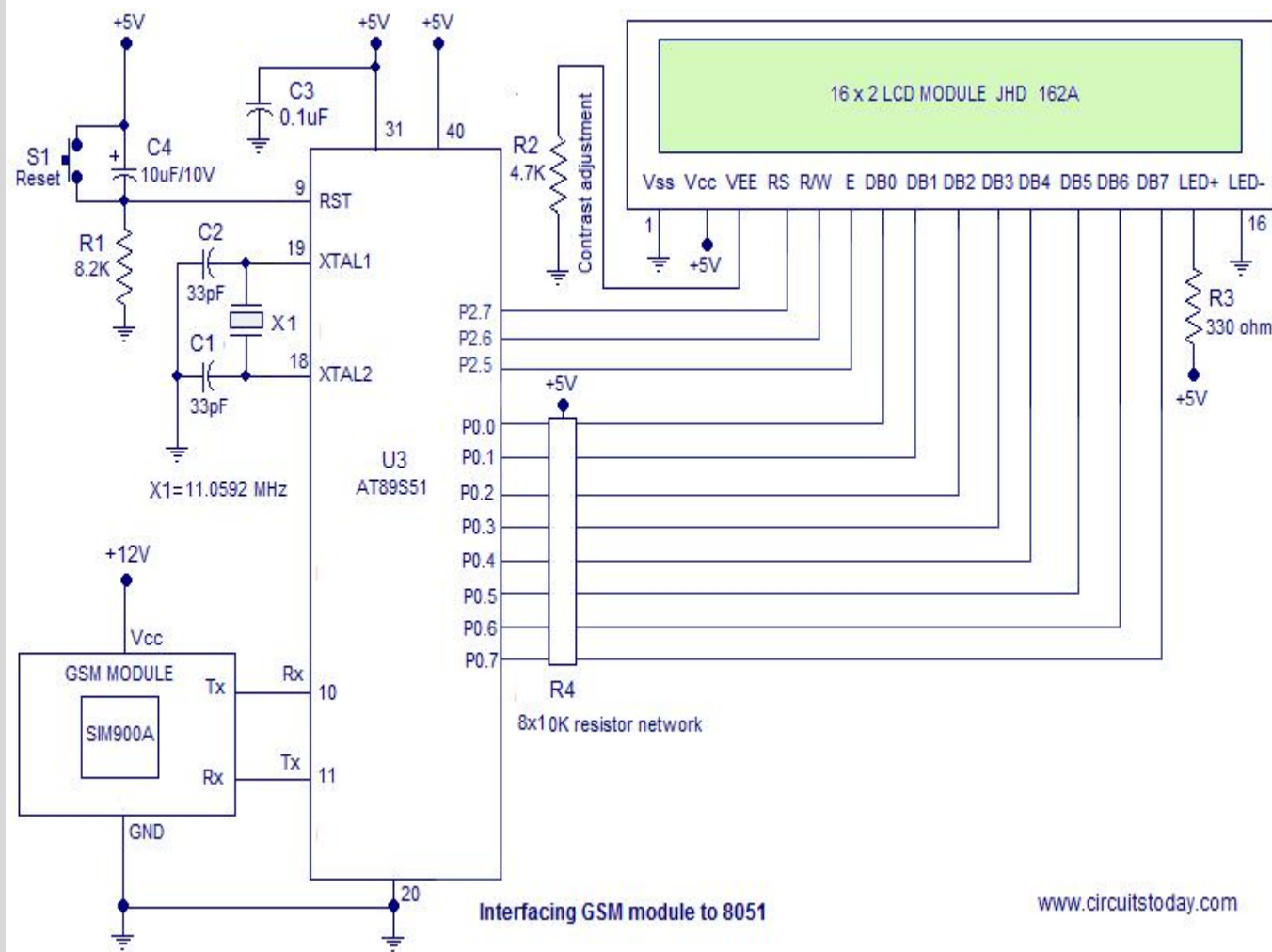
The SIM also contains a small amount of memory which can store up to 250 contacts, some SMS messages and other information used by the carrier who supplied the card



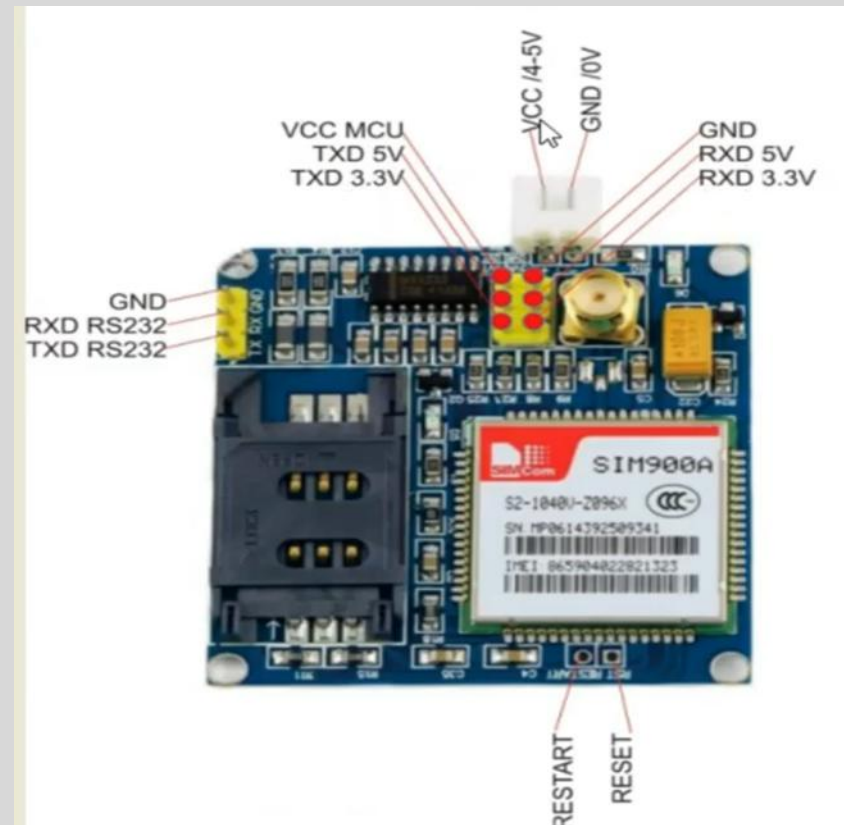
NOW WE
COMBINE ...

ALL....





12V, 9V, 5V
etc...



```

#include<reg51.h>
unsigned char *command = "AT";
unsigned char *echo = "ATE0";
unsigned char *msgConfig = "AT+CMGF=1";
unsigned char *number = "AT+CMGS=\[REDACTED]\\"; *\[REDACTED] no *\\
unsigned char *number = "AT+CMGS=\[REDACTED]\\"; *\\My number *\\
unsigned char *message = "hello";
unsigned char *CTRLZ = 0x1A;
void serial_init(void);
void serial(unsigned char);
void puts(unsigned char *p );
void delay(void);

void main()
{
    serial_init();
    puts(command);
    delay(); // delay of approx 1 sec
    puts(echo);
    delay();
    puts(msgConfig);
    delay();
    puts(number);
    delay();
    puts(message);
    delay();
    puts(CTRLZ);
    while(1);
}

void serial_init(void)
{
    TMOD=0x20; //timer 1, mode 2(8-bit autoreload) to set baud rate
    TH1=0xFD; //-3 to TH1 for 9600 baud rate
    SCON=0x50; // 8 bit txion, 1 start 1 stop bit, REN enable for both txfr and rxve
    TR1=1; // start timer
}

```

```

void puts(char *p)
{
char *temp = p; /*temp pointer so that the actual pointer is not displaced */
while(*temp != 0x00)
{
serial(*temp);
temp++;
}
}

void serial(unsigned char x)
{

SBUF=x;
while(TI==0);
TI=0;

}
void delay(void) // delay for approx 1 sec
{
int i;
TMOD=0x01; // timer 0 in mode 1
for(i=0;i<142;i++)
{
TL0=0x00; // starting value from 0
TH0=0x00;
TR0=1; // start timer
while(TF0==0); // polling TF flag for high
TR0=0; // stop timer
TF0=0; // clear flag TF0

}
}

```

Command – Operation

AT+CSMS	–	Select message service.
AT+CMGF	–	Message format.
AT+CMGL	–	List messages.
AT+CMGR	–	Read message.
AT+CMGS	–	Send message.
AT+CMGD	–	Delete message.
ATA	–	Answer a call.
ATD	–	Dial a number.
ATDL	–	Dial the last outgoing number.
ATH	–	Hang up the call.

CODE : Of the Project -

```
ORG 000H
MOV TMOD,#00100001B
MOV TH1,#253D
MOV SCON,#50H
SETB TR1
```

```
RS EQU P2.7
RW EQU P2.6
E EQU P2.5
```

```
MAIN: ACALL DINT
```

```
MOV A,#"A"
ACALL SEND
MOV A,#"T"
ACALL SEND
MOV A,#0DH
ACALL SEND
ACALL DELAY1
```

```
MOV A,#"A"
ACALL SEND
MOV A,#"T"
ACALL SEND
MOV A,#"+"
ACALL SEND
MOV A,#"C"
ACALL SEND
MOV A,#"M"
ACALL SEND
MOV A,#"G"
ACALL SEND
MOV A,#"F"
ACALL SEND
MOV A,#"="
ACALL SEND
MOV A,#"1"
ACALL SEND
MOV A,#0DH
ACALL SEND
ACALL DELAY1
```

```
MOV A,#"A"
```

```
ACALL SEND
MOV A,#"T"
ACALL SEND
MOV A,#"A"
ACALL SEND
MOV A,#"C"
ACALL SEND
MOV A,#"M"
ACALL SEND
MOV A,#"G"
ACALL SEND
MOV A,#"S"
ACALL SEND
MOV A,#"="
ACALL SEND
MOV A,#34D
ACALL SEND
MOV A,#"+"
ACALL SEND
MOV A,#"9"
ACALL SEND
MOV A,#"1"
ACALL SEND
MOV A,#"9"
ACALL SEND
MOV A,#"5"
ACALL SEND
MOV A,#"4"
ACALL SEND
MOV A,#"4"
ACALL SEND
MOV A,#"3"
ACALL SEND
MOV A,#"4"
ACALL SEND
MOV A,#"0"
ACALL SEND
MOV A,#"0"
ACALL SEND
MOV A,#"7"
ACALL SEND
MOV A,#"7"
ACALL SEND
MOV A,#34D
```

```
ACALL SEND
MOV A,#0DH
ACALL SEND
ACALL DELAY1
MOV A,#"H"
ACALL SEND
MOV A,#"E"
ACALL SEND
MOV A,#"L"
ACALL SEND
MOV A,#"L"
ACALL SEND
MOV A,#"O"
ACALL SEND
ACALL DELAY1
```

```
MOV A,#1AH
ACALL SEND
```

```
ACALL DELAY1
```

```
ACALL DINT
ACALL TEXT1
ACALL DELAY1
HERE1:SJMP HERE1
```

```
SEND:CLR TI
      MOV SBUF,A
WAIT:JNB TI,WAIT
      RET
```

```
DELAY1:MOV R6,#15D
BACK: MOV TH0,#00000000B
      MOV TL0,#00000000B
      SETB TR0
HERE: JNB TF0,HERE
      CLR TR0
      CLR TF0
      DJNZ R6,BACK
      RET
```

OUTPUT : will be received in mobile

