Sending SMS using AT Commands

This article describes sending a SMS from a Microcontroller through a GSM modem using AT Commands. through RS232. AT commands can be sent to a GSM modem via a computer serial port or from the serial port of a 8051 Microcontroller.

Computer serial port

1. Connect your GSM modem to the computer serial port. New systems nowadays dot not have serial port, hence you would need to buy a USB to serial converter and connect the GSM modem to it.

2. If you are using Windows XP® OS, open Programs -> Accessories -> Communications -> HyperTerminal

3. Select the COM port you have the modem connected to.

4. Check the port settings, make sure that the baud rate matches with that of the GSM modem and and also that the Flow Control is set to "None".

5. Type the AT Commands below in the HyperTerminal Editor

AT+CMGF = 1 and Enter
AT+CMGS="+919449XXXXXX" and Enter
"Electronicprojs.Blogspot.com" and Hit CTRL+Z

6. In Windows 7® OS though, there is no hyperterminal program inbuilt. You will have to download a similar one. There are many good free programs available.

7. You would need to note that in this case we are connecting the serial port cable from Female pin (GSM Modem) to Male pin (Computer). Hence no crossing is required. In other words, we should use a straight cable.

8051 MicroController Serial port

1. 8051 SFR's are programmed for a baud rate of 9600.

2. Send the AT commands in #5 above through code. For Enter use escape character "\r" and for CTRL+Z use ASCII 0x1A.

3. The serial port cable is connecting from Female pin (GSM Modem) to Female (8051 Board). So you need a have a female to male converter. Also, make sure that the Rx and Tx inside in the converter are crossed. In otherwords, the Rx pin of GSM Modem should go to Tx pin of 8051 microntroller. This is called a crossed cable.

4. Below is the C program using Keil® Compiler for 8051. We added infinite while loop after the sendsms() routine because otherwise we found that the code compiled was sending the SMS in a infinite loop.

Note : Beginners always complain about their code working through serial port of a computer but the same not working through chip. It is very essential to understand the difference between crossed cable and straight cable as well as the pins configurations of male and female connectors before starting.

C program

```
#include <REGX51.H>
#include <AT89X51.H>

unsigned char *command_AT = "AT\r";
unsigned char *command_CMGF = "AT+CMGF=1\r";
unsigned char *command_CMGS = "AT+CMGS=\"+919449XXXXX\"\r";
unsigned char *message = "electronicprojs.blogspot.com";
unsigned char CTRLZ = 0x1A;

void puts(unsigned char* ptr);
void putc(unsigned char chr);
void sendsms(void);
void initialize();

main()
{
 initialize();
 sendsms();
 while(1);
}

void initialize()
{
 SCON  = 0x50;   /*SCON: mode 1, 8-bit UART, enable receive      */
 TMOD |= 0x20;   /*TMOD: timer 1, mode 2, 8-bit                  */
 TH1   = 0xFD;   /*TH1:  for 9600 baud                           */
 TR1   = 1;      /*TR1:  timer 1 run                             */
}

void sendsms()
{
 puts(command_AT);
 puts(command_CMGF);
 puts(command_CMGS);
 puts(message);
 putc(CTRLZ);
}

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

```
void putc(unsigned char chr)
{
 SBUF = chr;
 while(TI==0);            /*Wait until the character is completely sent */
 TI=0;                    /*Reset the flag */
}
```