**Send and Read SMS through a GSM Modem using AT Commands**

By [**Syeda Anila Nusrat**](http://www.codeproject.com/script/Membership/View.aspx?mid=4881408) | 30 Aug 2010

**In this article, you can send and read SMS through a GSM modem using AT commands and without using any third party library.**

Top of Form





* [Download application - 21.6 KB](http://www.codeproject.com/KB/IP/Send_and_Read_SMS/SMSapplication.zip)

**Introduction**

There are many different kinds of applications **SMS** applications in the market today, and many others are being developed. Applications in which **SMS** messaging can be utilized are virtually unlimited. Some common examples of these are given below:

* Person-to-person text messaging is the most commonly used **SMS** application, and it is what the **SMS** technology was originally designed for.
* Many content providers make use of **SMS** text messages to send information such as news, weather report, and financial data to their subscribers.
* **SMS** messages can carry binary data, and so **SMS** can be used as the transport medium of wireless downloads. Objects such as ringtones, wallpapers, pictures, and operator logos can be encoded in **SMS** messages.
* **SMS** is a very suitable technology for delivering alerts and notifications of important events.
* **SMS** messaging can be used as a marketing tool.

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

1. 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.
2. Connect the computer / PC to the **SMS** center (**SMS**C) or **SMS** gateway of a wireless carrier or **SMS** service provider. Then send **SMS** messages using a protocol / interface supported by the **SMS**C or **SMS** gateway.

In this article, I will explain the first way to send, read, and delete **SMS** using AT commands. But before starting, I would like to explain a little bit about AT commands.

**AT Commands**

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. There are two types of AT commands:

1. Basic commands are AT commands that do not start with a "+". For example, D (Dial), A (Answer), H (Hook control), and O (Return to online data state) are the basic commands.
2. Extended commands are AT commands that start with a "+". All GSM AT commands are extended commands. For example, +CMGS (Send **SMS** message), +CMGL (List **SMS** messages), and +CMGR (Read **SMS** messages) are extended commands.

If you want to get more information about AT commands, then you can get it on my other article on CodeProject here: <http://www.codeproject.com/KB/system/IntroductiontoATcommands.aspx>.

**Operating Modes**

The **SMS** specification has defined two modes in which a GSM/GPRS modem or mobile phone can operate. They are called **SMS** text mode and **SMS** PDU mode. (PDU stands for Protocol Data Unit.) The mode that a GSM/GPRS modem or mobile phone is operating in determines the syntax of some **SMS** AT commands and the format of the responses returned after execution.

I am using **SMS** text mode in this article.

**How to Test GSM Modem Connectivity Using Hyper Terminal**

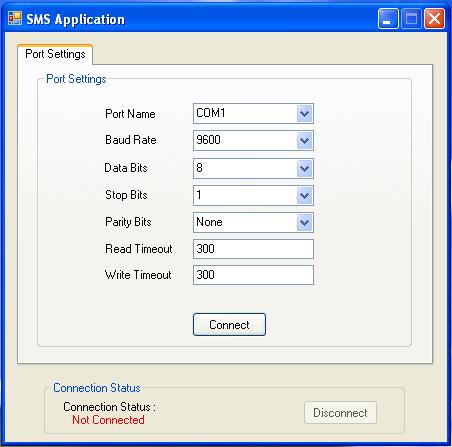
* First, find the best GSM modem that suits the needs. I tested this application with a Wavecom FASTRACK M1206.
* Understand the AT Command set required to communicate with the modem.
* Connect the modem to the computer according to the setup guide specified in the manual provided with the GSM modem.
* Put a valid SIM card into the mobile phone or GSM/GPRS modem.
* Connect your mobile phone or GSM/GPRS modem to a computer, and set up the corresponding wireless modem driver.
* Run the 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.
* 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.
* 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, consult the manual of your mobile phone or GSM/GPRS modem.
* Type "AT" in the main window. A response "OK" should be returned from the mobile phone or GSM/GPRS modem.
* If “OK” returns, it means your mobile phone or GSM/GPRS modem is connected successfully.

After successful connection of the GSM /GPRS modem with PC, you are ready to run this application. Download the attached project and run the application.

**Sending SMS through GSM Modem using AT Commands**

**Port Settings**

In this tab, you will have to do port settings which will be the same as you did in the hyper terminal and then click the OK button. If the modem is connected successfully, a message box will appear with the message “Modem is connected”.



http://www.codeproject.com/images/minus.gifCollapse | [Copy Code](http://www.codeproject.com/KB/IP/Send_and_Read_SMS.aspx)

public SerialPort OpenPort(string p\_strPortName,

int p\_uBaudRate, int p\_uDataBits,

int p\_uReadTimeout, int p\_uWriteTimeout)

{

receiveNow = new AutoResetEvent(false);

SerialPort port = new SerialPort();

try

{

port.PortName = p\_strPortName; *//COM1*

port.BaudRate = p\_uBaudRate; *//9600*

port.DataBits = p\_uDataBits; *//8*

port.StopBits = StopBits.One; *//1*

port.Parity = Parity.None; *//None*

port.ReadTimeout = p\_uReadTimeout; *//300*

port.WriteTimeout = p\_uWriteTimeout; *//300*

port.Encoding = Encoding.GetEncoding("iso-8859-1");

port.DataReceived += new SerialDataReceivedEventHandler

(port\_DataReceived);

port.Open();

port.DtrEnable = true;

port.RtsEnable = true;

}

catch (Exception ex)

{

throw ex;

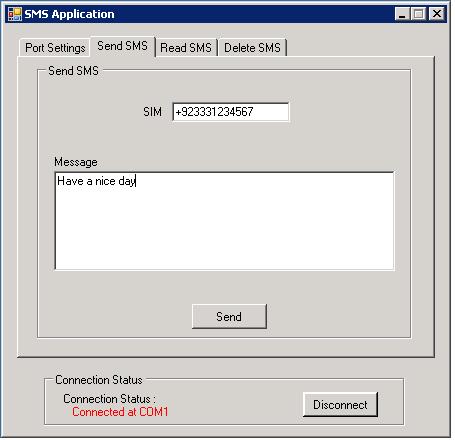
}

return port;

}

**Send SMS**

In the second tab, you can send **SMS**:



http://www.codeproject.com/images/minus.gifCollapse | [Copy Code](http://www.codeproject.com/KB/IP/Send_and_Read_SMS.aspx)

public bool sendMsg(SerialPort port, string PhoneNo, string Message)

{

bool isSend = false;

try

{

string recievedData = ExecCommand(port,"AT", 300, "No phone connected");

recievedData = ExecCommand(port,"AT+CMGF=1", 300,

"Failed to set message format.");

String command = "AT+CMGS=\"" + PhoneNo + "\"";

recievedData = ExecCommand(port,command, 300,

"Failed to accept phoneNo");

command = Message + char.ConvertFromUtf32(26) + "\r";

recievedData = ExecCommand(port,command, 3000,

"Failed to send message"); //3 seconds

if (recievedData.EndsWith("\r\nOK\r\n"))

{

isSend = true;

}

else if (recievedData.Contains("ERROR"))

{

isSend = false;

}

return isSend;

}

catch (Exception ex)

{

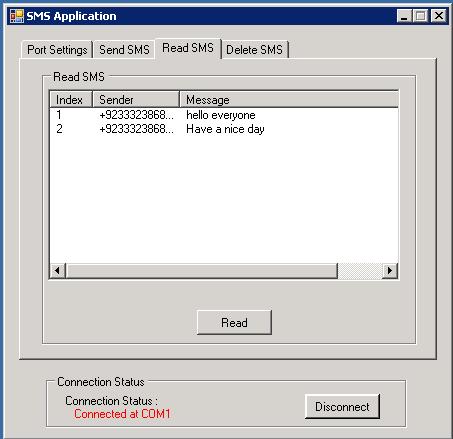
throw new Exception(ex.Message);

}

}

**Read SMS**

In the third tab, you can read **SMS**:



http://www.codeproject.com/images/minus.gifCollapse | [Copy Code](http://www.codeproject.com/KB/IP/Send_and_Read_SMS.aspx)

public ShortMessageCollection Read**SMS**(SerialPort port)

{

*// Set up the phone and read the messages*

ShortMessageCollection messages = null;

try

{

#region Execute Command

*// Check connection*

ExecCommand(port,"AT", 300, "No phone connected");

*// Use message format "Text mode"*

ExecCommand(port,"AT+CMGF=1", 300, "Failed to set message format.");

*// Use character set "PCCP437"*

ExecCommand(port,"AT+CSCS=\"PCCP437\"", 300,

"Failed to set character set.");

*// Select SIM storage*

ExecCommand(port,"AT+CPMS=\"SM\"", 300,

"Failed to select message storage.");

*// Read the messages*

string input = ExecCommand(port,"AT+CMGL=\"ALL\"", 5000,

"Failed to read the messages.");

#endregion

#region Parse messages

messages = ParseMessages(input);

#endregion

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

if (messages != null)

return messages;

else

return null;

}

**Delete SMS**

In the fourth and last tab, you can count the number of **SMS** and delete **SMS** as well.



http://www.codeproject.com/images/minus.gifCollapse | [Copy Code](http://www.codeproject.com/KB/IP/Send_and_Read_SMS.aspx)

public bool DeleteMsg(SerialPort port , string p\_strCommand)

{

bool isDeleted = false;

try

{

#region Execute Command

string recievedData = ExecCommand(port,"AT", 300, "No phone connected");

recievedData = ExecCommand(port,"AT+CMGF=1", 300,

"Failed to set message format.");

String command = p\_strCommand;

recievedData = ExecCommand(port,command, 300, "Failed to delete message");

#endregion

if (recievedData.EndsWith("\r\nOK\r\n"))

{

isDeleted = true;

}

if (recievedData.Contains("ERROR"))

{

isDeleted = false;

}

return isDeleted;

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

**Points of Interest**

I'm not using any third party library or anything else in this project.

Bottom of Form