

Tutorial

Creating SMS-based Applications

Part-2: *Sending and Receiving SMS using Visual Basic .NET*

By Haider Ali

May 2008

Introduction

In part-1 I gave some basics of SMS, the AT commands for SMS and then demonstrated sending/receiving SMS from HyperTerminal. Before reading this article I hope you have studied part 1 (Using HyperTerminal to send and receive SMS messages) of this series.

The SMS gateway application has two main components:

- Sending SMS from the PC.
- Reading the received SMS.

In this article I will write the first simple application that sends and receives SMS.

Messages stored in GSM/GPRS Modem or Mobile Phone Storage

Before going into the details of how to send and receive an SMS let me give some explanation of the messages stored in the mobile phone storage. The message stored in the mobile phone memory has the following four parts.

- Mobile Phone No.
- Message No.
- Status
- Message

The Mobile Phone No. is the number from which the message is received or to which the message is sent or to be sent.

The Message No. is a serial number given to each message.

The Status can be one of the following four:

STO UNSENT: Message stored for sending but not yet sent.

STO SENT: Message Sent.

STO UNREAD: Message received but not yet read.

REC READ: Message received and read.

The Message contains the actual message text.

For convenience I have created a `SMSMessage` class in this application. The code for the `SMSMessage` class is given in the following listing.

```
Public Class SMSMessage
    Public MessageNo As Integer
    Public Status As String
    Public MobileNo As String
    Public Message As String

    Public Sub New(ByVal mobileNo As String, ByVal message As String, _
        Optional ByVal MessageNo As Integer = 1,
        Optional ByVal status As String = "STO")
        Me.MessageNo = MessageNo
        Me.Status = status
        Me.MobileNo = mobileNo
        Me.Message = message
    End Sub
End Class
```

The SMSServices Class

The `SIMServices` class is created to put all the common functions related with SIM services like Send SMS, receive SMS and check the SIM status etc.

```
' -----
' SIMServices
' Author: Haider Ali
' College of Arts and Science at Wadi Ad-Dawasir
' King Saud University
' e-mail: haiderkhalil@yahoo.com
' The class provide interface to SMS Services of the Mobile Phone
' The Mobile phone may be connected via IrDA
' -----

Imports System.Threading

Public Class SIMServices
    Private CommPort As String
    Private Message As New SMSMessage
    Public AllMessages, UnreadMessages, SentMessages, UnSentMessages, _
        ReadMessages As Collection

    Public Sub New(Optional ByVal commPort As String = "COM4")
        Me.CommPort = commPort
    End Sub

    Public Sub SendSMS(ByVal MobNo As String, ByVal Mess As String)
        Dim Incoming, SubStr As String
        Dim MessageNo, N As Integer

        Using comm As IO.Ports.SerialPort = _
            My.Computer.Ports.OpenSerialPort(CommPort, 9600)

            comm.DtrEnable = True
            comm.Write("AT" & vbCrLf)
```

```

comm.Write("AT+cmgf=1" & vbCrLf)

' chr(34)= "
If Left(MobNo, 1) = Chr(34) Then
    comm.Write("AT+CMGW=" & MobNo & vbCrLf)
Else
    comm.Write("AT+CMGW=" & Chr(34) & MobNo & Chr(34) & vbCrLf)
End If

comm.Write(Mess & Chr(26) & vbCrLf) ' Ctrl-Z Character

Thread.Sleep(2000)

' Find the Message No. in Mobile Memory
Incoming = comm.ReadExisting
N = Incoming.IndexOf("+CMGW:")
N = N + 6
SubStr = Incoming.Substring(N)
MessageNo = Val(SubStr)
Debug.Print("Incoming: " & Incoming)

comm.WriteLine("AT+CMSS=" & MessageNo & vbCrLf)
Thread.Sleep(2000)

comm.Close()
End Using
End Sub

Public Sub SendSMS(ByVal Message As SMSMessage)
    SendSMS(Message.MobileNo, Message.Message)
End Sub

Public Sub DeleteSMS(ByVal MessNo As String)
    Thread.Sleep(2000)
    Using comm As IO.Ports.SerialPort = _
        My.Computer.Ports.OpenSerialPort(CommPort, 9600)

        comm.DtrEnable = True
        comm.Write("AT" & vbCrLf)
        comm.Write("AT+CMGF=1" & vbCrLf)
        comm.Write("AT+CMGD=" & MessNo & vbCrLf)
        Thread.Sleep(2000)
    End Using
End Sub

Public Sub DeleteAllSMS()
    Thread.Sleep(2000)
    Using comm As IO.Ports.SerialPort = _
        My.Computer.Ports.OpenSerialPort(CommPort, 9600)

        comm.DtrEnable = True
        comm.Write("AT" & vbCrLf)
        Thread.Sleep(2000)
        comm.Write("AT+CMGF=1" & vbCrLf)
        Thread.Sleep(2000)
        comm.Write("AT+CMGD=1,4" & vbCrLf)
        Thread.Sleep(2000)
    End Using
End Sub

```

```

Public Function Status() As String
    Dim Incoming As String
    Using comm As IO.Ports.SerialPort = _
        My.Computer.Ports.OpenSerialPort(CommPort, 9600)

        Incoming = comm.ReadExisting()
        comm.Close()
    End Using
    Return Incoming
End Function

Public Function SMSMessages() As String
    Dim Incoming As String

    Using comm As IO.Ports.SerialPort = _
        My.Computer.Ports.OpenSerialPort(CommPort, 9600)

        comm.DtrEnable = True
        comm.Write("AT" & vbCrLf)
        comm.Write("AT+CMGF=1" & vbCrLf)
        Thread.Sleep(2000)
        comm.Write("AT+CMGL=" & Chr(34) & "ALL" & Chr(34) & vbCrLf)
        Thread.Sleep(3000)
        Incoming = comm.ReadExisting
    End Using
    Return Incoming
End Function

Public Sub ReadAllSMSMessages()
    Dim i As Integer = 0
    AllMessages = New Collection
    Dim RemMessage As String 'Remaing Message
    Dim Position As Integer

    'Read Messages as string
    RemMessage = SMSMessages()

    Do
        Message = New SMSMessage
        'Find the location of the first SMS Message by finding +CMGL:
        Position = RemMessage.IndexOf("+CMGL:")
        If Position = -1 Then
            ' There are no more messages
            GoTo LoopEnd
        End If

        Position = Position + 6 'To skip string "+CMGL:

        RemMessage = RemMessage.Substring(Position)

        ' find the 1st comma
        Position = RemMessage.IndexOf(",")
        Message.MessageNo = Val(RemMessage.Substring(0, Position))
        'MsgBox(Message.MessageNo)

        ' Status
        RemMessage = RemMessage.Substring(Position + 2) ' Skip ","
        Position = RemMessage.IndexOf(",") ' Find next
    ,

        Message.Status = RemMessage.Substring(0, Position - 1)
        'MsgBox(Message.Status)

```

```

        ' Mobile No
RemMessage = RemMessage.Substring(Position + 1)
Position = RemMessage.IndexOf(",")
Message.MobileNo = RemMessage.Substring(0, Position)
'MsgBox(Message.MobileNo) it alsop ptints " like "03215414"

        ' Message Body
RemMessage = RemMessage.Substring(Position + 2)
Position = RemMessage.IndexOf("+CMGL:")
If Left(Message.Status, 3) = "REC" Then
    RemMessage = RemMessage.Substring(22)
    If Position <> -1 Then Position = Position - 22
End If

If Position = -1 Then
    ' Last Message in List
    'Deletes OK
    If (Len(RemMessage) > 3) Then
        RemMessage = RemMessage.Substring _
            (0, Len(RemMessage) - 4)
    End If

    Message.Message = RemMessage
Else
    Message.Message = RemMessage.Substring(0, Position)
End If
AllMessages.Add(Message)
Loop
LoopEnd:
UpdateCollections()
End Sub

Public Sub UpdateCollections()
    UnreadMessages = New Collection()
    ReadMessages = New Collection()
    UnSentMessages = New Collection()
    SentMessages = New Collection()

    For Each Message In AllMessages
        'MsgBox(Message.Status)
        Select Case Message.Status
            Case "STO UNSENT"
                UnSentMessages.Add(Message)
            Case "STO SENT"
                SentMessages.Add(Message)
            Case "REC UNREAD"
                UnreadMessages.Add(Message)
            Case "REC READ"
                ReadMessages.Add(Message)
        End Select
    Next
End Sub
End Class

```

Send SMS from VB .NET application

For sending SMS I have created a simple form **frmSendSMS**. The screenshot of the form is given in figure-1.



Figure 1: Send SMS

This simple form contain three controls:

- Textbox **tbTo** contains the mobile no. to which the message has to be sent.
- Textbox **tbMessage** contains the tesxt message that has to be sent.
- Button **btnSend** when clicked sends the message to the destination phone no.

The following code is associated with the click event of the **btnSend** button:

```
Imports System.Threading

Public Class frmSendSMS

Private Sub btnSend_Click(ByVal sender As System.Object,
                          ByVal e As System.EventArgs)
    Handles btnSend.Click

    Dim Incoming, SubStr As String
    Dim MessageNo, N As Integer

    Using comm As IO.Ports.SerialPort =
        My.Computer.Ports.OpenSerialPort("COM4", 9600)
        comm.DtrEnable = True
```

```

comm.Write("AT" & vbCrLf)
comm.Write("AT+cmgf=1" & vbCrLf)

' chr(34)= "
comm.Write("AT+CMGW=" & Chr(34) & MobNo & Chr(34) & vbCrLf)

comm.Write(tbMessage & Chr(26) & vbCrLf) ' Ctrl-Z Character

Thread.Sleep(2000)

' Find the Message No. in Mobile Memory
Incoming = comm.ReadExisting
N = Incoming.IndexOf("+CMGW:")
N = N + 6
SubStr = Incoming.Substring(N)
MessageNo = Val(SubStr)

comm.WriteLine("AT+CMSS=" & MessageNo & vbCrLf)
Thread.Sleep(2000)

comm.Close()
End Using
End Sub
End Class

```

Now let me explain each line of the code.

```
Imports System.Threading
```

We need to import the System.Threading namespace in order to use Threas.sleep method.

```
Dim Incoming, SubStr As String
Dim MessageNo, N As Integer
```

Some variables that will be used in the application.

```
Using comm As IO.Ports.SerialPort =
    My.Computer.Ports.OpenSerialPort("COM4", 9600)
```

When we read/write to a serial port we need to open it for reading/writing this method is used to initialize a serial port at COM4. If your GSM/GPRS modem is installed on another port, say COM3, you should replace the port name to the actual comm port in this code.

```
comm.DtrEnable = True
```

Before reading/writig to serial port we need to enable its data transfer ready mode.

```
comm.Write("AT" & vbCrLf)
comm.Write("AT+cmgf=1" & vbCrLf)
```

These are the actual AT commands that we used to issue in HyperTerminal. AT+CMGF=1 is the command to instruct the mobile phone to operate in text mode.

```
comm.Write("AT+CMGW=" & Chr(34) & MobNo & Chr(34) & vbCrLf)
```

AT+CMGW is used to write an SMS text message to the message storage of the mobile phone. MobNo is the mobile phone no. to which the message has to be sent. Note that the mobile phone no. has to be enclosed in double quotes for which the chr(34) is used in the code.

```
comm.Write(tbMessage & Chr(26) & vbCrLf)
```

After issuing the `AT+CMGW` command the serial port waits for the user input and accepts the input until Ctrl-Z is entered. In this line of code the message in the `tbMessage` textbox is written to the serial port and the Ctrl-Z character represented by `Chr(26)` is sent. This line writes the message to the message storage of the mobile phone.

Now that the message has been successfully written to the mobile phone memory, we need to send it. For sending a message from the mobile phone memory to the destination mobile no. we need to find the Message No. of the message in the mobile phone storage. This is done by the following code:

```
Incoming = comm.ReadExisting
N = Incoming.IndexOf("+CMGW:")
N = N + 6
SubStr = Incoming.Substring(N)
MessageNo = Val(SubStr)
```

When the `MessageNo` is identified we can use it to send the message to the destination. This is done by the following code:

```
comm.WriteLine("AT+CMSS=" & MessageNo & vbCrLf)
```

And that's it. We have successfully sent a message to the destination from our application.

Receive SMS in VB .NET

Receiving a SMS in VB .NET is actual retrieving the received SMSs in a mobile phone or GSM/GPRS modem. For this purpose I have created a simple form `frmReceiveSMS`. The screenshot of this form is given in figure-2 is given as follows:

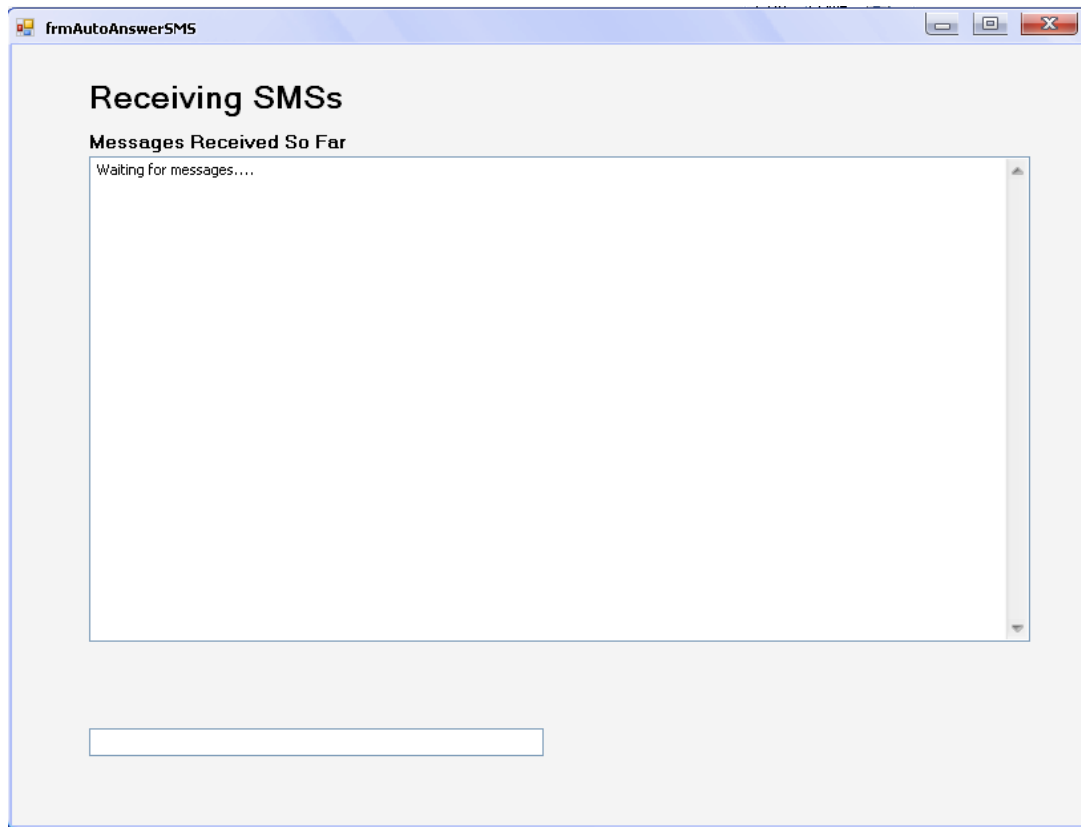


Figure 2: Receive SMS

This form has two label controls, two text boxes and one timer control (invisible). The instance of the SIMServices is created at the start of the class as follows:

```
Dim sim As New SIMServices("COM4")
```

The `Interval` property of the Timer control `Timer1` is set as 20000 so that the application checks the SIM at the interval of 20 seconds. The following code is executed on the `TimerTick` event of the `Timer1` object.

```
Private Sub Timer1_Tick(ByVal sender As System.Object,  
                       ByVal e As System.EventArgs)  
    Handles Timer1.Tick  
  
    Dim str As String  
    tbStatus.Text = "Reading SMSs"  
  
    ' Read All the Messages  
    sim.ReadAllSMSMessages()  
  
    Dim message, reply As New SMSMessage  
    tbStatus.Text = "Unread Messages: " & sim.UnreadMessages.Count
```

```
' Display all the unread messages in the text box tbMessage.  
For Each message In sim.UnreadMessages  
    str &= message.MobileNo & ", " & message.Message & vbCrLf  
    tbMessage.Text = str  
Next  
tbStatus.Text = "Idle..."  
End Sub
```

This code segment reads all the unread messages and displays them in the text box `tbMessage`.

Conclusion

In this small tutorial article I have explained to teach "How to create a small application in VB .NET for sending and receiving SMS. In the next article (part-3) I will write a small "SMS Auto Answering Application".

About the Author

Haider Ali did MSc in Computer Science from Department of Computer Science University of Peshawar, Pakistan in 1993. He has 14+ years of professional experience and is presently working as Lecturer in College of Arts and Science at Wadi Ad-Dawasir, King Saud University, Saudi Arabia. His email address is haiderkhalil@yahoo.com