

'Smart Body Check System interfacing software

'Written by: Dillian Wong, Apr 15 2004

'Purpose:

'It is a simple program target on demonstrating the connection between DTMF Receiver and PC through RS232 port. When the DTMF Receiver receiver the tone, it will send the tone data to PC. The valid data with include the start character "#" sign and stop character "*" sign. The first data with have many "#" sign for synchronization. Therefore, if the first start character is distorted, it still have second and third "#" for synchronization. When the following character not "#" sign, that means the character is data until receive "*" sign. After the valid data is extracted, the data will store in text file "txtSBSData.txt".

P R O T O C A L

' Example : ###ABC*#12345*

' ^ ^ ^ ^ ^ ^
' | | | | |
' | | | | | End of second data
' | | | | | second data
' | | | | | start of second data
' | | | | |
' | | | | | End of first data
' | | | | | first data
' | | | | |
' | | | | | synchronization

Option Explicit

'forces variables to be explicitly declared

Dim isDataFetching As Boolean

'indicate whether there have a data input stream

Dim isDataValidForExtract As Boolean

'indicate whether the data is ok for extract.

'when # sign indicate data is not ok

'when \$ sign indicate data is ok. ie we can get it!

Dim dataNo As Integer

'the data sequence number.

Dim bufferData(1000) As String

'the data will temporary store in buffer with size 1000

Dim isStart As Boolean

U P D A T A D A T A

'updata the data in VB label box, and write the data in text files.

Private Sub updata()

 Dim i As Integer

 If dataNo >= 2 Then

 dataNo = 0

 'reset buffer

 'display data

 For i = 1 To 2

 lstData(i).AddItem (bufferData(i))

 lblData(i).Caption = bufferData(i) 'for text file

 Next i

 lblData(3).Caption = Time

 'for text file

 lblData(4).Caption = Date

 'for text file

 lstData(3).AddItem (lblData(3).Caption)

 lstData(4).AddItem (lblData(4).Caption)

 'record the data in text files

 Open App.Path & "\" & "txtSBSData.txt" For Append Access Read Write As #1

 Print #1, lblData(1).Caption & " " & lblData(2).Caption & " " & lblData(3)

).Caption & " " & lblData(4).Caption

 Close #1

 End If

End Sub

Private Sub cmdStart_Click()

 isStart = True

End Sub

```

'when the program start establish the connection immediately
Private Sub Form_Load()
    ConnectTerminal
    isDataValidForExtract = False
    isDataFetching = False
End Sub

'Terminal connection routine code
Sub ConnectTerminal()
    On Error Resume Next
    With MSComm1
        If .PortOpen = True Then
            .PortOpen = False
        Else
            .PortOpen = True
            If Err.Number <> 0 Then
                MsgBox "Com" & .CommPort & " is not available." & vbCrLf & _
                    Err.Description
                Err.Clear
            End If
        End If
    End With
End Sub

'when the program terminate, disconnect the connection
Private Sub Form_Unload(Cancel As Integer)
    If (MSComm1.PortOpen) Then
        MSComm1.PortOpen = False
    End If
End Sub

'Info about the program
Private Sub mnuAbout_Click()
    MsgBox "Basic Terminal Program Version 0.01. " _
        & "It is a a simple program target on demonstrating the connection" _
        & " between two communication port. " _
        , , "About"
End Sub

'clear the terminal text board
Private Sub mnuClearLst_Click()
    txtResponse = ""
End Sub

'print out the characters received from RS232 port
Private Sub MSComm1_OnComm()
    Dim txtBuf As String
    Dim i As Integer
    Dim c As Integer
    Dim CaptureNumData As Integer
    Dim dataLength As Integer

    With MSComm1
        Select Case .CommEvent
            Case comEvReceive
                txtBuf = .Input
                For i = 1 To Len(txtBuf)
                    c = Asc(Mid(txtBuf, i, 1))
                    txtResponse = txtResponse & Chr(c)
                Next i
            Case comEvTransmit
            Case comEvReceiveError
            Case comEvReceiveFull
            Case comEvReceiveOverflow
            Case comEvReceiveParity
            Case comEvReceiveFraming
            Case comEvReceiveClock
            Case comEvReceiveData
            Case comEvReceiveBreak
            Case comEvReceiveSignal
            Case comEvReceiveStatus
            Case comEvReceiveReady
            Case comEvReceiveError
            Case comEvReceiveFull
            Case comEvReceiveOverflow
            Case comEvReceiveParity
            Case comEvReceiveFraming
            Case comEvReceiveClock
            Case comEvReceiveData
            Case comEvReceiveBreak
            Case comEvReceiveSignal
            Case comEvReceiveStatus
            Case comEvReceiveReady
        End Select
    End With

```

'check if input is particular, if particular place it

```

'in special label box.With synchronize capability.
Select Case Chr(c)
    Case "#"
        'start data fetching
        lblBuffer.Caption = ""
        isDataValidForExtract = False
        If Not (isDataFetching) Then
            isDataFetching = True
            dataNo = dataNo + 1    'data sequence increase by 1
        End If
    Case "*"
        'end of data
        isDataValidForExtract = True
        isDataFetching = False
End Select

If (isDataFetching) Then
    If Not Chr(c) = "#" Then
        lblBuffer = lblBuffer.Caption & Chr(c)
    End If
End If
If (isDataValidForExtract) Then
    'store the data in appropriate
    bufferData(dataNo) = lblBuffer.Caption 'sequence
    Call updata
End If

Next i
End Select
End With

txtResponse.SelStart = Len(txtResponse)
txtTextOut.SetFocus
End Sub

'send the character to the terminal
Private Sub txtTextOut_KeyPress(KeyAscii As Integer)
    MSComm1.Output = Chr(KeyAscii)
    txtTextOut.Text = ""
End Sub

```