

### The HX19 Access Program

*This program is supplied to help the user get started using the hx19, i.e. the only purpose is to make it easier for the programmer to write his own code.*

Please be aware that this demo software is written using Visual Basic 6, so there may be compatibility issues. VB6 is a dying programming language, and will not be supported in the future by Microsoft. The demo program made available is compiled using windows XP service pack 3. If you are attempting to install the program on XP service pack lower than sp3, then the DLL functions may be out of date and you may be asked if you wish to replace them. We don't recommend you replace the functions, if you don't there may be a compatibility issues when you run the hx19access program.

When the program hx19access (RF network access) is executed for the first time, a red window will pop up requesting the correct Com Port. The user can go to windows control panel and find "Device Manager". Under Ports (COM & LPT) the port number to which the Hx19ms is linked can be found. Visual basic may not find ports over 16. In case all ports are occupied, it may be possible to share one of the already allocated ports under 16. After typing in the serial port number then click on Ok, the port number is stored on a file called port.txt, if needed the port can be changed there.

The hx19ms must be connected to a USB port before the hx19access program is executed.

```
Dim linebuffer(100) As String      'only used for the scrolling routine, not linked to hx19 operations
Dim nn%
Dim engage As Boolean
```

```
Private Sub Form_Activate()
Dim cc As String
```

```
'The following code examines the com port selected to see if it is open and the device available
'if there is an error, it gives the user a chance to enter in the correct port number
'use Control Panel to determine which port is being used for the hx19 system
'this code section is not important for using the hx19
```

```
Text4.Visible = False
Command1.Visible = False
Check1.Visible = True
```

```
On Error GoTo fixit
Open "port.txt" For Input As 1
Input #1, Port
Close 1
GoTo allOK
fixit:
Check1.Visible = False
Text4.Visible = True
Command1.Visible = True
```

```
Text6 = "Type port number into the blue window above and click OK (find the correct port number under Device Manager)"
```

```
Do: DoEvents: Loop Until engage = True
Open "port.txt" For Output As 1
Print #1, Val(Text4)
Close 1
Text4.Visible = False
Command1.Visible = False
Check1.Visible = True
Port = Val(Text4)
```

```
allOK:
```

```
Text6 = ""
```

```
'END PORT NUMBER CHECK -----
```

```
'The following code segment is the heart of the interface to the hx19ms
```

```
Form1.MSComm1.CommPort = Port      'this port number must correspond to the hx19ms usb port
Form1.MSComm1.Settings = "256000,N,8,1" '256Kbaud, no parity, 8bit interchange and 1 stop bit
Form1.MSComm1.PortOpen = True
Form1.MSComm1.InputLen = 1          'get serial characters one by one
Form1.MSComm1.InBufferCount = 0     'make sure there is no data residue from last application in buffers
Form1.MSComm1.OutBufferCount = 0    'make sure output buffer is empty
ticut = 0
Do
Do:                                'loop until there's a character in input buffer
DoEvents
Loop While Form1.MSComm1.InBufferCount < 1

cc = Form1.MSComm1.Input
If cc = Chr(13) Then                'using MSComm1 control component supplied with visual basic 6
tScroll comline + vbCrLf            'input a character from hx19ms Mscomm1
If Check2.Value = 1 Then Print #1, comline 'if it is a data delimiter (carriage return) then display line
comline = ""                        'tScroll scrolls display down one line
Else                                'save data on file 1 when log option is selected
comline = comline + cc              'clear this data line and prepare for then next
Text3 = Text3 + cc                  'while there's no delimiter collect data into a line
End If                              'accumulate data into the text to be scrolled
Loop                                'continuous loop

End Sub
```

```
Private Sub Check1_Click()  
    'Any hx19ms receiving the command $ will initiate synchronized strobing  
    'the command % stops the sync sequence  
    If Check1.Value = 1 Then checkOut "M&$" Else checkOut "M&%"  
End Sub  
  
Private Sub Command3_Click()  
    checkOut Text6  
End Sub  
    'here the content of Text6 is transmitted via usb com port to the hx19ms  
  
Private Sub checkOut(temst as string)  
Dim xsum as integer, xx as string  
    'this routine sums up all Ascii characters entered, and creates an hx19 accepted checksum.  
    xsum = 0  
    For i = 1 To Len(temst)  
        xx = Mid(temst, i, 1)  
        xsum = xsum + Asc(xx)  
    Next  
    temst = temst + "/" + Hex(xsum)  
    Form1.MSComm1.Output = temst + Chr(13)  
End Sub  
    'compute the checksum of the string  
    'accumulate ASCII codes  
    'append the checksum in hexadecimal format  
    'sends a string to the hx19ms with the correct checksum attached  
  
Private Sub Text6_Change()  
    'This routine keeps track of the total characters entered, should not exceed 116 characters.  
    Text5 = Format(Len(Text6), "#")  
End Sub
```

### 'Remaining routines are unimportant and secondary to the understanding of the hx19

```
Private Sub Check2_Click()  
    'data coming from the hx19ms is text format and can be viewed using windows notepad  
    If Check2.Value = 1 Then  
        Open "hx19.log" For Output As 1  
        Else  
            Close 1  
        End If  
End Sub  
    'save incoming hx19ms text data on file called hx19.log  
  
Private Sub Command1_Click()  
    engage = True  
End Sub  
    'in case of a com error, user may input correction and then continue  
  
Private Sub tScroll(nline)  
Dim jj%  
    'routine scrolls down one line for display purposes only, it has otherwise nothing to do with hx19 system  
    linebuffer(nn) = nline  
    nn = (nn + 1) And 7  
    jj = nn + 1  
    Text3 = ""  
    Do  
        Text3 = Text3 + linebuffer(jj)  
        jj = (jj + 1) And 7  
    Loop Until jj = nn  
End Sub  
    'scrolls 16 lines of text through text window  
  
Private Sub Form_Terminate()  
    Close  
End Sub  
    'make sure no files are left open when the program ends  
    'if end isn't executed before termination, com port may be left open  
  
Private Sub Form_Unload(Cancel As Integer)  
    Close  
End Sub  
    'to be absolutely sure  
    'make sure no files are left open when the program ends  
    'if end isn't executed before termination, com port may be left open
```