RF Monolithics, Inc.

DM1800

Poll Demo and Application Software

Using

Microsoft Visual Basic 6.0

By

Bob Nelson

2/10/2006

Scope:

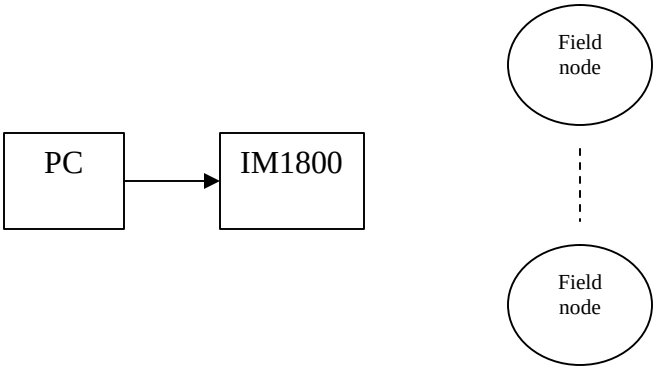
This application will demonstrate the ease of implementing a polling commands to interface to the IM1800 communication port and the DM1800 Base station. The software will show the user how to do the following;

1. Poll a field node
2. Receive power on message from a field node or router
3. Write a digital One to the digital out pin on the field node
4. Write a digital zero to the digital out pin on the field node

Software:

The application is written in Microsoft Visual Basic 6.0 and requires one USB port on a PC. Also the application assumes the up to 4 field nodes have been bound to the base station and are operational.

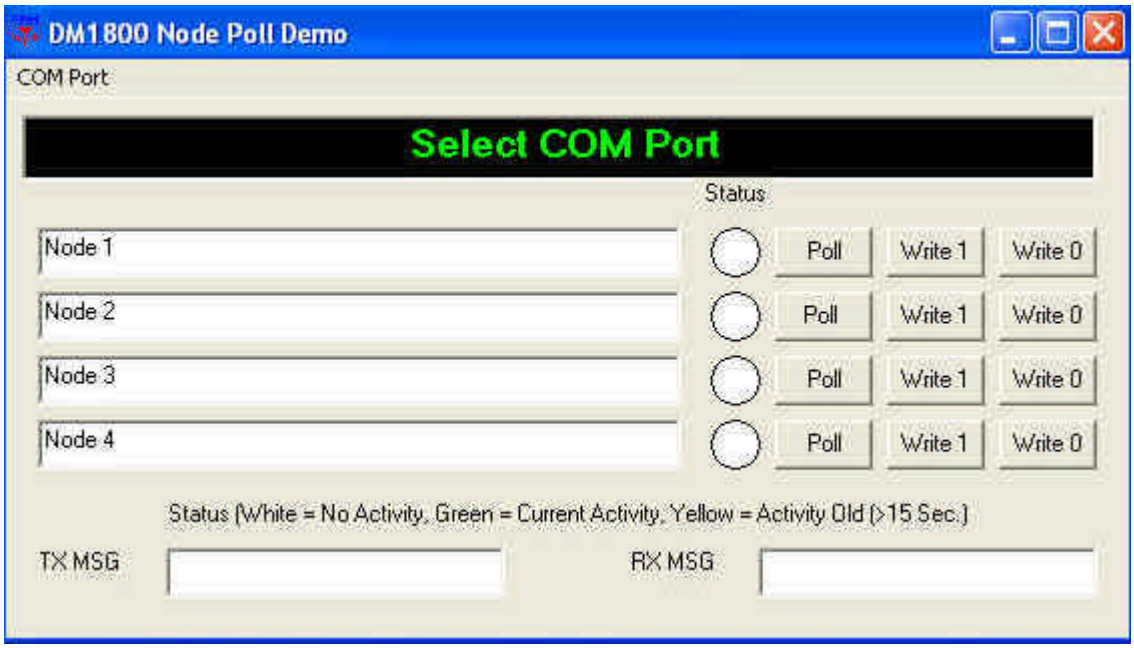
Connection:



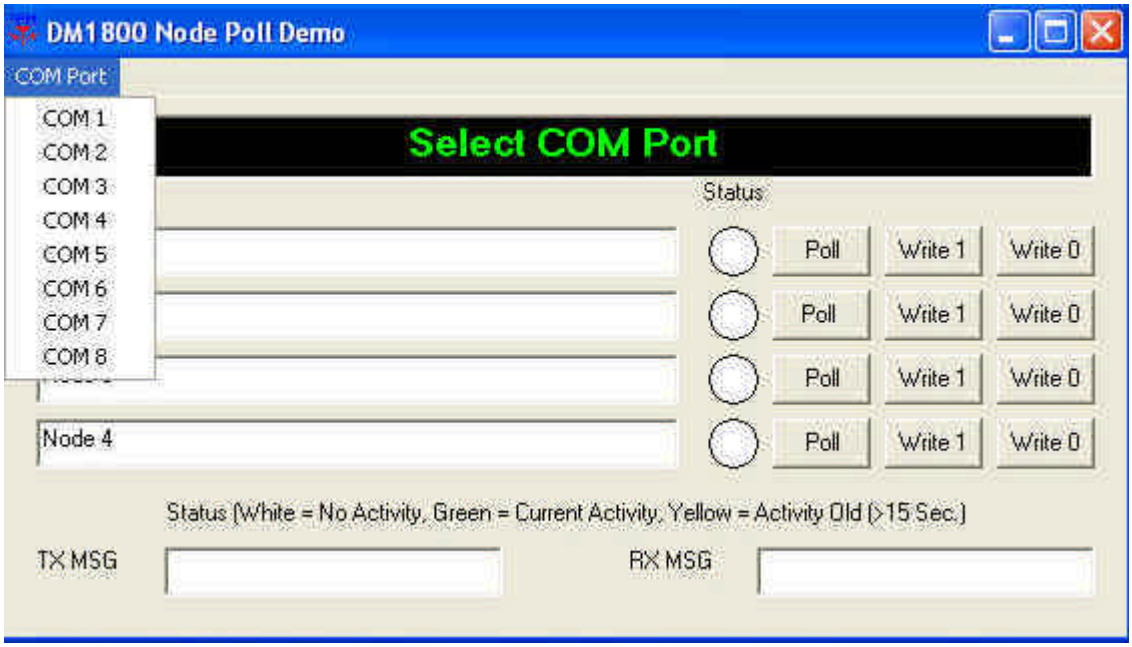
One PC, one IM1800 and 1 to 4 field nodes are required for this application. You may have up to seven routers but are not required.

Startup:

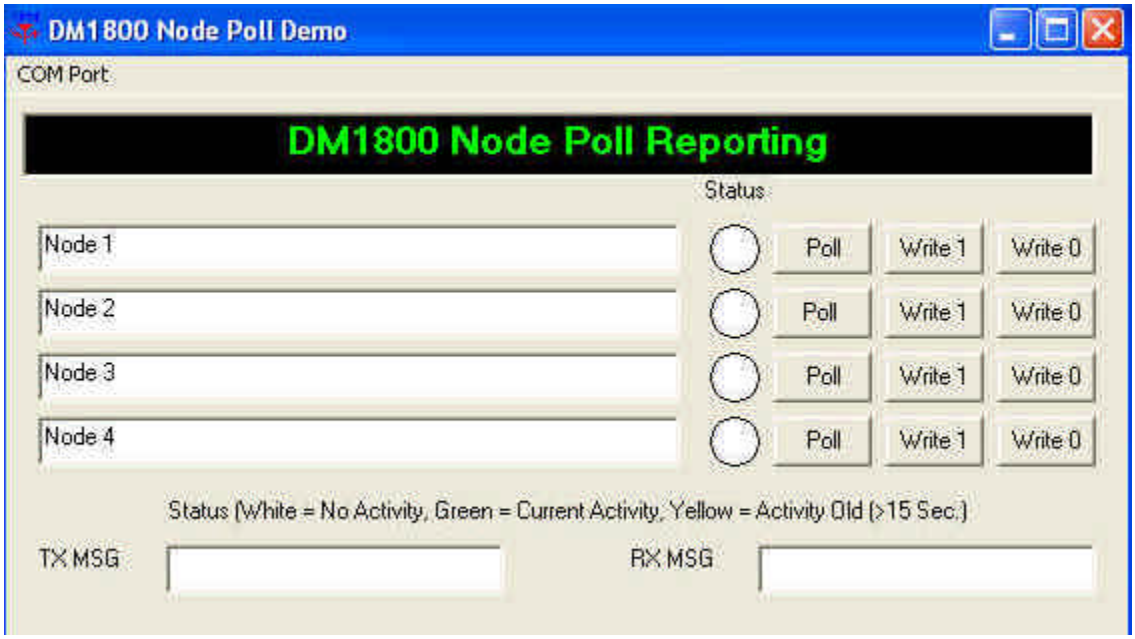
Copy "DM1800 Poll.exe" to a directory of your choice and then double click "DM1800 Poll" to run the program. You will see the following;



Click “Com Port” and select the comm. 1 though 8.



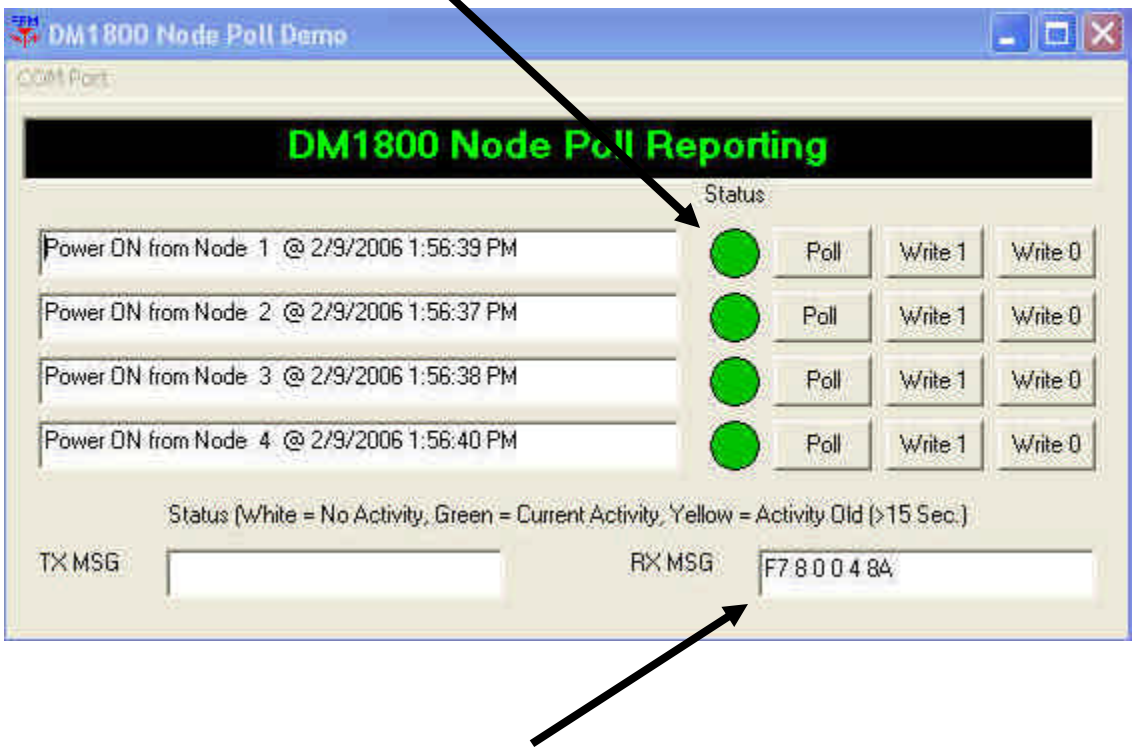
After selecting a valid com port you should see the following;



Operation:

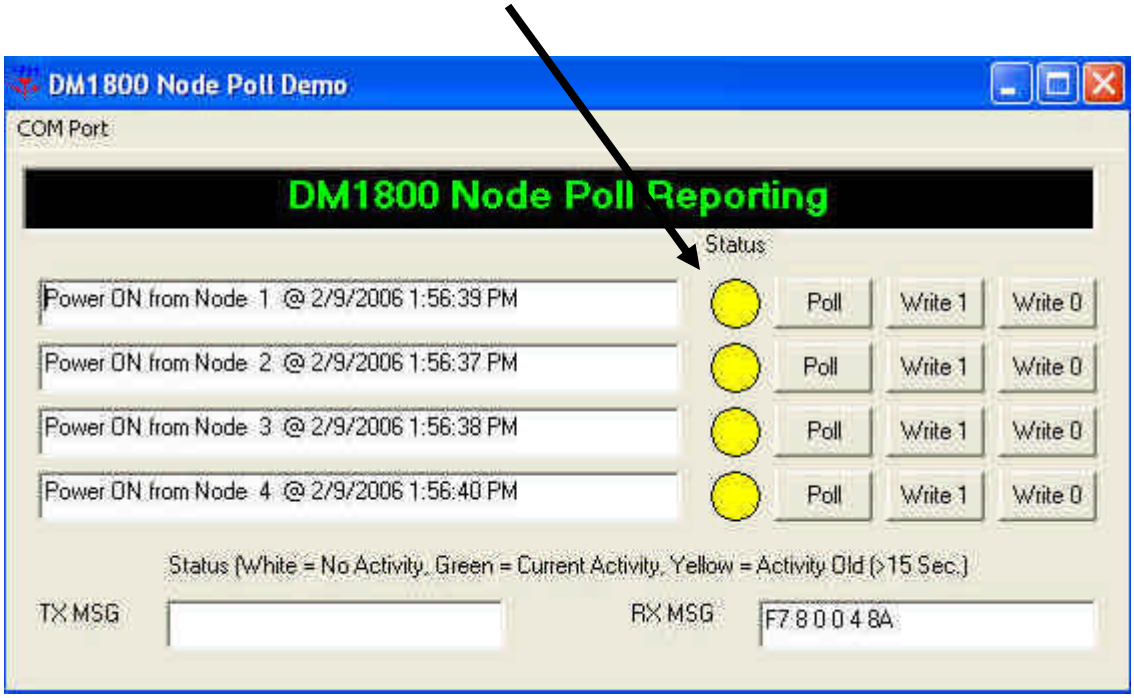
You may now turn on any of the field nodes and you should see the following;

The Green status light will stay green for approximately 15 seconds from any activity. After ~15 seconds the indictor will change to yellow if no activity is seen from that field node.

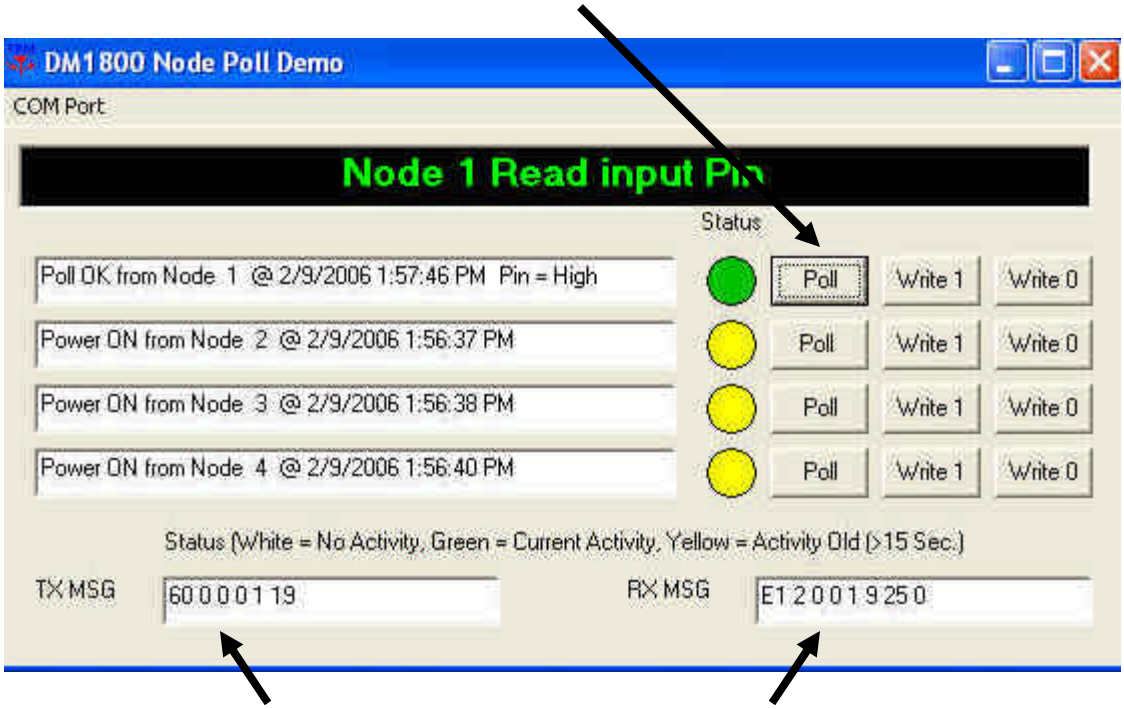


The RX MSG will display the most current message received from the field node in Hex with leading zeros suppressed. As an example the above message is F7 08 00 00 04 8A.

Example of no activity from any filed node for at least 15 seconds.

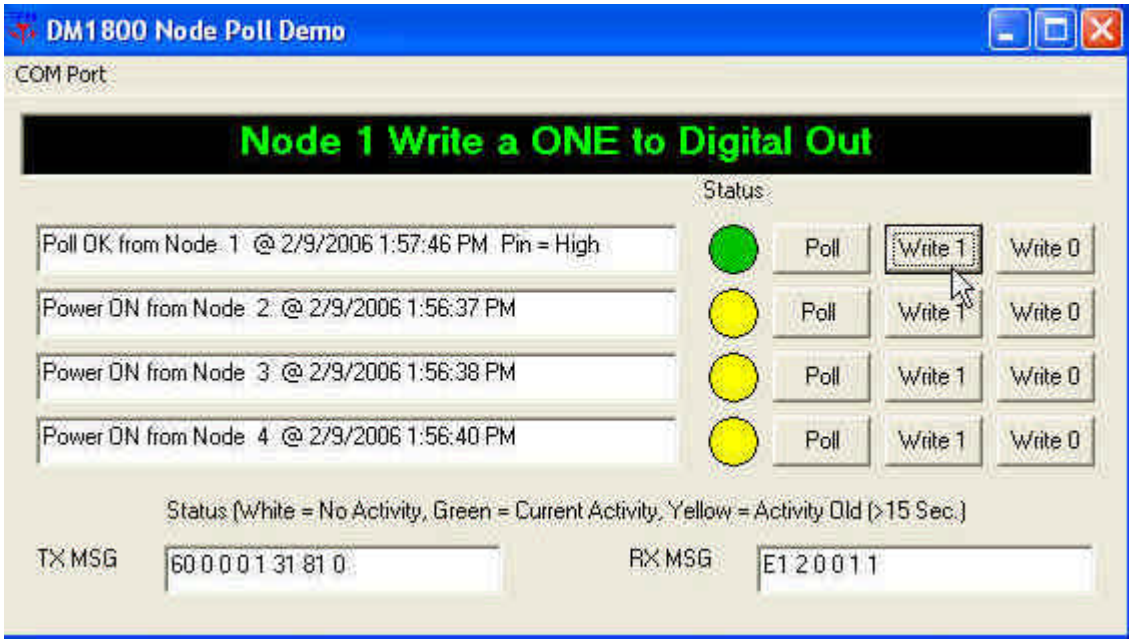


Pressing the “Poll” button for field node 1.

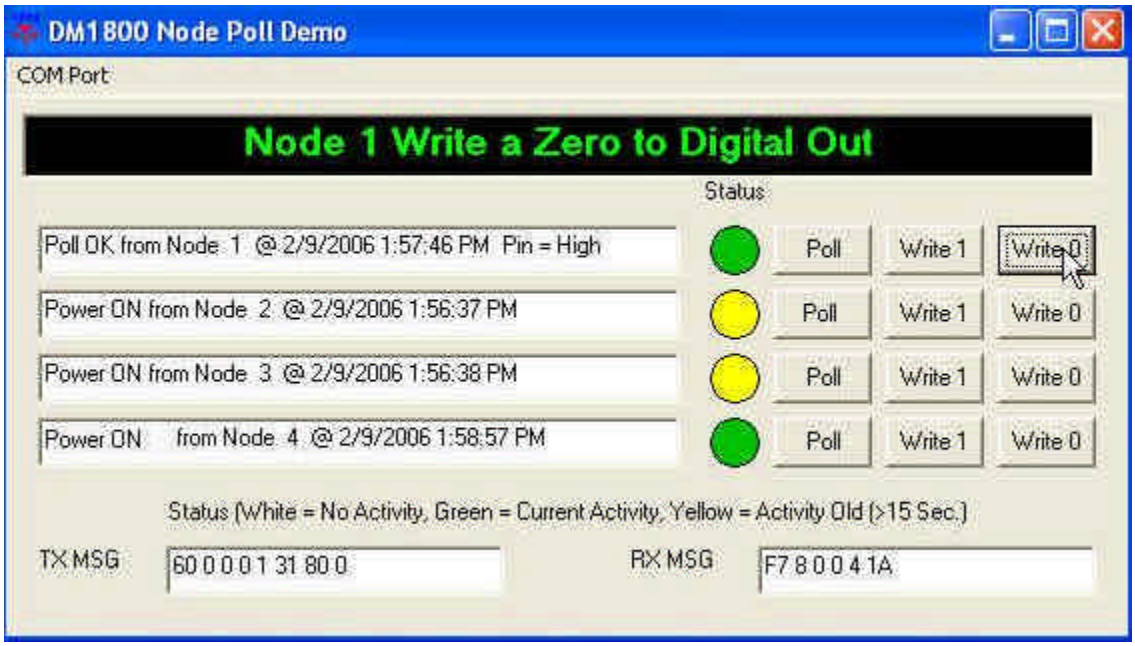


Notice the TX message for Poll and the response message from the field node.
The status results will be displayed, in this example the digital in pin is high.

Pressing the Write 1 button will write a one to the digital out pin.

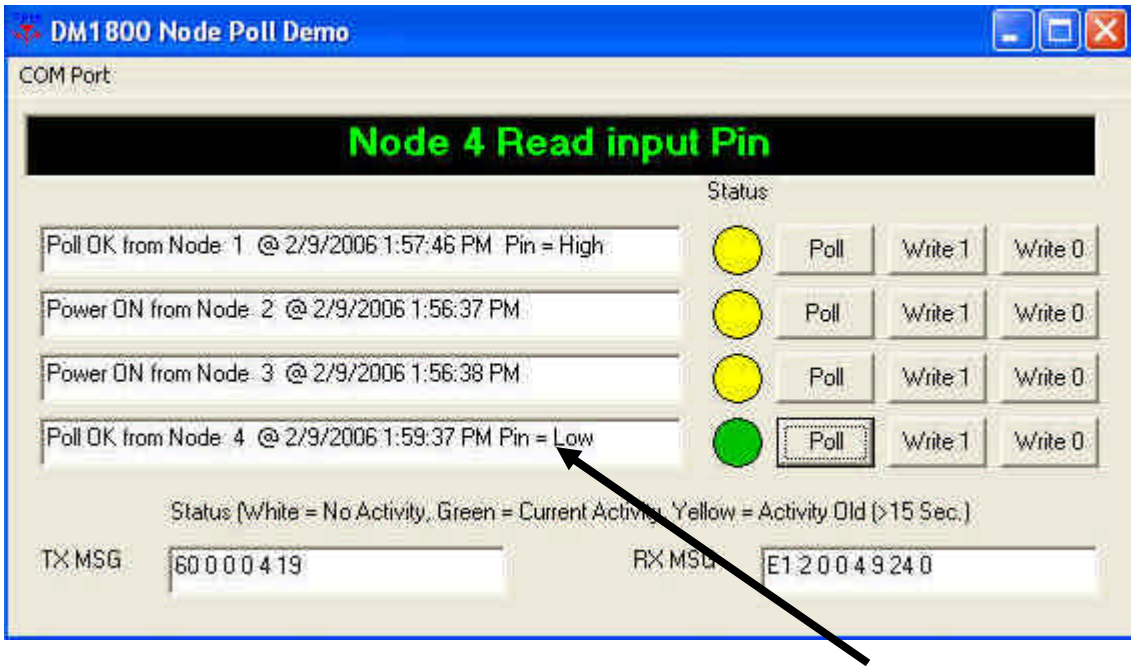


Press the Write 0 will write a zero to the digital out pin.

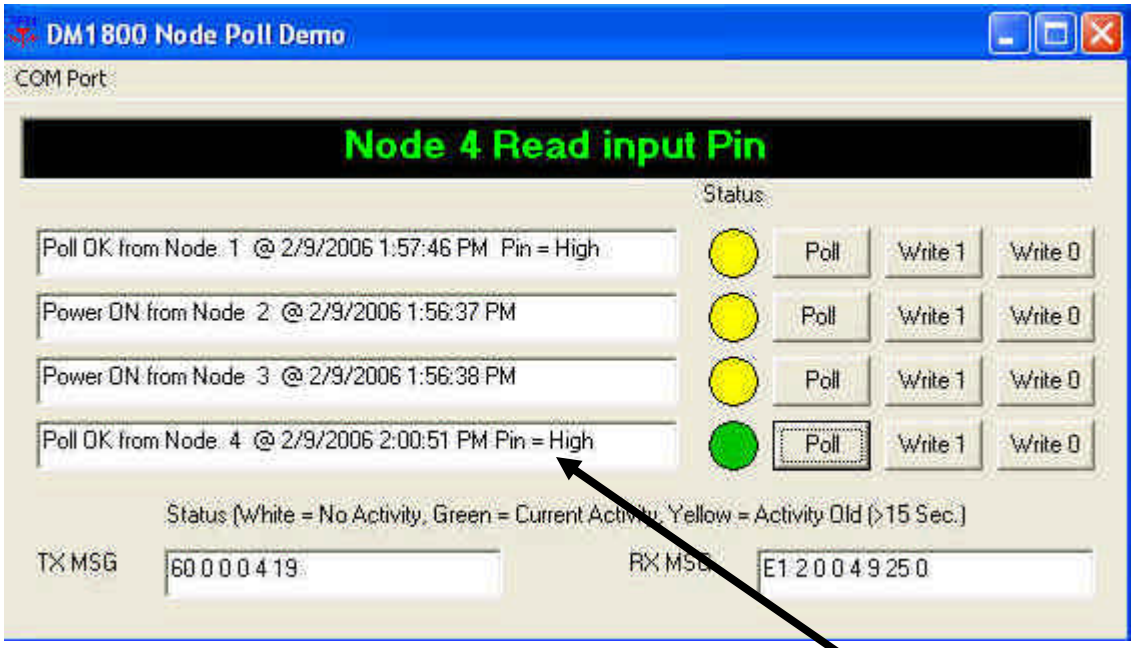


In this example, digital pin out on node one is connected to digital pin in on node 4.

By pressing “poll” for field node 4, the status of the digital pin will be displayed. In this case it will follow the digital out pin of field node 1.



The last command was to write a zero to digital out pin on node 1. Polling node 4 now shows the input to be low.



Pressing “write 1” on node 1 and then pressing “poll” on node 4 will now show high.

Software Listing:

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Begin VB.Form Form1
   Caption         =   "DM1800 Node Poll Demo"
   ClientHeight    =   4035
   ClientLeft      =   165
   ClientTop       =   555
   ClientWidth     =   8415
   FillColor       =   &H0000FFFF&
   ForeColor       =   &H00000000&
   Icon            =   "DM1800 Poll.frx":0000
   LinkTopic       =   "Form1"
   ScaleHeight     =   4035
   ScaleWidth      =   8415
   StartUpPosition =   1  'CenterOwner
   Begin VB.TextBox Text6
      Height          =   375
      Left            =   1200
      TabIndex        =   21
      Text            =   " "
      Top             =   3360
      Width           =   2535
   End
   Begin VB.TextBox Text5
      Height          =   375
      Left            =   5640
      TabIndex        =   19
      Text            =   " "
      Top             =   3360
      Width           =   2535
   End
   Begin VB.CommandButton Command12
      Caption         =   "Write 0"
      Height          =   375
      Left            =   7440
      TabIndex        =   18
      Top             =   2400
      Width           =   735
   End
   Begin VB.CommandButton Command11
      Caption         =   "Write 0"
      Height          =   375
      Left            =   7440
      TabIndex        =   17
      Top             =   1920
      Width           =   735
   End
   Begin VB.CommandButton Command10
      Caption         =   "Write 0"
      Height          =   375
      Left            =   7440
      TabIndex        =   16
      Top             =   1440
      Width           =   735
   End
   Begin VB.CommandButton Command9
      Caption         =   "Write 0"
      Height          =   375
      Left            =   7440
      TabIndex        =   15
      Top             =   960
      Width           =   735
   End
   Begin VB.CommandButton Command8
      Caption         =   "Write 1"
      Height          =   375
      Left            =   6600
      TabIndex        =   14
      Top             =   2400
      Width           =   735
   End
   Begin VB.CommandButton Command7
      Caption         =   "Write 1"
      Height          =   375
      Left            =   6600
      TabIndex        =   13
      Top             =   1920
```

```
   Width           =   735
End
Begin VB.CommandButton Command6
   Caption         =   "Write 1"
   Height          =   375
   Left            =   6600
   TabIndex        =   12
   Top             =   1440
   Width           =   735
End
Begin VB.CommandButton Command5
   Caption         =   "Write 1"
   Height          =   375
   Left            =   6600
   TabIndex        =   11
   Top             =   960
   Width           =   735
End
Begin VB.CommandButton Command4
   Caption         =   "Poll"
   Height          =   375
   Left            =   5760
   TabIndex        =   8
   Top             =   2400
   Width           =   735
End
Begin VB.CommandButton Command3
   Caption         =   "Poll"
   Height          =   375
   Left            =   5760
   TabIndex        =   7
   Top             =   1920
   Width           =   735
End
Begin VB.CommandButton Command2
   Caption         =   "Poll "
   Height          =   375
   Left            =   5760
   TabIndex        =   6
   Top             =   1440
   Width           =   735
End
Begin VB.CommandButton Command1
   Caption         =   "Poll"
   Height          =   375
   Left            =   5760
   TabIndex        =   5
   Top             =   960
   Width           =   735
End
Begin VB.TextBox Text4
   Height          =   375
   Left            =   240
   TabIndex        =   4
   Text            =   "Node 4"
   Top             =   2400
   Width           =   4815
End
Begin VB.TextBox Text3
   Height          =   375
   Left            =   240
   TabIndex        =   3
   Text            =   "Node 3"
   Top             =   1920
   Width           =   4815
End
Begin VB.TextBox Text2
   Height          =   375
   Left            =   240
   TabIndex        =   2
   Text            =   "Node 2"
   Top             =   1440
   Width           =   4815
End
Begin VB.TextBox Text1
   Height          =   375
   Left            =   240
   TabIndex        =   1
   Text            =   "Node 1"
   Top             =   960
   Width           =   4815
```

```
   End
   Begin MSCommLib.MSComm MSComm1
      Left            =   4560
      Top             =   120
      _ExtentX        =   1005
      _ExtentY        =   1005
      _Version        =   393216
      DTREnable       =   -1  'True
   End
   Begin VB.Timer Timer1
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   4080
      Top             =   120
   End
   Begin VB.Label Label5
      Caption         =   "TX MSG"
      Height          =   255
      Left            =   240
      TabIndex        =   22
      Top             =   3360
      Width           =   855
   End
   Begin VB.Label Label4
      Caption         =   "RX MSG"
      Height          =   255
      Left            =   4680
      TabIndex        =   20
      Top             =   3360
      Width           =   855
   End
   Begin VB.Label Label3
      Alignment       =   2  'Center
      Caption         =   "Status (White = No Activity, Green = Current Activity, Yellow = Activity
Old (>15 Sec.)"
      Height          =   255
      Left            =   240
      TabIndex        =   10
      Top             =   3000
      Width           =   7935
   End
   Begin VB.Label Label2
      Alignment       =   2  'Center
      Caption         =   "Status"
      Height          =   255
      Left            =   5160
      TabIndex        =   9
      Top             =   600
      Width           =   615
   End
   Begin VB.Shape Shape4
      FillColor       =   &H00FFFFFF&
      FillStyle       =   0  'Solid
      Height          =   375
      Left            =   5160
      Shape           =   3  'Circle
      Top             =   2400
      Width           =   615
   End
   Begin VB.Shape Shape3
      FillColor       =   &H00FFFFFF&
      FillStyle       =   0  'Solid
      Height          =   375
      Left            =   5160
      Shape           =   3  'Circle
      Top             =   1920
      Width           =   615
   End
   Begin VB.Shape Shape2
      FillColor       =   &H00FFFFFF&
      FillStyle       =   0  'Solid
      Height          =   375
      Left            =   5160
      Shape           =   3  'Circle
      Top             =   1440
      Width           =   615
   End
   Begin VB.Shape Shape1
      FillColor       =   &H00FFFFFF&
      FillStyle       =   0  'Solid
      Height          =   375
```

```
   Left            =   5280
   Shape           =   3  'Circle
   Top             =   960
   Width           =   375
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackColor       =   &H00000000&
   BorderStyle     =   1  'Fixed Single
   Caption         =   " Select COM Port"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   13.5
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FF00&
   Height          =   495
   Left            =   120
   TabIndex        =   0
   Top             =   120
   Width           =   8055
End
Begin VB.Menu mnuSerial
   Caption         =   "COM Port"
   Begin VB.Menu mnuCOM1
      Caption         =   "COM 1"
   End
   Begin VB.Menu mnuCOM2
      Caption         =   "COM 2"
   End
   Begin VB.Menu mnuCOM3
      Caption         =   "COM 3"
   End
   Begin VB.Menu mnuCOM4
      Caption         =   "COM 4"
   End
   Begin VB.Menu mnuCOM5
      Caption         =   "COM 5"
   End
   Begin VB.Menu mnuCOM6
      Caption         =   "COM 6"
   End
   Begin VB.Menu mnuCOM7
      Caption         =   "COM 7"
   End
   Begin VB.Menu mnuCOM8
      Caption         =   "COM 8"
   End
End
Begin VB.Menu mnuNode
   Caption         =   "Node"
   Enabled         =   0   'False
   Visible         =   0   'False
   Begin VB.Menu mnuNode1
      Caption         =   "Node1"
      Checked         =   -1  'True
      Enabled         =   0   'False
   End
   Begin VB.Menu mnuNode2
      Caption         =   "Node 2"
   End
   Begin VB.Menu mnuNode3
      Caption         =   "Node 3"
   End
   Begin VB.Menu mnuNode4
      Caption         =   "Node 4"
   End
   Begin VB.Menu mnuNode5
      Caption         =   "Node 5"
   End
   Begin VB.Menu mnuNode6
      Caption         =   "Node 6"
   End
   Begin VB.Menu mnuNode7
      Caption         =   "Node 7"
   End
   Begin VB.Menu mnuNode8
```

```
         Caption         =   "Node 8"
      End
      Begin VB.Menu mnuNode9
         Caption         =   "Node 9"
         Enabled         =   0   'False
         Visible         =   0   'False
      End
      Begin VB.Menu mnuNode10
         Caption         =   "Node 10"
         Enabled         =   0   'False
         Visible         =   0   'False
      End
   End
   Begin VB.Menu mnuUnits
      Caption         =   "Units"
      Visible         =   0   'False
      Begin VB.Menu mnuF
         Caption         =   "Deg F"
         Checked         =   -1  'True
      End
      Begin VB.Menu mnuC
         Caption         =   "Deg C"
      End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'*************************************************************************************
'*************************************************************************************
'*********************    DM1800 Poll.FRM, 02/9/2006          ********************
'*********************   DM1800 Poll tutorial software         ********************
'*********************             Poll demo                   ********************
'*********************  NO representation is made that this    ********************
'*********************  software is suitable for any purpose   ********************
'********************* Compiled in Microsoft Visual Basic 6.0  ********************
'*************************************************************************************
'*********************               Bob Nelson                ********************
'*********************                  RFM                    ********************
'*********************              Dallas, Texas              ********************
'*************************************************************************************
'*************************************************************************************
'
'*************************************************************************************
'***************************** Function of Program ***********************************
'*************************************************************************************
'This program looks for the following message from a node that has Event turned on
'
'For a Poll the tx msg looks like in hex for node 1;
' 60 00 00 00 01 19
'
'To Write one to Digial Pin on node 1;
' 60 00 00 00 01 31 81 00
'
'To Write zero to Digital pin on node 1;
' 60 00 00 00 01 31 80 00
'
'Responce for a write to digial pin node 1
' E1 02 00 00 01 01
'
'Responce for a Poll of node 1 digital pin high
' E1 02 00 00 01 09 25 00
'
'Responce for a Poll of node 1 digital pin low
' E1 02 00 00 01 09 24 00
'
'
'*************************************************************************************
'Remember each message must have the DLE and STX
' 10 03 message 10 02
'
'*************************************************************************************
'*************************************************************************************
'
'
'
' declare variables:
  Dim J As Integer                                          ' RPkt$ string position
  Dim K As Integer                                          ' current message length
```

```
  Dim Q As Integer                                        ' FIFO$ length
  Dim R As Integer                                        ' DLE flag
  Dim U As Integer                                        ' build packet flag
  Dim Y As Integer                                        ' FOR loop counter
  Dim ComData$                                            ' COM input string
  Dim C$                                                  ' RX character buffer
  Dim FIFO$                                               ' RX character FIFO
  Dim RPkt$                                               ' response message
  Dim DLE$                                                ' bi-sync frame character
  Dim STX$                                                ' bi-sync start character
  Dim ETX$                                                ' bi-sync end character
  Dim Core$                                               ' command Read bin input
  Dim ACore$                                              ' addressed core command
  Dim GTPkt$                                              ' bi-sync framed Read ADC command
  Dim dumprx$
  Dim dumptx$
  Dim N As Integer                                        ' string length
  Dim M As Integer                                        ' checksum accumulator
  Dim CK As Byte                                          ' checksum byte
  Dim AID As Byte                                         ' AID address (low byte)
  Dim X As Integer                                        ' array loop counter
  Dim P As Integer                                        ' COM port
  Dim NN$                                                 ' node number
  Dim Node1 As Integer
  Dim Node1timer As Integer
  Dim Node2 As Integer
  Dim Node2timer As Integer
  Dim Node3 As Integer
  Dim Node3timer As Integer
  Dim Node4 As Integer
  Dim Node4timer As Integer


Private Sub Command1_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&H19)          ' Node 1 Read input command
ACore$ = Core$                                            ' default addressed core command

Shape1.FillColor = &HFF&
Node1 = 1
Node1timer = 0

Call TXCmd                                                ' call to send TX command
Label1.Caption = "Node 1 Read input Pin"                  ' update display
End Sub

Private Sub Command10_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H2) + Chr$(&H31) + Chr$(&H80) + Chr$(&H0) ' Node 2 Write a Zero to
Digital port command

ACore$ = Core$                                             ' default addressed core command

Shape2.FillColor = &HFF&
Node2 = 1
Node2timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 2 Write a Zero to Digital Out"       ' update display

End Sub

Private Sub Command11_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H3) + Chr$(&H31) + Chr$(&H80) + Chr$(&H0) ' Node 3 Write a Zero to
Digital port command

ACore$ = Core$                                             ' default addressed core command

Shape3.FillColor = &HFF&
Node3 = 1
Node3timer = 0

Call TXCmd                                                  ' call to send TX command

Label1.Caption = "Node 3 Write a Zero to Digital Out"       ' update display
End Sub

Private Sub Command12_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
```

```
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H4) + Chr$(&H31) + Chr$(&H80) + Chr$(&H0) ' Node 4 Write a Zero to
Digital port command

ACore$ = Core$                                                  ' default addressed core command

Shape4.FillColor = &HFF&
Node4 = 1
Node4timer = 0

Call TXCmd                                                       ' call to send TX command

Label1.Caption = "Node 4 Write a Zero to Digital Out"           ' update display
End Sub

Private Sub Command2_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H2) + Chr$(&H19)          ' Node 2 Read input command
ACore$ = Core$                                              ' default addressed core command

Shape2.FillColor = &HFF&
Node2 = 1
Node2timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 2 Read input Pin"                    ' update display
End Sub

Private Sub Command3_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H3) + Chr$(&H19)          ' Node 3 Read input command
ACore$ = Core$                                              ' default addressed core command

Shape3.FillColor = &HFF&
Node3 = 1
Node3timer = 0


Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 3 Read input Pin"                    ' update display
End Sub

Private Sub Command4_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H4) + Chr$(&H19)          ' Node 4 Read input command

ACore$ = Core$                                              ' default addressed core command

Shape4.FillColor = &HFF&
Node4 = 1
Node4timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 4 Read input Pin"                    ' update display
End Sub

Private Sub Command5_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&H31) + Chr$(&H81) + Chr$(&H0) ' Node 1 Write a ONE to
Digital port command

ACore$ = Core$                                                ' default addressed core command

Shape1.FillColor = &HFF&
Node1 = 1
Node1timer = 0

Call TXCmd                                                     ' call to send TX command

Label1.Caption = "Node 1 Write a ONE to Digital Out"           ' update display

End Sub

Private Sub Command6_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H2) + Chr$(&H31) + Chr$(&H81) + Chr$(&H0) ' Node 2 Write a ONE to
Digital port command

ACore$ = Core$                                                 ' default addressed core command

Shape2.FillColor = &HFF&
Node2 = 1
```

```
Node2timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 2 Write a ONE to Digital Out"        ' update display
End Sub

Private Sub Command7_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H3) + Chr$(&H31) + Chr$(&H81) + Chr$(&H0) ' Node 3 Write a ONE to
Digital port command

ACore$ = Core$                                             ' default addressed core command

Shape3.FillColor = &HFF&
Node3 = 1
Node3timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 3 Write a ONE to Digital Out"        ' update display
End Sub

Private Sub Command8_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H4) + Chr$(&H31) + Chr$(&H81) + Chr$(&H0) ' Node 4 Write a ONE to
Digital port command

ACore$ = Core$                                             ' default addressed core command

Shape4.FillColor = &HFF&
Node4 = 1
Node4timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 4 Write a ONE to Digital Out"        ' update display
End Sub

Private Sub Command9_Click()
Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&H31) + Chr$(&H80) + Chr$(&H0) ' Node 1 Write a Zero to
Digital port command

ACore$ = Core$                                              ' default addressed core command

Shape1.FillColor = &HFF&
Node1 = 1
Node1timer = 0

Call TXCmd                                                  ' call to send TX command
Label1.Caption = "Node 1 Write a Zero to Digital Out"       ' update display
End Sub

Private Sub Form_Load()

' initialize variables:
  J = 1                                                    ' clear string position
  K = 0                                                    ' clear message length
  Q = 0                                                    ' clear FIFO$ length
  R = 0                                                    ' clear DLE$ flag
  U = 0                                                    ' clear build flag
  Y = 0                                                    ' clear FOR counter
  ComData$ = ""                                            ' clear COM input string
  C$ = ""                                                  ' clear RX character buffer
  FIFO$ = ""                                               ' clear FIFO string
  RPkt$ = ""                                               ' clear response string
  DLE$ = Chr$(&H10)                                        ' DLE = &H10
  STX$ = Chr$(&H2)                                         ' STX = &H02
  ETX$ = Chr$(&H3)                                         ' ETX = &H03
  Core$ = Chr$(&H60) + Chr$(&H0) _
  + Chr$(&H0) + Chr$(&H0) + Chr$(&H1) + Chr$(&H19)         ' core Read input command
  ACore$ = Core$                                           ' default addressed core command
  P = 1                                                    ' default COM port number
  N = 0                                                    ' clear command length
  M = 0                                                    ' clear checksum accumulator
  CK = 0                                                   ' clear checksum byte
  AID = 1                                                  ' set default node address
  X = 0                                                    ' clear loop counter
  T = 0                                                    ' set display default as deg F
  NN$ = ""

  Form1.Left = (Screen.Width - Form1.Width) / 1.2          ' put form at
  Form1.Top = (Screen.Height - Form1.Height) / 5           ' upper right
```

```
  Timer1.Interval = 200                                 ' ser timer interval to 200 ms
  Timer1.Enabled = False                                ' start timer when COM port selected

End Sub



Private Sub Timer1_Timer()
  If MSComm1.InBufferCount > 0 Then                     ' if bytes in COM input buffer
    Call GetRX                                          ' call to get RX bytes
  End If
'  Call TXCmd                                           ' call to send TX command

  If Node1 = 1 Then Node1timer = Node1timer + 1          ' start activity timer for node
  If Node1timer = 75 Then                               ' 30 seconds
     Shape1.FillColor = &HFFFF&                         ' change color to yellow
     Node1 = 0                                          ' clear flags
     Node1timer = 0
  End If

  If Node2 = 1 Then Node2timer = Node2timer + 1          ' start activity timer for node
  If Node2timer = 75 Then                               ' 30 seconds
     Shape2.FillColor = &HFFFF&                         ' change color to yellow
     Node2 = 0                                          ' clear flags
     Node2timer = 0
  End If

  If Node3 = 1 Then Node3timer = Node3timer + 1          ' start activity timer for node
  If Node3timer = 75 Then                               ' 30 seconds
     Shape3.FillColor = &HFFFF&                         ' change color to yellow
     Node3 = 0                                          ' clear flags
     Node3timer = 0
  End If

  If Node4 = 1 Then Node4timer = Node4timer + 1          ' start activity timer for node
  If Node4timer = 75 Then                               ' 30 seconds
     Shape4.FillColor = &HFFFF&                         ' change color to yellow
     Node4 = 0                                          ' clear flags
     Node4timer = 0
  End If




End Sub

Public Sub TXCmd()
  N = Len(ACore$)                                       ' get addressed core command length
  M = 0                                                 ' clear checksum
  P = 1                                                 ' nibble counter
  For Y = 1 To N                                        ' loop to add all ACore bytes
    M = M + Asc(Mid$(ACore$, Y, 1))                     ' add each ACore byte to checksum
    dumptx$ = dumptx$ + Hex(Asc(Mid$(ACore$, Y, 1)))    ' build buffer of tx msg being sent
    P = P + 1
    If P = 2 Then                                       ' byte yet
      dumptx$ = dumptx$ + " "                           ' add space between bytes
      P = 1
    End If

  Next Y
  CK = CByte(M Mod 256)                                 ' checksum is lower byte
  GTPkt$ = DLE$ + STX$ + ACore$                         ' assemble command up to checksum
  If (CK = &H10) Then                                   ' if checksum = DLE$
    GTPkt$ = GTPkt$ + Chr$(CK) + DLE$ + DLE$ + ETX$     ' add checksum, DLE$ escape, DLE$ + ETX$
  Else
    GTPkt$ = GTPkt$ + Chr$(CK) + DLE$ + ETX$            ' else just add checksum, DLE$ + ETX$
  End If
  MSComm1.Output = GTPkt$                               ' send command to read ADC
  Text6 = dumptx$                                       ' display msg sent in hex
  dumptx$ = ""                                          ' clear buffer
  Text5 = ""                                            ' clear receice msg box to indicate
waiting for responce
End Sub

Public Sub GetRX()
  On Error Resume Next                                  ' set up error handler
  ComData$ = MSComm1.Input                              ' load ComData$ from COM input buffer
  Call RXRsp                                            ' extract RX response message(s)
End Sub
```

```
Public Sub RXRsp()
  FIFO$ = FIFO$ & ComData$                                  ' add ComData$ to end of FIFO$ string
  ComData$ = ""                                             ' clear ComData$
  RPkt$ = ""                                                ' clear RPkt$
  J = 1                                                     ' point J to first character position
  R = 0                                                     ' clear DLE$ flag
  U = 0                                                     ' clear build message flag
  Q = Len(FIFO$)                                            ' load Q with number of characters in
FIFO$
  If (Q > 256) Then                                         ' if FIFO$ is overflowing (garbage)
    FIFO$ = ""                                              ' clear FIFO$
    Exit Sub                                                ' and exit
  End If
  If (Q < 2) Then                                           ' if FIFO$ less than 2 characters
    Exit Sub                                                ' exit
  End If
  For Y = Q To 1 Step -1                                    ' else loop to scan all FIFO$ characters
    C$ = Mid$(FIFO$, J, 1)                                  ' load C$ with next character
    If (R = 0) And (C$ = DLE$) Then                         ' if character is a DLE$
      R = 1                                                 ' set DLE$ flag
    ElseIf (R = 1) And (C$ = STX$) Then                     ' else if DLE$ & STX$ sequence
      R = 0                                                 ' clear DLE$ flag
      U = 1                                                 ' set build message flag
      RPkt$ = ""                                            ' clear RPkt$
    ElseIf (R = 0) And (U = 1) Then                         ' else if build flag set and DLE$ flag
clear
      RPkt$ = RPkt$ & C$                                    ' add character to message
    ElseIf (R = 1) And (U = 1) And (C$ = DLE$) Then         ' else if DLE$ & DLE$ sequence
      R = 0                                                 ' reset DLE$ flag
      RPkt$ = RPkt$ & C$                                    ' add only one DLE$ to RPkt$
    ElseIf (R = 1) And (U = 1) And (C$ = ETX$) Then         ' else if DLE$ & ETX$ sequence
      R = 0                                                 ' clear DLE$ flag
      U = 0                                                 ' clear build message flag
      K = J                                                 ' load K with current message length
      If RPkt$ <> "" Then                                   ' if RPkt$ is not null
        Call ShowEvent                                        ' filter for and display temperature
        RPkt = ""                                           ' clear RPkt$
      End If
    ElseIf (R = 1) Then                                     ' else if DLE$ sequence not used here
      R = 0                                                 ' clear DLE$ flag and move on
    End If
    J = J + 1                                               ' increment character position
  Next Y                                                    ' do FOR loop Q times
  If (K < Q) Then                                           ' if K is less Q after looping
    FIFO$ = Right$(FIFO$, (Q - K))                          ' load FIFO$ with remaining characters
  Else                                                      ' else
    FIFO$ = ""                                              ' clear FIFO$ string
  End If
End Sub

Public Sub ShowEvent()
' filter for valid Node Event response
  N = Len(RPkt$)                                            ' get response length
  M = 0                                                     ' clear checksum accumulator
  P = 1                                                     ' nibble counter
  For Y = 1 To (N - 1)                                      ' for all bytes received checksum
    M = M + Asc(Mid$(RPkt$, Y, 1))                          ' add bytes to checksum calculation
    dumprx$ = dumprx$ + Hex(Asc(Mid$(RPkt$, Y, 1)))         ' build buffer with hex received data
    P = P + 1
    If P = 2 Then                                           ' byte yet
      dumprx$ = dumprx$ + " "                               ' add spaces between bytes
      P = 1
    End If
   Next Y

  CK = CByte(M Mod 256)                                     ' checksum calculation is lower byte
  If (CK <> Asc(Right$(RPkt$, 1))) Then                     ' if bad checksum
    Exit Sub                                                ' exit
  End If

  Text5 = dumprx$                                           ' display received message
  dumprx$ = ""                                              ' clear buffer for next msg

'Node Event Reporting (byte 6 is type of msg being reported)
  If (Asc(Mid$(RPkt$, 6, 1)) = &H1A) Then                   'test for msg
     Call NodeEvent
  End If

'Node Power On Reporting
  If (Asc(Mid$(RPkt$, 6, 1)) = &H8A) Then                  'test for power on msg
```

```
     Call NodeonEvent
  End If

'Node Poll Reporting
  If (Asc(Mid$(RPkt$, 6, 1)) = &H9) Then                    'test for result of poll msg
     Call NodepollEvent
  End If

'Node Poll Reporting
  If (Asc(Mid$(RPkt$, 6, 1)) = &H1) Then                    'test for responce to write msg
     Call Noderesponce
  End If

End Sub

Public Sub NodeEvent()
' filter for valid Node Event response

   NN$ = Str(Asc(Mid$(RPkt$, 5, 1)))                        ' get node number
   If (NN$ = 1) Then
    Text1 = "Event Report from Node " & NN$ & "  @ " & Now 'if node 1 display it
        Shape1.FillColor = &HC000&
        Node1 = 1
        Node1timer = 0
   End If
   If (NN$ = 2) Then
    Text2 = "Event Report from Node " & NN$ & "  @ " & Now 'if node 2 display it
        Shape2.FillColor = &HC000&
        Node2 = 1
        Node2timer = 0
   End If
   If (NN$ = 3) Then
    Text3 = "Event Report from Node " & NN$ & "  @ " & Now 'if node 3 display it
        Shape3.FillColor = &HC000&
        Node3 = 1
        Node3timer = 0
   End If
   If (NN$ = 4) Then
    Text4 = "Event Report from Node " & NN$ & "  @ " & Now 'if node 4 display it
        Shape4.FillColor = &HC000&
        Node4 = 1
        Node4timer = 0
   End If

End Sub
Public Sub NodeonEvent()
' filter for valid Node power on Event response

   NN$ = Str(Asc(Mid$(RPkt$, 5, 1)))                        ' get node number
   If (NN$ = 1) Then
        Text1 = "Power ON from Node " & NN$ & "  @ " & Now 'if node 1 display it
        Shape1.FillColor = &HC000&
        Node1 = 1
        Node1timer = 0
   End If
   If (NN$ = 2) Then
        Text2 = "Power ON from Node " & NN$ & "  @ " & Now 'if node 2 display it
        Shape2.FillColor = &HC000&
        Node2 = 1
        Node2timer = 0
   End If
   If (NN$ = 3) Then
        Text3 = "Power ON from Node " & NN$ & "  @ " & Now 'if node 3 display it
        Shape3.FillColor = &HC000&
        Node3 = 1
        Node3timer = 0
   End If
   If (NN$ = 4) Then
         Text4 = "Power ON from Node " & NN$ & "  @ " & Now 'if node 4 display it
        Shape4.FillColor = &HC000&
        Node4 = 1
        Node4timer = 0
   End If

End Sub
Public Sub NodepollEvent()
' filter for valid Node POll response

   NN$ = Str(Asc(Mid$(RPkt$, 5, 1)))                        ' get node number
   If (NN$ = 1) Then
        If (Asc(Mid$(RPkt$, 7, 1)) = &H25) Then
```

```
          Text1 = "Poll OK from Node " & NN$ & "  @ " & Now & "  Pin = High" 'if node 1 display it
        End If
        If (Asc(Mid$(RPkt$, 7, 1)) = &H24) Then
          Text1 = "Poll OK from Node " & NN$ & "  @ " & Now & "  Pin = Low " 'if node 1 display it
        End If
        Shape1.FillColor = &HC000&
        Node1 = 1
        Node1timer = 0
   End If
   If (NN$ = 2) Then
        If (Asc(Mid$(RPkt$, 7, 1)) = &H25) Then
          Text2 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = High "  'if node 2 display it
        End If
        If (Asc(Mid$(RPkt$, 7, 1)) = &H24) Then
          Text2 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = Low  "  'if node 2 display it
        End If
        Shape2.FillColor = &HC000&
        Node2 = 1
        Node2timer = 0
   End If
   If (NN$ = 3) Then
        If (Asc(Mid$(RPkt$, 7, 1)) = &H25) Then
          Text3 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = High "  'if node 2 display it
        End If
        If (Asc(Mid$(RPkt$, 7, 1)) = &H24) Then
          Text3 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = Low  "  'if node 2 display it
        End If
        Shape3.FillColor = &HC000&
        Node3 = 1
        Node3timer = 0
   End If
   If (NN$ = 4) Then
        If (Asc(Mid$(RPkt$, 7, 1)) = &H25) Then
          Text4 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = High "  'if node 2 display it
        End If
        If (Asc(Mid$(RPkt$, 7, 1)) = &H24) Then
          Text4 = "Poll OK from Node " & NN$ & "  @ " & Now & " Pin = Low  "  'if node 2 display it
        End If
        Shape4.FillColor = &HC000&
        Node4 = 1
        Node4timer = 0
   End If

End Sub
Public Sub Noderesponce()
' filter for valid Node response
   NN$ = Str(Asc(Mid$(RPkt$, 5, 1)))                    ' get node number and set color for status
   If (NN$ = 1) Then
        Shape1.FillColor = &HC000&
        Node1 = 1
        Node1timer = 0
   End If
   If (NN$ = 2) Then
        Shape2.FillColor = &HC000&
        Node2 = 1
        Node2timer = 0
   End If
   If (NN$ = 3) Then
        Shape3.FillColor = &HC000&
        Node3 = 1
        Node3timer = 0
   End If
   If (NN$ = 4) Then
        Shape4.FillColor = &HC000&
        Node4 = 1
        Node4timer = 0
   End If


 End Sub




Private Sub mnuCOM1_Click()
  P = 1                                                 ' COM port 1
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM1.Checked = True                                ' set this checkmark
```

```
End Sub

Private Sub mnuCOM2_Click()
  P = 2                                                 ' COM port 2
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM2.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM3_Click()
  P = 3                                                 ' COM port 3
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM3.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM4_Click()
  P = 4                                                 ' COM port 4
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM4.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM5_Click()
  P = 5                                                 ' COM port 5
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM5.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM6_Click()
  P = 6                                                 ' COM port 6
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM6.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM7_Click()
  P = 7                                                 ' COM port 7
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM7.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuCOM8_Click()
  P = 8                                                 ' COM port 8
  Call SetCOM                                           ' open COM port
  Call ResetCOM                                         ' clear all checkmarks
  mnuCOM8.Checked = True                                ' set this checkmark
End Sub

Public Sub SetCOM()
  On Error GoTo Skip                                     ' invalid COM port recovery
  If (MSComm1.PortOpen = True) Then                      ' if COM port open
    MSComm1.PortOpen = False                             ' close it
    Timer1.Enabled = False                               ' stop timer
  End If
  MSComm1.CommPort = P                                   ' initialize COM port 5
  MSComm1.Settings = "9600,N,8,1"                        ' at 9600 b/s
  MSComm1.RThreshold = 0                                 ' poll only, no interrupts
  MSComm1.InputLen = 0                                   ' read all bytes
  MSComm1.PortOpen = True                                ' open COM port
  Label1.Caption = " DM1800 Node Poll Reporting "        ' update display
  Timer1.Enabled = True                                  ' start timer
  Exit Sub
Skip:
  Label1.Caption = " Invalid COM  "                      ' update display
End Sub

Public Sub ResetCOM()
    mnuCOM1.Checked = False                              ' clear checkmark
    mnuCOM2.Checked = False                              ' clear checkmark
    mnuCOM3.Checked = False                              ' clear checkmark
    mnuCOM4.Checked = False                              ' clear checkmark
    mnuCOM5.Checked = False                              ' clear checkmark
    mnuCOM6.Checked = False                              ' clear checkmark
    mnuCOM7.Checked = False                              ' clear checkmark
    mnuCOM8.Checked = False                              ' clear checkmark
End Sub

Private Sub mnuNode1_Click()
```

```
  ACore$ = Core                                          ' default address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode1.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode2_Click()
  AID = 2                                                ' node 2
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode2.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode3_Click()
  AID = 3                                                ' node 3
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode3.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode4_Click()
  AID = 4                                                ' node 4
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode4.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode5_Click()
  AID = 5                                                ' node 5
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode5.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode6_Click()
  AID = 6                                                ' node 6
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode6.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode7_Click()
  AID = 7                                                ' node 7
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode7.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode8_Click()
  AID = 8                                                ' node 8
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode8.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode9_Click()
  AID = 9                                                ' node 9
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode9.Checked = True                                ' set this checkmark
End Sub

Private Sub mnuNode10_Click()
  AID = 10                                               ' node 10
  ACore$ = Left$(Core$, 4) + Chr$(AID) + Right$(Core$, 1)   ' insert new address
  Call ResetNode                                         ' clear all checkmarks
  mnuNode10.Checked = True                               ' set this checkmark
End Sub

Public Sub ResetNode()
    mnuNode1.Checked = False                             ' clear checkmark
    mnuNode2.Checked = False                             ' clear checkmark
    mnuNode3.Checked = False                             ' clear checkmark
    mnuNode4.Checked = False                             ' clear checkmark
    mnuNode5.Checked = False                             ' clear checkmark
    mnuNode6.Checked = False                             ' clear checkmark
    mnuNode7.Checked = False                             ' clear checkmark
    mnuNode8.Checked = False                             ' clear checkmark
    mnuNode9.Checked = False                             ' clear checkmark
    mnuNode10.Checked = False                            ' clear checkmark
End Sub
```

```
Private Sub mnuC_Click()
    mnuC.Checked = True                                 ' check deg C
    mnuF.Checked = False                                ' uncheck deg F
    T = 1                                               ' show deg C
End Sub

Private Sub mnuF_Click()
    mnuC.Checked = False                                ' uncheck deg C
    mnuF.Checked = True                                 ' check deg F
    T = 0                                               ' show deg F
End Sub

Private Sub Form_Unload(Cancel As Integer)
  If (MSComm1.PortOpen = True) Then                     ' if COM port open
    MSComm1.PortOpen = False                            ' close it
  End If
  End                                                   ' done
End Sub
```

Bob Nelson
RF Monolithics
2/10/2006