frmConfig - 1

```
Option Explicit

Private Sub cmdOK_Click()
'    Dim i As Integer
'
'    If Right(Me.Caption, 1) = "1" Then
'        frmMain.txtDev1Dial = txtDial(0)
'        frmMain.txtDev1Settings = txtSettings(0)
'        frmMain.txtDev1Port = txtPort(0)
'        frmMain.txtDev1Rate = txtRate
''        For i = 0 To 24
''            frmMain.txtData0(i) = txtReg(i)
''        Next i
'    Else
'        frmMain.txtDev2Dial = txtDial(0)
'        frmMain.txtDev2Settings = txtSettings(0)
'        frmMain.txtDev2Port = txtPort(0)
'        frmMain.txtDev2Rate = txtRate
''        For i = 0 To 24
''            frmMain.txtData1(i) = txtReg(i)
''        Next i
'    End If
''    If frmMain.cmdConnect(0).Caption = "Disconnect" Or frmMain.cmdConnect(1).Caption = "Discon
nect" Then
''        frmMain.tmrRefresh.Enabled = True
''    End If
    Me.Hide
End Sub
```

frmDetails - 1

```
Option Explicit

Private Sub Command2_Click()
    Me.Hide
End Sub
```

frmDeviceStatus - 1

```
Option Explicit

Private Sub Command1_Click()
    Me.Hide
End Sub

Private Sub Command2_Click()
    frmDeviceStatus.EnableDisplay = "Enabled"
End Sub

Private Sub Command3_Click()
    frmDeviceStatus.EnableDisplay = "Disabled"
End Sub
```

```
Private Sub Command1_Click()
    Me.Hide
End Sub

Private Sub Form_Load()
    Text2.Text = "1.0.0001"
End Sub
```

```
Option Explicit
Private Sub cmdDetails1_Click()
    frmDeviceStatus.Show
End Sub

Private Sub GetStatus()
    Dim retAry() As Double, i As Integer
    '30340 = kWH accumlated
    '30445 = Runtime, days
    '30446 = Runtime, hours
    '30447 = Runtime, minutes
    'StartAdd = offset in Modbus register table
    If frmMain.txtStatus = "Connected" Then
        retAry = returnValues(2, frmDeviceStatus.FNC.Text, frmDeviceStatus.StartAdd.Text, 10)
        If frmDeviceStatus.EnableDisplay = "Enabled" Then
            For i = 0 To UBound(retAry)
                frmDeviceStatus.retAry(i) = retAry(i)
            Next i
            If UBound(retAry) >= 5 Then
                frmDeviceStatus.txtKWH = retAry(0)
                frmDeviceStatus.txtRuntimeDays = retAry(4)
                frmDeviceStatus.txtRuntimeHours = retAry(5)
            End If
        End If
    End If
End Sub
Private Sub cmdStartEngine_Click()
    Dim retAry() As Double
    retAry = returnValues(2, 5, 26, 65280)
End Sub
Private Sub cmdStopEngine_Click()
    Dim retAry() As Double
    retAry = returnValues(2, 5, 26, 0)
End Sub
Private Sub Command1_Click()
    frmSelectSite.Show
End Sub
Private Sub Command2_Click()
    frmMain.txtStatus = "Disconnecting"
    Timer1.Enabled = True ' this waits two seconds then calls Timer1_Timer()
End Sub
Private Sub Form_Unload(Cancel As Integer)
On Error Resume Next
    MSComm1.CommPort = 9
    MSComm1.PortOpen = True
End Sub
Private Sub imgCH_Click()
    Dim retAry() As Double

End Sub
Private Sub mnuHelpAbout_Click()
    frmHelpAbout.Show
End Sub
Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent
        Case comEvSend:
        Case comEvReceive: 'handleIncoming
        Case comEventBreak:
        Case comEventFrame:
        Case comEventOverrun:
        Case comEventRxOver:
        Case comEventRxParity:
        Case comEventTxFull:
        Case comEventDCB:
    End Select
End Sub
Private Sub UpdateMain()
'    Dim retAry() As Double
'    If txtStatus = "Connected" Then
```

```
'
'        'Get Runtime
'        retAry = returnValues(2, 4, 444, 2)
'        If retAry(0) <> -1 Then
'            frmMain.txtRuntimeDay = retAry(0)
'            frmMain.txtRuntimeHours = retAry(1)
'        End If
'
'        'Get Engine running status
'        retAry = returnValues(2, 2, 15, 1)
'        If retAry(0) <> -1 Then
'            If retAry(0) = 161 Then
'                txtEngineStatus.Caption = "Running"
'            Else
'                txtEngineStatus.Caption = "Stopped"
'            End If
'        Else
'            MsgBox ("Comm Error")
'        End If
'
'        'Get Generator Voltage
'        retAry = returnValues(2, 4, 317, 1)
'        If retAry(0) <> -1 Then
'            frmMain.txtGenVolt = retAry(0)
'        Else
'            MsgBox ("Comm Error")
'        End If
'
'        'Get Generator Frequency
'        retAry = returnValues(2, 4, 329, 2)
'        If retAry(0) <> -1 Then
'            frmMain.txtGenFreq = retAry(0) + (retAry(1) / 65536)
'        End If
'
'        'Get Utility Voltage
'        retAry = returnValues(2, 4, 119, 1)
'        If retAry(0) <> -1 Then
'            frmMain.txtUtilVolt = retAry(0)
'        End If
'
'        'Get Utility Frequency
'        retAry = returnValues(2, 4, 334, 2)
'        If retAry(0) <> -1 Then
'            frmMain.txtUtilFreq = retAry(0) + (retAry(1) / 65536)
'        End If
'
'      End If
'
End Sub
Private Sub tmrRefresh_Timer()
'    Command4_Click
    If frmDeviceStatus.Enabled = True Then
        GetStatus
    End If
    UpdateMain

End Sub
Private Sub txtStatus1_Change(Index As Integer)

End Sub
```

```
Private Sub Command2_Click()
    Me.Hide
End Sub

Private Sub Dir1_Change()
    Dir1.Path = Drive1.Drive   ' Set directory path.
End Sub

Private Sub File1_Click()
    File1.Path = Dir1.Path   ' Set file path.
End Sub
```

frmSelectSite - 1

```
Option Explicit

Private Sub Command1_Click()
    frmSiteStatus.Show
    frmSelectSite.Text1.BackColor = &H8000000F
    frmSelectSite.Text1.Locked = True
    frmSelectSite.Text2.BackColor = &H8000000F
    frmSelectSite.Text2.Locked = True
    frmSelectSite.Text3.BackColor = &H8000000F
    frmSelectSite.Text3.Locked = True

    frmSiteStatus.Text1.Text = frmSelectSite.Text1.Text
End Sub

Private Sub Command2_Click()
    frmSelectSite.Hide
    frmSelectSite.Text1.BackColor = &H8000000F
    frmSelectSite.Text1.Locked = True
    frmSelectSite.Text2.BackColor = &H8000000F
    frmSelectSite.Text2.Locked = True
    frmSelectSite.Text3.BackColor = &H8000000F
    frmSelectSite.Text3.Locked = True
End Sub

Private Sub Command3_Click()
    frmSelectSite.Text1.BackColor = &H8000000E
    frmSelectSite.Text1.Locked = False
    frmSelectSite.Text2.BackColor = &H8000000E
    frmSelectSite.Text2.Locked = False
    frmSelectSite.Text3.BackColor = &H8000000E
    frmSelectSite.Text3.Locked = False
End Sub

Private Sub Command4_Click()
    frmSaveSite.Show
End Sub
```

frmSiteStatus - 1

```
Private Sub Command1_Click()
    frmSiteStatus.Hide
End Sub
Private Sub Command10_Click()

    Dim iWait As Integer

    MSComm1.CommPort = frmConfig.Combo1.Text
    MSComm1.Settings = frmConfig.Text1.Text
    MSComm1.Tag = frmSelectSite.Text2.Text
    MSComm1.PortOpen = True
    frmSiteStatus.Label5.Caption = "Dialing"
    UpdateStatus ("Dialing")
    MSComm1.InBufferCount = 0
    MSComm1.OutBufferCount = 0
    MSComm1.InputMode = comInputModeText
    MSComm1.Output = "ATDT" & MSComm1.Tag & vbCr
    frmSiteStatus.Label5.Caption = "Waiting for connection..."
    UpdateStatus ("Waiting for connection")

    iWait = 450   'preset to 45.0 seconds
    Do While iWait >= 0
        frmSiteStatus.Label5.Caption = frmSiteStatus.Label5.Caption + GetData
        Pause 100
        iWait = iWait - 1
    Loop

End Sub
Private Sub Command11_Click()
    frmSiteStatus.Label5.Caption = "Disconnecting"
    UpdateStatus ("Disconnecting")
    Timer1.Enabled = True ' this waits two seconds then calls Timer1_Timer()
End Sub
Private Sub Timer1_Timer()
    Timer1.Enabled = False
    If MSComm1.PortOpen = True Then
        MSComm1.Output = "+++"
        UpdateStatus ("+++")
    End If

    Timer2.Enabled = True ' this waits two seconds then calls Timer2_Timer()
End Sub
Private Sub Timer2_Timer()
    Dim ret

    Timer2.Enabled = False
    If MSComm1.PortOpen = True Then
        MSComm1.Output = "ATH" & vbCr
        UpdateStatus ("ATH")
        MSComm1.PortOpen = False
        MSComm1.InBufferCount = 0
        MSComm1.OutBufferCount = 0
        ret = MSComm1.DTREnable     ' Save the current setting.
        MSComm1.DTREnable = True    ' Turn DTR on.
        MSComm1.DTREnable = False   ' Turn DTR off.
        MSComm1.DTREnable = ret     ' Restore the old setting.
    End If
    frmSiteStatus.Label5.Caption = "Offline"
    UpdateStatus ("Offline")
    Screen.MousePointer = vbDefault
End Sub
Private Sub Command2_Click()
    frmSiteStatus.Hide
End Sub
Private Sub Command9_Click()
    frmDetails.Show
End Sub
```

```
Option Explicit
Public Declare Function GetTickCount Lib "kernel32" () As Long
Public Function returnValues(ByVal Node As String, ByVal FNC As String, ByVal ST_ADD As String,
ByVal Num As String) As Double()

    Dim iBC As Integer, iOld As Integer, Data() As Byte, dblTemp1 As Double, dblTemp2 As Double,
 CRC() As Byte
    Dim strBuffer As String, ii As Integer, i As Integer, Results() As Double, j As Integer
    ReDim Results(5)        '"pre-dimension" in case of an error (the array will have already be
en dimensioned)

    mb_Rcv Node, FNC, ST_ADD, Num ' send data to read/write

    iOld = -1
    iBC = frmMain.MSComm1.InBufferCount
    Do While iOld <> iBC
        iOld = iBC
        Pause 100 ' milliseconds
        iBC = frmMain.MSComm1.InBufferCount
    Loop
        If frmMain.MSComm1.InBufferCount > 0 Then
            ii = frmMain.MSComm1.InBufferCount
            strBuffer = frmMain.MSComm1.Input
            If ii > 5 Then
                ' assuming ii (length of receipt) is 7, then
                ReDim Data(ii - 2)                              ' Data() is redimmed to 5
                ReDim CRC(ii)                                   ' CRC() is redimmed to 7
                For i = 0 To ii - 1                             ' i= 0 to 6
                    CRC(i) = Asc(Mid(strBuffer, i + 1, 1))      ' assign every value into CRC()
                    If i < ii - 2 Then                          ' if i < 5 use it (weed out the
CRC)
                        Data(i) = Asc(Mid(strBuffer, i + 1, 1)) ' assign all but the CRC to Data
()
                    End If
                Next i

                frmMain.CrcCtrl1.CRCType = Modbus           ' set which algorithm to use and cal
c it
                frmMain.CrcCtrl1.DoCRCByteArray Data, ii - 2, True, 65535
                dblTemp1 = frmMain.CrcCtrl1.GetCRCDecStr \ 256
                dblTemp2 = frmMain.CrcCtrl1.GetCRCDecStr - ((frmMain.CrcCtrl1.GetCRCDecStr) \ 25
6) * 256

                Select Case FNC
                    Case 1: ReDim Results((ii - 4) / 2)         ' Function Call 01 returns 2 HDR
 & 2 CRC
                            j = 2                               ' start of data is at CRC(2)
                    Case 2: ReDim Results((ii - 4) / 2)         ' Function Call 02 returns 2 HDR
 & 2 CRC
                            j = 2                               ' start of data is at CRC(2)
                    Case 3: ReDim Results((ii - 5) / 2)         ' Function Call 03 returns 3 HDR
 & 2 CRC
                            j = 3                               ' start of data is at CRC(3)
                    Case 4: ReDim Results((ii - 5) / 2)         ' Function Call 04 returns 3 HDR
 & 2 CRC
                            j = 3                               ' start of data is at CRC(3)
                    Case 5: ReDim Results((ii - 4) / 2)         ' Function Call 05 returns 2 HDR
 & 2 CRC
                            j = 2                               ' start of data is at CRC(2)
                    Case Else: ReDim Results(ii)                ' Function Call ?? returns ? HDR
 & 2 CRC
                            j = 2                               ' start of data is at CRC(2)
                End Select

                ' check the CRC returned
                If (dblTemp1 = CRC(ii - 2)) And (dblTemp2 = CRC(ii - 1)) Then
                    For i = 0 To UBound(Results) - 1            ' put the data into the return a
rray
                        dblTemp1 = CRC(j)                       ' this is the high order bit
```

```
                        dblTemp2 = dblTemp1 * 256               ' multiply the high order bit by
 256
                        Results(i) = dblTemp2 + CRC(j + 1)      ' this is the total number in de
cimal
                        j = j + 2                               ' skip to the next set of return
ed data
                    Next i
                    frmMain.txtRetAry0 = Results(0)
                Else

                    For i = 0 To UBound(Results) - 1            ' populate the return array with
 -1s
                        Results(i) = -1                         ' if there is a problem withthe
CRC
                    Next i
                Results(0) = -1                                 ' return the values
                End If
            Else
                Results(0) = -1
            End If
        Else
            ReDim Results(1)
            Results(0) = -1
        End If

    returnValues = Results()
End Function
Sub Pause(lMilliseconds As Long)
    Dim start
    start = GetTickCount
    While start + lMilliseconds > GetTickCount
        DoEvents
    Wend
End Sub
Public Function mb_Rcv(ByVal Node As String, ByVal FNC As String, ByVal ST_ADD As String, ByVal
Num As String)
    Dim Data(8) As Byte, lTemp As Long, intWaits As Integer

    frmMain.MSComm1.InBufferCount = 0 ' clears out any contents of the Input buffer

    Data(0) = Node
    Data(1) = FNC
    Data(2) = ST_ADD \ 256
    Data(3) = ST_ADD - ((ST_ADD \ 256) * 256)
    Data(4) = Num \ 256
    Data(5) = Num - ((Num \ 256) * 256)

    ' set which algorithm to use
    frmMain.CrcCtrl1.CRCType = Modbus

    ' do the calculation
    frmMain.CrcCtrl1.DoCRCByteArray Data, 6, True, 65535

    ' display the result in a label
    lTemp = (frmMain.CrcCtrl1.GetCRCDecStr) \ 256
    Data(6) = lTemp
    lTemp = (frmMain.CrcCtrl1.GetCRCDecStr) - ((frmMain.CrcCtrl1.GetCRCDecStr \ 256) * 256)
    Data(7) = lTemp

    frmMain.MSComm1.InBufferCount = 0 ' clears out any contents of the Input buffer
    frmMain.MSComm1.OutBufferCount = 0 ' clears out any contents of the output buffer
    frmMain.MSComm1.PortOpen = False
    frmMain.MSComm1.OutBufferSize = 8 ' sets the number of bytes to send
    frmMain.MSComm1.PortOpen = True
    frmMain.MSComm1.OutBufferCount = 0 ' clears out any contents of the output buffer

    frmMain.MSComm1.Output = Data
End Function
Sub Main()
```

Math - 3

```
    Load frmConfig
    frmMain.Show
End Sub
Function SendCommand(ByVal sCommand As String, ByVal lTimeout As Long) As String

Dim vResp As Variant
Dim vNumOfBytesRead As Variant
   Dim abyData() As Byte
   Dim i As Integer

   sCommand = sCommand & vbCrLf
   ReDim abyData(0 To Len(sCommand) - 1)

   For i = 0 To Len(sCommand) - 1
     abyData(i) = Asc(Mid$(sCommand, i + 1, 1))
   Next i

'   m_ModbusServer.TxRxRawData CVar(abyData), Len(sCommand), vResp, 0, lTimeout, vNumOfBytesRead


  For i = 0 To vNumOfBytesRead - 1
     SendCommand = SendCommand & Chr$(vResp(i))
   Next i

End Function
Sub UpdateStatus(ByVal Data)
    Dim DT As String
    DT = Now

    frmMain.Text1.Text = frmMain.Text1.Text + vbCrLf + DT + " " + Data

End Sub
```