

PROJECT SCADA 2007

กล่าวนำ

ปัจจุบันนี้นับได้ว่า เข้าสู่ยุคของการแข่งขันทางธุรกิจอย่างแท้จริง สถานประกอบการน้อยใหญ่จะต้องปรับประสิทธิภาพระบบการจัดการในองค์กรให้ได้ประสิทธิภาพสูงสุด การนำเทคโนโลยีข้อมูลข่าวสารทางอิเล็กทรอนิกส์มาประยุกต์ใช้ในการบริหารจัดการ น่าจะเป็นทางออกที่ดีที่สุด

วัตถุประสงค์

เพื่อให้เข้าใจการออกแบบระบบควบคุม และ การเชื่อมโยงฐานข้อมูลที่มีผ่านระบบสื่อสารแบบดิจิทัล ที่สามารถเข้าถึงข้อมูลได้จากทุกมุมโลก

ขั้นตอนการทำงาน

พัฒนาโปรแกรมระบบควบคุมอัตโนมัติ พร้อม สร้างฐานข้อมูลที่ต้องการตรวจสอบและเปลี่ยนแปลงแก้ไขจากระบบสื่อสารได้

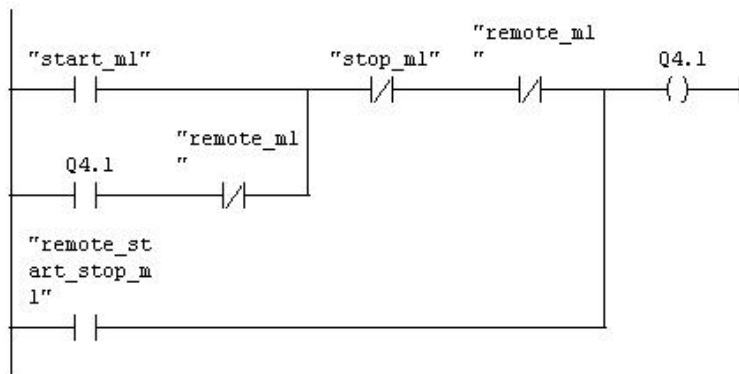
พัฒนาโปรแกรมระบบสื่อสารเพื่อเข้าไปแก้ไขและตรวจสอบข้อมูลที่สามารถเข้าถึงได้ทุกที่ ทุกเวลา จากทุกมุมโลก

ตัวอย่างระบบควบคุม

-พัฒนาโปรแกรมระบบควบคุมอัตโนมัติ พร้อม สร้างฐานข้อมูล โดยใช้ PLC

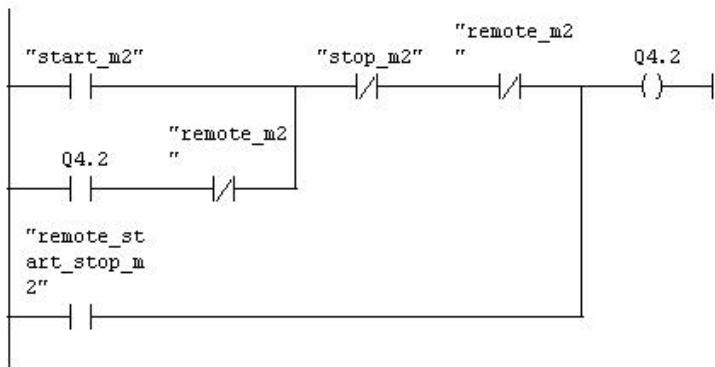
Network 1 : Title:

Comment:



Network 2 : Title:

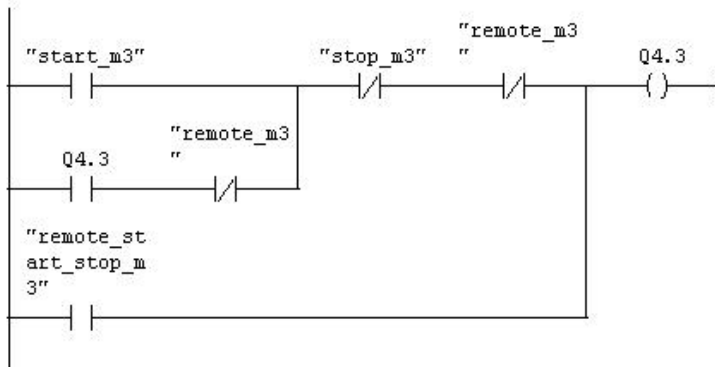
Comment:



Handwritten squiggly marks

Network 3 : Title:

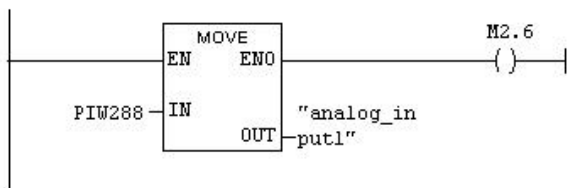
Comment:



Handwritten squiggly marks

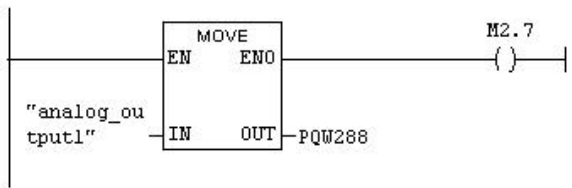
Network 4 : Title:

Comment:



Network 5 : Title:

Comment:



Handwritten squiggly marks

Network 6 : Title:

Comment:



Network 7 : Title:

Comment:



Network 8 : Title:

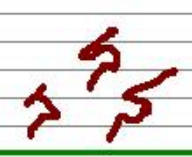
Comment:



รูป project scada 2007_1 โปรแกรมควบคุมใน PLC

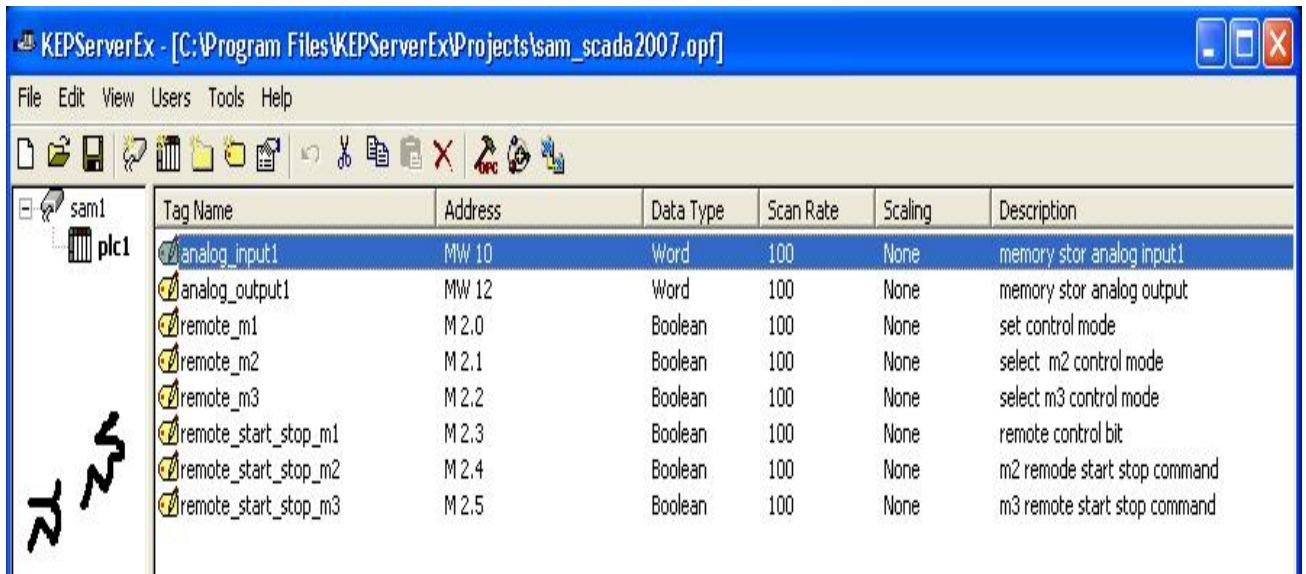
จากโปรแกรมในรูป project scada 2007_1 จะประกอบด้วยฐานข้อมูลที่สามารถเข้าถึงและควบคุมจากภายนอก (REMOTE CONTROL) ดังนี้คือ

| | Status | Symbol | Address | Data type | Comment |
|----|--------|----------------------|---------|-----------|--------------------------|
| 1 | | start_m1 | I 0.0 | BOOL | |
| 2 | | stop_m1 | I 0.1 | BOOL | |
| 3 | | stop_m2 | I 0.2 | BOOL | |
| 4 | | start_m2 | I 0.3 | BOOL | |
| 5 | | start_m3 | I 0.4 | BOOL | |
| 6 | | stop_m3 | I 0.5 | BOOL | |
| 7 | | remote_m1 | M 2.0 | BOOL | for remote control m1 |
| 8 | | remote_m2 | M 2.1 | BOOL | for remote control m2 |
| 9 | | remote_m3 | M 2.2 | BOOL | for remote control m3 |
| 10 | | remote_start_stop_m1 | M 2.3 | BOOL | remote start and stop m1 |
| 11 | | remote_start_stop_m2 | M 2.4 | BOOL | remote start and stop m2 |
| 12 | | remote_start_stop_m3 | M 2.5 | BOOL | remote start and stop m3 |
| 13 | | status_m1 | M 3.0 | BOOL | |
| 14 | | status_m2 | M 3.1 | BOOL | |
| 15 | | status_m3 | M 3.2 | BOOL | |
| 16 | | analog_input1 | MV 10 | WORD | |
| 17 | | analog_output1 | MV 12 | WORD | |
| 18 | | | | | |



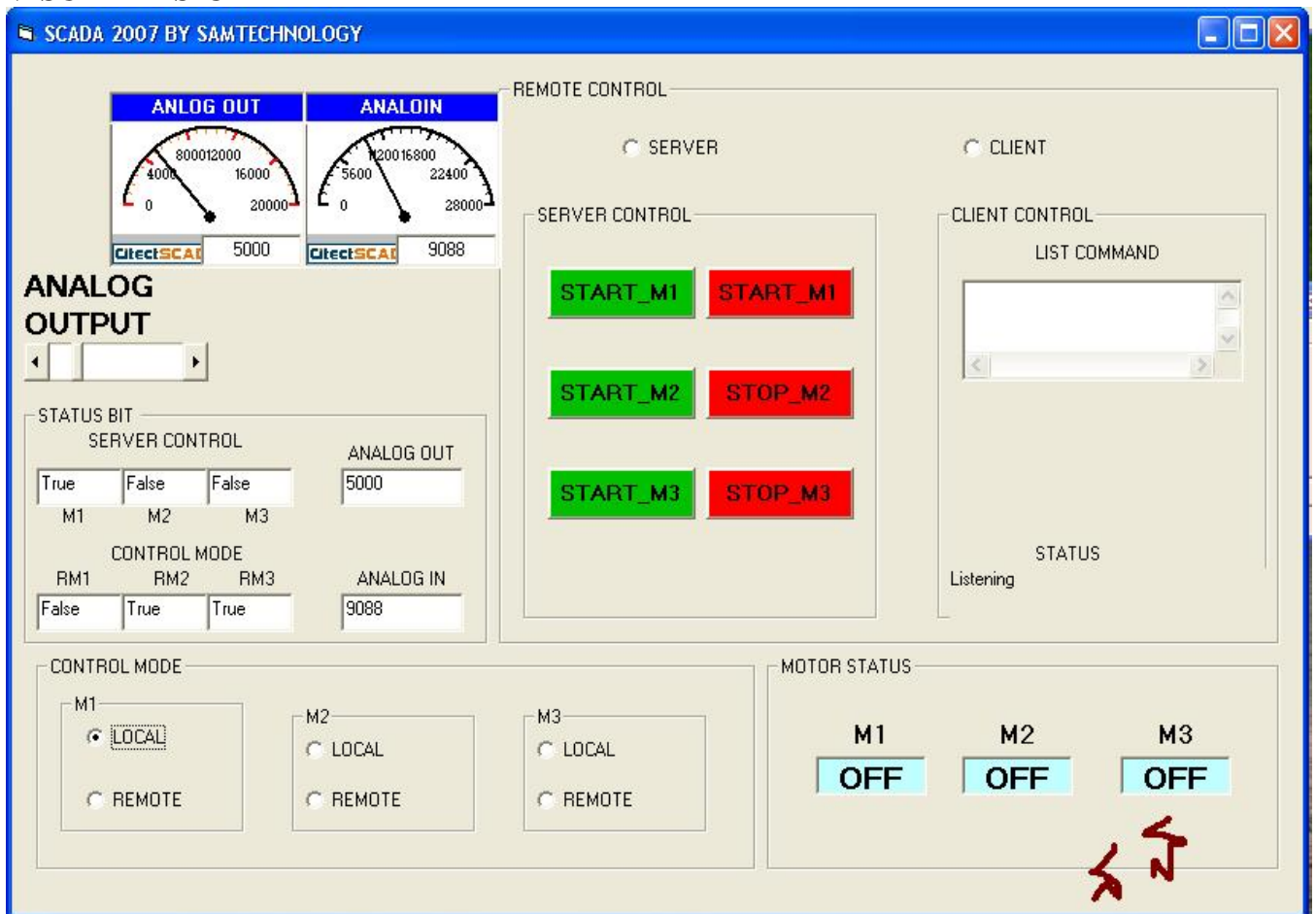
รูป project scada 2007_2 ในกรอบสีเขียวคือพื้นที่หน่วยความจำที่สามารถเข้าถึงจากโลกภายนอก

ใช้ SOFTWARE KEPSERWARE เป็นตัวกลางในการเชื่อมต่อฐานข้อมูลระหว่าง PLC กับ PROGRAM VISUAL BASIC ในการพัฒนาระบบควบคุม

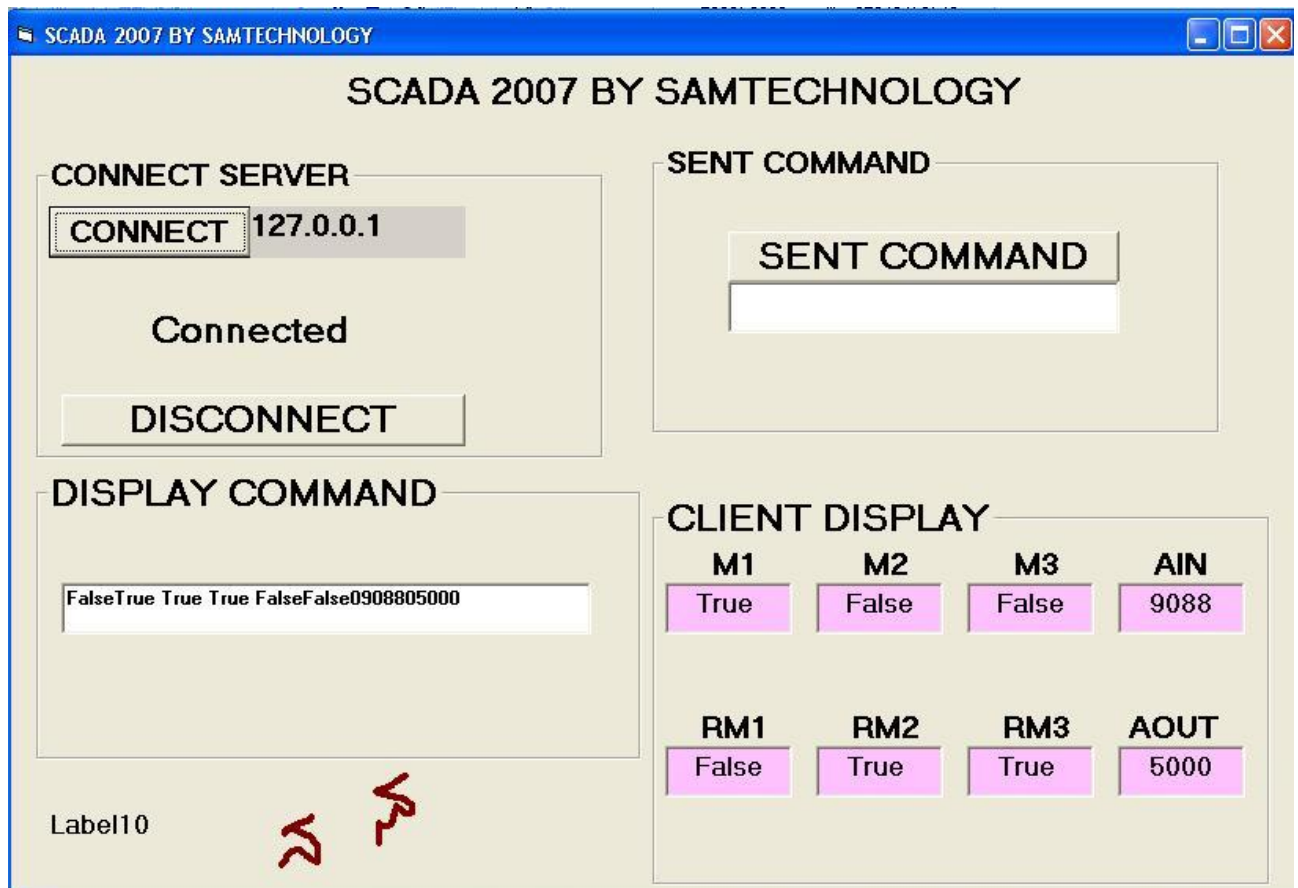


รูป project scada 2007_3 กำหนด tag ในส่วนที่ต้องการใช้เป็นฐานข้อมูล

พัฒนาโปรแกรมระบบสื่อสารเพื่อเข้าไปแก้ไขและตรวจสอบข้อมูลที่สามารถเข้าถึงได้ทุกที่ ทุกเวลา จากทุกมุมโลก โดยใช้ VISUAL BASIC



รูป project scada 2007_3 GUI SERVER



รูป project scada 2007_3 GUI CLIENT

PROGRAM ในส่วนของ GUI SERVER

Option Base 1

Dim CONNECT1 As Boolean

Dim CONNECT2 As Boolean

Dim CONNECT3 As Boolean

Dim WithEvents AnOPCServer As OPCServer

Dim WithEvents ConnectedOPCServer As OPCServer

Dim ConnectedServerGroup As OPCGroups

Dim WithEvents ConnectedGroup As OPCGroup

Dim OPCItemCollection As OPCItems

Dim OneOPCItem As OPCItem

Dim ItemCount As Long

Dim OPCItemIDs(10) As String

Dim ItemServerHandles() As Long

Dim ItemServerErrors() As Long

Dim ClientHandles(10) As Long

Dim ANOUT_VALUE

Dim ANALOGOUT\$

Sub CLIENT_START_M1()

ItemCount = 1

Dim SyncItemValues(1) As Variant

Dim SyncItemServerHandles(1) As Long

Dim SyncItemServerErrors() As Long

```
    SyncItemServerHandles(1) = ItemServerHandles(4)
    SyncItemValues(1) = 1
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
    SyncItemServerErrors
```

End Sub

```
Sub CLIENT_START_M2()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(5)
    SyncItemValues(1) = 1
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
    SyncItemServerErrors
```

End Sub

```
Sub CLIENT_START_M3()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(6)
    SyncItemValues(1) = 1
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
    SyncItemServerErrors
```

End Sub

```
Sub CLIENT_STOP_M1()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(4)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
    SyncItemServerErrors
```

End Sub

```
Sub CLIENT_STOP_M2()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
```

```
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(5)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Sub CLIENT_STOP_M3()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(6)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Sub M1_LOCAL_CONTROL()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(1)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Sub M1_REMOTE_CONTROL()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(1)
    SyncItemValues(1) = 1
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Sub M2_LOCAL_CONTROL()
```

```
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
```

```
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(2)
SyncItemValues(1) = 0
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
End Sub
```

```
Sub M3_LOCAL_CONTROL()
```

```
ItemCount = 1
Dim SyncItemValues(1) As Variant
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(3)
SyncItemValues(1) = 0
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

```
End Sub
```

```
Sub m2_remote_control()
```

```
ItemCount = 1
Dim SyncItemValues(1) As Variant
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(2)
SyncItemValues(1) = 1
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

```
End Sub
```

```
Sub M3_remote_CONTROL()
```

```
ItemCount = 1
Dim SyncItemValues(1) As Variant
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(3)
SyncItemValues(1) = 1
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
ItemCount = 1
Dim SyncItemValues(1) As Variant
```



```
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(1)
SyncItemValues(1) = 1
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Private Sub Command2_Click()
ItemCount = 1
Dim SyncItemValues(1) As Variant
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(1)
SyncItemValues(1) = 0
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Sub WRITE_ANOUT()
ItemCount = 1
Dim SyncItemValues(1) As Variant
Dim SyncItemServerHandles(1) As Long
Dim SyncItemServerErrors() As Long
SyncItemServerHandles(1) = ItemServerHandles(8)
SyncItemValues(1) = ANOUT_VALUE
ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors
```

End Sub

```
Private Sub ConnectedGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems
As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long,
TimeStamps() As Date)
Dim i
On Error Resume Next
For i = 1 To NumItems
Text4(ClientHandles(i)).Text = ItemValues(i)
Next i
End Sub
```

```
Private Sub Form_Activate()
Winsock.LocalPort = 456
Winsock.Listen
HScroll1.Value = Val(Text4(7).Text)
End Sub
```

```

Private Sub Form_Load()
scada2007.Caption = "SCADA 2007 BY SAMTECHNOLOGY"
'=====conect server=====
Set ConnectedOPCServer = New OPCServer
ConnectedOPCServer.Connect "KEPware.KEPServerEx.V4" ', ""

'=====conected grop=====
Set ConnectedServerGroup = ConnectedOPCServer.OPCGroups
ConnectedServerGroup.DefaultGroupIsActive = True
ConnectedServerGroup.DefaultGroupDeadband = 0
Set ConnectedGroup = ConnectedServerGroup.Add("g1")
ConnectedGroup.UpdateRate = 100
ConnectedGroup.IsSubscribed = True
'=====add item=====
ItemCount = 8
OPCItemIDs(1) = "sam1.plc1.remote_m1"
OPCItemIDs(2) = "sam1.plc1.remote_m2"
OPCItemIDs(3) = "sam1.plc1.remote_m3"
OPCItemIDs(4) = "sam1.plc1.remote_start_stop_m1"
OPCItemIDs(5) = "sam1.plc1.remote_start_stop_m2"
OPCItemIDs(6) = "sam1.plc1.remote_start_stop_m3"
OPCItemIDs(7) = "sam1.plc1.analog_input1"
OPCItemIDs(8) = "sam1.plc1.analog_output1"
'-----
ClientHandles(1) = 0
ClientHandles(2) = 1
ClientHandles(3) = 2
ClientHandles(4) = 3
ClientHandles(5) = 4
ClientHandles(6) = 5
ClientHandles(7) = 6
ClientHandles(8) = 7
'-----
Set OPCItemCollection = ConnectedGroup.OPCItems
OPCItemCollection.DefaultIsActive = True
OPCItemCollection.AddItem ItemCount, OPCItemIDs, ClientHandles, ItemServerHandles,
ItemServerErrors

'=====CHECK M1_LOCAL=====
If M1_LOCAL.Value = True Then

M1_LOCAL_CONTROL

End If
'=====CHECK M1_REMOTE=====
If M1_REMOTE.Value = True Then

M1_REMOTE_CONTROL

```

End If

'=====

End Sub

Private Sub STOP_M1_Click()

End Sub

Private Sub START_M3_Click()

End Sub

Private Sub HScroll1_Change()

ItemCount = 1

Dim SyncItemValues(1) As Variant

Dim SyncItemServerHandles(1) As Long

Dim SyncItemServerErrors() As Long

SyncItemServerHandles(1) = ItemServerHandles(8)

SyncItemValues(1) = Val(HScroll1.Value)

**ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors**

End Sub

Private Sub M1_LOCAL_Click()

If M1_LOCAL.Value = True Then

M1_LOCAL_CONTROL

End If

End Sub

Private Sub M1_REMOTE_Click()

M1_REMOTE_CONTROL

End Sub

Private Sub M2_LOCAL_Click()

If M2_LOCAL.Value = True Then

M2_LOCAL_CONTROL

End If

End Sub

Private Sub M2_REMOTE_Click()

If M2_REMOTE.Value = True Then

m2_remote_control

End If

End Sub

Private Sub M3_LOCAL_Click()

If M3_LOCAL.Value = True Then

M3_LOCAL_CONTROL

End If

End Sub

Private Sub M3_REMOTE_Click()

If M3_REMOTE.Value = True Then

M3_remote_CONTROL

End If

End Sub

Private Sub Option4_Click()

End Sub

Private Sub Option3_Click()

End Sub

Private Sub SERVER_REMOTE_Click()

If SERVER_REMOTE.Value = True Then

Winsock.Close

Winsock.Listen

End If

End Sub

```
Private Sub SERVER_START_M1_Click()  
    ItemCount = 1  
    Dim SyncItemValues(1) As Variant  
    Dim SyncItemServerHandles(1) As Long  
    Dim SyncItemServerErrors() As Long  
    SyncItemServerHandles(1) = ItemServerHandles(4)  
    SyncItemValues(1) = 1  
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,  
    SyncItemServerErrors
```

End Sub

```
Private Sub SERVER_START_M2_Click()  
    ItemCount = 1  
    Dim SyncItemValues(1) As Variant  
    Dim SyncItemServerHandles(1) As Long  
    Dim SyncItemServerErrors() As Long  
    SyncItemServerHandles(1) = ItemServerHandles(5)  
    SyncItemValues(1) = 1  
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,  
    SyncItemServerErrors
```

End Sub

```
Private Sub SERVER_START_M3_Click()  
    ItemCount = 1  
    Dim SyncItemValues(1) As Variant  
    Dim SyncItemServerHandles(1) As Long  
    Dim SyncItemServerErrors() As Long  
    SyncItemServerHandles(1) = ItemServerHandles(6)  
    SyncItemValues(1) = 1  
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,  
    SyncItemServerErrors
```

End Sub

```
Private Sub SERVER_STOP_M1_Click()  
    ItemCount = 1  
    Dim SyncItemValues(1) As Variant  
    Dim SyncItemServerHandles(1) As Long  
    Dim SyncItemServerErrors() As Long  
    SyncItemServerHandles(1) = ItemServerHandles(4)  
    SyncItemValues(1) = 0  
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,  
    SyncItemServerErrors
```

End Sub

```

Private Sub SERVER_STOP_M2_Click()
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(5)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors

End Sub

```

```

Private Sub SERVER_STOP_M3_Click()
    ItemCount = 1
    Dim SyncItemValues(1) As Variant
    Dim SyncItemServerHandles(1) As Long
    Dim SyncItemServerErrors() As Long
    SyncItemServerHandles(1) = ItemServerHandles(6)
    SyncItemValues(1) = 0
    ConnectedGroup.SyncWrite ItemCount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors

End Sub

```

```

Private Sub Text4_Change(Index As Integer)
    TechMeter2.Value = Val(Text4(7).Text)
    TechMeter1.Value = Val(Text4(6).Text)
    HScroll1.Value = Val(Text4(7).Text)
    If Winsock.State = sckConnected Then ' make sure we're connected before sending

    TechMeter2.Value = Val(Text4(7).Text)
    TechMeter1.Value = Val(Text4(6).Text)
    '=====add 0000000000
    If Len(Text4(6).Text) <> 5 Then

    Do While Len(Text4(6).Text) <> 5
        Text4(6).Text = "0" & Text4(6).Text

    Loop

    End If
    '=====
    If Len(Text4(7).Text) <> 5 Then

    Do While Len(Text4(7).Text) <> 5
        Text4(7).Text = "0" & Text4(7).Text

```

Loop

End If

```
'=====
If Text4(0).Text = "True" Then Text4(0).Text = Text4(0).Text & " "
  If Text4(1).Text = "True" Then Text4(1).Text = Text4(1).Text & " "
    If Text4(2).Text = "True" Then Text4(2).Text = Text4(2).Text & " "
      If Text4(3).Text = "True" Then Text4(3).Text = Text4(3).Text & " "
        If Text4(4).Text = "True" Then Text4(4).Text = Text4(4).Text & " "
          If Text4(5).Text = "True" Then Text4(5).Text = Text4(5).Text & " "
```

```
Winsock.SendData Text4(0).Text & Text4(1).Text & Text4(2).Text & Text4(3).Text &  
Text4(4).Text & Text4(5).Text & Text4(6).Text & Text4(7).Text
```

End If

```
'LIST_COMMAND.Text = vbNullString           'clear after send
```

End Sub

Private Sub Timer1_Timer()

```
Select Case Winsock.State
```

```
Case sckClosed
```

```
  lblStatus.Caption = "Connection closed"
```

```
Case sckOpen
```

```
  lblStatus.Caption = "Open"
```

```
Case sckListening
```

```
  lblStatus.Caption = "Listening"
```

```
Case sckConnectionPending
```

```
  lblStatus.Caption = "Connection pending"
```

```
Case sckResolvingHost
```

```
  lblStatus.Caption = "Resolving host"
```

```
Case sckHostResolved
```

```
  lblStatus.Caption = "Host resolved"
```

```
Case sckConnecting
```

```
  lblStatus.Caption = "Connecting"
```

```
Case sckConnected
```

```
  lblStatus.Caption = "Connected"
```

```
Case sckClosing
```

```
  lblStatus.Caption = "Peer is closing the connection"
```

```
Case sckError
```

```
  lblStatus.Caption = "Error"
```

End Select

End Sub

Private Sub Winsock1_Close()

Winsock.Close

Winsock.Listen

End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)

If Winsock.State <> sckClosed Then Winsock.Close

Winsock.Accept requestID

End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

Dim incomming As String

Winsock.GetData incomming

LIST_COMMAND.SelStart = Len(txtChat.Text) ' set the selection to the end

LIST_COMMAND.SelText = incomming & vbCrLf ' append the text

LIST_COMMAND.SelStart = Len(txtChat.Text) ' scroll to the bottom

If Left\$(incomming, 4) = "quit" Then Winsock_Close

End Sub

Private Sub Timer2_Timer()

If Winsock.State = sckConnected Then ' make sure we're connected before sending

 '=====add 0000000000

If Len(Text4(6).Text) <> 5 Then

Do While Len(Text4(6).Text) <> 5

Text4(6).Text = "0" & Text4(6).Text

Loop

End If

 '=====

If Len(Text4(7).Text) <> 5 Then

Do While Len(Text4(7).Text) <> 5

Text4(7).Text = "0" & Text4(7).Text

Loop

End If

 '=====

If Text4(0).Text = "True" Then Text4(0).Text = Text4(0).Text & " "

If Text4(1).Text = "True" Then Text4(1).Text = Text4(1).Text & " "

If Text4(2).Text = "True" Then Text4(2).Text = Text4(2).Text & " "

If Text4(3).Text = "True" Then Text4(3).Text = Text4(3).Text & " "


```

    If Text4(4).Text = "True" Then Text4(4).Text = Text4(4).Text & " "
    If Text4(5).Text = "True" Then Text4(5).Text = Text4(5).Text & " "

    Winsock.SendData Text4(0).Text & Text4(1).Text & Text4(2).Text & Text4(3).Text &
    Text4(4).Text & Text4(5).Text & Text4(6).Text & Text4(7).Text

    End If
    'LIST_COMMAND.Text = vbNullString          'clear after send
End Sub

Private Sub Winsock_Close()
    Winsock.Close
    Winsock.Listen
End Sub

Private Sub Winsock_ConnectionRequest(ByVal requestID As Long)
If SERVER_REMOTE.Value = True Then Exit Sub

    If Winsock.State <> sckClosed Then Winsock.Close
    Winsock.Accept requestID
    SERVER_REMOTE.Value = False
    CLOSE_WINSOCK.Value = True

End Sub

Private Sub Winsock_DataArrival(ByVal bytesTotal As Long)
    Dim incomming As String

    Winsock.GetData incomming
    LIST_COMMAND.SelStart = Len(LIST_COMMAND.Text)      ' set the selection to the
end
    LIST_COMMAND.SelText = incomming & vbCrLf          ' append the text
    LIST_COMMAND.SelStart = Len(LIST_COMMAND.Text)      ' scroll to the bottom

'LIST_COMMAND.Text = incomming + Chr$(10) + Chr$(13)
If Left$(incomming, 4) = "quit" Then Winsock_Close

'=====command CONTROL M1r=====
    If Left$(incomming, 5) = "m1_on" Or Left$(incomming, 5) = "M1_ON" Then
CLIENT_START_M1
    If Left$(incomming, 6) = "m1_off" Or Left$(incomming, 6) = "M1_OFF" Then
CLIENT_STOP_M1

'===== COMMAND CONTROL
M2=====
    If Left$(incomming, 5) = "m2_on" Or Left$(incomming, 5) = "M2_ON" Then
CLIENT_START_M2
    If Left$(incomming, 6) = "m2_off" Or Left$(incomming, 6) = "M2_OFF" Then
CLIENT_STOP_M2

```

```

'=====COMAND CONTROL
M3=====
  If Left$(incomming, 5) = "m3_on" Or Left$(incomming, 5) = "M3_ON" Then
CLIENT_START_M3
  If Left$(incomming, 6) = "m3_off" Or Left$(incomming, 6) = "M3_OFF" Then
CLIENT_STOP_M3
'===== COMMAND ANALOG
OUT=====
  If Left$(incomming, 5) = "anout" Or Left$(incomming, 5) = "ANOUT" Then
ANALOGOUT$ = Mid$(incomming, 6, 5)

  ANOUT_VALUE = Val(ANALOGOUT$)
  WRITE_ANOUT
  End If
End Sub

```

=====

โปรแกรมในส่วนของ GUI CLIENT

```

Dim DATA$
Private Sub CONNECT_SERVER_Click()
  If LenB(IP_ADDRESS.Text) = 0 Then
    MsgBox "Enter the IP address to which you wish to connect."
    Exit Sub
  End If

  If Winsock.State <> sckConnected Then
    Winsock.LocalPort = 0 ' get a free random port
    Winsock.Connect IP_ADDRESS.Text, 456
  End If
End Sub

Private Sub DIS_CONNECT_Click()
  If MsgBox("Are you sure you want to Disconnect?", vbYesNo, "Confirm Disconnect") =
vbYes Then
    If LBSTATUS.Caption = "Connection closed" Then Exit Sub
    Winsock.SendData "quit"
  End If
End Sub

Private Sub Form_Load()

End Sub

Private Sub SENT_Click()
  If Winsock.State = sckConnected Then
    Winsock.SendData SENT_TXT.Text
    'DIS_PLAY.SelText = SENT_TXT.Text
    DIS_PLAY.SelStart = Len(DIS_PLAY.Text) 'scroll to bottom
  Else

```

```

    MsgBox "Not Connected"
End If
SENT_TXT.Text = vbNullString 'clear after send

End Sub

Private Sub Timer1_Timer()
    Select Case Winsock.State
        Case sckClosed
            LBSTATUS.Caption = "Connection closed"

        Case sckOpen
            LBSTATUS.Caption = "Open"
            Winsock.Close

        Case sckListening
            LBSTATUS.Caption = "Listening"

        Case sckConnectionPending
            LBSTATUS.Caption = "Connection pending"

        Case sckResolvingHost
            LBSTATUS.Caption = "Resolving host"

        Case sckHostResolved
            LBSTATUS.Caption = "Host resolved"

        Case sckConnecting
            LBSTATUS.Caption = "Connecting"

        Case sckConnected
            LBSTATUS.Caption = "Connected"

        Case sckClosing
            LBSTATUS.Caption = "Peer is closing the connection"

        Case sckError
            LBSTATUS.Caption = "Error"
            Winsock.Close

    End Select

End Sub

Private Sub Winsock_Close()
    Winsock.Close
End Sub

Private Sub Winsock_DataArrival(ByVal bytesTotal As Long)
    Dim sIncomming As String

```

```
DIS_PLAY.Text = ""  
Winsock.GetData sIncomming  
DIS_PLAY.Text = sIncomming  
DATA$ = DIS_PLAY.Text  
Text5.Text = Mid$(DATA$, 1, 5)  
Text6.Text = Mid$(DATA$, 6, 5)  
Text7.Text = Mid$(DATA$, 11, 5)  
Text8.Text = Val(Mid$(DATA$, 36, 5))
```

```
Text1.Text = Mid$(DATA$, 16, 5)  
Text2.Text = Mid$(DATA$, 21, 5)  
Text3.Text = Mid$(DATA$, 26, 5)  
Text4.Text = Val(Mid$(DATA$, 31, 5))
```

End Sub