

```

Attribute VB_Name = "AdRoutine"
Option Explicit

'Public Const AD_MaxVolt = 5      'バイポーラ -5~ +5V
Public Const AD_MaxVolt = 10      'バイポーラ -10~+10V

Public Function ConvInt(vInData As Single) As Integer
    ConvInt = vInData - IIf(vInData > 32767, 65536, 0)
End Function

Public Function ConvData(vInData As Integer, vChNo As Integer) As Single
    Dim dummy As Single
    dummy = vInData
    Select Case (vChNo)
        Case 0 To 5
            ConvData = dummy * AD_MaxVolt / 32768
        Case Else
            ConvData = 0
    End Select
End Function

Public Function ConvBin(vInData As Single, vChNo As Integer) As Integer
    Dim dummy As Single
    Select Case (vChNo)
        Case 0 To 5
            dummy = vInData * 32768 / AD_MaxVolt
        Case Else
            dummy = 0
    End Select
    If dummy < -32768 Then
        dummy = -32768
    ElseIf dummy > 32767 Then
        dummy = 32767
    End If
    ConvBin = CInt(dummy)
End Function

```

AdRoutine.bas x 2

```

Attribute VB_Name = "AdRoutine"
Option Explicit

'Public Const AD_MaxVolt = 5      'バイポーラ -5~ +5V
Public Const AD_MaxVolt = 10      'バイポーラ -10~+10V

Public Function ConvInt(vInData As Single) As Integer
    ConvInt = vInData - IIf(vInData > 32767, 65536, 0)
End Function

Public Function ConvData(vInData As Integer, vChNo As Integer) As Single
    Dim dummy As Single
    dummy = vInData
    Select Case (vChNo)
        Case 0 To 5
            ConvData = dummy * AD_MaxVolt / 32768
        Case Else
            ConvData = 0
    End Select
End Function

Public Function ConvBin(vInData As Single, vChNo As Integer) As Integer
    Dim dummy As Single
    Select Case (vChNo)
        Case 0 To 5
            dummy = vInData * 32768 / AD_MaxVolt
        Case Else
            dummy = 0
    End Select
    If dummy < -32768 Then
        dummy = -32768
    ElseIf dummy > 32767 Then
        dummy = 32767
    End If
    ConvBin = CInt(dummy)
End Function

```

```

PVERSION 1.0 CLASS
BEGIN
  MultiUse = -1 ' True
  Persistable = 0 ' NotPersistable
  DataBindingBehavior = 0 ' vbNone
  DataSourceBehavior = 0 ' vbNone
  MTSTransactionMode = 0 ' NotAnMTSObject
END
Attribute VB_Name = "CMeasData"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" , "Yes"
Attribute VB_Ext_KEY = "Top_Level" , "Yes"
Option Explicit

```

```

Private DataAD(40000, 5) As Integer
Private DataKine(16999, 0 To 5) As Integer

```

```

Private DataPoint As Long
Private RecDataN As Long
Private TrgLevel1BinL As Integer, TrgLevel1BinH As Integer
Private TrgLevel2BinL As Integer, TrgLevel2BinH As Integer

```

```

Private HeadData As DataFileHead

```

```

'プロパティ値を保持するためのローカル変数。
Private mvarTriggerTime As Date 'ローカル コピー
Private mvarFileName As Variant 'ローカル コピー
Private mvarTrgLevel1 As Single, mvarTrgLevel2 As Single 'ローカル コピー

```

```

Private mvarDataDir As String 'ローカル コピー
Private mvarTimeLag As Integer 'ローカル コピー
Private mvarActMode As Integer 'ローカル コピー

```

```

Private Sub Class_Initialize()
  DataPoint = 0
End Sub

```

```

Public Sub DataFileGet()
  Dim FileNo As Integer

```

```

VERSION 1.0 CLASS
BEGIN
  MultiUse = -1 ' True
  Persistable = 0 ' NotPersistable
  DataBindingBehavior = 0 ' vbNone
  DataSourceBehavior = 0 ' vbNone
  MTSTransactionMode = 0 ' NotAnMTSObject
END
Attribute VB_Name = "CMeasData"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" , "Yes"
Attribute VB_Ext_KEY = "Top_Level" , "Yes"
Option Explicit

```

```

Private DataAD(100000, 3) As Integer
Private DataKine(59999, 0 To 2) As Integer
Private DataKine() As Integer

```

```

Private DataPoint As Long
Private RecDataN As Long
Private TrgLevel1BinL As Integer, TrgLevel1BinH As Integer
Private TrgLevel2BinL As Integer, TrgLevel2BinH As Integer
Private TrgLevel3BinL As Integer, TrgLevel3BinH As Integer
Private TrgLevel4BinL As Integer, TrgLevel4BinH As Integer

```

```

Private AbsMaxDataBin(3) As Long '新宅 追加
Private AbsL10DataBin(3) As Long '新宅 追加

```

```

Private HeadData As DataFileHead

```

```

'プロパティ値を保持するためのローカル変数。
Private mvarTriggerTime As Date 'ローカル コピー
Private mvarFileName As Variant 'ローカル コピー
Private mvarTrgLevel1 As Single, mvarTrgLevel2 As Single 'ローカル コピー
Private mvarTrgLevel3 As Single, mvarTrgLevel4 As Single 'ローカル コピー

```

```

Private mvarDataDir As String 'ローカル コピー
Private mvarTimeLag As Integer 'ローカル コピー
Private mvarActMode As Integer 'ローカル コピー
Private mvarGalVRange1 As Single, mvarGalVRange2 As Single 'ローカル コピー
Private mvarGalVRange3 As Single, mvarGalVRange4 As Single 'ローカル コピー
Private mvarKeiho1 As Single, mvarKeiho2 As Single 'ローカル コピー
Private mvarKeiho3 As Single, mvarKeiho4 As Single 'ローカル コピー
Private mvarKeiho11 As Single, mvarKeiho12 As Single 'ローカル コピー
Private mvarKeiho13 As Single, mvarKeiho14 As Single 'ローカル コピー

```

```

Private mvarSampSec As Single 'ローカル コピー
Private mvarRecDataSec As Single, mvarRecPreDataSec As Single 'ローカル コピー

```

```

Private Sub Class_Initialize()
  DataPoint = 0
End Sub

```

```

Public Sub DataFileGet()
  Dim FileNo As Integer

```

```

FileNo = FreeFile
Open FileName For Binary As FileNo
Get FileNo, 1, HeadData
Get FileNo, 1 + 256, DataKine
Close FileNo
End Sub

```

```

Public Sub DataFilePut()
Dim FileNo As Integer
FileNo = FreeFile
Open FileName For Binary As FileNo
Put FileNo, 1, HeadData
Put FileNo, 1 + 256, DataKine
Close FileNo
End Sub
Public Sub RecStart(ByVal vStartTime As Date)
Dim pos As Long

```

```

CreateFileName vStartTime
FileOpen
HeadData.観測開始_年 = CInt(Format(vStartTime, "yyyy"))
HeadData.観測開始_月 = CByte(Format(vStartTime, "mm"))
HeadData.観測開始_日 = CByte(Format(vStartTime, "dd"))
HeadData.観測開始_時 = CByte(Format(vStartTime, "hh"))
HeadData.観測開始_分 = CByte(Format(vStartTime, "nn"))
HeadData.観測開始_秒 = CByte(Format(vStartTime, "ss"))

```

```

RecDataN = 0
AppendLog "観測開始": Grig_AppendStatus "観測開始"
ActMode = e動作Mode.収録中
For pos = DataPoint - 2000 To DataPoint - 1
RecDataPut (pos)
Next

```

```

End Sub
Private Sub RecDataPut(vPos As Long)
Dim DimPos As Long, i As Integer
DimPos = (vPos + 40000) Mod 40000
For i = 0 To gMaxChNo
DataKine(RecDataN, i) = DataAD(DimPos, i)
Next
RecDataN = RecDataN + 1
If RecDataN = 17000 Then
RecEnd
End If

```

```

FileNo = FreeFile
Open FileName For Binary As FileNo
Get FileNo, 1, HeadData
Get FileNo, 1 + 256, DataKine
Close FileNo
End Sub

```

```

Public Sub DataFilePut()
Dim FileNo As Integer
FileNo = FreeFile
Open FileName For Binary As FileNo
Put FileNo, 1, HeadData
Put FileNo, 1 + 256, DataKine
Close FileNo
End Sub
Public Sub RecStart(ByVal vStartTime As Date)
Dim pos As Long

```

```

Dim DataNum As Long
Dim DataPreTrgNum As Long

DataNum = Int(RecDataSec / SampSec)
ReDim DataKine(DataNum, 0 To 3) As Integer
CreateFileName vStartTime
FileOpen
HeadData.観測開始_年 = CInt(Format(vStartTime, "yyyy"))
HeadData.観測開始_月 = CByte(Format(vStartTime, "mm"))
HeadData.観測開始_日 = CByte(Format(vStartTime, "dd"))
HeadData.観測開始_時 = CByte(Format(vStartTime, "hh"))
HeadData.観測開始_分 = CByte(Format(vStartTime, "nn"))
HeadData.観測開始_秒 = CByte(Format(vStartTime, "ss"))
HeadData.CH数_Int = gMaxChNo
HeadData.サンプリングタイム_秒 = SampSec
HeadData.収録データ数_LONG = Int(RecDataSec / SampSec)
HeadData.プリトリガデータ数_LONG = Int(RecPreDataSec / SampSec)
HeadData.CH1レンジ_gal = GalVRange1
HeadData.CH2レンジ_gal = GalVRange2
HeadData.CH3レンジ_gal = GalVRange3
HeadData.CH4レンジ_gal = GalVRange4

```

```

RecDataN = 0
AppendLog "観測開始": Grig_AppendStatus "観測開始"
ActMode = e動作Mode.収録中
DataPreTrgNum = Int(RecPreDataSec / SampSec)
For pos = DataPoint - DataPreTrgNum To DataPoint - 1
RecDataPut (pos)
Next

```

```

End Sub
Private Sub RecDataPut(vPos As Long)
Dim DimPos As Long, i As Integer
DimPos = (vPos + 100000) Mod 100000
For i = 0 To gMaxChNo
DataKine(RecDataN, i) = DataAD(DimPos, i)
Next
RecDataN = RecDataN + 1
If RecDataN = Int(RecDataSec / SampSec) + 1 Then '収録時間
If RecDataN = 17000 Then
RecEnd
End If

```

End Sub

Public Sub RecEnd()

```
DataFilePut
AppendLog "観測終了": Grig_AppendStatus "観測終了"
ActMode = e動作Mode.監視中
DataFileListUpdate
If MoveFileN < 100 Then
    MoveFile(MoveFileN) = mvarFileName
    MoveFileN = MoveFileN + 1
End If
```

End Sub

```
Public Function PuthData(rData() As Integer, vCount As Long, vTime As Date) As Boolean
    Dim i As Integer, j As Integer, Offset As Long
```

```
If fMainForm.ManualStart = True Then
    PuthData = True
    fMainForm.ManualStart = False
Else
    PuthData = False
End If
```

```
Offset = DataPoint
For i = 0 To vCount - 1
    For j = 0 To gMaxChNo
        If j < 3 Then
```

```
            If rData(i, j) < TrgLevel1BinL Or rData(i, j) > TrgLevel1BinH Then
                PuthData = True
            End If
```

Else

```
            If rData(i, j) < TrgLevel2BinL Or rData(i, j) > TrgLevel2BinH Then
                PuthData = True
            End If
```

End Sub

Public Sub RecEnd()

```
DataFilePut
AppendLog "観測終了": Grig_AppendStatus "観測終了"
ActMode = e動作Mode.監視中
DataFileListUpdate
If MoveFileN < 100 Then
    MoveFile(MoveFileN) = mvarFileName
    MoveFileN = MoveFileN + 1
End If
```

End Sub

```
Public Function PuthData(rData() As Integer, vCount As Long, vTime As Date) As Boolean
    Dim i As Integer, j As Integer, Offset As Long
```

```
    Dim tmpDat As Long
    Dim valDat As Single
    Dim intDat As Integer
```

```
If fMainForm.ManualStart = True Then
    PuthData = True
    fMainForm.ManualStart = False
Else
    PuthData = False
End If
```

```
Offset = DataPoint
For i = 0 To vCount - 1
    For j = 0 To gMaxChNo
```

```
        'トリガレベル変更 2003/07/17 新宅
```

```
        If j = 0 Then
```

```
            'Lmax抽出
```

```
            If rData(i, j) < TrgLevel1BinL Or rData(i, j) > TrgLevel1BinH Then
                PuthData = True
            End If
```

End If

```
            tmpDat = rData(i, j)
```

```
            tmpDat = Abs(tmpDat)
```

```
            If tmpDat > AbsMaxDataBin(0) Then AbsMaxDataBin(0) = tmpDat
```

```
            'L10抽出
```

```
            valDat = 40 * tmpDat / 32768 * 10 + GalVRange1 - 100
```

```
            valDat = Int(valDat * 10)
```

```
            If 0 < DataRing(RingCount, 0) And DataRing(RingCount, 0) <= 1500 Then '旧データのカウン
```

```
                intDat = DataRing(RingCount, 0)
```

```
                L10DataBox(intDat, 0) = L10DataBox(intDat, 0) - 1
```

```
            End If
```

```
            DataRing(RingCount, 0) = valDat '新データのカウンTBOXをインクリメント
```

```
            If 0 < valDat And valDat <= 1500 Then
```

```
                intDat = valDat
```

```
                L10DataBox(intDat, 0) = L10DataBox(intDat, 0) + 1
```

```
            End If
```

```
        ElseIf j = 1 Then
```

```
            If rData(i, j) < TrgLevel2BinL Or rData(i, j) > TrgLevel2BinH Then
                PuthData = True
            End If
```

End If

```
            tmpDat = rData(i, j)
```

```
            tmpDat = Abs(tmpDat)
```

```
            If tmpDat > AbsMaxDataBin(1) Then AbsMaxDataBin(1) = tmpDat
```

```
            'L10抽出
```

```
            valDat = 40 * tmpDat / 32768 * 10 + GalVRange2 - 100
```

```
            valDat = Int(valDat * 10)
```

```
            If 0 < DataRing(RingCount, 1) And DataRing(RingCount, 1) <= 1500 Then '旧データのカウン
```

```
                intDat = DataRing(RingCount, 1)
```

```
                L10DataBox(intDat, 1) = L10DataBox(intDat, 1) - 1
```

```

End If
DataAD(DataPoint, j) = rData(i, j)
Next

```

```

If fMainForm.chkWatchTime(Hour(vTime)).Value = 0 Then '監視時間外はトリガーをオフする
    PuthData = False
End If

```

```

End If
DataRing(RingCount, 1) = valDat '新データのカウンタBOXをインクリメント
If 0 < valDat And valDat <= 1500 Then
    intDat = valDat
    L10DataBox(intDat, 1) = L10DataBox(intDat, 1) + 1
End If
ElseIf j = 2 Then
    If rData(i, j) < TrgLevel3BinL Or rData(i, j) > TrgLevel3BinH Then
        PuthData = True
    End If
    tmpDat = rData(i, j)
    tmpDat = Abs(tmpDat)
    If tmpDat > AbsMaxDataBin(2) Then AbsMaxDataBin(2) = tmpDat
    'L10抽出
    valDat = 40 * tmpDat / 32768 * 10 + GalVRange3 - 100
    valDat = Int(valDat * 10)
    If 0 < DataRing(RingCount, 2) And DataRing(RingCount, 2) <= 1500 Then '旧データのカウン
        intDat = DataRing(RingCount, 2)
        L10DataBox(intDat, 2) = L10DataBox(intDat, 2) - 1
    End If
    DataRing(RingCount, 2) = valDat '新データのカウンタBOXをインクリメント
    If 0 < valDat And valDat <= 1500 Then
        intDat = valDat
        L10DataBox(intDat, 2) = L10DataBox(intDat, 2) + 1
    End If
ElseIf j = 3 Then
    If rData(i, j) < TrgLevel4BinL Or rData(i, j) > TrgLevel4BinH Then
        PuthData = True
    End If
    tmpDat = rData(i, j)
    tmpDat = Abs(tmpDat)
    If tmpDat > AbsMaxDataBin(3) Then AbsMaxDataBin(3) = tmpDat
    'L10抽出
    valDat = 40 * tmpDat / 32768 * 10 + GalVRange4 - 100
    valDat = Int(valDat * 10)
    If 0 < DataRing(RingCount, 3) And DataRing(RingCount, 3) <= 1500 Then '旧データのカウン
        intDat = DataRing(RingCount, 3)
        L10DataBox(intDat, 3) = L10DataBox(intDat, 3) - 1
    End If
    DataRing(RingCount, 3) = valDat '新データのカウンタBOXをインクリメント
    If 0 <= valDat And valDat <= 1500 Then
        intDat = valDat
        L10DataBox(intDat, 3) = L10DataBox(intDat, 3) + 1
    End If

    If RingCount >= 5999 Then
        RingCount = 0
    Else
        RingCount = RingCount + 1
    End If
End If

```

```

End If
DataAD(DataPoint, j) = rData(i, j)
Next

```

```

tmpDat = rData(vCount - 1, 0) : LvDataGal(0) = 40 * tmpDat / 32768 * 10 + GalVRange1 - 100
tmpDat = rData(vCount - 1, 1) : LvDataGal(1) = 40 * tmpDat / 32768 * 10 + GalVRange2 - 100
tmpDat = rData(vCount - 1, 2) : LvDataGal(2) = 40 * tmpDat / 32768 * 10 + GalVRange3 - 100
tmpDat = rData(vCount - 1, 3) : LvDataGal(3) = 40 * tmpDat / 32768 * 10 + GalVRange4 - 100

```

```

If fMainForm.chkWatchTime(Hour(vTime)).Value = 0 Then '監視時間外はトリガーをオフする
    PuthData = False
End If

```

```

If PuthData = True And ActMode = e動作Mode.監視中 Then ' 観測スタート
    RecStart vTime
End If
If ActMode = e動作Mode.収録中 Then
    RecDataPut (DataPoint)
End If
DataPoint = (DataPoint + 1) Mod 40000
Next

End Function
Public Sub CreateFileName(ByVal vTriggerTime As Date)
    Dim strBuf As String
    strBuf = Format(vTriggerTime, "yyyymmddhhnss")
    mvarFileName = DataDir & "¥" & strBuf & ".WDT"

End Sub
Public Property Let FileName(ByVal vData As Variant)
    mvarFileName = vData
End Property
Public Property Set FileName(ByVal vData As Variant)
    Set mvarFileName = vData
End Property
Public Property Get FileName() As Variant
    If IsObject(mvarFileName) Then
        Set FileName = mvarFileName
    Else
        FileName = mvarFileName
    End If
End Property
Public Property Let TriggerTime(ByVal vData As Date)
    mvarTriggerTime = vData
End Property
Public Property Get TriggerTime() As Date
    TriggerTime = mvarTriggerTime
End Property
Public Property Let DataDir(ByVal vData As String)
    mvarDataDir = vData
End Property
Public Property Get DataDir() As String
    DataDir = mvarDataDir
End Property

Public Property Let TrgLevel1(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel1 = vData
    TrgLevel1BinL = ConvBin(-vData, 0)
    TrgLevel1BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel1() As Single
    TrgLevel1 = mvarTrgLevel1
End Property
Public Property Let TrgLevel2(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel2 = vData

```

```

If Right(Time, 5) = "00:00" Then PuthData = True

If PuthData = True And ActMode = e動作Mode.監視中 Then ' 観測スタート
    RecStart vTime
End If
If ActMode = e動作Mode.収録中 Then
    RecDataPut (DataPoint)
End If
DataPoint = (DataPoint + 1) Mod 100000
Next

End Function
Public Sub CreateFileName(ByVal vTriggerTime As Date)
    Dim strBuf As String
    strBuf = Format(vTriggerTime, "yyyymmddhhnss")
    If PointNo > 0 Then
        mvarFileName = DataDir & "¥" & PointNo & "_" & strBuf & ".WDT"
    Else
        mvarFileName = DataDir & "¥" & strBuf & ".WDT"
    End If
    mvarFileName = DataDir & "¥" & strBuf & ".WDT"

End Sub
Public Property Let FileName(ByVal vData As Variant)
    mvarFileName = vData
End Property
Public Property Set FileName(ByVal vData As Variant)
    Set mvarFileName = vData
End Property
Public Property Get FileName() As Variant
    If IsObject(mvarFileName) Then
        Set FileName = mvarFileName
    Else
        FileName = mvarFileName
    End If
End Property
Public Property Let TriggerTime(ByVal vData As Date)
    mvarTriggerTime = vData
End Property
Public Property Get TriggerTime() As Date
    TriggerTime = mvarTriggerTime
End Property
Public Property Let DataDir(ByVal vData As String)
    mvarDataDir = vData
End Property
Public Property Get DataDir() As String
    DataDir = mvarDataDir
End Property

Public Property Let TrgLevel1(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel1 = vData
    TrgLevel1BinL = ConvBin(-vData, 0)
    TrgLevel1BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel1() As Single
    TrgLevel1 = mvarTrgLevel1
End Property
Public Property Let TrgLevel2(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel2 = vData

```

```

TrgLevel2BinL = ConvBin(-vData, 0)
TrgLevel2BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel2() As Single
    TrgLevel2 = mvarTrgLevel2
End Property

```

```

TrgLevel2BinL = ConvBin(-vData, 0)
TrgLevel2BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel2() As Single
    TrgLevel2 = mvarTrgLevel2
End Property
Public Property Let TrgLevel3(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel3 = vData
    TrgLevel3BinL = ConvBin(-vData, 0)
    TrgLevel3BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel3() As Single
    TrgLevel3 = mvarTrgLevel3
End Property
Public Property Let TrgLevel4(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarTrgLevel4 = vData
    TrgLevel4BinL = ConvBin(-vData, 0)
    TrgLevel4BinH = ConvBin(vData, 0)
End Property
Public Property Get TrgLevel4() As Single
    TrgLevel4 = mvarTrgLevel4
End Property
Public Property Let GalVRange1(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarGalVRange1 = vData
End Property
Public Property Get GalVRange1() As Single
    GalVRange1 = mvarGalVRange1
End Property
Public Property Let GalVRange2(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarGalVRange2 = vData
End Property
Public Property Get GalVRange2() As Single
    GalVRange2 = mvarGalVRange2
End Property
Public Property Let GalVRange3(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarGalVRange3 = vData
End Property
Public Property Get GalVRange3() As Single
    GalVRange3 = mvarGalVRange3
End Property
Public Property Let GalVRange4(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarGalVRange4 = vData
End Property
Public Property Get GalVRange4() As Single
    GalVRange4 = mvarGalVRange4
End Property
Public Property Let Keiho1(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho1 = vData
End Property
Public Property Get Keiho1() As Single
    Keiho1 = mvarKeiho1
End Property
Public Property Let Keiho2(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho2 = vData

```

```

End Property
Public Property Get Keiho2() As Single
    Keiho2 = mvarKeiho2
End Property
Public Property Let Keiho3(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho3 = vData
End Property
Public Property Get Keiho3() As Single
    Keiho3 = mvarKeiho3
End Property
Public Property Let Keiho4(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho4 = vData
End Property
Public Property Get Keiho4() As Single
    Keiho4 = mvarKeiho4
End Property
Public Property Let Keiho11(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho11 = vData
End Property
Public Property Get Keiho11() As Single
    Keiho11 = mvarKeiho11
End Property
Public Property Let Keiho12(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho12 = vData
End Property
Public Property Get Keiho12() As Single
    Keiho12 = mvarKeiho12
End Property
Public Property Let Keiho13(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho13 = vData
End Property
Public Property Get Keiho13() As Single
    Keiho13 = mvarKeiho13
End Property
Public Property Let Keiho14(ByVal vData As Single)
    If vData < 0 Then vData = vData * -1
    mvarKeiho14 = vData
End Property
Public Property Get Keiho14() As Single
    Keiho14 = mvarKeiho14
End Property

```

```

Public Property Let ActMode(ByVal vData As Integer)
    mvarActMode = vData
End Property
Public Property Get ActMode() As Integer
    ActMode = mvarActMode
End Property

```

```

Public Property Let ActMode(ByVal vData As Integer)
    mvarActMode = vData
End Property
Public Property Get ActMode() As Integer
    ActMode = mvarActMode
End Property
Public Property Let SampSec(ByVal vData As Single)
    mvarSampSec = vData

```

```

End Property
Public Property Get SampSec() As Single
    SampSec = mvarSampSec
End Property
Public Property Let RecPreDataSec(ByVal vData As Single)
    mvarRecPreDataSec = vData
End Property
Public Property Get RecPreDataSec() As Single
    RecPreDataSec = mvarRecPreDataSec
End Property
Public Property Let RecDataSec(ByVal vData As Single)
    mvarRecDataSec = vData
End Property
Public Property Get RecDataSec() As Single
    RecDataSec = mvarRecDataSec
End Property

```

```

Public Property Get ActModeStr() As String
    Select Case mvarActMode
    Case 0:
        ActModeStr = "プログラム停止"
    Case 1:
        ActModeStr = "データ監視中"
    Case 2:
        ActModeStr = "データ収録中"
    Case Else
        ActModeStr = "未定義"
    End Select
End Property

```

```

Public Property Get ActModeStr() As String
    Select Case mvarActMode
    Case 0:
        ActModeStr = "プログラム停止"
    Case 1:
        ActModeStr = "データ監視中"
    Case 2:
        ActModeStr = "データ収録中"
    Case Else
        ActModeStr = "未定義"
    End Select
End Property

```

```

Public Function MaxData(ByVal vCh As Integer)
    MaxData = AbsMaxDataBin(vCh)
End Function
Public Function L05Data(ByVal vCh As Long)
    Dim i As Long
    Dim j As Long

    ' If IntvPitch = 1 Then
    '     For i = 1500 To 0 Step -1
    '         j = j + L10DataBox(i, vCh)
    '         If j >= 60 Then L10Data = i / 10: Exit For
    '     Next i
    ' ElseIf IntvPitch = 10 Then
    '     For i = 1500 To 0 Step -1
    '         j = j + L10DataBox(i, vCh)
    '         If j >= 300 Then
    '             L05Data = i / 10
    '             Exit For
    '         End If
    '     Next i
    ' End If
End Function
Public Function L10Data(ByVal vCh As Long)
    Dim i As Long
    Dim j As Long

    ' If IntvPitch = 1 Then
    '     For i = 1500 To 0 Step -1

```

```
        j = j + L10DataBox(i, vCh)
        If j >= 60 Then L10Data = i / 10: Exit For
    Next i
    ElseIf IntvPitch = 10 Then
        For i = 1500 To 0 Step -1
            j = j + L10DataBox(i, vCh)
            If j >= 600 Then
                L10Data = i / 10
                Exit For
            End If
        Next i
    End If
End Function

Public Function L10DataClr (ByVal vCh As Integer)

Erase L10DataBox

End Function
Public Function MaxDataClr (ByVal vCh As Integer)

AbsMaxDataBin(vCh) = 0
AbsL10DataBin(vCh) = 0
End Function
```

```
Attribute VB_Name = "CtrlSub01"  
Option Explicit  
'ListViewnのヘッダ - インデックス番号取得  
Public Function GetListHardIndex(rListView As ListView, vKey As String) As Integer  
    GetListHardIndex = rListView.ColumnHeaders(vKey).Index - 1  
End Function
```

CtrlSub01.bas x 2

```
Attribute VB_Name = "CtrlSub01"  
Option Explicit  
'ListViewnのヘッダ - インデックス番号取得  
Public Function GetListHardIndex(rListView As ListView, vKey As String) As Integer  
    GetListHardIndex = rListView.ColumnHeaders(vKey).Index - 1  
End Function
```

```

Attribute VB_Name = "FileSub"
Option Explicit

Public Sub LogFileClear()
    Dim fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Output As #fno
    Close #fno
    fMainForm.IsvActionLog.ListItems.Clear
End Sub

Public Sub AppendLog(vActionStr As String)
    Dim fno As Integer
    Dim TimeStr As String, ActionStr As String
    fno = FreeFile()
    Open LogFilePathName For Append As #fno
    TimeStr = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
    ActionStr = vActionStr
    Write #fno, TimeStr, ActionStr
    Close #fno

    With fMainForm
        Dim itmX As ListItem
        Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
        itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        .IsvActionLog.Refresh
    End With
End Sub

Public Sub ReadLog()
    Dim TimeStr As String, ActionStr As String
    Dim itmX As ListItem, fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Append As #fno: Close #fno
    Open LogFilePathName For Input As #fno

    With fMainForm
        .IsvActionLog.ListItems.Clear
        Do While Not EOF(fno)
            Input #fno, TimeStr, ActionStr
            Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
            itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        Loop
    End With
    Close #fno
End Sub

Public Sub DataFileListUpdate()
    Dim DataPahtDir As String, FileName As String, buf2 As String

    With fMainForm
        .IsvMeasList.ListItems.Clear
        DataPahtDir = MstData.DataDir & "¥*.wdt"

        FileName = Dir(DataPahtDir) ' 最初のフォルダ名を返します。
        Do While FileName <> ""
            Dim itmX As ListItem
            Set itmX = .IsvMeasList.ListItems.Add(, , FileName)
        Loop
    End With
End Sub

```

FileSub.bas x 2

```

Attribute VB_Name = "FileSub"
Option Explicit

Public Sub LogFileClear()
    Dim fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Output As #fno
    Close #fno
    fMainForm.IsvActionLog.ListItems.Clear
End Sub

Public Sub AppendLog(vActionStr As String)
    Dim fno As Integer
    Dim TimeStr As String, ActionStr As String
    fno = FreeFile()
    Open LogFilePathName For Append As #fno
    TimeStr = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
    ActionStr = vActionStr
    Write #fno, TimeStr, ActionStr
    Close #fno

    With fMainForm
        Dim itmX As ListItem
        Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
        itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        .IsvActionLog.Refresh
    End With
End Sub

Public Sub ReadLog()
    Dim TimeStr As String, ActionStr As String
    Dim itmX As ListItem, fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Append As #fno: Close #fno
    Open LogFilePathName For Input As #fno

    With fMainForm
        .IsvActionLog.ListItems.Clear
        Do While Not EOF(fno)
            Input #fno, TimeStr, ActionStr
            Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
            itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        Loop
    End With
    Close #fno
End Sub

Public Sub DataFileListUpdate()
    Dim DataPahtDir As String, FileName As String, buf2 As String

    With fMainForm
        .IsvMeasList.ListItems.Clear
        DataPahtDir = MstData.DataDir & "¥*.wdt"

        FileName = Dir(DataPahtDir) ' 最初のフォルダ名を返します。
        Do While FileName <> ""
            Dim itmX As ListItem
            Set itmX = .IsvMeasList.ListItems.Add(, , FileName)
        Loop
    End With
End Sub

```

```
buf2 = FileNameToTimeAndID(FileName)
itm.SubItems(1) = buf2
FileName = Dir
Loop
End With
End Sub
```

```
FileSub.bas x 2
buf2 = FileNameToTimeAndID(FileName)
itm.SubItems(1) = buf2
FileName = Dir
Loop
End With
End Sub
```

```

VERSION 5.00
Begin VB.Form frmAbout
    BorderStyle = 3 '固定ダイアログ'
    Caption = "バージョン情報 TunnelClient"
    ClientHeight = 2175
    ClientLeft = 45
    ClientTop = 330
    ClientWidth = 5850
    ClipControls = 0 'False'
    Icon = "frmAbout.frx":0000
    LinkTopic = "Form1"
    MaxButton = 0 'False'
    MinButton = 0 'False'
    Picture = "frmAbout.frx":08CA
    ScaleHeight = 2175
    ScaleWidth = 5850
    ShowInTaskbar = 0 'False'
    StartUpPosition = 1 'オーナー フォームの中央'
    Tag = "バージョン情報 KoiTunnelClient"
    Begin VB.PictureBox picIcon
        BackColor = &H00C0C0C0&
        ClipControls = 0 'False'
        Height = 960
        Left = 60
        Picture = "frmAbout.frx":0C0C
        ScaleHeight = 900
        ScaleMode = 0 'ユーザー'
        ScaleWidth = 960
        TabIndex = 5
        TabStop = 0 'False'
        Top = 60
        Width = 1020
    End
    Begin VB.CommandButton cmdOK
        Cancel = -1 'True'
        Caption = "OK"
        Default = -1 'True'
        Height = 375
        Left = 4320
        TabIndex = 0
        Tag = "OK"
        Top = 1200
        Width = 1335
    End
    Begin VB.CommandButton cmdSysInfo
        Caption = "システム情報 (&S)..."
        Height = 375
        Left = 4320
        TabIndex = 1
        Tag = "システム情報 (&S)..."
        Top = 1680
        Width = 1335
    End
    Begin VB.Label lblCopyright
        AutoSize = -1 'True'
        Caption = "Copyright (C)"
        BeginProperty Font
            Name = "MS Pゴシック"
            Size = 9.75
            Charset = 128
            Weight = 400
            Underline = 0 'False'
        EndProperty
    End

```

```

VERSION 5.00
Begin VB.Form frmAbout
    BorderStyle = 3 '固定ダイアログ'
    Caption = "バージョン情報 TunnelClient"
    ClientHeight = 2175
    ClientLeft = 45
    ClientTop = 330
    ClientWidth = 5850
    ClipControls = 0 'False'
    Icon = "frmAbout.frx":0000
    LinkTopic = "Form1"
    MaxButton = 0 'False'
    MinButton = 0 'False'
    Picture = "frmAbout.frx":08CA
    ScaleHeight = 2175
    ScaleWidth = 5850
    ShowInTaskbar = 0 'False'
    StartUpPosition = 1 'オーナー フォームの中央'
    Tag = "バージョン情報 KoiTunnelClient"
    Begin VB.PictureBox picIcon
        BackColor = &H00C0C0C0&
        ClipControls = 0 'False'
        Height = 960
        Left = 60
        Picture = "frmAbout.frx":0C0C
        ScaleHeight = 900
        ScaleMode = 0 'ユーザー'
        ScaleWidth = 960
        TabIndex = 5
        TabStop = 0 'False'
        Top = 60
        Width = 1020
    End
    Begin VB.CommandButton cmdOK
        Cancel = -1 'True'
        Caption = "OK"
        Default = -1 'True'
        Height = 375
        Left = 4320
        TabIndex = 0
        Tag = "OK"
        Top = 1200
        Width = 1335
    End
    Begin VB.CommandButton cmdSysInfo
        Caption = "システム情報 (&S)..."
        Height = 375
        Left = 4320
        TabIndex = 1
        Tag = "システム情報 (&S)..."
        Top = 1680
        Width = 1335
    End
    Begin VB.Label lblCopyright
        AutoSize = -1 'True'
        Caption = "Copyright (C)"
        BeginProperty Font
            Name = "MS Pゴシック"
            Size = 9.75
            Charset = 128
            Weight = 400
            Underline = 0 'False'
        EndProperty
    End

```

```

Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 195
Left = 2160
TabIndex = 9
Tag = "著作権"
Top = 840
Width = 1305

```

```

End
Begin VB.Label lblCompany
AutoSize = -1 'True
Caption = "1992-2002 Nextforest"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
Height = 195
Left = 3540
TabIndex = 8
Tag = "会社名"
Top = 840
Width = 2085

```

```

End
Begin VB.Label Label1
Alignment = 1 '右揃え
AutoSize = -1 'True
Caption = "Nextforest"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = -1 'True
Strikethrough = 0 'False

```

```

EndProperty
Height = 195
Left = 1200
TabIndex = 7
Tag = "プラットフォーム"
Top = 60
Width = 900

```

```

End
Begin VB.Label lblPlatform
Alignment = 1 '右揃え
AutoSize = -1 'True
Caption = "for Windows NT/2000"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty

```

```

Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 195
Left = 1560
TabIndex = 9
Tag = "著作権"
Top = 840
Width = 1305

```

```

End
Begin VB.Label lblCompany
AutoSize = -1 'True
Caption = "2004 KeisokuResearchConsultant"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
Height = 195
Left = 2880
TabIndex = 8
Tag = "会社名"
Top = 840
Width = 2865

```

```

End
Begin VB.Label Label1
Alignment = 1 '右揃え
AutoSize = -1 'True
Caption = "License to (株)計測リサーチコンサルタント"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = -1 'True
Strikethrough = 0 'False

```

```

EndProperty
Height = 195
Left = 1320
TabIndex = 7
Tag = "プラットフォーム"
Top = 60
Width = 2805

```

```

End
Begin VB.Label lblPlatform
Alignment = 1 '右揃え
AutoSize = -1 'True
Caption = "for Windows NT/2000/XP"

```

```

BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty

```

```

Height      = 195
Left       = 1560
TabIndex   = 6
Tag        = "プラットフォーム"
Top        = 600
Width      = 1830

```

End

```

Begin VB.Label lblTitle
Caption    = "アプリケーション タイトル"
BeginProperty Font
Name      = "MS Pゴシック"
Size      = 12
Charset   = 128
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty

```

```

ForeColor  = &H00000000&
Height     = 285
Left       = 1200
TabIndex   = 4
Tag        = "アプリケーション タイトル"
Top        = 300
Width      = 4095

```

End

```

Begin VB.Line Line1
BorderColor = &H00808080&
BorderStyle = 6 '実線 (ふちどり)
Index       = 1
X1          = 225
X2          = 5657
Y1          = 1080
Y2          = 1080

```

End

```

Begin VB.Line Line1
BorderColor = &H00FFFFFF&
BorderWidth = 2
Index       = 0
X1          = 240
X2          = 5657
Y1          = 1080
Y2          = 1080

```

End

```

Begin VB.Label lblVersion
Caption    = "バージョン"
Height     = 225
Left       = 3600
TabIndex   = 3
Tag        = "バージョン"
Top        = 600
Width      = 1695

```

End

```

Begin VB.Label lblDisclaimer
Caption    = "$"frmAbout.frx":4EA8
ForeColor  = &H00000000&
Height     = 765
Left       = 255
TabIndex   = 2
Tag        = "警告: ..."
Top        = 1200
Width      = 3870

```

```

Height      = 195
Left       = 1215
TabIndex   = 6
Tag        = "プラットフォーム"
Top        = 600
Width      = 2175

```

End

```

Begin VB.Label lblTitle
Caption    = "アプリケーション タイトル"
BeginProperty Font
Name      = "MS Pゴシック"
Size      = 12
Charset   = 128
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty

```

```

ForeColor  = &H00000000&
Height     = 285
Left       = 1200
TabIndex   = 4
Tag        = "アプリケーション タイトル"
Top        = 300
Width      = 4095

```

End

```

Begin VB.Line Line1
BorderColor = &H00808080&
BorderStyle = 6 '実線 (ふちどり)
Index       = 1
X1          = 225
X2          = 5657
Y1          = 1080
Y2          = 1080

```

End

```

Begin VB.Line Line1
BorderColor = &H00FFFFFF&
BorderWidth = 2
Index       = 0
X1          = 240
X2          = 5657
Y1          = 1080
Y2          = 1080

```

End

```

Begin VB.Label lblVersion
Caption    = "バージョン"
Height     = 225
Left       = 3600
TabIndex   = 3
Tag        = "バージョン"
Top        = 600
Width      = 1695

```

End

```

Begin VB.Label lblDisclaimer
Caption    = "$"frmAbout.frx":4EA8
ForeColor  = &H00000000&
Height     = 765
Left       = 255
TabIndex   = 2
Tag        = "警告: ..."
Top        = 1200
Width      = 3870

```

```

End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' レジストリ キー セキュリティ オプション...
Const KEY_ALL_ACCESS = &H2003F

' レジストリ キー ROOT 型...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Null 文字で終わる Unicode 文字列
Const REG_DWORD = 4 ' 32 ビット数値

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal Path As String, ByVal Options As Long, ByVal SecurityAttributes As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal Path As String, ByVal Options As Long, ByVal DataTypeInfo As Long, ByVal Data As String, ByVal DataSize As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub Form_Load()
    lblVersion.Caption = "バージョン " & App.Major & "." & App.Minor & "." & App.Revision
    lblTitle.Caption = App.Title
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    ' レジストリからシステム情報プログラムのパス名前を取得します...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
        ' レジストリからシステム情報プログラムのパス名のみを取得します...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
            ' 既に存在するはずの 32 ビット バージョンのファイルを確認します
            If (Dir(SysInfoPath & "%MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "%MSINFO32.EXE"

            ' エラー - ファイルが見つかりません...
            Else
                GoTo SysInfoErr
            End If
        ' エラー - レジストリのエントリが見つかりません...
    Else
        GoTo SysInfoErr
    End Sub

```

```

End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' レジストリ キー セキュリティ オプション...
Const KEY_ALL_ACCESS = &H2003F

' レジストリ キー ROOT 型...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Null 文字で終わる Unicode 文字列
Const REG_DWORD = 4 ' 32 ビット数値

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal Path As String, ByVal Options As Long, ByVal SecurityAttributes As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal Path As String, ByVal Options As Long, ByVal DataTypeInfo As Long, ByVal Data As String, ByVal DataSize As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub Form_Load()
    lblVersion.Caption = "バージョン " & App.Major & "." & App.Minor & "." & App.Revision
    lblTitle.Caption = App.Title
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    ' レジストリからシステム情報プログラムのパス名前を取得します...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
        ' レジストリからシステム情報プログラムのパス名のみを取得します...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
            ' 既に存在するはずの 32 ビット バージョンのファイルを確認します
            If (Dir(SysInfoPath & "%MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "%MSINFO32.EXE"

            ' エラー - ファイルが見つかりません...
            Else
                GoTo SysInfoErr
            End If
        ' エラー - レジストリのエントリが見つかりません...
    Else
        GoTo SysInfoErr
    End Sub

```

```

End If
Call Shell (SysInfoPath, vbNormalFocus)
Exit Sub
SysInfoErr:
MsgBox "現時点ではシステム情報を使用できません", vbOKOnly
End Sub

Public Function GetKeyValue (KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef KeyVal As String)
Dim i As Long ' ループ カウンタ
Dim rc As Long ' 戻り値のコード
Dim hKey As Long ' オープンしたレジストリ キーのハンドル
Dim hDepth As Long
Dim KeyValType As Long ' レジストリ キーのデータ型
Dim tmpVal As String ' レジストリ キー値の一時保存領域
Dim KeyValSize As Long ' レジストリ キー変数のサイズ

' -----
' ルート キー {HKEY_LOCAL_MACHINE...} にあるレジストリ キーを開きます。
rc = RegOpenKeyEx (KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' レジストリ キーを開く

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' ハンドル エラー...

tmpVal = String$(1024, 0) ' 変数領域の割り当て
KeyValSize = 1024 ' 変数のサイズを記憶

' -----
' レジストリ キー値を取得します...
rc = RegQueryValueEx (hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize) ' キー値の取得/イ

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' ハンドル エラー

tmpVal = VBA.Left (tmpVal, InStr (tmpVal, Chr (0)) - 1)

' 変換のために、キー値の型を調べます...

Select Case KeyValType ' データ型検索...
Case REG_SZ ' String レジストリ キー データ型
KeyVal = tmpVal ' String 値をコピー
Case REG_DWORD ' Double Word レジストリ キー データ型
For i = Len (tmpVal) To 1 Step -1 ' 各ビットの変換
KeyVal = KeyVal + Hex (Asc (Mid (tmpVal, i, 1))) ' Char ごとに値を作成
Next
KeyVal = Format$("&h" + KeyVal) ' Double Word を String に変換
End Select

GetKeyValue = True ' 正常終了
rc = RegCloseKey (hKey) ' レジストリ キーをクローズ
Exit Function ' 終了

GetKeyError: ' エラー発生後の後始末...
KeyVal = "" ' 戻り値の値を空文字列に設定
GetKeyValue = False ' 異常終了
rc = RegCloseKey (hKey) ' レジストリ キーをクローズ
End Function

```

VERSION 5.00

```

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB} #1.2#0"; "COMDLG32.OCX"
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1} #2.0#0"; "MSCOMCTL.OCX"
Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A} #1.1#0"; "TABCTL32.OCX"
Object = "{32F26F51-FE61-11D0-A042-000021D96098} #2.3#0"; "Acxaioc.ocx"
Object = "{1BEF5480-3BEA-11D0-B06E-00A024A3D7D7} #1.0#0"; "ZGLOCK.OCX"
Object = "{3D44DB00-69EE-11D0-B073-00A024A3D7D7} #1.0#0"; "ZAGLOCK.OCX"
Object = "{415ABC80-F52E-11CF-B069-00A024A3D7D7} #1.0#0"; "ZCalendar.ocx"
Object = "{COA63B80-4B21-11D3-BD95-D426EF2C7949} #1.0#0"; "Vsflex7L.ocx"

```

Begin VB.Form frmMain

```

Caption = "発破振動観測プログラム"
ClientHeight = 7860
ClientLeft = 165
ClientTop = 735
ClientWidth = 12105
Icon = "frmMain.frx":0000
LinkTopic = "Form1"
ScaleHeight = 7860
ScaleWidth = 12105

```

StartupPosition = 3 'Windows の既定値

Begin ACXAIOLib.AcxAio AcxAio1

```

Left = 1320
Top = 2520
_Version = 131075
_ExtentX = 847
_ExtentY = 847
_StockProps = 0
BoardName = "AD16-16 (PCI) E"
ObjName = "AI000"
InputRange = 3
NumOfAIChannel = 32
NumOfAOChannel = 1
NumOfDIBit = 4
NumOfDOBit = 4
IoAddress = 0
IrqLevel = 9
DeviceName = "AI000"

```

VERSION 5.00

```

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB} #1.2#0"; "COMDLG32.OCX"
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1} #2.0#0"; "mscomctl.ocx"
Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A} #1.1#0"; "TABCTL32.OCX"
Object = "{1BEF5480-3BEA-11D0-B06E-00A024A3D7D7} #1.0#0"; "ZGLOCK.OCX"
Object = "{3D44DB00-69EE-11D0-B073-00A024A3D7D7} #1.0#0"; "ZAGLOCK.OCX"
Object = "{415ABC80-F52E-11CF-B069-00A024A3D7D7} #1.0#0"; "ZCalendar.ocx"
Object = "{COA63B80-4B21-11D3-BD95-D426EF2C7949} #1.0#0"; "Vsflex7L.ocx"
Object = "{5D378AC1-85AA-11D0-A949-0040051A7A82} #1.0#0"; "CBGraph.ocx"
Object = "{32F26F51-FE61-11D0-A042-000021D96098} #3.4#0"; "Acxaioc.ocx"

```

Begin VB.Form frmMain

```

Caption = "騒音振動監視プログラム"
ClientHeight = 8700
ClientLeft = 225
ClientTop = 870
ClientWidth = 14070
Icon = "frmMain.frx":0000
LinkTopic = "Form1"
ScaleHeight = 8700
ScaleWidth = 14070

```

StartupPosition = 3 'Windows の既定値

Begin ACXAIOLib.AcxAio AcxAio1

```

Left = 3720
Top = 7380
_Version = 196612
_ExtentX = 847
_ExtentY = 847
_StockProps = 0
NumOfAIChannel = 16

```

```

IoAddress = 0
IrqLevel = 0

```

End

Begin VB.CommandButton Command4

```

Caption = ">"
Height = 255
Left = 2160
TabIndex = 107
Top = 480
Width = 315

```

End

Begin VB.CommandButton Command3

```

Caption = "<"
Height = 255
Left = 0
TabIndex = 106
Top = 480
Width = 375

```

End

Begin VB.Timer Timer3

```

Left = 3120
Top = 2760

```

End

Begin VB.Timer Timer2

```

Left = 2520
Top = 2640

```

```

End
Begin VB.CommandButton cmdRun
Caption = "実行"
Height = 375
Left = 3900
TabIndex = 27
Top = 3360
Visible = 0 'False
Width = 495

```

```

End
Begin VSFlex7LCtl.VSFlexGrid gridStatus

```

```

Height = 2115
Left = 120
TabIndex = 45
Top = 5100
Width = 4275
_cx = 7541
_cy = 3731

```

```

_ConvInfo = 1
Appearance = 1
BorderStyle = 1
Enabled = -1 'True

```

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

```

```

Name = "MS ゴシック"
Size = 8.25
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty

```

```

MousePointer = 0
BackColor = -2147483643
ForeColor = -2147483640
BackColorFixed = -2147483633
ForeColorFixed = -2147483630
BackColorSel = -2147483635
ForeColorSel = -2147483634
BackColorBkg = -2147483636
BackColorAlternate = -2147483643
GridColor = -2147483633
GridColorFixed = -2147483632
TreeColor = -2147483632
FloodColor = 192
SheetBorder = -2147483642
FocusRect = 1
HighLight = 1
AllowSelection = -1 'True
AllowBigSelection = -1 'True
AllowUserResizing = 0
SelectionMode = 0
GridLines = 1
GridLinesFixed = 2
GridLineWidth = 1
Rows = 2
Cols = 2
FixedRows = 1
FixedCols = 0
RowHeightMin = 0
RowHeightMax = 0
ColWidthMin = 0
ColWidthMax = 0

```

```

End
Begin VB.CommandButton cmdRun
Caption = "実行"
Height = 375
Left = 3900
TabIndex = 27
Top = 2880

```

```

Visible = 0 'False
Width = 495

```

```

End
Begin VSFlex7LCtl.VSFlexGrid gridStatus

```

```

Height = 1395
Left = 120
TabIndex = 45
Top = 5820
Width = 4275
_cx = 7541
_cy = 2461

```

```

_ConvInfo = 1
Appearance = 1
BorderStyle = 1
Enabled = -1 'True

```

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

```

```

Name = "MS ゴシック"
Size = 8.25
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```

EndProperty

```

```

MousePointer = 0
BackColor = -2147483643
ForeColor = -2147483640
BackColorFixed = -2147483633
ForeColorFixed = -2147483630
BackColorSel = -2147483635
ForeColorSel = -2147483634
BackColorBkg = -2147483636
BackColorAlternate = -2147483643
GridColor = -2147483633
GridColorFixed = -2147483632
TreeColor = -2147483632
FloodColor = 192
SheetBorder = -2147483642
FocusRect = 1
HighLight = 1
AllowSelection = -1 'True
AllowBigSelection = -1 'True
AllowUserResizing = 0
SelectionMode = 0
GridLines = 1
GridLinesFixed = 2
GridLineWidth = 1
Rows = 2
Cols = 2
FixedRows = 1
FixedCols = 0
RowHeightMin = 0
RowHeightMax = 0
ColWidthMin = 0
ColWidthMax = 0

```

```

ExtendLastCol = -1 ' True
FormatString = $"frmMain. frx":0E42
ScrollTrack = 0 ' False
ScrollBars = 3
ScrollTips = 0 ' False
MergeCells = 0
MergeCompare = 0
AutoSize = -1 ' True
AutoSizeMode = 0
AutoSearch = 0
AutoSearchDelay = 2
MultiTotals = -1 ' True
SubtotalPosition = 1
OutlineBar = 0
OutlineCol = 0
Ellipsis = 0
ExplorerBar = 0
PicturesOver = 0 ' False
FillStyle = 0
RightToLeft = 0 ' False
PictureType = 0
TabBehavior = 0
OwnerDraw = 0
Editable = 0
ShowComboButton = -1 ' True
WordWrap = 0 ' False
TextStyle = 0
TextStyleFixed = 0
OleDragMode = 0
OleDropMode = 0
ComboSearch = 3
AutoSizeMouse = -1 ' True
FrozenRows = 0
FrozenCols = 0
AllowUserFreezing = 0
BackColorFrozen = 0
ForeColorFrozen = 0
WallPaperAlignment = 9

```

End

Begin VB.CommandButton cmdManualStart

```

Caption = "マニュアル観測開始"
Height = 375
Left = 120
TabIndex = 44
Top = 4380
Width = 4215

```

End

Begin VB.TextBox txtAD\_Status

```

Height = 270
Left = 1560
TabIndex = 43
Text = "Text1"
Top = 3780
Width = 2775

```

End

Begin MSComctlLib.ProgressBar prbAdBuf

```

Height = 255
Left = 1560
TabIndex = 28
Top = 4080
Width = 2775
_ExtentX = 4895

```

```

ExtendLastCol = -1 ' True
FormatString = $"frmMain. frx":0E42
ScrollTrack = 0 ' False
ScrollBars = 3
ScrollTips = 0 ' False
MergeCells = 0
MergeCompare = 0
AutoSize = -1 ' True
AutoSizeMode = 0
AutoSearch = 0
AutoSearchDelay = 2
MultiTotals = -1 ' True
SubtotalPosition = 1
OutlineBar = 0
OutlineCol = 0
Ellipsis = 0
ExplorerBar = 0
PicturesOver = 0 ' False
FillStyle = 0
RightToLeft = 0 ' False
PictureType = 0
TabBehavior = 0
OwnerDraw = 0
Editable = 0
ShowComboButton = -1 ' True
WordWrap = 0 ' False
TextStyle = 0
TextStyleFixed = 0
OleDragMode = 0
OleDropMode = 0
ComboSearch = 3
AutoSizeMouse = -1 ' True
FrozenRows = 0
FrozenCols = 0
AllowUserFreezing = 0
BackColorFrozen = 0
ForeColorFrozen = 0
WallPaperAlignment = 9

```

End

Begin VB.CommandButton cmdManualStart

```

Caption = "マニュアル観測開始"
Height = 375
Left = 120
TabIndex = 44
Top = 4440
Width = 4215

```

End

Begin VB.TextBox txtAD\_Status

```

Height = 270
Left = 1560
TabIndex = 43
Text = "Text1"
Top = 3300
Width = 2775

```

End

Begin MSComctlLib.ProgressBar prbAdBuf

```

Height = 255
Left = 1560
TabIndex = 28
Top = 3600
Width = 2775
_ExtentX = 4895

```

```

_ExtentY      = 450
_Version      = 393216
Appearance    = 1

```

```

End
Begin VB.Timer Timer1
Enabled       = 0 'False
Left         = 1860
Top          = 2460

```

```

End
Begin VB.TextBox txtActMode
BeginProperty Font
Name         = "MS ゴシック"
Size         = 15.75
Charset      = 128
Weight       = 400
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
Height       = 390
Left         = 1560
TabIndex     = 14
Text         = "観測監視中"
Top          = 3360
Width        = 2775

```

```

End
Begin ZCLOCKLib.ZClock ZClock1

```

```

Height       = 495
Left         = 2520
TabIndex     = 9
Top          = 2640
Width        = 1815
_Version     = 65536
_ExtentX     = 3201
_ExtentY     = 873
_StockProps  = 65
BackColor    = 65535
DigitalClockColor= 8388608

```

```

BeginProperty DigitalClockFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
Name         = "MS ゴシック"
Size         = 12
Charset      = 128
Weight       = 700
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
ShowSecond   = -1 'True
DigitalClockMode= 1
DigitalClockEffect= 0

```

```

End
Begin ZACLOCKLib.ZAClock ZAClock1

```

```

Height       = 2175
Left         = 2520
TabIndex     = 8
Top          = 480
Width        = 1815
_Version     = 65536
_ExtentX     = 3201
_ExtentY     = 3836
_StockProps  = 1
BackColor    = 65535

```

```

_ExtentY      = 450
_Version      = 393216
Appearance    = 1

```

```

End
Begin VB.Timer Timer1
Enabled       = 0 'False
Left         = 1860
Top          = 2460

```

```

End
Begin VB.TextBox txtActMode
BeginProperty Font
Name         = "MS ゴシック"
Size         = 15.75
Charset      = 128
Weight       = 400
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
Height       = 390
Left         = 1560
TabIndex     = 14
Text         = "観測監視中"
Top          = 2880
Width        = 2775

```

```

End
Begin ZCLOCKLib.ZClock ZClock1

```

```

Height       = 375
Left         = 2520
TabIndex     = 9
Top          = 2400
Width        = 1815
_Version     = 65536
_ExtentX     = 3201
_ExtentY     = 661
_StockProps  = 65
BackColor    = 65535
DigitalClockColor= 8388608

```

```

BeginProperty DigitalClockFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
Name         = "MS ゴシック"
Size         = 11.99
Charset      = 128
Weight       = 700
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False

```

```

EndProperty
ShowSecond   = -1 'True
DigitalClockMode= 1
DigitalClockEffect= 0

```

```

End
Begin ZACLOCKLib.ZAClock ZAClock1

```

```

Height       = 2055
Left         = 2520
TabIndex     = 8
Top          = 480
Width        = 1815
_Version     = 65536
_ExtentX     = 3201
_ExtentY     = 3625
_StockProps  = 1
BackColor    = 65535

```

```

End
Begin TabDlg.SSTab tabMain
  Height = 6735
  Left = 4440
  TabIndex = 2
  Top = 480
  Width = 7575
  _ExtentX = 13361
  _ExtentY = 11880
  _Version = 393216
  Tab = 2

  TabHeight = 520
  TabCaption(0) = "観測一覧"
  TabPicture(0) = "frmMain.frx":0E7F
  Tab(0).ControlEnabled= 0 'False
  Tab(0).Control(0) = "frmTab(0)"
  Tab(0).ControlCount= 1
  TabCaption(1) = "動作ログ"
  TabPicture(1) = "frmMain.frx":0E9B
  Tab(1).ControlEnabled= 0 'False
  Tab(1).Control(0) = "frmTab(1)"
  Tab(1).ControlCount= 1
  TabCaption(2) = "アラート入力設定"
  TabPicture(2) = "frmMain.frx":0EB7
  Tab(2).ControlEnabled= -1 'True
  Tab(2).Control(0) = "frmTab(2)"
  Tab(2).Control(0).Enabled= 0 'False
  Tab(2).ControlCount= 1

```

```

End
Begin TabDlg.SSTab tabMain
  Height = 8355
  Left = 4380
  TabIndex = 2
  Top = 60
  Width = 9675
  _ExtentX = 17066
  _ExtentY = 14737
  _Version = 393216
  Tabs = 5
  Tab = 3
  TabsPerRow = 5
  TabHeight = 520
  TabCaption(0) = "観測一覧"
  TabPicture(0) = "frmMain.frx":0E7F
  Tab(0).ControlEnabled= 0 'False
  Tab(0).Control(0) = "frmTab(0)"
  Tab(0).ControlCount= 1
  TabCaption(1) = "動作ログ"
  TabPicture(1) = "frmMain.frx":0E9B
  Tab(1).ControlEnabled= 0 'False
  Tab(1).Control(0) = "frmTab(1)"
  Tab(1).ControlCount= 1
  TabCaption(2) = "アラート入力設定"
  TabPicture(2) = "frmMain.frx":0EB7
  Tab(2).ControlEnabled= 0 'False
  Tab(2).Control(0) = "frmTab(2)"
  Tab(2).Control(0).Enabled= 0 'False
  Tab(2).ControlCount= 1
  TabCaption(3) = "La05, Lv10の経時変化"
  TabPicture(3) = "frmMain.frx":0ED3
  Tab(3).ControlEnabled= -1 'True
  Tab(3).Control(0) = "frmTab(3)"
  Tab(3).Control(0).Enabled= 0 'False
  Tab(3).ControlCount= 1
  TabCaption(4) = "過去のLa05, Lv10"
  TabPicture(4) = "frmMain.frx":0EEF
  Tab(4).ControlEnabled= 0 'False
  Tab(4).Control(0) = "frmTab(4)"
  Tab(4).Control(0).Enabled= 0 'False
  Tab(4).ControlCount= 1
  Begin VB.Frame frmTab
    Height = 7635
    Index = 4
    Left = -74880
    TabIndex = 101
    Top = 480
    Width = 9255
    Begin VB.CheckBox Check1
      Caption = "自動的に管理画面にもどらない"
      Height = 180
      Left = 6180
      TabIndex = 108
      Top = 300
      Width = 2895
    End
  End
  Begin VB.TextBox Text1
    Height = 315
    Left = 3120
    TabIndex = 105
    Top = 480

```

```

Width          = 1815
End
Begin VB.CommandButton Command2
Caption        = "O K"
Height        = 315
Left          = 1980
TabIndex      = 104
Top           = 480
Width         = 915
End
Begin CBGRAPHLib.CBGraph CBGraph2
Height        = 6735
Left          = 180
TabIndex      = 102
Top           = 840
Width         = 8955
_Version      = 65536
_ExtentX     = 15796
_ExtentY     = 11880
_StockProps  = 225
Appearance   = 1
FileName      = "D:\騒音振動dB監視4CH\MAX24HOUR.CBG"
IncludeFile   = -1 ' True
GraphDesign  = "frmMain.frx":0F0B
End
Begin VB.Label txtPointNo00
Caption       = "測点"
BeginProperty Font
Name         = "MS ゴシック"
Size        = 15.75
Charset     = 128
Weight     = 400
Underline   = 0 ' False
Italic     = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height      = 315
Left        = 5160
TabIndex    = 116
Top         = 480
Width       = 1095
End
Begin VB.Label Label24
Caption     = "←カレンダーで表示したい月日を選択してOKをクリックしてください"
BeginProperty Font
Name       = "MS ゴシック"
Size      = 8.25
Charset   = 128
Weight    = 400
Underline = 0 ' False
Italic    = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height    = 195
Left     = 180
TabIndex = 109
Top      = 180
Width    = 5115
End
Begin VB.Label Label29
Caption     = "$"frmMain.frx":26B0
Height     = 375

```

```

        Left      = 240
        TabIndex  = 103
        Top       = 420
        Width     = 1455
    End
End
Begin VB.Frame frmTab
    Height      = 7875
    Index       = 3
    Left        = 120
    TabIndex    = 71
    Top         = 420
    Width       = 9255
    Begin VB.CommandButton Command1
        Caption   = "警報クリア"
        Height    = 375
        Left      = 8220
        TabIndex  = 80
        Top       = 1080
        Visible   = 0 'False
        Width     = 975
    End
    Begin CBGRAPHLib.CBGraph CBGraph1
        Height    = 6675
        Left      = 120
        TabIndex  = 72
        Top       = 1080
        Width     = 8775
        _Version  = 65536
        _ExtentX  = 15478
        _ExtentY  = 11774
        _StockProps = 225
        Appearance = 1
        FileName   = "D:¥騒音振動dB監視4CH¥MAX24HOUR. CBG"
        IncludeFile = -1 'True
        GraphDesign = "frmMain.frx":26D2
    End
    Begin VB.Label txtPointNo
        Caption   = "0"
        BeginProperty Font
            Name      = "MS Pゴシック"
            Size      = 20.25
            Charset   = 128
            Weight    = 400
            Underline  = 0 'False
            Italic    = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height    = 375
        Left      = 480
        TabIndex  = 115
        Top       = 600
        Width     = 495
    End
    Begin VB.Label Label25
        Caption   = "測点"
        BeginProperty Font
            Name      = "MS Pゴシック"
            Size      = 20.25
            Charset   = 128
            Weight    = 400
            Underline  = 0 'False

```

```

        Italic          = 0 'False
        Strikethrough   = 0 'False
    EndProperty
    Height             = 375
    Left               = 180
    TabIndex           = 114
    Top                = 180
    Width              = 975
End
Begin VB.Label Label23
    Appearance         = 0 'フラット
    BackColor          = &H80000005&
    Caption            = "Label23"
    ForeColor          = &H80000008&
    Height             = 195
    Left               = 6840
    TabIndex           = 113
    Top                = 180
    Visible            = 0 'False
    Width              = 1155
End
Begin VB.Label txtChName
    Caption            = "CH3"
    Height             = 375
    Index              = 3
    Left               = 6000
    TabIndex           = 112
    Top                = 180
    Width              = 1215
End
Begin VB.Label txtChName
    Caption            = "CH3"
    Height             = 375
    Index              = 2
    Left               = 4920
    TabIndex           = 111
    Top                = 180
    Width              = 1215
End
Begin VB.Label txtChName
    Caption            = "CH2"
    Height             = 375
    Index              = 1
    Left               = 3840
    TabIndex           = 110
    Top                = 180
    Width              = 1215
End
Begin VB.Label txtChName
    Caption            = "CH1"
    Height             = 375
    Index              = 0
    Left               = 2760
    TabIndex           = 100
    Top                = 180
    Width              = 1215
End
Begin VB.Label Label22
    Caption            = "時間率評価時間[分] 10"
    Height             = 255
    Left               = 7200
    TabIndex           = 99

```

```

Top           = 720
Width        = 1935
End
Begin VB.Label Label21
Caption      = "レベル瞬時値   La, Lv [dB]"
Height      = 375
Left        = 1440
TabIndex    = 98
Top         = 180
Width       = 1215
End
Begin VB.Label Label5
Alignment   = 1 ' 右揃え
BackColor   = &H00FFFFFF&
BorderStyle = 1 ' 実線
Caption     = "0.0"
Height     = 255
Index      = 7
Left       = 6000
TabIndex   = 97
Top        = 600
Width     = 735
End
Begin VB.Label Label5
Alignment   = 1 ' 右揃え
BackColor   = &H00FFFFFF&
BorderStyle = 1 ' 実線
Caption     = "0.0"
Height     = 255
Index      = 6
Left       = 4920
TabIndex   = 96
Top        = 600
Width     = 735
End
Begin VB.Label Label5
Alignment   = 1 ' 右揃え
BackColor   = &H00FFFFFF&
BorderStyle = 1 ' 実線
Caption     = "0.0"
Height     = 255
Index      = 5
Left       = 3840
TabIndex   = 95
Top        = 600
Width     = 735
End
Begin VB.Label Label5
Alignment   = 1 ' 右揃え
BackColor   = &H00FFFFFF&
BorderStyle = 1 ' 実線
Caption     = "0.0"
Height     = 255
Index      = 4
Left       = 2700
TabIndex   = 94
Top        = 600
Width     = 735
End
Begin VB.Label Label20
Caption     = "Label20"
Height     = 195

```

```

Left = 8520
TabIndex = 93
Top = 660
Visible = 0 'False
Width = 675
End
Begin VB.Label Label18
Caption = "Label18"
Height = 195
Left = 8520
TabIndex = 92
Top = 240
Visible = 0 'False
Width = 675
End
Begin VB.Label Label8
Caption = "時間率レベル L a 5, L v 10[dB]"
Height = 495
Left = 1320
TabIndex = 79
Top = 600
Width = 1215
End
Begin VB.Label Label7
Caption = "**"
Height = 255
Left = 7920
TabIndex = 78
Top = 180
Visible = 0 'False
Width = 375
End
Begin VB.Label Label6
Caption = "グラフ表示ビ ッチ[分] 1"
Height = 255
Left = 7200
TabIndex = 77
Top = 360
Width = 1875
End
Begin VB.Label Label5
Alignment = 1 '右揃え
BackColor = &H00FFFFFF&
BorderStyle = 1 '実線
Caption = "Label5"
Height = 255
Index = 3
Left = 6000
TabIndex = 76
Top = 840
Width = 735
End
Begin VB.Label Label5
Alignment = 1 '右揃え
BackColor = &H00FFFFFF&
BorderStyle = 1 '実線
Caption = "Label5"
Height = 255
Index = 2
Left = 4920
TabIndex = 75
Top = 840

```

```

Begin VB.Frame frmTab
    Height      = 6255
    Index       = 0
    Left        = -74880
    TabIndex    = 6
    Top         = 360
    Width       = 7335
    Begin MSComctlLib.ListView lsvMeasList
        Height      = 5895
        Left        = 120
        TabIndex    = 12
        Top         = 240
        Width       = 7095
        _ExtentX    = 12515
        _ExtentY    = 10398
        View        = 3
        LabelWrap   = -1 'True
        HideSelection = -1 'True
        FullRowSelect = -1 'True
        GridLines   = -1 'True
        _Version    = 393217
        ForeColor   = -2147483640
        BackColor   = -2147483643
        BorderStyle = 1
        Appearance  = 1
        NumItems    = 3
        BeginProperty ColumnHeader (1) {BDD1F052-858B-11D1-B16A-00C0F0283628}
            Key      = "FILENAME"
            Text     = "データファイル名"
            Object.Width = 3528
        EndProperty
        BeginProperty ColumnHeader (2) {BDD1F052-858B-11D1-B16A-00C0F0283628}
            SubItemIndex = 1
            Key          = "TRGTIME"
            Text         = "トリガー時間"
        EndProperty
    End
End

```

```

Width      = 735
End
Begin VB.Label Label5
    Alignment = 1 '右揃え
    BackColor = &H00FFFFFF&
    BorderStyle = 1 '実線
    Caption    = "Label5"
    Height     = 255
    Index      = 1
    Left       = 3840
    TabIndex   = 74
    Top        = 840
    Width      = 735
End
Begin VB.Label Label5
    Alignment = 1 '右揃え
    BackColor = &H00FFFFFF&
    BorderStyle = 1 '実線
    Caption    = "Label5"
    Height     = 255
    Index      = 0
    Left       = 2700
    TabIndex   = 73
    Top        = 840
    Width      = 735
End
End
Begin VB.Frame frmTab
    Height      = 6255
    Index       = 0
    Left        = -74880
    TabIndex    = 6
    Top         = 660
    Width       = 7335
    Begin MSComctlLib.ListView lsvMeasList
        Height      = 5895
        Left        = 120
        TabIndex    = 12
        Top         = 240
        Width       = 7095
        _ExtentX    = 12515
        _ExtentY    = 10398
        View        = 3
        LabelWrap   = -1 'True
        HideSelection = -1 'True
        FullRowSelect = -1 'True
        GridLines   = -1 'True
        _Version    = 393217
        ForeColor   = -2147483640
        BackColor   = -2147483643
        BorderStyle = 1
        Appearance  = 1
        NumItems    = 3
        BeginProperty ColumnHeader (1) {BDD1F052-858B-11D1-B16A-00C0F0283628}
            Key      = "FILENAME"
            Text     = "データファイル名"
            Object.Width = 3528
        EndProperty
        BeginProperty ColumnHeader (2) {BDD1F052-858B-11D1-B16A-00C0F0283628}
            SubItemIndex = 1
            Key          = "TRGTIME"
            Text         = "トリガー時間"
        EndProperty
    End
End

```

```

Object.Width = 7056
EndProperty
BeginProperty ColumnHeader (3) {BDD1F052-858B-11D1-B16A-00C0F0283628}
SubItemIndex = 2
Key = "STATUS"
Text = "ステータス"
Object.Width = 1764
EndProperty
End
End
Begin VB.Frame frmTab
Height = 6255
Index = 2
Left = 120
TabIndex = 5
Top = 360
Width = 7335
Begin VB.Frame Frame1
Caption = "監視時間"
Height = 2835
Left = 1920
TabIndex = 46
Top = 2520
Width = 5055
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 23
Left = 3570
TabIndex = 70
Top = 2445
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 22
Left = 3570
TabIndex = 69
Top = 2130
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 21
Left = 3570
TabIndex = 68
Top = 1815
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 20
Left = 3570
TabIndex = 67
Top = 1500
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"

```

```

Object.Width = 7056
EndProperty
BeginProperty ColumnHeader (3) {BDD1F052-858B-11D1-B16A-00C0F0283628}
SubItemIndex = 2
Key = "STATUS"
Text = "ステータス"
Object.Width = 1764
EndProperty
End
End
Begin VB.Frame frmTab
Height = 5655
Index = 2
Left = -74880
TabIndex = 5
Top = 660
Width = 7335
Begin VB.Frame Frame1
Caption = "監視時間"
Height = 2835
Left = 1920
TabIndex = 46
Top = 2520
Width = 5055
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 23
Left = 3570
TabIndex = 70
Top = 2445
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 22
Left = 3570
TabIndex = 69
Top = 2130
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 21
Left = 3570
TabIndex = 68
Top = 1815
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"
Height = 315
Index = 20
Left = 3570
TabIndex = 67
Top = 1500
Width = 1275
End
Begin VB.CheckBox chkWatchTime
Caption = "00:00~01:59"

```

```

Height      = 315
Index       = 19
Left        = 3570
TabIndex    = 66
Top         = 1185
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 18
Left       = 3570
TabIndex   = 65
Top        = 870
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 17
Left       = 3570
TabIndex   = 64
Top        = 555
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 16
Left       = 3570
TabIndex   = 63
Top        = 240
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 15
Left       = 1875
TabIndex   = 62
Top        = 2445
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 14
Left       = 1875
TabIndex   = 61
Top        = 2130
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 13
Left       = 1875
TabIndex   = 60
Top        = 1815
Width      = 1275
End
Begin VB.CheckBox chkWatchTime

```

```

Height      = 315
Index       = 19
Left        = 3570
TabIndex    = 66
Top         = 1185
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 18
Left       = 3570
TabIndex   = 65
Top        = 870
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 17
Left       = 3570
TabIndex   = 64
Top        = 555
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 16
Left       = 3570
TabIndex   = 63
Top        = 240
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 15
Left       = 1875
TabIndex   = 62
Top        = 2445
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 14
Left       = 1875
TabIndex   = 61
Top        = 2130
Width      = 1275
End
Begin VB.CheckBox chkWatchTime
Caption     = "00:00~01:59"
Height     = 315
Index      = 13
Left       = 1875
TabIndex   = 60
Top        = 1815
Width      = 1275
End
Begin VB.CheckBox chkWatchTime

```

```

Caption      = "00:00~01:59"
Height      = 315
Index       = 12
Left        = 1875
TabIndex    = 59
Top         = 1500
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 11
Left        = 1875
TabIndex    = 58
Top         = 1185
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 10
Left        = 1875
TabIndex    = 57
Top         = 870
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 9
Left        = 1875
TabIndex    = 56
Top         = 555
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 8
Left        = 1875
TabIndex    = 55
Top         = 240
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 7
Left        = 180
TabIndex    = 54
Top         = 2445
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 6
Left        = 180
TabIndex    = 53
Top         = 2130
Width       = 1275
End

```

```

Caption      = "00:00~01:59"
Height      = 315
Index       = 12
Left        = 1875
TabIndex    = 59
Top         = 1500
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 11
Left        = 1875
TabIndex    = 58
Top         = 1185
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 10
Left        = 1875
TabIndex    = 57
Top         = 870
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 9
Left        = 1875
TabIndex    = 56
Top         = 555
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 8
Left        = 1875
TabIndex    = 55
Top         = 240
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 7
Left        = 180
TabIndex    = 54
Top         = 2445
Width       = 1275
End
Begin VB.CheckBox chkWatchTime
Caption      = "00:00~01:59"
Height      = 315
Index       = 6
Left        = 180
TabIndex    = 53
Top         = 2130
Width       = 1275
End

```

```

Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 5
    Left         = 180
    TabIndex     = 52
    Top          = 1815
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 4
    Left         = 180
    TabIndex     = 51
    Top          = 1500
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 3
    Left         = 180
    TabIndex     = 50
    Top          = 1185
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 2
    Left         = 180
    TabIndex     = 49
    Top          = 870
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 1
    Left         = 180
    TabIndex     = 48
    Top          = 555
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 0
    Left         = 180
    TabIndex     = 47
    Top          = 240
    Width        = 1275
End
End
Begin MSComctlLib.ProgressBar prbAdData
    Height       = 200
    Index        = 0
    Left         = 2880
    TabIndex     = 36
    Top          = 240
    Width        = 4335

```

```

Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 5
    Left         = 180
    TabIndex     = 52
    Top          = 1815
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 4
    Left         = 180
    TabIndex     = 51
    Top          = 1500
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 3
    Left         = 180
    TabIndex     = 50
    Top          = 1185
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 2
    Left         = 180
    TabIndex     = 49
    Top          = 870
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 1
    Left         = 180
    TabIndex     = 48
    Top          = 555
    Width        = 1275
End
Begin VB.CheckBox chkWatchTime
    Caption       = "00:00~01:59"
    Height       = 315
    Index        = 0
    Left         = 180
    TabIndex     = 47
    Top          = 240
    Width        = 1275
End
End
Begin MSComctlLib.ProgressBar prbAdData
    Height       = 200
    Index        = 0
    Left         = 2880
    TabIndex     = 36
    Top          = 240
    Width        = 4335

```

```

_ExtentX      = 7646
_ExtentY      = 344
_Version      = 393216
Appearance    = 0
Max           = 65535
Scrolling     = 1
End
Begin VB.CommandButton LoadProperty
Caption       = "フロパティ読み込み"
Height       = 450
Left         = 240
TabIndex     = 23
Top          = 4080
Width        = 1485
End
Begin VB.CommandButton SaveProperty
Caption       = "フロパティ保存"
Height       = 450
Left         = 240
TabIndex     = 22
Top          = 3600
Width        = 1485
End
Begin VB.CommandButton AD_Close
Caption       = "終了処理"
Height       = 450
Left         = 240
TabIndex     = 21
Top          = 3120
Width        = 1485
End
Begin VB.CommandButton GetAIStatus
Caption       = "動作ステータス"
Height       = 450
Left         = 240
TabIndex     = 20
Top          = 2640
Width        = 1485
End
Begin VB.CommandButton StopAI
Caption       = "変換停止"
Height       = 450
Left         = 240
TabIndex     = 19
Top          = 2160
Width        = 1485
End
Begin VB.CommandButton StartAI
Caption       = "変換開始"
Height       = 450
Left         = 240
TabIndex     = 18
Top          = 1680
Width        = 1485
End
Begin VB.CommandButton Open
Caption       = "初期化"
Height       = 450
Left         = 240
TabIndex     = 17
Top          = 1200
Width        = 1485

```

```

_ExtentX      = 7646
_ExtentY      = 344
_Version      = 393216
Appearance    = 0
Max           = 65535
Scrolling     = 1
End
Begin VB.CommandButton LoadProperty
Caption       = "フロパティ読み込み"
Height       = 450
Left         = 240
TabIndex     = 23
Top          = 4080
Width        = 1485
End
Begin VB.CommandButton SaveProperty
Caption       = "フロパティ保存"
Height       = 450
Left         = 240
TabIndex     = 22
Top          = 3600
Width        = 1485
End
Begin VB.CommandButton AD_Close
Caption       = "終了処理"
Height       = 450
Left         = 240
TabIndex     = 21
Top          = 3120
Width        = 1485
End
Begin VB.CommandButton GetAIStatus
Caption       = "動作ステータス"
Height       = 450
Left         = 240
TabIndex     = 20
Top          = 2640
Width        = 1485
End
Begin VB.CommandButton StopAI
Caption       = "変換停止"
Height       = 450
Left         = 240
TabIndex     = 19
Top          = 2160
Width        = 1485
End
Begin VB.CommandButton StartAI
Caption       = "変換開始"
Height       = 450
Left         = 240
TabIndex     = 18
Top          = 1680
Width        = 1485
End
Begin VB.CommandButton Open
Caption       = "初期化"
Height       = 450
Left         = 240
TabIndex     = 17
Top          = 1200
Width        = 1485

```

```

End
Begin VB.CommandButton cmdSetUp
Caption = "設定"
BeginProperty Font
Name = "MS ゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 450
Left = 240
TabIndex = 16
Top = 720
Width = 1485
End
Begin VB.CommandButton ShowProperty
Caption = "プロパティ表示"
BeginProperty Font
Name = "MS ゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 450
Left = 240
TabIndex = 15
Top = 240
Width = 1485
End
Begin MSComctlLib.ProgressBar prbAdData
Height = 195
Index = 1
Left = 2880
TabIndex = 37
Top = 420
Width = 4335
_ExtentX = 7646
_ExtentY = 344
_Version = 393216
Appearance = 0
Max = 65535
Scrolling = 1
End
Begin MSComctlLib.ProgressBar prbAdData
Height = 195
Index = 2
Left = 2880
TabIndex = 38
Top = 600
Width = 4335
_ExtentX = 7646
_ExtentY = 344
_Version = 393216
Appearance = 0
Max = 65535
Scrolling = 1

```

```

End
Begin VB.CommandButton cmdSetUp
Caption = "設定"
BeginProperty Font
Name = "MS ゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 450
Left = 240
TabIndex = 16
Top = 720
Width = 1485
End
Begin VB.CommandButton ShowProperty
Caption = "プロパティ表示"
BeginProperty Font
Name = "MS ゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 450
Left = 240
TabIndex = 15
Top = 240
Width = 1485
End
Begin MSComctlLib.ProgressBar prbAdData
Height = 195
Index = 1
Left = 2880
TabIndex = 37
Top = 420
Width = 4335
_ExtentX = 7646
_ExtentY = 344
_Version = 393216
Appearance = 0
Max = 65535
Scrolling = 1
End
Begin MSComctlLib.ProgressBar prbAdData
Height = 195
Index = 2
Left = 2880
TabIndex = 38
Top = 600
Width = 4335
_ExtentX = 7646
_ExtentY = 344
_Version = 393216
Appearance = 0
Max = 65535
Scrolling = 1

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 3
    Left = 2880
    TabIndex = 39
    Top = 780
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 4
    Left = 2880
    TabIndex = 40
    Top = 960
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 5
    Left = 2880
    TabIndex = 41
    Top = 1140
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```
End
```

```

Begin VB.Label lblAdData
    AutoSize = -1 'True
    Caption = "-99.9999V"

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 3
    Left = 2880
    TabIndex = 39
    Top = 780
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 4
    Left = 2880
    TabIndex = 40
    Top = 960
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```

End
Begin MSComctlLib.ProgressBar prbAdData
    Height = 195
    Index = 5
    Left = 2880
    TabIndex = 41
    Top = 1140
    Width = 4335
    _ExtentX = 7646
    _ExtentY = 344
    _Version = 393216
    Appearance = 0
    Max = 65535
    Scrolling = 1

```

```
End
```

```

Begin VB.Label txtOpenFileNum
    Caption = "Label20"
    Height = 255
    Left = 5760
    TabIndex = 91
    Top = 5400
    Width = 1215

```

```

End
Begin VB.Label Label19
    Caption = "Openファイル個数"
    Height = 255
    Left = 3840
    TabIndex = 90
    Top = 5400
    Width = 1575

```

```
End
```

```

Begin VB.Label lblAdData
    AutoSize = -1 'True
    Caption = "-99.9999V"

```

```

BeginProperty Font
  Name      = "MS ゴシック"
  Size      = 9
  Charset   = 128
  Weight    = 400
  Underline = 0 'False
  Italic    = 0 'False
  Strikethrough = 0 'False
EndProperty
Height      = 180
Index      = 5
Left       = 1920
TabIndex   = 35
Top        = 1140
Width      = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height    = 180
  Index     = 4
  Left     = 1920
 TabIndex  = 34
  Top      = 960
  Width    = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height    = 180
  Index     = 3
  Left     = 1920
 TabIndex  = 33
  Top      = 780
  Width    = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
  EndProperty

```

```

BeginProperty Font
  Name      = "MS ゴシック"
  Size      = 9
  Charset   = 128
  Weight    = 400
  Underline = 0 'False
  Italic    = 0 'False
  Strikethrough = 0 'False
EndProperty
Height      = 180
Index      = 5
Left       = 1920
TabIndex   = 35
Top        = 1140
Width      = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height    = 180
  Index     = 4
  Left     = 1920
 TabIndex  = 34
  Top      = 960
  Width    = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height    = 180
  Index     = 3
  Left     = 1920
 TabIndex  = 33
  Top      = 780
  Width    = 810
End
Begin VB.Label lblAdData
  AutoSize  = -1 'True
  Caption   = "-99.9999V"
  BeginProperty Font
    Name      = "MS ゴシック"
    Size      = 9
    Charset   = 128
    Weight    = 400
  EndProperty

```

```

Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 2
Left = 1920
TabIndex = 32
Top = 600
Width = 810
End
Begin VB.Label lblAdData
AutoSize = -1 'True
Caption = "-99.9999V"
BeginProperty Font
Name = "MS ゴシック"
Size = 9
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 1
Left = 1920
TabIndex = 31
Top = 420
Width = 810
End
Begin VB.Label lblAdData
AutoSize = -1 'True
Caption = "-99.9999V"
BeginProperty Font
Name = "MS ゴシック"
Size = 9
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 0
Left = 1920
TabIndex = 30
Top = 240
Width = 810
End
Begin VB.Label dspData
BorderStyle = 1 '実線
Height = 330
Left = 1920
TabIndex = 26
Top = 1380
Width = 5295
End
Begin VB.Label Info
BorderStyle = 1 '実線
Height = 330
Index = 0
Left = 1920

```

```

Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 2
Left = 1920
TabIndex = 32
Top = 600
Width = 810
End
Begin VB.Label lblAdData
AutoSize = -1 'True
Caption = "-99.9999V"
BeginProperty Font
Name = "MS ゴシック"
Size = 9
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 1
Left = 1920
TabIndex = 31
Top = 420
Width = 810
End
Begin VB.Label lblAdData
AutoSize = -1 'True
Caption = "-99.9999V"
BeginProperty Font
Name = "MS ゴシック"
Size = 9
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
EndProperty
Height = 180
Index = 0
Left = 1920
TabIndex = 30
Top = 240
Width = 810
End
Begin VB.Label dspData
BorderStyle = 1 '実線
Height = 330
Left = 1920
TabIndex = 26
Top = 1380
Width = 5295
End
Begin VB.Label Info
BorderStyle = 1 '実線
Height = 330
Index = 0
Left = 1920

```

```

TabIndex      = 25
Top           = 1740
Width        = 5295
End
Begin VB.Label Info
BorderStyle   = 1 '実線
Height       = 330
Index        = 1
Left         = 1920
TabIndex     = 24
Top          = 2100
Width        = 5295

```

End

End

Begin VB.Frame frmTab

```

Height       = 6255
Index        = 1
Left         = -74880
TabIndex     = 4
Top          = 360
Width        = 7335

```

Begin VB.CommandButton cmdLogClear

```

Caption      = "クリアー"
Height      = 375
Left        = 120
TabIndex    = 10
Top         = 5760
Width       = 7095

```

End

Begin MSComctlLib.ListView lsvActionLog

```

Height      = 5415
Left        = 60
TabIndex    = 11
Top         = 180
Width       = 7095
_ExtentX   = 12515
_ExtentY   = 9551
View        = 3
LabelWrap   = -1 'True
HideSelection = -1 'True
FullRowSelect = -1 'True
GridLines   = -1 'True
_Version    = 393217
ForeColor   = -2147483640
BackColor   = -2147483643
BorderStyle = 1
Appearance  = 1

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

```

Name        = "MS ゴシック"
Size        = 9
Charset     = 128
Weight      = 400
Underline   = 0 'False
Italic      = 0 'False
Strikethrough = 0 'False

```

EndProperty

NumItems = 2

BeginProperty ColumnHeader (1) {BDD1F052-858B-11D1-B16A-00C0F0283628}

```

Key         = "TIME"
Text        = "時刻"
Object.Width = 3175

```

EndProperty

```

TabIndex      = 25
Top           = 1740
Width        = 5295
End
Begin VB.Label Info
BorderStyle   = 1 '実線
Height       = 330
Index        = 1
Left         = 1920
TabIndex     = 24
Top          = 2100
Width        = 5295

```

End

End

Begin VB.Frame frmTab

```

Height       = 6255
Index        = 1
Left         = -74880
TabIndex     = 4
Top          = 660
Width        = 7335

```

Begin VB.CommandButton cmdLogClear

```

Caption      = "クリアー"
Height      = 375
Left        = 120
TabIndex    = 10
Top         = 5760
Width       = 7095

```

End

Begin MSComctlLib.ListView lsvActionLog

```

Height      = 5415
Left        = 60
TabIndex    = 11
Top         = 180
Width       = 7095
_ExtentX   = 12515
_ExtentY   = 9551
View        = 3
LabelWrap   = -1 'True
HideSelection = -1 'True
FullRowSelect = -1 'True
GridLines   = -1 'True
_Version    = 393217
ForeColor   = -2147483640
BackColor   = -2147483643
BorderStyle = 1
Appearance  = 1

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

```

Name        = "MS ゴシック"
Size        = 9
Charset     = 128
Weight      = 400
Underline   = 0 'False
Italic      = 0 'False
Strikethrough = 0 'False

```

EndProperty

NumItems = 2

BeginProperty ColumnHeader (1) {BDD1F052-858B-11D1-B16A-00C0F0283628}

```

Key         = "TIME"
Text        = "時刻"
Object.Width = 3175

```

EndProperty

```

BeginProperty ColumnHeader (2) {BDD1F052-858B-11D1-B16A-00C0F0283628}
  SubItemIndex = 1
  Key = "ACTION"
  Text = "内容"
  Object.Width = 8819
EndProperty
End
End
End
Begin MSComctlLib.Toolbar tbToolBar
  Align = 1 '上揃え'
  Height = 420
  Left = 0
  TabIndex = 1
  Top = 0
  Width = 12105
  ExtentX = 21352
  ExtentY = 741
  ButtonWidth = 609
  ButtonHeight = 582
  Appearance = 1
  ImageList = "imlToolBarIcons"
  _Version = 393216
BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
  NumButtons = 4
  BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Style = 3
  EndProperty
  BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Key = "フ ロハ ティ"
    Object.ToolTipText = "フ ロハ ティ"
    ImageKey = "Properties"
  EndProperty
  BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Style = 3
  EndProperty
  BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Key = "ヘルプ"
    Object.ToolTipText = "ヘルプ"
    ImageKey = "Help"
  EndProperty
EndProperty
End
Begin MSComctlLib.StatusBar sbStatusBar
  Align = 2 '下揃え'
  Height = 270
  Left = 0
  TabIndex = 0
  Top = 7590
  Width = 12105
  ExtentX = 21352
  ExtentY = 476
  _Version = 393216
BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}
  NumPanels = 3
  BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}
    AutoSize = 1
    Object.Width = 15716
    Text = "ステータス"
    TextSave = "ステータス"
  EndProperty
  BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-00C0F0283628}

```

```

BeginProperty ColumnHeader (2) {BDD1F052-858B-11D1-B16A-00C0F0283628}
  SubItemIndex = 1
  Key = "ACTION"
  Text = "内容"
  Object.Width = 8819
EndProperty
End
End
End
Begin MSComctlLib.Toolbar tbToolBar
  Align = 1 '上揃え'
  Height = 420
  Left = 0
  TabIndex = 1
  Top = 0
  Width = 14070
  ExtentX = 24818
  ExtentY = 741
  ButtonWidth = 609
  ButtonHeight = 582
  Appearance = 1
  ImageList = "imlToolBarIcons"
  _Version = 393216
BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
  NumButtons = 4
  BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Style = 3
  EndProperty
  BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Key = "フ ロハ ティ"
    Object.ToolTipText = "フ ロハ ティ"
    ImageKey = "Properties"
  EndProperty
  BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Style = 3
  EndProperty
  BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
    Key = "ヘルプ"
    Object.ToolTipText = "ヘルプ"
    ImageKey = "Help"
  EndProperty
EndProperty
End
Begin MSComctlLib.StatusBar sbStatusBar
  Align = 2 '下揃え'
  Height = 270
  Left = 0
  TabIndex = 0
  Top = 8430
  Width = 14070
  ExtentX = 24818
  ExtentY = 476
  _Version = 393216
BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}
  NumPanels = 3
  BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}
    AutoSize = 1
    Object.Width = 19156
    Text = "ステータス"
    TextSave = "ステータス"
  EndProperty
  BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-00C0F0283628}

```

```

Style = 6
AutoSize = 2
TextSave = "2003/07/07"
EndProperty
BeginProperty Panel3 {8E3867AB-8586-11D1-B16A-00C0F0283628}
Style = 5
AutoSize = 2
TextSave = "0:13"
EndProperty
EndProperty
End
Begin MSComDlg.CommonDialog dlgCommonDialog
Left = 120
Top = 2520
_ExtentX = 847
_ExtentY = 847
_Version = 393216
End
Begin MSComctlLib.ImageList imlToolbarIcons
Left = 600
Top = 2520
_ExtentX = 1005
_ExtentY = 1005
BackColor = -2147483643
ImageWidth = 16
ImageHeight = 16
MaskColor = 12632256
_Version = 393216
BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
NumListImages = 2
BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture = "frmMain.frx":0ED3
Key = "Properties"
EndProperty
BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture = "frmMain.frx":0FE5
Key = "Help"
EndProperty
EndProperty
End
Begin CZCalendarLib.ZCalendar ZCalendar1
Height = 2655
Left = 0
TabIndex = 7
Top = 480
Width = 2535
_Version = 65536
_ExtentX = 4471
_ExtentY = 4683
_StockProps = 65
BackColor = 65535
CurrentDateColor = 32768
ShowYear = -1 'True
ShowMonth = -1 'True
ShowHorizontalLine = -1 'True
ShowVerticalLine = -1 'True
ReverseCurrentDate = -1 'True
BeginProperty TitleFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 400

```

```

Style = 6
AutoSize = 2
TextSave = "2013/04/25"
EndProperty
BeginProperty Panel3 {8E3867AB-8586-11D1-B16A-00C0F0283628}
Style = 5
AutoSize = 2
TextSave = "10:47"
EndProperty
EndProperty
End
Begin MSComDlg.CommonDialog dlgCommonDialog
Left = 120
Top = 2520
_ExtentX = 847
_ExtentY = 847
_Version = 393216
End
Begin MSComctlLib.ImageList imlToolbarIcons
Left = 600
Top = 2520
_ExtentX = 1005
_ExtentY = 1005
BackColor = -2147483643
ImageWidth = 16
ImageHeight = 16
MaskColor = 12632256
_Version = 393216
BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
NumListImages = 2
BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture = "frmMain.frx":3E77
Key = "Properties"
EndProperty
BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
Picture = "frmMain.frx":3F89
Key = "Help"
EndProperty
EndProperty
End
Begin CZCalendarLib.ZCalendar ZCalendar1
Height = 2295
Left = 0
TabIndex = 7
Top = 480
Width = 2535
_Version = 65536
_ExtentX = 4471
_ExtentY = 4048
_StockProps = 65
BackColor = 65535
CurrentDateColor = 32768
ShowYear = -1 'True
ShowMonth = -1 'True
ShowHorizontalLine = -1 'True
ShowVerticalLine = -1 'True
ReverseCurrentDate = -1 'True
BeginProperty TitleFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 400

```

```
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
BeginProperty WeekTitleFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
```

```
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
BeginProperty DayFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
```

```
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
CellEffect = 1
```

```
End
```

```
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
BeginProperty WeekTitleFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
```

```
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 400
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
BeginProperty DayFont {0BE35203-8F91-11CE-9DE3-00AA004BB851}
```

```
Name = "MS ゴシック"
Size = 12
Charset = 128
Weight = 700
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
CellEffect = 1
```

```
End
```

```
Begin VB.Label Label17
```

```
Caption = "DataDir CopyDir"
Height = 615
Left = 120
TabIndex = 89
Top = 4920
Width = 615
```

```
End
```

```
Begin VB.Label Label16
```

```
BorderStyle = 1 '実線
Caption = "Label16"
Height = 285
Left = 840
TabIndex = 88
Top = 5280
Width = 3495
```

```
End
```

```
Begin VB.Label Label15
```

```
BorderStyle = 1 '実線
Caption = "Label15"
Height = 255
Left = 840
TabIndex = 87
Top = 4920
Width = 3495
```

```
End
```

```
Begin VB.Label Label14
```

```
BorderStyle = 1 '実線
Caption = "Label14"
Height = 255
Left = 2400
TabIndex = 86
Top = 4200
Width = 735
```

```
End
```

```
Begin VB.Label Label13
```

```
Caption = "収録時間(フリリカ含む)[sec]"
Height = 195
```

```

Begin VB.Label Label4
  AutoSize      = -1 'True
  Caption       = "ステータス"
  BeginProperty Font
    Name        = "MS Pゴシック"
    Size        = 12
    Charset     = 128
    Weight      = 700
    Underline   = 0 'False
    Italic      = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height        = 240
  Left          = 120
  TabIndex     = 3
  Top          = 4860
  Width        = 1110
End
Begin VB.Label Label3
  AutoSize      = -1 'True
  Caption       = "AD Status"
  Height        = 180
  Left          = 180

```

```

Left          = 180
TabIndex     = 85
Top          = 4260
Width        = 2175
End
Begin VB.Label Label12
  BorderStyle   = 1 '実線
  Caption       = "Label12"
  Height        = 255
  Left          = 3600
  TabIndex     = 84
  Top          = 3960
  Width        = 735
End
Begin VB.Label Label11
  Caption       = "PreTrigger[sec]"
  Height        = 195
  Left          = 2220
  TabIndex     = 83
  Top          = 3960
  Width        = 1215
End
Begin VB.Label Label10
  BorderStyle   = 1 '実線
  Caption       = "Label10"
  Height        = 255
  Left          = 1260
  TabIndex     = 82
  Top          = 3960
  Width        = 735
End
Begin VB.Label Label9
  Caption       = "Sampling time[sec]"
  Height        = 375
  Left          = 180
  TabIndex     = 81
  Top          = 3840
  Width        = 1035
End
Begin VB.Label Label4
  AutoSize      = -1 'True
  Caption       = "ステータス"
  BeginProperty Font
    Name        = "MS Pゴシック"
    Size        = 12
    Charset     = 128
    Weight      = 700
    Underline   = 0 'False
    Italic      = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height        = 240
  Left          = 120
  TabIndex     = 3
  Top          = 5520
  Width        = 1110
End
Begin VB.Label Label3
  AutoSize      = -1 'True
  Caption       = "AD Status"
  Height        = 180
  Left          = 180

```

```

TabIndex      = 42
Top           = 3840
Width        = 795
End
Begin VB.Label Label1
    AutoSize   = -1 'True
    Caption    = "AD board buffer"
    Height     = 180
    Left       = 180
    TabIndex   = 29
Top           = 4080
Width        = 1230
End
Begin VB.Label Label2
    AutoSize   = -1 'True
    Caption    = "動作モード"
BeginProperty Font
    Name       = "MS Pゴシック"
    Size       = 14.25
    Charset    = 128
    Weight     = 700
    Underline  = 0 'False
    Italic     = 0 'False
    Strikethrough = 0 'False
EndProperty
    Height     = 285
    Left       = 120
    TabIndex   = 13
Top           = 3420
Width        = 1365
End
Begin VB.Menu mnuFile
    Caption    = "ファイル(&F)"
Begin VB.Menu mnuFileProperties
    Caption    = "プロパティ(&I)"
End
Begin VB.Menu mnuFileBar0
    Caption    = "-"
End
Begin VB.Menu mnuFileExit
    Caption    = "終了(&X)"
End
End
Begin VB.Menu mnuSetting
    Caption    = "設定(&S)"
End
Begin VB.Menu mnuView
    Caption    = "表示(&V)"
Begin VB.Menu mnuViewToolBar
    Caption    = "ツールバー(&T)"
    Checked    = -1 'True
End
Begin VB.Menu mnuViewStatusBar
    Caption    = "ステータスバー(&B)"
    Checked    = -1 'True
End
End
Begin VB.Menu mnuHelp
    Caption    = "ヘルプ(&H)"
Begin VB.Menu mnuHelpAbout
    Caption    = "バージョン情報(&A)"
End
End

```

```

TabIndex      = 42
Top           = 3360
Width        = 795
End
Begin VB.Label Label1
    AutoSize   = -1 'True
    Caption    = "AD board buffer"
    Height     = 180
    Left       = 180
    TabIndex   = 29
Top           = 3600
Width        = 1230
End
Begin VB.Label Label2
    AutoSize   = -1 'True
    Caption    = "動作モード"
BeginProperty Font
    Name       = "MS Pゴシック"
    Size       = 14.25
    Charset    = 128
    Weight     = 700
    Underline  = 0 'False
    Italic     = 0 'False
    Strikethrough = 0 'False
EndProperty
    Height     = 285
    Left       = 120
    TabIndex   = 13
Top           = 2940
Width        = 1365
End
Begin VB.Menu mnuFile
    Caption    = "ファイル(&F)"
Begin VB.Menu mnuFileProperties
    Caption    = "プロパティ(&I)"
End
Begin VB.Menu mnuFileBar0
    Caption    = "-"
End
Begin VB.Menu mnuFileExit
    Caption    = "終了(&X)"
End
End
Begin VB.Menu mnuSetting
    Caption    = "設定(&S)"
End
Begin VB.Menu mnuView
    Caption    = "表示(&V)"
Begin VB.Menu mnuViewToolBar
    Caption    = "ツールバー(&T)"
    Checked    = -1 'True
End
Begin VB.Menu mnuViewStatusBar
    Caption    = "ステータスバー(&B)"
    Checked    = -1 'True
End
End
Begin VB.Menu mnuHelp
    Caption    = "ヘルプ(&H)"
Begin VB.Menu mnuHelpAbout
    Caption    = "バージョン情報(&A)"
End
End

```

```

End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

```

```

Private m_Repeat&, m_host$
'Dim Ret As Integer
Private Buffer(262144) As Single, DataSec(1000, 5) As Integer

```

```

Public ManualStart As Boolean
Public Timer2_Interval As Integer
Private Sub AD_Close_Click()
Dim Ret As Integer
'コントロールの終了処理
Ret = AcxAio1.Close
End Sub

```

```

Private Sub cmdLogClear_Click()
LogFileClear
End Sub

```

```

Private Sub cmdManualStart_Click()
ManualStart = True
End Sub

```

```

Private Sub cmdRun_Click()
cmdSetUp_Click
Open_Click
StartAI_Click
End Sub

```

```

Private Sub cmdSetUp_Click()
Timer1.Interval = 0
AcxAio1.TriggerMode = 0
AcxAio1.ChannelString = "0-" + Trim(Str(gMaxChNo))
AcxAio1.InputRange = 1 '1:2' sコンプレメントハ イナリ
AcxAio1.UnitType = 1
AcxAio1.UnitFreq = 1
AcxAio1.Frequency = 1

```

```

AcxAio1.ClockSource = 0
AcxAio1.InterruptFactor = 2
AcxAio1.ScanNumber = -1
prbAdBuf.Min = 0
prbAdBuf.Max = 262144 / AcxAio1.ChannelNumber '43690
End Sub

```

```

End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

```

```

Private m_Repeat&, m_host$
'Dim Ret As Integer
Private Buffer(262144) As Single, DataSec(1000, 5) As Integer
Private Buffer(262144) As Single, DataSec(1000, 3) As Integer

```

```

Public MaxDataClrFlg As Boolean '追加 新宅 最大値をクリアするフラグ
Public ManualStart As Boolean
Public Timer2_Interval As Integer
Private Sub AD_Close_Click()
Dim Ret As Integer
'コントロールの終了処理
Ret = AcxAio1.Close
End Sub

```

```

Private Sub cmdLogClear_Click()
LogFileClear
End Sub

```

```

Private Sub cmdManualStart_Click()
ManualStart = True
End Sub

```

```

Private Sub cmdRun_Click()
cmdSetUp_Click
Open_Click
StartAI_Click
End Sub

```

```

Private Sub cmdSetUp_Click()
Timer1.Interval = 0
AcxAio1.TriggerMode = 0
AcxAio1.ChannelString = "0-" + Trim(Str(gMaxChNo))
AcxAio1.InputRange = 1 '1:2' sコンプレメントハ イナリ
AcxAio1.UnitType = 1

```

```

If MstData.SampSec <= 0.001 Then
AcxAio1.UnitFreq = 1
AcxAio1.Frequency = 1 / MstData.SampSec / 1000
Else
AcxAio1.UnitFreq = 0
AcxAio1.Frequency = 1 / MstData.SampSec
End If
SampHz = 1000 ^ AcxAio1.UnitFreq * AcxAio1.Frequency
AcxAio1.ClockSource = 0
AcxAio1.InterruptFactor = 2
AcxAio1.ScanNumber = -1
prbAdBuf.Min = 0
prbAdBuf.Max = 262144 / AcxAio1.ChannelNumber '43690
End Sub

```

```

Private Sub Command1_Click() '警報クリア

```

```

Timer2.Enabled = False
    CBGraph1.Canvas.Color = RGB(255, 255, 255)
    AcxAio1.WriteBitData 0, 0
    KeihoFlg = True
End Sub

Private Sub Command2_Click() '任意月データ読み込み
Dim ii As Long
Dim PlotStart As Long 'データの開始位置
Dim PlotEnd As Long 'データの終わり位置
Dim sOFileName As String
Dim iiCount As Long 'データ個数
Dim tmpDateValue As Single

txtPointNo00 = "測点" & PointNo
Text1.Text = ZCalender1.Year & "/" & ZCalender1.Month & "/" & ZCalender1.Day

' If ZCalender1.Month < 10 Then
'     sOFileName = ZCalender1.Year & "0" & ZCalender1.Month
' Else
'     sOFileName = ZCalender1.Year & ZCalender1.Month
' End If
iiCount = MaxDataLoad0(Text1.Text)

If iiCount = 0 Then Text1.Text = "Uuknow File": Exit Sub
tmpDateValue = DateValue(Text1.Text)
For ii = 0 To iiCount - 1
    If tmpDateValue >= MaxDataX0(ii) Then
        PlotStart = ii
    End If
    If tmpDateValue + 1 > MaxDataX0(ii) Then
        PlotEnd = ii
    End If
Next ii
ii = MaxDataPlot2(PlotStart, PlotEnd)
If ii = 0 Then Text1.Text = "Uuknow Data": Exit Sub

End Sub

Private Sub Command3_Click()
If ZCalender1.Month = 1 Then
    ZCalender1.Year = ZCalender1.Year - 1
    ZCalender1.Month = 12
Else
    ZCalender1.Month = ZCalender1.Month - 1
End If
End Sub

Private Sub Command4_Click()
If ZCalender1.Month = 12 Then
    ZCalender1.Year = ZCalender1.Year + 1
    ZCalender1.Month = 1
Else

```

```

        ZCalender1.Month = ZCalender1.Month + 1
    End If
End Sub

Private Sub Form_Load()
    Dim Ret As Integer, i As Integer
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)

    For i = 0 To 23
        chkWatchTime(i).Caption = Format(i, "00") + ":00~" + Format(i, "00") + ":59"
        chkWatchTime(i).Value = GetSetting(App.Title, "Settings", "WatchTime" + Format(i, "00"), 1)
    Next

    For i = 0 To tabMain.Tabs - 1: frmTab(i).BorderStyle = 0: Next
    ZClock1.ShowSecond = True
    ManualStart = False
    Timer1.Interval = 100

    '設定ファイルの読み込み
    If Dir(AppDir + "¥ACXAIO.INI") <> "" Then
        Ret = AcxAio1.LoadProperty(AppDir + "¥ACXAIO.INI")
    End If
End Sub

cmdRun_Click
SetActiveTabFrame 0
Timer1.Enabled = True

```

```

        ZCalender1.Month = ZCalender1.Month + 1
    End If
End Sub

Private Sub Form_Load()
    Dim Ret As Integer, i As Integer
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)

    For i = 0 To 23
        chkWatchTime(i).Caption = Format(i, "00") + ":00~" + Format(i, "00") + ":59"
        chkWatchTime(i).Value = GetSetting(App.Title, "Settings", "WatchTime" + Format(i, "00"), 1)
    Next

    For i = 0 To tabMain.Tabs - 1: frmTab(i).BorderStyle = 0: Next
    ZClock1.ShowSecond = True
    ManualStart = False
    Timer1.Interval = 100

    '設定ファイルの読み込み
    If Dir(AppDir + "¥ACXAIO.INI") <> "" Then
        Ret = AcxAio1.LoadProperty(AppDir + "¥ACXAIO.INI")
    End If
End Sub

Para_Read
cmdRun_Click
SetActiveTabFrame 3
Timer1.Enabled = True
Timer2.Enabled = False
Timer2.Interval = 1000
Timer3.Enabled = False
Timer3.Interval = 1000

MaxDataClrFlg = False
KeihoFlg1(0) = True
KeihoFlg1(1) = True
KeihoFlg1(2) = True
KeihoFlg1(3) = True
KeihoFlg(0) = True
KeihoFlg(1) = True
KeihoFlg(2) = True
KeihoFlg(3) = True
Label7.Caption = IntvPitch
CBGraph1.Canvas.axes("軸1").startvalue = GraphYScale(0)
CBGraph1.Canvas.axes("軸1").endvalue = 0
CBGraph1.Canvas.axes("軸2").startvalue = GraphYScale(1)
CBGraph1.Canvas.axes("軸2").endvalue = 0
CBGraph1.Canvas.axes("軸3").startvalue = GraphYScale(2)
CBGraph1.Canvas.axes("軸3").endvalue = 0
CBGraph1.Canvas.axes("軸4").startvalue = GraphYScale(3)
CBGraph1.Canvas.axes("軸4").endvalue = 0

CBGraph1.Canvas.axes("軸1").scalecount = GraphYScaleCount(0)
CBGraph1.Canvas.axes("軸2").scalecount = GraphYScaleCount(1)
CBGraph1.Canvas.axes("軸3").scalecount = GraphYScaleCount(2)
CBGraph1.Canvas.axes("軸4").scalecount = GraphYScaleCount(3)
CBGraph1.Canvas.grids("格子1").HorzGridCount = GraphYScaleCount(0) - 1
CBGraph1.Canvas.grids("格子2").HorzGridCount = GraphYScaleCount(1) - 1
CBGraph1.Canvas.grids("格子3").HorzGridCount = GraphYScaleCount(2) - 1
CBGraph1.Canvas.grids("格子4").HorzGridCount = GraphYScaleCount(3) - 1

```

```

CBGraph1.Canvas.labels("ラベル1").Text = ChName(0)
CBGraph1.Canvas.labels("ラベル2").Text = ChName(1)
CBGraph1.Canvas.labels("ラベル3").Text = ChName(2)
CBGraph1.Canvas.labels("ラベル4").Text = ChName(3)

CBGraph2.Canvas.axes("軸1").startvalue = GraphYScale(0)
CBGraph2.Canvas.axes("軸1").endvalue = 0
CBGraph2.Canvas.axes("軸2").startvalue = GraphYScale(1)
CBGraph2.Canvas.axes("軸2").endvalue = 0
CBGraph2.Canvas.axes("軸3").startvalue = GraphYScale(2)
CBGraph2.Canvas.axes("軸3").endvalue = 0
CBGraph2.Canvas.axes("軸4").startvalue = GraphYScale(3)
CBGraph2.Canvas.axes("軸4").endvalue = 0

CBGraph2.Canvas.axes("軸1").scalecount = GraphYScaleCount(0)
CBGraph2.Canvas.axes("軸2").scalecount = GraphYScaleCount(1)
CBGraph2.Canvas.axes("軸3").scalecount = GraphYScaleCount(2)
CBGraph2.Canvas.axes("軸4").scalecount = GraphYScaleCount(3)
CBGraph2.Canvas.grids("格子1").HorzGridCount = GraphYScaleCount(0) - 1
CBGraph2.Canvas.grids("格子2").HorzGridCount = GraphYScaleCount(1) - 1
CBGraph2.Canvas.grids("格子3").HorzGridCount = GraphYScaleCount(2) - 1
CBGraph2.Canvas.grids("格子4").HorzGridCount = GraphYScaleCount(3) - 1
CBGraph2.Canvas.Color = RGB(230, 230, 230)

CBGraph2.Canvas.labels("ラベル1").Text = ChName(0)
CBGraph2.Canvas.labels("ラベル2").Text = ChName(1)
CBGraph2.Canvas.labels("ラベル3").Text = ChName(2)
CBGraph2.Canvas.labels("ラベル4").Text = ChName(3)

' CBGraph2.Canvas.plotdata("データ1").makercolor = RGB(0, 255, 255)

Label15.Caption = MstData.DataDir
Label16.Caption = strMaxDataCopyDir
If intCopyFlg = 0 Then Label16.Caption = ""
If IntLvX = 0 Then
    Label8.Caption = "時間率レベル L a5, L v 10[dB]"
ElseIf IntLvX = 1 Then
    Label8.Caption = "時間率レベル L amax L v max[dB]"
End If
KeihoEmailFlg = False
KeihoEmailFlg1 = False
If KeihoOffDisp = 1 Then
    Label5(0).Visible = False
    Label5(1).Visible = False
    Label5(2).Visible = False
    Label5(3).Visible = False
End If

txtChName(0) = ChName(0)
txtChName(1) = ChName(1)
txtChName(2) = ChName(2)
txtChName(3) = ChName(3)

```

End Sub

```

Private Sub Form_Resize()
    Dim i As Integer, TabIndex
    If Me.WindowState = vbNormal Or Me.WindowState = vbMaximized Then
        TabIndex = tabMain.Tab
        If Me.Width < 12000 Then Me.Width = 12000
        If Me.Height < 7000 Then Me.Height = 7000

        gridStatus.Height = Me.ScaleHeight - gridStatus.Top - sbStatusBar.Height - 0

        tabMain.Top = tbToolBar.Height
        tabMain.Height = Me.Height - (sbStatusBar.Height + tbToolBar.Height + 700)
        tabMain.Left = 4350
        tabMain.Width = Me.Width - 120 - 4350
        frmTab(TabIndex).Top = 360
        frmTab(TabIndex).Height = tabMain.Height - 525
        frmTab(TabIndex).Left = 120
        frmTab(TabIndex).Width = tabMain.Width - 225

        Select Case TabIndex
            Case 0
                lsvMeasList.Width = frmTab(TabIndex).Width - 240
                lsvMeasList.Height = frmTab(TabIndex).Height - 400
            Case 1
                lsvActionLog.Width = frmTab(TabIndex).Width - 240
                lsvActionLog.Height = frmTab(TabIndex).Height - 840
                cmdLogClear.Top = lsvActionLog.Top + lsvActionLog.Height + 105
                cmdLogClear.Width = lsvActionLog.Width
            Case 2
                'close all sub forms
                For i = Forms.Count - 1 To 1 Step -1
                    Unload Forms(i)
                Next
                AppendLog "プログラム終了"

                If Me.WindowState <> vbMinimized Then
                    SaveSetting App.Title, "Settings", "MainLeft", Me.Left
                    SaveSetting App.Title, "Settings", "MainTop", Me.Top
                    SaveSetting App.Title, "Settings", "MainWidth", Me.Width
                End If
            End Select
        End If
    End Sub

```

End Sub

```

Private Sub Form_Resize()
    Dim i As Integer, TabIndex
    If Me.WindowState = vbNormal Or Me.WindowState = vbMaximized Then
        TabIndex = tabMain.Tab
        If Me.Width < 12000 Then Me.Width = 12000
        If Me.Height < 7000 Then Me.Height = 7000

        gridStatus.Height = Me.ScaleHeight - gridStatus.Top - sbStatusBar.Height - 0

        tabMain.Top = tbToolBar.Height
        tabMain.Height = Me.Height - (sbStatusBar.Height + tbToolBar.Height + 700)
        tabMain.Left = 4350
        tabMain.Width = Me.Width - 120 - 4350
        frmTab(TabIndex).Top = 360
        frmTab(TabIndex).Height = tabMain.Height - 525
        frmTab(TabIndex).Left = 120
        frmTab(TabIndex).Width = tabMain.Width - 225

        Select Case TabIndex
            Case 0
                lsvMeasList.Width = frmTab(TabIndex).Width - 240
                lsvMeasList.Height = frmTab(TabIndex).Height - 400
            Case 1
                lsvActionLog.Width = frmTab(TabIndex).Width - 240
                lsvActionLog.Height = frmTab(TabIndex).Height - 840
                cmdLogClear.Top = lsvActionLog.Top + lsvActionLog.Height + 105
                cmdLogClear.Width = lsvActionLog.Width
            Case 2
                'close all sub forms
                For i = Forms.Count - 1 To 1 Step -1
                    Unload Forms(i)
                Next
                AppendLog "プログラム終了"

                If Me.WindowState <> vbMinimized Then
                    SaveSetting App.Title, "Settings", "MainLeft", Me.Left
                    SaveSetting App.Title, "Settings", "MainTop", Me.Top
                    SaveSetting App.Title, "Settings", "MainWidth", Me.Width
                End If
            End Select
        End If
    End Sub

```

Case 3

```

CBGraph1.Height = frmTab(TabIndex).Height - 400
CBGraph1.Width = frmTab(TabIndex).Width - 240
CBGraph2.Height = frmTab(TabIndex).Height - 400
CBGraph2.Width = frmTab(TabIndex).Width - 240

```

End Select

End If

End Sub

```

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer
    Timer1.Interval = 0

    For i = 0 To 23
        SaveSetting App.Title, "Settings", "WatchTime" + Format(i, "00"), chkWatchTime(i).Value
    Next

    'close all sub forms
    For i = Forms.Count - 1 To 1 Step -1
        Unload Forms(i)
    Next
    AppendLog "プログラム終了"

    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
    End If
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer
    Timer1.Interval = 0

    For i = 0 To 23
        SaveSetting App.Title, "Settings", "WatchTime" + Format(i, "00"), chkWatchTime(i).Value
    Next

    'close all sub forms
    For i = Forms.Count - 1 To 1 Step -1
        Unload Forms(i)
    Next
    AppendLog "プログラム終了"

    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
    End If
End Sub

```

SaveSetting App.Title, "Settings", "MainHeight", Me.Height

End If  
End Sub

Private Sub GetAIStatus\_Click()

Dim ScanN As Long

'動作ステータス取得

Info(0).Caption = "動作ステータス = " &amp; AcxAio1.GetAIStatus

AcxAio1.GetScanNumber ScanN

Info(1).Caption = "スキャン回数 = " &amp; Right(" " + Format(ScanN, "####0"), 5)

' Me.Caption = "Fifo スキャン回数 = " &amp; ScanN

End Sub

Private Sub LoadProperty\_Click()

'プロパティをファイルから読み込み

AcxAio1.LoadProperty ""

End Sub

Private Sub Open\_Click()

Dim Ret As Integer

'Timer1.Interval = 0

'コントロールの初期化

Ret = AcxAio1.Open

If Ret = 0 Then

Info(0).Caption = "初期化成功"

End If

End Sub

Private Sub SaveProperty\_Click()

'プロパティをファイルへ保存

AcxAio1.SaveProperty ""

End Sub

Private Sub ShowProperty\_Click()

'プロパティページを表示

AcxAio1.ShowProperty

End Sub

Private Sub StartAI\_Click()

Dim Ret As Integer

'サンプリング開始

Ret = AcxAio1.StartAI

Info(0).Caption = "StartAI = " &amp; Ret

DspData.Caption = ""

If Ret = 0 Then

DspData.Caption = "サンプリング中 ..."

End If

Timer1.Interval = 100

MstData.ActMode = e動作Mode.監視中

SaveSetting App.Title, "Settings", "MainHeight", Me.Height

End If  
End Sub

Private Sub GetAIStatus\_Click()

Dim ScanN As Long

'動作ステータス取得

Info(0).Caption = "動作ステータス = " &amp; AcxAio1.GetAIStatus

AcxAio1.GetScanNumber ScanN

Info(1).Caption = "スキャン回数 = " &amp; Right(" " + Format(ScanN, "####0"), 5)

Info(1).Caption = "スキャン回数 = " &amp; Right(" " + Format(ScanN, "####0"), 3)

' Me.Caption = "Fifo スキャン回数 = " &amp; ScanN

End Sub

Private Sub LoadProperty\_Click()

'プロパティをファイルから読み込み

AcxAio1.LoadProperty ""

End Sub

Private Sub Open\_Click()

Dim Ret As Integer

'Timer1.Interval = 0

'コントロールの初期化

Ret = AcxAio1.Open()

If Ret = 0 Then

Info(0).Caption = "初期化成功"

End If

End Sub

Private Sub SaveProperty\_Click()

'プロパティをファイルへ保存

AcxAio1.SaveProperty ""

End Sub

Private Sub ShowProperty\_Click()

'プロパティページを表示

AcxAio1.ShowProperty

End Sub

Private Sub StartAI\_Click()

Dim Ret As Integer

'サンプリング開始

Ret = AcxAio1.StartAI

Info(0).Caption = "StartAI = " &amp; Ret

DspData.Caption = ""

If Ret = 0 Then

DspData.Caption = "サンプリング中 ..."

End If

Timer1.Interval = 100

MstData.ActMode = e動作Mode.監視中

Label10.Caption = MstData.SampSec ' サンプリングタイム  
Label12.Caption = MstData.RecPreDataSec ' プリトリガ秒数  
Label14.Caption = MstData.RecDataSec ' 収録秒数

```

End Sub
Private Sub StopAI_Click()
    Timer1.Interval = 0
    AcxAio1.StopAI
    DoEvents
End Sub
Private Sub tabMain_Click(PreviousTab As Integer)
    SetActiveTabFrame tabMain.Tab
    Form_Resize
End Sub
Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "フ ロハ ティ"
            mnuFileProperties_Click
        Case "ヘルプ"
            mnuHelpAbout_Click
    End Select

```

```

End Sub
Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub
Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

```

```

Private Sub mnuViewToolBar_Click()
    mnuViewToolBar.Checked = Not mnuViewToolBar.Checked
    tbToolBar.Visible = mnuViewToolBar.Checked
End Sub

```

```

Private Sub mnuFileExit_Click()
    'プログラムの終了
    If Dir(AppDir + "¥ACXAIO.INI") <> "" Then Kill AppDir + "¥ACXAIO.INI"
    AcxAio1.SaveProperty(AppDir + "¥ACXAIO.INI")
    AD_Close_Click
    Unload Me

```

```

End Sub
Private Sub mnuFileProperties_Click()
    frmProperty.Show vbModeless, Me 'vbModeless, vbModal

```

```

End Sub
Public Sub SetActiveTabFrame(Index As Integer)
    Dim i As Integer
    tabMain.Tab = Index
    For i = 0 To tabMain.Tabs - 1
        frmTab(i).Enabled = (tabMain.Tab = i)
        frmTab(i).Visible = (tabMain.Tab = i)
    Next
    Select Case (tabMain.Tab)
        Case 0
        Case 1
        Case 2
        Case 3
    End Select
End Sub

```

```

Private Sub Timer1_Timer()

```

```

End Sub
Private Sub StopAI_Click()
    Timer1.Interval = 0
    AcxAio1.StopAI
    DoEvents
End Sub
Private Sub tabMain_Click(PreviousTab As Integer)
    SetActiveTabFrame tabMain.Tab
    Form_Resize
End Sub
Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "フ ロハ ティ"
            mnuFileProperties_Click
        Case "ヘルプ"
            mnuHelpAbout_Click
    End Select

```

```

End Sub
Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub
Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

```

```

Private Sub mnuViewToolBar_Click()
    mnuViewToolBar.Checked = Not mnuViewToolBar.Checked
    tbToolBar.Visible = mnuViewToolBar.Checked
End Sub

```

```

Private Sub mnuFileExit_Click()
    'プログラムの終了
    If Dir(AppDir + "¥ACXAIO.INI") <> "" Then Kill AppDir + "¥ACXAIO.INI"
    AcxAio1.SaveProperty(AppDir + "¥ACXAIO.INI")
    AD_Close_Click
    Unload Me

```

```

End Sub
Private Sub mnuFileProperties_Click()
    frmProperty.Show vbModeless, Me 'vbModeless, vbModal

```

```

End Sub
Public Sub SetActiveTabFrame(Index As Integer)
    Dim i As Integer
    tabMain.Tab = Index
    For i = 0 To tabMain.Tabs - 1
        frmTab(i).Enabled = (tabMain.Tab = i)
        frmTab(i).Visible = (tabMain.Tab = i)
    Next
    Select Case (tabMain.Tab)
        Case 0
        Case 1
        Case 2
        Case 3
    End Select
End Sub

```

```

Private Sub Timer1_Timer()

```

```

Dim Ret As Integer
Dim ii As Integer, jj As Integer, DataCompare As Long
Dim a As Double, kk As Integer, Volt As Double
Dim DateTime As Date

```

```
GetAIStatus_Click
```

```
If AcxAio1.GetAIStatus <> 1 Then
    cmdRun_Click
End If
```

```
Dim ScanN As Long, aaa As Long
```

```
AcxAio1.GetScanNumber ScanN
```

```
prbAdBuf.Value = ScanN
```

```
If ScanN > 1000 Then
```

```
    Ret = AcxAio1.GetAIData(1000, Buffer)
```

```
    For jj = 0 To 999
```

```
        For ii = 0 To gMaxChNo
```

```
            DataSec(jj, ii) = ConvInt(Buffer(jj * (gMaxChNo + 1) + ii))
```

```
        Next
```

```
    Next
```

```
    DateTime = (Date + Time) - Int(ScanN / 1000) / 86400
```

```
    MstData.PuthData DataSec, 1000, DateTime
```

```
    For ii = 0 To gMaxChNo
```

```
        Volt = ConvData(DataSec(0, ii), ii)
```

```
        lblAdData(ii) = Right(" " + Format(Volt, "##0.0000"), 8) & "V"
```

```
        prbAdData(ii).Value = DataSec(0, ii) + 32768
```

```
    Next
```

```
End If
```

```
If ScanN > 10000 Then
```

```
    Me.Show
```

```
    If Me.WindowState = vbMinimized Then
```

```
        Me.WindowState = vbNormal
```

```
    End If
```

```
End If
```

```
DispStatus
```

```
txtActMode = MstData.ActModeStr
```

```
If Format(Time, "nn:ss") = "00:00" Then
```

```
    ZCalender1.Year = Format(Date, "yyyy")
```

```
    ZCalender1.Month = Format(Date, "mm")
```

```
    ZCalender1.Day = Format(Date, "dd")
```

```
    ZCalender1.Refresh
```

```
Dim Ret As Integer
```

```
Dim ii As Integer, jj As Integer, DataCompare As Long
```

```
Dim a As Double, kk As Integer, Volt As Double
```

```
Dim DateTime As Date
```

```
Dim tempSampHz As Long
```

```
Static Keiho3Count(2) As Long
```

```
Dim MaxData1 As Integer
```

```
GetAIStatus_Click
```

```
If AcxAio1.GetAIStatus <> 1 Then
```

```
    cmdRun_Click
```

```
End If
```

```
Dim ScanN As Long, aaa As Long
```

```
AcxAio1.GetScanNumber ScanN
```

```
prbAdBuf.Value = ScanN
```

```
tempSampHz = Int(1# / MstData.SampSec)
```

```
If ScanN > tempSampHz Then
```

```
    Ret = AcxAio1.GetAIData(tempSampHz, Buffer)
```

```
    For jj = 0 To tempSampHz - 1
```

```
        For ii = 0 To gMaxChNo
```

```
            DataSec(jj, ii) = ConvInt(Buffer(jj * (gMaxChNo + 1) + ii))
```

```
        Next
```

```
    Next
```

```
    DateTime = (Date + Time) - Int(ScanN / tempSampHz) / 86400
```

```
    MstData.PuthData DataSec, tempSampHz, DateTime
```

```
    For ii = 0 To gMaxChNo
```

```
        Volt = ConvData(DataSec(0, ii), ii)
```

```
        lblAdData(ii) = Right(" " + Format(Volt, "##0.0000"), 8) & "V"
```

```
        prbAdData(ii).Value = DataSec(0, ii) + 32768
```

```
    Next
```

```
End If
```

```
If ScanN > 10000 Then
```

```
    If ScanN > 10000 Then
```

```
        Me.Show
```

```
        If Me.WindowState = vbMinimized Then
```

```
            Me.WindowState = vbNormal
```

```
        End If
```

```
    End If
```

```
DispStatus
```

```
txtActMode = MstData.ActModeStr
```

```
If Format(Time, "nn:ss") = "00:00" Then
```

```
    ZCalender1.Year = Format(Date, "yyyy")
```

```
    ZCalender1.Month = Format(Date, "mm")
```

```
    ZCalender1.Day = Format(Date, "dd")
```

```
    ZCalender1.Refresh
```

```
Para_Read
```

```
End If
```

```
MaxDataGal(0) = 40 * MstData.MaxData(0) / 32768 * 10 + MstData.GalVRange1 - 100
```

```
MaxDataGal(1) = 40 * MstData.MaxData(1) / 32768 * 10 + MstData.GalVRange2 - 100
```

```
MaxDataGal(2) = 40 * MstData.MaxData(2) / 32768 * 10 + MstData.GalVRange3 - 100
```

```
MaxDataGal(3) = 40 * MstData.MaxData(3) / 32768 * 10 + MstData.GalVRange4 - 100
```

```
L10DataGal(0) = MstData.L10Data(0)
```

```
L10DataGal(1) = MstData.L10Data(1)
```

```
L10DataGal(2) = MstData.L10Data(2)
```

```
L10DataGal(3) = MstData.L10Data(3)
```

```

MaxDataGal(3) = MstData.MaxData(3) * MstData.GalVRange4 / 32768 * 10
If IntLvX = 0 Then
  For ii = 0 To 3
    Label5(ii) = Format(L10DataGal(ii), "###0.0")
    Label5(ii + 4) = Format(LvDataGal(ii), "###0.0")
  Next ii
ElseIf IntLvX = 1 Then
  For ii = 0 To 3
    Label5(ii) = Format(MaxDataGal(ii), "###0.0")
  Next ii
End If
Label10.Caption = MstData.SampSec ' サンプルングタイム
Label12.Caption = MstData.RecPreDataSec ' プリトリガ秒数
Label14.Caption = MstData.RecDataSec ' 収録秒数
If (ZCalender1.Day = "01" And Format(Time, "nn:ss") = "00:00") Or MaxDataCount > 44640 Then
  MaxDataCount = 0
End If

警報判別

' 最大値をクリア 1分, 10分, 1時間
' If IntvPitch = 1 Then
'   If Right(Time, 2) = "00" Then
'     If MaxDataClrFlg = False Then
'       Command1_Click
'       警報判別
'       MaxDataClrFlg = True
'       最大値保存
'       If intCopyFlg = 1 And IntCopyMin = Val(Mid(Right(Time, 5), 1, 2)) Then
'         最大値保存Copy
'       End If
'
'       KeihoFlg1 = True
'       KeihoFlg = True
'       MaxDataPlot
'
'       MstData.MaxDataClr(0) ' 最大値クリア, L10クリア
'       MstData.MaxDataClr(1)
'       MstData.MaxDataClr(2)
'       MstData.MaxDataClr(3)
'       KeihoEmailFlg = False
'       If KeihoOffDisp = 1 Then
'         Label5(0).Visible = False
'         Label5(1).Visible = False
'         Label5(2).Visible = False
'         Label(3).Visible = False
'       Else
'         Label5(0).Visible = True
'         Label5(1).Visible = True
'         Label5(2).Visible = True
'         Label(3).Visible = True
'       End If
'     End If
'   End If
' ElseIf Right(Time, 2) = "01" Then
'   MaxDataClrFlg = False
' End If
' ElseIf IntvPitch = 10 Then
'   If Right(Time, 2) = "00" Or Right(Time, 2) = "01" Then ' グラフ表示, L10データ保存ピッチ: 1分
'     If MaxDataClrFlg = False Then
'       Command1_Click
'       警報判別

```

```

MaxDataClrFlg = True
Keiho3IntvFlg = False
If Keiho3Intv = 1 Then ' 毎分3連続データで判別
    If L10DataGal(0) >= MstData.Keiho11 Then
        Keiho3Count(0) = Keiho3Count(0) + 1
    Else
        Keiho3Count(0) = 0
    End If
    If L10DataGal(1) >= MstData.Keiho12 Then
        Keiho3Count(1) = Keiho3Count(1) + 1
    Else
        Keiho3Count(1) = 0
    End If
    If L10DataGal(2) >= MstData.Keiho13 Then
        Keiho3Count(2) = Keiho3Count(2) + 1
    Else
        Keiho3Count(2) = 0
    End If
    If Keiho3Count(0) >= 3 Or Keiho3Count(1) >= 3 Or Keiho3Count(2) >= 3 Then
        Keiho3IntvFlg = True
        警報判別
        Keiho3IntvFlg = False
    End If
End If

最大値保存
If intCopyFlg = 1 And IntCopyMin = Val(Mid(Right(Time, 5), 1, 2)) Then
On Error GoTo Error1
    最大値保存Copy
On Error GoTo 0
End If
''
    KeihoFlg1 = True
''
    KeihoFlg = True
MaxDataPlot

'''
    MstData.MaxDataClr (0)
'''
    MstData.MaxDataClr (1)
'''
    MstData.MaxDataClr (2)
'''
    MstData.MaxDataClr (3)
MaxDataClrFlg = True
If KeihoOffDisp = 1 Then
    Label5(0).Visible = False
    Label5(1).Visible = False
    Label5(2).Visible = False
    Label5(3).Visible = False
Else
    Label5(0).Visible = True
    Label5(1).Visible = True
    Label5(2).Visible = True
    Label5(3).Visible = True
End If
MstData.MaxDataClr (0) ' 最大値クリア
MstData.MaxDataClr (1)
MstData.MaxDataClr (2)
MstData.MaxDataClr (3)
If Check1.Value <> 1 Then ' 強制的に管理画面にもどる
    ZCalender1.Year = Year(Date)
    ZCalender1.Month = Month(Date)
    ZCalender1.Day = Day(Date)
    tabMain.Tab = 3
End If
End If

```

```

ElseIf Right(Time, 2) = "02" Or Right(Time, 2) = "03" Then
    MaxDataClrFlg = False
End If
' ElseIf IntvPitch = 60 Then
'     If Format(Time, "nn:ss") = "00:00" Then
'         If MaxDataClrFlg = False Then
'             MaxDataClrFlg = True
'             最大値保存
'             If intCopyFlg = 1 And IntCopyMin = Val(Mid(Right(Time, 5), 1, 2)) Then
'                 最大値保存Copy
'             End If
'             KeihoFlg1 = True
'             KeihoFlg = True
'             MaxDataPlot
'
'             MstData.MaxDataClr (0)
'             MstData.MaxDataClr (1)
'             MstData.MaxDataClr (2)
'             MstData.MaxDataClr (3)
'             MaxDataClrFlg = True
'         End If
'     ElseIf Format(Time, "nn:ss") = "00:01" Then
'         MaxDataClrFlg = False
'     End If
' End If
Label7.Caption = IntvPitch
Label16.Caption = ""
If intCopyFlg <> 0 Then Label16.Caption = strMaxDataCopyDir
If IntLvX = 0 Then
    Label8.Caption = "時間率レベル   L a5, L v 10[dB]"
ElseIf IntLvX = 1 Then
    Label8.Caption = "時間率レベル   L amax L v max[dB]"
End If

' 日 1 回画面クリア
Dim GraphPlotDatX(10) As Single
Dim GraphPlotDatY(10) As Single
Dim GraphP As Object
Dim DatCount As Integer

If Time = "0:00:00" Or Time = "0:00:01" Then
    If GraphOverWrite = 1 Then
        CBGraph1.Canvas.plotdataCollection("プロットデータ1").DataCount = 0
        CBGraph1.Canvas.plotdataCollection("プロットデータ2").DataCount = 0
        CBGraph1.Canvas.plotdataCollection("プロットデータ3").DataCount = 0
        CBGraph1.Canvas.plotdataCollection("プロットデータ4").DataCount = 0
        For ii = 0 To 0
            GraphPlotDatY(ii) = L10DataY(MaxDataCount0 + ii, 0)
        Next ii
        Set GraphP = CBGraph1.Canvas.PlotDataCollection("プロットデータ1")
        GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
        CBGraph1.Canvas.PlotDataCollection("プロットデータ1").ShowMarker = False
        For ii = 0 To 0
            CBGraph1.Canvas.PlotDataCollection("プロットデータ1").MarkerState(ii) = False
        Next ii
    End If
    MstData.L10DataClr (0) ' L10クリア
    MstData.L10DataClr (1) ' L10クリア
    MstData.L10DataClr (2) ' L10クリア

```

End If

```

End Sub

Public Sub DispStatus()

    Select Case AcxAio1.GetAIStatus
    Case 0:
        txtAD_Status = "正常終了"
    Case 1:
        txtAD_Status = "サンプリング中"
    Case 4:
        txtAD_Status = "StopAI による強制停止"
    Case 12000:
        txtAD_Status = "バッファオーバーフロー"
    Case 12001:
        txtAD_Status = "サンプリングクロックエラー"
    Case 12002:
        txtAD_Status = "スタートトリガエラー"
    Case -1:
        txtAD_Status = "ステータスの取得に失敗した"
    Case Else
        txtAD_Status = "未定義"
    End Select

End Sub

```

```
Public Sub DispStatus()
```

```

    Select Case AcxAio1.GetAIStatus
    Case 0:
        txtAD_Status = "正常終了"
    Case 1:
        txtAD_Status = "サンプリング中"
    Case 4:
        txtAD_Status = "StopAI による強制停止"
    Case 12000:
        txtAD_Status = "バッファオーバーフロー"
    Case 12001:
        txtAD_Status = "サンプリングクロックエラー"
    Case 12002:
        txtAD_Status = "スタートトリガエラー"
    Case -1:
        txtAD_Status = "ステータスの取得に失敗した"
    Case Else
        txtAD_Status = "未定義"
    End Select

```

```
End Sub
```

```

'名前解決した
Private Sub Ping1_Resolved(ByVal RemoteIpAddress As String)
    '再送のためにホストのIPアドレスを記憶しておく
    m_host = RemoteIpAddress
End Sub

```

```

    MstData.L10DataClr (3) 'L10クリア
End If
' If KeihoStopFlg = True Then
'     Command1_Click
'     KeihoStopFlg = False
' End If

DoEvents
txtPointNo.Caption = PointNo
' Label23.Caption = RingCount
Exit Sub
Error1:
Dim i As Integer
i = i

```

```
Public Sub DispStatus()
```

```

    Select Case AcxAio1.GetAIStatus
    Case 0:
        txtAD_Status = "正常終了"
    Case 1:
        txtAD_Status = "サンプリング中"
    Case 4:
        txtAD_Status = "StopAI による強制停止"
    Case 12000:
        txtAD_Status = "バッファオーバーフロー"
    Case 12001:
        txtAD_Status = "サンプリングクロックエラー"
    Case 12002:
        txtAD_Status = "スタートトリガエラー"
    Case -1:
        txtAD_Status = "ステータスの取得に失敗した"
    Case Else
        txtAD_Status = "未定義"
    End Select

```

```
End Sub
```

```

'名前解決した
Private Sub Ping1_Resolved(ByVal RemoteIpAddress As String)
    '再送のためにホストのIPアドレスを記憶しておく
    m_host = RemoteIpAddress
End Sub

```

```

Public Function MaxDataPlot()
Dim i As Integer
Dim GraphP As Object
Dim DatCount As Integer
Static GraphPlotDatX(10080) As Single, GraphPlotDatY(10080) As Single
'データの取得 MaxDataGal→MaxDataY()は最大値保存で実施
'グラフに最大値をプロット
On Error GoTo Error1
Dim FileNo As Integer
FileNo = FreeFile
txtOpenFileNum.Caption = Str(FileNo)

If MaxDataCount <= MaxDataCount0 Then Exit Function
If MaxDataCount - MaxDataCount0 >= 1440 Then Exit Function

```

```

CBGraph1.Canvas.axes("軸1").startvalue = GraphYScale(0)
CBGraph1.Canvas.axes("軸2").startvalue = GraphYScale(1)
CBGraph1.Canvas.axes("軸3").startvalue = GraphYScale(2)
CBGraph1.Canvas.axes("軸4").startvalue = GraphYScale(3)

CBGraph1.Canvas.axes("軸1").scalecount = GraphYScaleCount(0)
CBGraph1.Canvas.axes("軸2").scalecount = GraphYScaleCount(1)
CBGraph1.Canvas.axes("軸3").scalecount = GraphYScaleCount(2)
CBGraph1.Canvas.axes("軸4").scalecount = GraphYScaleCount(3)
CBGraph1.Canvas.grids("格子1").HorzGridCount = GraphYScaleCount(0) - 1
CBGraph1.Canvas.grids("格子2").HorzGridCount = GraphYScaleCount(1) - 1
CBGraph1.Canvas.grids("格子3").HorzGridCount = GraphYScaleCount(2) - 1
CBGraph1.Canvas.grids("格子4").HorzGridCount = GraphYScaleCount(3) - 1

DatCount = MaxDataCount - MaxDataCount0
For i = 0 To DatCount
    GraphPlotDatX(i) = (MaxDataX(MaxDataCount0 + i) - Int(MaxDataX(MaxDataCount0 + i))) * 24
Next i
If IntLvX = 0 Then 'Lv10をプロット
    For i = 0 To DatCount
        If KeihoOffDisp = 1 And L10DataY(MaxDataCount0 + i, 0) >= MstData.Keiho11 Then
            GraphPlotDatY(i) = -99
            GraphPlotDatX(i) = 0
        Else
            GraphPlotDatY(i) = L10DataY(MaxDataCount0 + i, 0)
        End If
    Next i
    Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ1")
    GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
    CBGraph1.Canvas.plotdataCollection("プロットデータ1").ShowMarker = True
    For i = 0 To DatCount - 1
        CBGraph1.Canvas.plotdataCollection("プロットデータ1").MarkerState(i) = True
    Next i

    For i = 0 To DatCount
        If KeihoOffDisp = 1 And L10DataY(MaxDataCount0 + i, 1) >= MstData.Keiho12 Then
            GraphPlotDatY(i) = -99
            GraphPlotDatX(i) = 0
        Else
            GraphPlotDatY(i) = L10DataY(MaxDataCount0 + i, 1)
        End If
    Next i
    Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ2")
    GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
    CBGraph1.Canvas.plotdataCollection("プロットデータ2").ShowMarker = True
    For i = 0 To DatCount - 1
        CBGraph1.Canvas.plotdataCollection("プロットデータ2").MarkerState(i) = True
    Next i

    For i = 0 To DatCount
        If KeihoOffDisp = 1 And L10DataY(MaxDataCount0 + i, 2) >= MstData.Keiho13 Then
            GraphPlotDatY(i) = -99
            GraphPlotDatX(i) = 0
        Else
            GraphPlotDatY(i) = L10DataY(MaxDataCount0 + i, 2)
        End If
    Next i
    Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ3")
    GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
    CBGraph1.Canvas.plotdataCollection("プロットデータ3").ShowMarker = True

```

```

For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ3").MarkerState(i) = True
Next i

For i = 0 To DatCount
    If KeihoOffDisp = 1 And L10DataY(MaxDataCount0 + i, 3) >= MstData.Keiho14 Then
        GraphPlotDatY(i) = -99
        GraphPlotDatX(i) = 0
    Else
        GraphPlotDatY(i) = L10DataY(MaxDataCount0 + i, 3)
    End If
Next i
Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ4")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
CBGraph1.Canvas.plotdataCollection("プロットデータ4").ShowMarker = True
For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ4").MarkerState(i) = True
Next i

ElseIf IntLvX = 1 Then 'Lvmaxをプロット
For i = 0 To DatCount
    If KeihoOffDisp = 1 And MaxDataY(MaxDataCount0 + i, 0) >= MstData.Keiho11 Then
        GraphPlotDatY(i) = -99
        GraphPlotDatX(i) = 0
    Else
        GraphPlotDatY(i) = MaxDataY(MaxDataCount0 + i, 0)
    End If
Next i
Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ1")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
CBGraph1.Canvas.plotdataCollection("プロットデータ1").ShowMarker = True
For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ1").MarkerState(i) = True
Next i

For i = 0 To DatCount
    If KeihoOffDisp = 1 And MaxDataY(MaxDataCount0 + i, 1) >= MstData.Keiho12 Then
        GraphPlotDatY(i) = -99
        GraphPlotDatX(i) = 0
    Else
        GraphPlotDatY(i) = MaxDataY(MaxDataCount0 + i, 1)
    End If
Next i
Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ2")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
CBGraph1.Canvas.plotdataCollection("プロットデータ2").ShowMarker = True
For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ2").MarkerState(i) = True
Next i

For i = 0 To DatCount
    If KeihoOffDisp = 1 And MaxDataY(MaxDataCount0 + i, 2) >= MstData.Keiho13 Then
        GraphPlotDatY(i) = -99
        GraphPlotDatX(i) = 0
    Else
        GraphPlotDatY(i) = MaxDataY(MaxDataCount0 + i, 2)
    End If

```

```

Next i
Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ3")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
CBGraph1.Canvas.plotdataCollection("プロットデータ3").ShowMarker = True
For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ3").MarkerState(i) = True
Next i

For i = 0 To DatCount
    If KeihoOffDisp = 1 And MaxDataY(MaxDataCount0 + i, 3) >= MstData.Keiho14 Then
        GraphPlotDatY(i) = -99
        GraphPlotDatX(i) = 0
    Else
        GraphPlotDatY(i) = MaxDataY(MaxDataCount0 + i, 3)
    End If
Next i
Set GraphP = CBGraph1.Canvas.plotdataCollection("プロットデータ4")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
CBGraph1.Canvas.plotdataCollection("プロットデータ4").ShowMarker = True
For i = 0 To DatCount - 1
    CBGraph1.Canvas.plotdataCollection("プロットデータ4").MarkerState(i) = True
Next i

End If

For i = 0 To 3
    CBGraph1.Canvas.lines("線" & Right(i + 1, 1)).startX = TimeValue(Time) * 2300 + 250
    CBGraph1.Canvas.lines("線" & Right(i + 1, 1)).startY = i * 500 + 100
    CBGraph1.Canvas.lines("線" & Right(i + 1, 1)).endX = TimeValue(Time) * 2300 + 250
    CBGraph1.Canvas.lines("線" & Right(i + 1, 1)).endY = i * 500 + 550
Next i

If GraphYScale(0) >= MstData.Keiho1 Then
    CBGraph1.Canvas.lines("線11").Show = True
    CBGraph1.Canvas.lines("線11").startX = 250
    CBGraph1.Canvas.lines("線11").startY = 550 - MstData.Keiho1 / GraphYScale(0) * 450
    CBGraph1.Canvas.lines("線11").endX = 2550
    CBGraph1.Canvas.lines("線11").endY = 550 - MstData.Keiho1 / GraphYScale(0) * 450
Else
    CBGraph1.Canvas.lines("線11").Show = False
End If

If GraphYScale(1) >= MstData.Keiho2 Then
    CBGraph1.Canvas.lines("線12").Show = True
    CBGraph1.Canvas.lines("線12").startX = 250
    CBGraph1.Canvas.lines("線12").startY = 1050 - MstData.Keiho2 / GraphYScale(1) * 450
    CBGraph1.Canvas.lines("線12").endX = 2550
    CBGraph1.Canvas.lines("線12").endY = 1050 - MstData.Keiho2 / GraphYScale(1) * 450
Else
    CBGraph1.Canvas.lines("線12").Show = False
End If

If GraphYScale(2) >= MstData.Keiho3 Then
    CBGraph1.Canvas.lines("線13").Show = True
    CBGraph1.Canvas.lines("線13").startX = 250
    CBGraph1.Canvas.lines("線13").startY = 1550 - MstData.Keiho3 / GraphYScale(2) * 450
    CBGraph1.Canvas.lines("線13").endX = 2550
    CBGraph1.Canvas.lines("線13").endY = 1550 - MstData.Keiho3 / GraphYScale(2) * 450

```

```

Else
  CBGraph1.Canvas.Lines("線13").Show = False
End If

If GraphYScale(3) >= MstData.Keiho4 Then
  CBGraph1.Canvas.Lines("線14").Show = True
  CBGraph1.Canvas.Lines("線14").startX = 250
  CBGraph1.Canvas.Lines("線14").startY = 2050 - MstData.Keiho4 / GraphYScale(3) * 450
  CBGraph1.Canvas.Lines("線14").endX = 2550
  CBGraph1.Canvas.Lines("線14").endY = 2050 - MstData.Keiho4 / GraphYScale(3) * 450
Else
  CBGraph1.Canvas.Lines("線14").Show = False
End If

If GraphYScale(0) >= MstData.Keiho11 Then
  CBGraph1.Canvas.Lines("線21").Show = True
  CBGraph1.Canvas.Lines("線21").startX = 250
  CBGraph1.Canvas.Lines("線21").startY = 550 - MstData.Keiho11 / GraphYScale(0) * 450
  CBGraph1.Canvas.Lines("線21").endX = 2550
  CBGraph1.Canvas.Lines("線21").endY = 550 - MstData.Keiho11 / GraphYScale(0) * 450
Else
  CBGraph1.Canvas.Lines("線21").Show = False
End If

If GraphYScale(1) >= MstData.Keiho12 Then
  CBGraph1.Canvas.Lines("線22").Show = True
  CBGraph1.Canvas.Lines("線22").startX = 250
  CBGraph1.Canvas.Lines("線22").startY = 1050 - MstData.Keiho12 / GraphYScale(1) * 450
  CBGraph1.Canvas.Lines("線22").endX = 2550
  CBGraph1.Canvas.Lines("線22").endY = 1050 - MstData.Keiho12 / GraphYScale(1) * 450
Else
  CBGraph1.Canvas.Lines("線13").Show = False
End If

If GraphYScale(2) >= MstData.Keiho13 Then
  CBGraph1.Canvas.Lines("線23").Show = True
  CBGraph1.Canvas.Lines("線23").startX = 250
  CBGraph1.Canvas.Lines("線23").startY = 1550 - MstData.Keiho13 / GraphYScale(2) * 450
  CBGraph1.Canvas.Lines("線23").endX = 2550
  CBGraph1.Canvas.Lines("線23").endY = 1550 - MstData.Keiho13 / GraphYScale(2) * 450
Else
  CBGraph1.Canvas.Lines("線23").Show = False
End If

If GraphYScale(3) >= MstData.Keiho14 Then
  CBGraph1.Canvas.Lines("線24").Show = True
  CBGraph1.Canvas.Lines("線24").startX = 250
  CBGraph1.Canvas.Lines("線24").startY = 2050 - MstData.Keiho14 / GraphYScale(3) * 450
  CBGraph1.Canvas.Lines("線24").endX = 2550
  CBGraph1.Canvas.Lines("線24").endY = 2050 - MstData.Keiho14 / GraphYScale(3) * 450
Else
  CBGraph1.Canvas.Lines("線24").Show = False
End If

Exit Function
Error1:
i = i
End Function
Public Function MaxDataPlot2(ByVal DCount0 As Long, ByVal DCount As Long) As Integer 'DCount0;

```

```

Dim i As Integer
Dim GraphP As Object
Dim DatCount As Integer
Static GraphPlotDatX(10080) As Single, GraphPlotDatY(10080) As Single
'データの取得 MaxDataGal→MaxDataY()は最大値保存で実施
'グラフに最大値をプロット
On Error GoTo Error1
Dim FileNo As Integer
FileNo = FreeFile
txtOpenFileNum.Caption = Str(FileNo)
MaxDataPlot2 = 1
If DCount <= DCount0 Then: MaxDataPlot2 = 0: Exit Function
If DCount - DCount0 >= 1440 Then: MaxDataPlot2 = 0: Exit Function

CBGraph2.Canvas.plotdataCollection("プロットデータ1").DataCount = 0
CBGraph2.Canvas.plotdataCollection("プロットデータ2").DataCount = 0
CBGraph2.Canvas.plotdataCollection("プロットデータ3").DataCount = 0
CBGraph2.Canvas.plotdataCollection("プロットデータ4").DataCount = 0

CBGraph2.Canvas.axes("軸1").startvalue = GraphYScale(0)
CBGraph2.Canvas.axes("軸2").startvalue = GraphYScale(1)
CBGraph2.Canvas.axes("軸3").startvalue = GraphYScale(2)
CBGraph2.Canvas.axes("軸4").startvalue = GraphYScale(3)

CBGraph2.Canvas.axes("軸1").scalecount = GraphYScaleCount(0)
CBGraph2.Canvas.axes("軸2").scalecount = GraphYScaleCount(1)
CBGraph2.Canvas.axes("軸3").scalecount = GraphYScaleCount(2)
CBGraph2.Canvas.axes("軸4").scalecount = GraphYScaleCount(3)
CBGraph2.Canvas.grids("格子1").HorzGridCount = GraphYScaleCount(0) - 1
CBGraph2.Canvas.grids("格子2").HorzGridCount = GraphYScaleCount(1) - 1
CBGraph2.Canvas.grids("格子3").HorzGridCount = GraphYScaleCount(2) - 1
CBGraph2.Canvas.grids("格子4").HorzGridCount = GraphYScaleCount(3) - 1

DatCount = DCount - DCount0
For i = 0 To DatCount
    GraphPlotDatX(i) = (MaxDataX0(DCount0 + i) - Int(MaxDataX0(DCount0 + i))) * 24
Next i
If IntLvX = 0 Then 'Lv10をプロット
    For i = 0 To DatCount
        GraphPlotDatY(i) = L10DataY0(DCount0 + i, 0)
    Next i
    Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ1")
    GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
    CBGraph2.Canvas.plotdataCollection("プロットデータ1").ShowMarker = True
    For i = 0 To DatCount - 1
        CBGraph2.Canvas.plotdataCollection("プロットデータ1").MarkerState(i) = True
    Next i

    For i = 0 To DatCount
        GraphPlotDatY(i) = L10DataY0(DCount0 + i, 1)
    Next i
    Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ2")
    GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
    CBGraph2.Canvas.plotdataCollection("プロットデータ2").ShowMarker = True
    For i = 0 To DatCount - 1
        CBGraph2.Canvas.plotdataCollection("プロットデータ2").MarkerState(i) = True
    Next i

    For i = 0 To DatCount
        GraphPlotDatY(i) = L10DataY0(DCount0 + i, 2)
    Next i
    Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ3")

```

```

GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ3").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ3").MarkerState(i) = True
Next i

For i = 0 To DatCount
  GraphPlotDatY(i) = L10DataY0(DCount0 + i, 3)
Next i
Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ4")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ4").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ4").MarkerState(i) = True
Next i

ElseIf IntLvX = 1 Then 'Lvmaxをプロット
For i = 0 To DatCount
  GraphPlotDatY(i) = MaxDataY0(DCount0 + i, 0)
Next i
Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ1")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ1").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ1").MarkerState(i) = True
Next i

For i = 0 To DatCount
  GraphPlotDatY(i) = MaxDataY0(DCount0 + i, 1)
Next i
Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ2")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ2").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ2").MarkerState(i) = True
Next i

For i = 0 To DatCount
  GraphPlotDatY(i) = MaxDataY0(DCount0 + i, 2)
Next i
Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ3")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ3").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ3").MarkerState(i) = True
Next i

For i = 0 To DatCount
  GraphPlotDatY(i) = MaxDataY0(DCount0 + i, 3)
Next i
Set GraphP = CBGraph2.Canvas.plotdataCollection("プロットデータ4")
GraphP.SetData GraphPlotDatX, GraphPlotDatY, DatCount
  CBGraph2.Canvas.plotdataCollection("プロットデータ4").ShowMarker = True
For i = 0 To DatCount - 1
  CBGraph2.Canvas.plotdataCollection("プロットデータ4").MarkerState(i) = True
Next i
End If

For i = 0 To 3
  CBGraph2.Canvas.lines("線" & Right(i + 1, 1)).startX = TimeValue(Time) * 2200 + 250
  CBGraph2.Canvas.lines("線" & Right(i + 1, 1)).startY = i * 550 + 100

```

```

        CBGraph2.Canvas.Lines("線" & Right(i + 1, 1)).endX = TimeValue(Time) * 2200 + 250
        CBGraph2.Canvas.Lines("線" & Right(i + 1, 1)).endY = i * 550 + 600
    Next i

    If GraphYScale(0) >= MstData.Keiho1 Then
        CBGraph2.Canvas.Lines("線11").Show = True
        CBGraph2.Canvas.Lines("線11").startX = 250
        CBGraph2.Canvas.Lines("線11").startY = 550 - MstData.Keiho1 / GraphYScale(0) * 450
        CBGraph2.Canvas.Lines("線11").endX = 2550
        CBGraph2.Canvas.Lines("線11").endY = 550 - MstData.Keiho1 / GraphYScale(0) * 450
    Else
        CBGraph2.Canvas.Lines("線11").Show = False
    End If

    If GraphYScale(1) >= MstData.Keiho2 Then
        CBGraph2.Canvas.Lines("線12").Show = True
        CBGraph2.Canvas.Lines("線12").startX = 250
        CBGraph2.Canvas.Lines("線12").startY = 1050 - MstData.Keiho2 / GraphYScale(1) * 450
        CBGraph2.Canvas.Lines("線12").endX = 2550
        CBGraph2.Canvas.Lines("線12").endY = 1050 - MstData.Keiho2 / GraphYScale(1) * 450
    Else
        CBGraph2.Canvas.Lines("線12").Show = False
    End If

    If GraphYScale(2) >= MstData.Keiho3 Then
        CBGraph2.Canvas.Lines("線13").Show = True
        CBGraph2.Canvas.Lines("線13").startX = 250
        CBGraph2.Canvas.Lines("線13").startY = 1550 - MstData.Keiho3 / GraphYScale(2) * 450
        CBGraph2.Canvas.Lines("線13").endX = 2550
        CBGraph2.Canvas.Lines("線13").endY = 1550 - MstData.Keiho3 / GraphYScale(2) * 450
    Else
        CBGraph2.Canvas.Lines("線13").Show = False
    End If

    If GraphYScale(3) >= MstData.Keiho4 Then
        CBGraph2.Canvas.Lines("線14").Show = True
        CBGraph2.Canvas.Lines("線14").startX = 250
        CBGraph2.Canvas.Lines("線14").startY = 2050 - MstData.Keiho4 / GraphYScale(3) * 450
        CBGraph2.Canvas.Lines("線14").endX = 2550
        CBGraph2.Canvas.Lines("線14").endY = 2050 - MstData.Keiho4 / GraphYScale(3) * 450
    Else
        CBGraph2.Canvas.Lines("線14").Show = False
    End If

    If GraphYScale(0) >= MstData.Keiho11 Then
        CBGraph2.Canvas.Lines("線21").Show = True
        CBGraph2.Canvas.Lines("線21").startX = 250
        CBGraph2.Canvas.Lines("線21").startY = 550 - MstData.Keiho11 / GraphYScale(0) * 450
        CBGraph2.Canvas.Lines("線21").endX = 2550
        CBGraph2.Canvas.Lines("線21").endY = 550 - MstData.Keiho11 / GraphYScale(0) * 450
    Else
        CBGraph2.Canvas.Lines("線21").Show = False
    End If

    If GraphYScale(1) >= MstData.Keiho12 Then
        CBGraph2.Canvas.Lines("線22").Show = True
        CBGraph2.Canvas.Lines("線22").startX = 250
        CBGraph2.Canvas.Lines("線22").startY = 1050 - MstData.Keiho12 / GraphYScale(1) * 450
        CBGraph2.Canvas.Lines("線22").endX = 2550
        CBGraph2.Canvas.Lines("線22").endY = 1050 - MstData.Keiho12 / GraphYScale(1) * 450
    End If

```

```

Else
    CBGraph2.Canvas.Lines("線13").Show = False
End If

If GraphYScale(2) >= MstData.Keiho13 Then
    CBGraph2.Canvas.Lines("線23").Show = True
    CBGraph2.Canvas.Lines("線23").startX = 250
    CBGraph2.Canvas.Lines("線23").startY = 1550 - MstData.Keiho13 / GraphYScale(2) * 450
    CBGraph2.Canvas.Lines("線23").endX = 2550
    CBGraph2.Canvas.Lines("線23").endY = 1550 - MstData.Keiho13 / GraphYScale(2) * 450
Else
    CBGraph2.Canvas.Lines("線23").Show = False
End If

If GraphYScale(3) >= MstData.Keiho14 Then
    CBGraph2.Canvas.Lines("線24").Show = True
    CBGraph2.Canvas.Lines("線24").startX = 250
    CBGraph2.Canvas.Lines("線24").startY = 2050 - MstData.Keiho14 / GraphYScale(3) * 450
    CBGraph2.Canvas.Lines("線24").endX = 2550
    CBGraph2.Canvas.Lines("線24").endY = 2050 - MstData.Keiho14 / GraphYScale(3) * 450
Else
    CBGraph2.Canvas.Lines("線24").Show = False
End If
Exit Function
Error1:
i = i
End Function
Public Function 警報判別()
    'Static Keiho3Count(2) As Long
    Dim EmailSend

On Error GoTo Error1
If KeihoCheck <> 0 Then '警報非表示の場合
    KeihoFlg1(0) = True: KeihoFlg1(1) = True: KeihoFlg1(2) = True: KeihoFlg1(3) = True
    KeihoFlg(0) = True: KeihoFlg(1) = True: KeihoFlg(2) = True: KeihoFlg(3) = True
Exit Function
End If
If Keiho3Intv = 1 Then '3連続表示値管理の場合
    '-----注意判別-----
    If Keiho3IntvFlg = True Then '起動判別
        If KeihoFlg1(0) = True Then
            If L10DataGal(0) >= MstData.Keiho11 And L10DataGal(0) < MstData.Keiho1 Then
                CBGraph1.Canvas.Color = RGB(255, 255, 0)
                Label5(0).BackColor = RGB(255, 255, 0)
                Timer2.Enabled = True
                KeihoFlg1(0) = False
                AcxAio1.WriteBitData(BitNumber as Integer, Flag as Integer) as Integer
                AcxAio1.WriteBitData 0, 1
                AppendLog "A 注意報起動": Grig_AppendStatus "A 注意報起動"
                EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
                KeihoEmailFlg = True
                Timer3.Enabled = True
            End If
        End If
        If KeihoFlg1(1) = True Then
            If L10DataGal(1) >= MstData.Keiho12 And L10DataGal(1) < MstData.Keiho2 Then
                CBGraph1.Canvas.Color = RGB(255, 255, 0)
                Label5(1).BackColor = RGB(255, 255, 0)
                Timer2.Enabled = True
                KeihoFlg1(1) = False
                AcxAio1.WriteBitData 0, 1
                AppendLog "B 注意報起動": Grig_AppendStatus "B 注意報起動"
            End If
        End If
    End If

```

```

        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
        KeihoEmailFlg = True
        Timer3.Enabled = True
    End If
End If
If KeihoFlg1(2) = True Then
    If L10DataGal(2) >= MstData.Keiho13 And L10DataGal(2) < MstData.Keiho3 Then
        Label5(2).BackColor = RGB(255, 255, 0)
        Timer2.Enabled = True
        KeihoFlg1(2) = False
        AcxAio1.WriteBitData 0, 1
        AppendLog "C 注意報起動": Grig_AppendStatus "C 注意報起動"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = True
    End If
End If
If KeihoFlg(0) = True Then
    If L10DataGal(0) >= MstData.Keiho1 Then
        Timer2.Enabled = False
        Label5(0).BackColor = RGB(255, 0, 0)
        Timer2.Enabled = True
        KeihoFlg(0) = False
        AcxAio1.WriteBitData 0, 1
        AppendLog "A 警報起動": Grig_AppendStatus "A 警報起動"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
        KeihoEmailFlg = True
    End If
End If
If KeihoFlg(1) = True Then
    If L10DataGal(1) >= MstData.Keiho2 Then
        Timer2.Enabled = False
        Label5(1).BackColor = RGB(255, 0, 0)
        Timer2.Enabled = True
        KeihoFlg(1) = False
        AcxAio1.WriteBitData 0, 1
        AppendLog "B 警報起動": Grig_AppendStatus "B 警報起動"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
        KeihoEmailFlg = True
    End If
End If
If KeihoFlg(2) = True Then
    If L10DataGal(2) >= MstData.Keiho3 Then
        Timer2.Enabled = False
        Label5(2).BackColor = RGB(255, 0, 0)
        Timer2.Enabled = True
        KeihoFlg(2) = False
        AcxAio1.WriteBitData 0, 1
        AppendLog "C 警報起動": Grig_AppendStatus "C 警報起動"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = True
    End If
End If
If KeihoFlg(3) = True Then
    If L10DataGal(3) >= MstData.Keiho4 Then
        Timer2.Enabled = False
        Label5(3).BackColor = RGB(255, 0, 0)
        Timer2.Enabled = True
        KeihoFlg(3) = False
        AcxAio1.WriteBitData 0, 1
        AppendLog "C 警報起動": Grig_AppendStatus "C 警報起動"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = True
    End If
End If

```

```

End If
End If

End If

If KeihoFlg1(0) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005. 3. 8
' keiho11
If KeihoFlg1(0) = True And L10DataGal(0) < MstData.Keiho11 Then
Label5(0).BackColor = RGB(255, 255, 255)
End If
Else ' 注意報終了処理
If L10DataGal(0) < MstData.Keiho11 Then
' CBGraph1.Canvas.Color = RGB(255, 255, 255)
Label5(0).BackColor = RGB(255, 255, 255)
' Timer2.Enabled = False
KeihoFlg1(0) = True
' AcxAio1.WriteBitData(BitNumber as Integer, Flag as Integer) as Integer
AcxAio1.WriteBitData 0, 0
AppendLog "A 注意報終了": Grig_AppendStatus "A 注意報終了"
' EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
KeihoEmailFlg = False
' Timer3.Enabled = True
End If
Label5(0).BackColor = RGB(255, 255, 0)
End If

If KeihoFlg1(1) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005. 3. 8
' keiho12
If KeihoFlg1(1) = True And L10DataGal(1) < MstData.Keiho12 Then
Label5(1).BackColor = RGB(255, 255, 255)
End If
Else ' 注意報終了処理
If L10DataGal(1) < MstData.Keiho12 Then
Label5(1).BackColor = RGB(255, 255, 255)
' Timer2.Enabled = False
KeihoFlg1(1) = True
AcxAio1.WriteBitData 0, 0
AppendLog "B 注意報終了": Grig_AppendStatus "B 注意報終了"
' EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
KeihoEmailFlg = False
' Timer3.Enabled = True
End If
Label5(1).BackColor = RGB(255, 255, 0)
End If

If KeihoFlg1(2) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005. 3. 8
' keiho13
If KeihoFlg1(2) = True And L10DataGal(2) < MstData.Keiho13 Then
Label5(2).BackColor = RGB(255, 255, 255)
End If
Else ' 注意報終了処理
If L10DataGal(2) < MstData.Keiho13 Then
Label5(2).BackColor = RGB(255, 255, 255)
' Timer2.Enabled = False
KeihoFlg1(2) = True
AcxAio1.WriteBitData 0, 0
AppendLog "C 注意報終了": Grig_AppendStatus "C 注意報終了"
' EmailSend = Shell("d:¥振動data¥csv_open.bat")
KeihoEmailFlg = False
End If

```

```

Label5(2).BackColor = RGB(255, 255, 0)
End If
If KeihoFlg1(3) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho13
    If KeihoFlg1(3) = True And L10DataGal(3) < MstData.Keiho14 Then
        Label5(3).BackColor = RGB(255, 255, 255)
    End If
Else ' 注意報終了処理
    If L10DataGal(3) < MstData.Keiho14 Then
        Label5(3).BackColor = RGB(255, 255, 255)
        Timer2.Enabled = False
        KeihoFlg1(3) = True
        AcxAio1.WriteBitData 0, 0
        AppendLog "C 注意報終了": Grig_AppendStatus "C 注意報終了"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = False
    End If
    Label5(3).BackColor = RGB(255, 255, 0)
End If

' -----警報判別-----
If KeihoFlg(0) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho1
Else ' 警報終了処理
    If L10DataGal(0) < MstData.Keiho1 Then
        Timer2.Enabled = False
        Label5(0).BackColor = RGB(255, 255, 255)
        Timer2.Enabled = False
        KeihoFlg(0) = True
        AcxAio1.WriteBitData 0, 0
        AppendLog "A 警報終了": Grig_AppendStatus "A 警報終了"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
        KeihoEmailFlg = False
    End If
    Label5(0).BackColor = RGB(255, 0, 0)
End If

If KeihoFlg(1) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho2
Else ' 警報終了処理
    If L10DataGal(1) < MstData.Keiho2 Then
        Timer2.Enabled = False
        Label5(1).BackColor = RGB(255, 255, 255)
        Timer2.Enabled = True
        KeihoFlg(1) = True
        AcxAio1.WriteBitData 0, 0
        AppendLog "B 警報終了": Grig_AppendStatus "B 警報終了"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
        KeihoEmailFlg = False
    End If
    Label5(1).BackColor = RGB(255, 0, 0)
End If
If KeihoFlg(2) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho2
Else ' 警報終了処理
    If L10DataGal(2) < MstData.Keiho3 Then
        Timer2.Enabled = False
        Label5(2).BackColor = RGB(255, 255, 255)
        Timer2.Enabled = False

```

```

        KeihoFlg(2) = True
        AcxAio1.WriteBitData 0, 0
        AppendLog "C 警報終了": Grig_AppendStatus "C 警報終了"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = False
    End If
    Label5(2).BackColor = RGB(255, 0, 0)
End If
If KeihoFlg(3) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho2

Else ' 警報終了処理
    If L10DataGal(3) < MstData.Keiho4 Then
        Timer2.Enabled = False
        Label5(3).BackColor = RGB(255, 255, 255)
        Timer2.Enabled = False
        KeihoFlg(3) = True
        AcxAio1.WriteBitData 0, 0
        AppendLog "C 警報終了": Grig_AppendStatus "C 警報終了"
        EmailSend = Shell("d:¥振動data¥csv_open.bat")
        KeihoEmailFlg = False
    End If
    Label5(3).BackColor = RGB(255, 0, 0)
End If

' 画面色
If KeihoFlg1(0) = True And KeihoFlg1(1) = True And KeihoFlg1(2) = True And _
KeihoFlg(0) = True And KeihoFlg(1) = True And KeihoFlg(2) = True Then
    CBGraph1.Canvas.Color = RGB(255, 255, 255)
    KeihoEmailFlg1 = False 'メールフラグ1クリア
ElseIf KeihoFlg(0) = False Or KeihoFlg(1) = False Or KeihoFlg(2) = False Then
    If Right(Time, 1) Mod 2 = 1 Then 'フリッカー動作
        CBGraph1.Canvas.Color = RGB(255, 0, 0) '赤
    Else
        CBGraph1.Canvas.Color = RGB(255, 255, 255)
    End If
ElseIf KeihoFlg1(0) = False Or KeihoFlg1(1) = False Or KeihoFlg1(2) = False Then
    If Right(Time, 1) Mod 2 = 1 Then 'フリッカー動作
        CBGraph1.Canvas.Color = RGB(255, 255, 0) '黄
    Else
        CBGraph1.Canvas.Color = RGB(255, 255, 255)
    End If
End If

' -----Emeal-----
If KeihoEmailFlg = True Then
    If KeihoEmail = 1 And KeihoEmailFlg1 = False Then
        AppendLog "メール送信": Grig_AppendStatus "メール送信"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
    End If
    KeihoEmailFlg = False
    KeihoEmailFlg1 = True
End If
Exit Function
End If

' If Keiho3Intv = 1 Then' 警報タイマー
' If L10DataGal(0) >= MstData.Keiho11 Then
'     Keiho3Count(0) = Keiho3Count(0) + 1
' Else
'     Keiho3Count(0) = 0
' End If

```

```

' If L10DataGal(1) >= MstData.Keiho12 Then
'     Keiho3Count(1) = Keiho3Count(1) + 1
' Else
'     Keiho3Count(1) = 0
' End If
' If L10DataGal(2) >= MstData.Keiho13 Then
'     Keiho3Count(2) = Keiho3Count(2) + 1
' Else
'     Keiho3Count(2) = 0
' End If
' If Keiho3Count(0) < 3 And Keiho3Count(1) < 3 And Keiho3Count(2) < 3 Then Exit Function
' End If
-----注意判別-----
If KeihoFlg1(0) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho11
' If L10DataGal(0) >= MstData.Keiho11 And L10DataGal(0) < MstData.Keiho1 Then
'     CBGraph1.Canvas.Color = RGB(255, 255, 0)
'     Label5(0).BackColor = RGB(255, 255, 0)
'     Timer2.Enabled = True
'     KeihoFlg1(0) = False
'     AcxAio1.WriteBitData(BitNumber as Integer, Flag as Integer) as Integer
'     AcxAio1.WriteBitData 0, 1
'     AppendLog "A 注意報起動": Grig_AppendStatus "A 注意報起動"
'     EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
'     KeihoEmailFlg = True
'     Timer3.Enabled = True
' End If
' If KeihoFlg1(0) = True And L10DataGal(0) < MstData.Keiho11 Then
'     Label5(0).BackColor = RGB(255, 255, 255)
' End If
Else ' 注意報終了処理
' If L10DataGal(0) < MstData.Keiho11 Then
'     CBGraph1.Canvas.Color = RGB(255, 255, 255)
'     Label5(0).BackColor = RGB(255, 255, 255)
'     Timer2.Enabled = False
'     KeihoFlg1(0) = True
'     AcxAio1.WriteBitData(BitNumber as Integer, Flag as Integer) as Integer
'     AcxAio1.WriteBitData 0, 0
'     AppendLog "A 注意報終了": Grig_AppendStatus "A 注意報終了"
'     EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
'     KeihoEmailFlg = False
'     Timer3.Enabled = True
' End If
' Label5(0).BackColor = RGB(255, 255, 0)
End If
If KeihoFlg1(1) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho12
' If L10DataGal(1) >= MstData.Keiho12 And L10DataGal(1) < MstData.Keiho2 Then
'     CBGraph1.Canvas.Color = RGB(255, 255, 0)
'     Label5(1).BackColor = RGB(255, 255, 0)
'     Timer2.Enabled = True
'     KeihoFlg1(1) = False
'     AcxAio1.WriteBitData 0, 1
'     AppendLog "B 注意報起動": Grig_AppendStatus "B 注意報起動"
'     EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
'     KeihoEmailFlg = True
'     Timer3.Enabled = True
' End If
' If KeihoFlg1(1) = True And L10DataGal(1) < MstData.Keiho12 Then
'     Label5(1).BackColor = RGB(255, 255, 255)
' End If

```

```

Else '注意報終了処理
If L10DataGal(1) < MstData.Keiho12 Then
Label5(1).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg1(1) = True
AcxAio1.WriteBitData 0, 0
AppendLog "B 注意報終了": Grig_AppendStatus "B 注意報終了"
EmailSend = Shell("D:\¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
KeihoEmailFlg = False
Timer3.Enabled = True
End If
Label5(1).BackColor = RGB(255, 255, 0)
End If
If KeihoFlg1(2) = True Then '警報判別はMaxDataGal→L10DataGalに変更2005.3.8
'keiho13
If L10DataGal(2) >= MstData.Keiho13 And L10DataGal(2) < MstData.Keiho3 Then
Label5(2).BackColor = RGB(255, 255, 0)
Timer2.Enabled = True
KeihoFlg1(2) = False
AcxAio1.WriteBitData 0, 1
AppendLog "C 注意報起動": Grig_AppendStatus "C 注意報起動"
EmailSend = Shell("d:\¥振動data¥csv_open.bat")
KeihoEmailFlg = True
End If
If KeihoFlg1(2) = True And L10DataGal(2) < MstData.Keiho13 Then
Label5(2).BackColor = RGB(255, 255, 255)
End If
Else '注意報終了処理
If L10DataGal(2) < MstData.Keiho13 Then
Label5(2).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg1(2) = True
AcxAio1.WriteBitData 0, 0
AppendLog "C 注意報終了": Grig_AppendStatus "C 注意報終了"
EmailSend = Shell("d:\¥振動data¥csv_open.bat")
KeihoEmailFlg = False
End If
Label5(2).BackColor = RGB(255, 255, 0)
End If
If KeihoFlg1(3) = True Then '警報判別はMaxDataGal→L10DataGalに変更2005.3.8
'keiho13
If L10DataGal(3) >= MstData.Keiho13 And L10DataGal(3) < MstData.Keiho4 Then
Label5(3).BackColor = RGB(255, 255, 0)
Timer2.Enabled = True
KeihoFlg1(3) = False
AcxAio1.WriteBitData 0, 1
AppendLog "C 注意報起動": Grig_AppendStatus "C 注意報起動"
EmailSend = Shell("d:\¥振動data¥csv_open.bat")
KeihoEmailFlg = True
End If
If KeihoFlg1(3) = True And L10DataGal(3) < MstData.Keiho14 Then
Label5(3).BackColor = RGB(255, 255, 255)
End If
Else '注意報終了処理
If L10DataGal(3) < MstData.Keiho14 Then
Label5(3).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg1(3) = True
AcxAio1.WriteBitData 0, 0
AppendLog "C 注意報終了": Grig_AppendStatus "C 注意報終了"
EmailSend = Shell("d:\¥振動data¥csv_open.bat")
KeihoEmailFlg = False

```

```

End If
Label5(3).BackColor = RGB(255, 255, 0)
End If
'-----警報判別-----
If KeihoFlg(0) = True Then '警報判別はMaxDataGal→L10DataGalに変更2005.3.8
'keiho1
If L10DataGal(0) >= MstData.Keiho1 Then
Timer2.Enabled = False
Label5(0).BackColor = RGB(255, 0, 0)
Timer2.Enabled = True
KeihoFlg(0) = False
AcxAio1.WriteBitData 0, 1
AppendLog "A 警報起動": Grig_AppendStatus "A 警報起動"
EmailSend = Shell("D:\振動dB監視3CH\AutoMail\Adialup_.bat")
KeihoEmailFlg = True
End If
Else '警報終了処理
If L10DataGal(0) < MstData.Keiho1 Then
Timer2.Enabled = False
Label5(0).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg(0) = True
AcxAio1.WriteBitData 0, 0
AppendLog "A 警報終了": Grig_AppendStatus "A 警報終了"
EmailSend = Shell("D:\振動dB監視3CH\AutoMail\Adialup_.bat")
KeihoEmailFlg = False
End If
Label5(0).BackColor = RGB(255, 0, 0)
End If
If KeihoFlg(1) = True Then '警報判別はMaxDataGal→L10DataGalに変更2005.3.8
'keiho2
If L10DataGal(1) >= MstData.Keiho2 Then
Timer2.Enabled = False
Label5(1).BackColor = RGB(255, 0, 0)
Timer2.Enabled = True
KeihoFlg(1) = False
AcxAio1.WriteBitData 0, 1
AppendLog "B 警報起動": Grig_AppendStatus "B 警報起動"
EmailSend = Shell("D:\振動dB監視3CH\AutoMail\Adialup_.bat")
KeihoEmailFlg = True
End If
Else '警報終了処理
If L10DataGal(1) < MstData.Keiho2 Then
Timer2.Enabled = False
Label5(1).BackColor = RGB(255, 255, 255)
Timer2.Enabled = True
KeihoFlg(1) = True
AcxAio1.WriteBitData 0, 0
AppendLog "B 警報終了": Grig_AppendStatus "B 警報終了"
EmailSend = Shell("D:\振動dB監視3CH\AutoMail\Adialup_.bat")
KeihoEmailFlg = False
End If
Label5(1).BackColor = RGB(255, 0, 0)
End If
If KeihoFlg(2) = True Then '警報判別はMaxDataGal→L10DataGalに変更2005.3.8
'keiho2
If L10DataGal(2) >= MstData.Keiho3 Then
Timer2.Enabled = False
Label5(2).BackColor = RGB(255, 0, 0)

```

```

Timer2.Enabled = True
KeihoFlg(2) = False
AcxAio1.WriteBitData 0, 1
AppendLog "C 警報起動": Grig_AppendStatus "C 警報起動"
EmailSend = Shell("d:¥振動data¥csv_open.bat")
KeihoEmailFlg = True
End If
Else ' 警報終了処理
If L10DataGal(2) < MstData.Keiho3 Then
Timer2.Enabled = False
Label5(2).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg(2) = True
AcxAio1.WriteBitData 0, 0
AppendLog "C 警報終了": Grig_AppendStatus "C 警報終了"
EmailSend = Shell("d:¥振動data¥csv_open.bat")
KeihoEmailFlg = False
End If
Label5(2).BackColor = RGB(255, 0, 0)
End If
If KeihoFlg(3) = True Then ' 警報判別はMaxDataGal→L10DataGalに変更2005.3.8
' keiho2
If L10DataGal(3) >= MstData.Keiho4 Then
Timer2.Enabled = False
Label5(3).BackColor = RGB(255, 0, 0)
Timer2.Enabled = True
KeihoFlg(3) = False
AcxAio1.WriteBitData 0, 1
AppendLog "C 警報起動": Grig_AppendStatus "C 警報起動"
EmailSend = Shell("d:¥振動data¥csv_open.bat")
KeihoEmailFlg = True
End If
Else ' 警報終了処理
If L10DataGal(3) < MstData.Keiho4 Then
Timer2.Enabled = False
Label5(3).BackColor = RGB(255, 255, 255)
Timer2.Enabled = False
KeihoFlg(3) = True
AcxAio1.WriteBitData 0, 0
AppendLog "C 警報終了": Grig_AppendStatus "C 警報終了"
EmailSend = Shell("d:¥振動data¥csv_open.bat")
KeihoEmailFlg = False
End If
Label5(3).BackColor = RGB(255, 0, 0)
End If
' 画面色
If KeihoFlg1(0) = True And KeihoFlg1(1) = True And KeihoFlg1(2) = True And KeihoFlg1(3) = True
KeihoFlg(0) = True And KeihoFlg(1) = True And KeihoFlg(2) = True And KeihoFlg(3) = True Then
CBGraph1.Canvas.Color = RGB(255, 255, 255)
KeihoEmailFlg1 = False ' メールフラグ1クリア
ElseIf KeihoFlg(0) = False Or KeihoFlg(1) = False Or KeihoFlg(2) = False Or KeihoFlg(3) = False
If Right(Time, 1) Mod 2 = 1 Then ' フリッカー動作
CBGraph1.Canvas.Color = RGB(255, 0, 0) ' 赤
Else
CBGraph1.Canvas.Color = RGB(255, 255, 255)
End If
ElseIf KeihoFlg1(0) = False Or KeihoFlg1(1) = False Or KeihoFlg1(2) = False Or KeihoFlg1(3) = False
If Right(Time, 1) Mod 2 = 1 Then ' フリッカー動作
CBGraph1.Canvas.Color = RGB(255, 255, 0) ' 黄
Else
CBGraph1.Canvas.Color = RGB(255, 255, 255)

```

```

End If
End If
'-----Emeal-----
If KeihoEmailFlg = True Then
    If KeihoEmail = 1 And KeihoEmailFlg1 = False Then
        AppendLog "Eメール送信": Grig_AppendStatus "Eメール送信"
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
    End If
    KeihoEmailFlg = False
    KeihoEmailFlg1 = True
End If
Exit Function
Error1:
Dim i As Integer
i = i
End Function
Private Sub Timer2_Timer()
Static t2count As Long
Static tempcount As Integer
Exit Sub
Dim EmailSend
On Error GoTo Error1
If CBGraph1.Canvas.Color = RGB(254, 254, 254) Then
    CBGraph1.Canvas.Color = RGB(255, 255, 0)
    tempcount = tempcount + 1
ElseIf CBGraph1.Canvas.Color = RGB(255, 255, 255) Then
    CBGraph1.Canvas.Color = RGB(255, 0, 0)
    tempcount = tempcount + 1
ElseIf CBGraph1.Canvas.Color = RGB(255, 255, 0) Then
    CBGraph1.Canvas.Color = RGB(254, 254, 254)
    tempcount = tempcount + 1
ElseIf CBGraph1.Canvas.Color = RGB(255, 0, 0) Then
    CBGraph1.Canvas.Color = RGB(255, 255, 255)
    tempcount = tempcount + 1
End If
If KeihoEmailFlg = True Then
    KeihoEmailFlg = False
    If KeihoEmail = 1 Then
        EmailSend = Shell("D:¥振動dB監視3CH¥AutoMail¥Adialup_.bat")
    End If
End If
If t2count > IntvPitch * 60 Then
    t2count = 0
    CBGraph1.Canvas.Color = RGB(255, 255, 255)
    AcxAio1.WriteBitData 0, 0
    Timer2.Enabled = False
    AppendLog "警報終了": Grig_AppendStatus "警報終了"
End If
t2count = t2count + 1
Label18.Caption = tempcount
Label20.Caption = t2count
Exit Sub
Error1:
Dim i As Integer
i = i
End Sub
Private Sub Para_Read() 'パラメータを更新
Dim FileNo1 As Integer
Dim dFilename As String
Dim a As String
Dim i As Integer

```

```
Dim j As Integer
On Error GoTo Error1
dFilename = strMaxDataCopyDir & "¥" & PointNo & "_" & "para.txt"

FileNo1 = FreeFile

Open dFilename For Input As #FileNo1
txtOpenFileNum.Caption = Str(FileNo1)
Input #FileNo1, a
Input #FileNo1, i
KeihoCheck = i
Input #FileNo1, a
Input #FileNo1, i
If i = 1 Then
    Command1_Click
End If
If KeihoCheck = 0 Or KeihoCheck = 1 Then SaveSetting App.Title, "Settings", "Keiho", KeihoCheck
Input #FileNo1, a
Input #FileNo1, i
Input #FileNo1, j
IntCopyPitch = i ' 最大値コピーピッチ分
IntCopyMin = j ' 最大値コピー分

Close #FileNo1

Exit Sub
Error1:

Close #FileNo1
End Sub

' Private Sub Timer3_Timer()
' Static t3count As Long
' t3count = t3count + 1
' If t3count > IntvPitch * 60 Then
'     t3count = 0
'
'     KeihoStopFlg = True
'     Timer3.Enabled = False
' End If
' End Sub
```

```

VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
Begin VB.Form frmOptions
  BorderStyle = 3 '固定ダイアログ'
  Caption = "オプション"
  ClientHeight = 5040
  ClientLeft = 45
  ClientTop = 330
  ClientWidth = 6150
  KeyPreview = -1 'True'
  LinkTopic = "Form1"
  MaxButton = 0 'False'
  MinButton = 0 'False'
  ScaleHeight = 5040
  ScaleWidth = 6150
  ShowInTaskbar = 0 'False'
  StartUpPosition = 1 'オーナーフォームの中央'
  Tag = "オプション"
Begin VB.CommandButton cmdOK
  Caption = "OK"
  Height = 375
  Left = 2490
  TabIndex = 1
  Tag = "OK"
  Top = 4455
  Width = 1095
End
Begin VB.CommandButton cmdCancel
  Cancel = -1 'True'
  Caption = "キャンセル"
  Height = 375
  Left = 3720
  TabIndex = 3
  Tag = "キャンセル"
  Top = 4455
  Width = 1095
End
Begin VB.CommandButton cmdApply
  Caption = "適用(&A)"
  Height = 375
  Left = 4920
  TabIndex = 5
  Tag = "適用(&A)"
  Top = 4455
  Width = 1095
End
Begin VB.PictureBox picOptions
  BorderStyle = 0 'なし'
  Height = 3780
  Index = 3
  Left = -20000
  ScaleHeight = 3840.968
  ScaleMode = 0 'ユーザー'
  ScaleWidth = 5745.64
  TabIndex = 7
  TabStop = 0 'False'
  Top = 480
  Width = 5685
Begin VB.Frame fraSample4
  Caption = "サンプル 4"
  Height = 2022
  Left = 505

```

```

VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
Begin VB.Form frmOptions
  BorderStyle = 3 '固定ダイアログ'
  Caption = "オプション"
  ClientHeight = 5040
  ClientLeft = 45
  ClientTop = 330
  ClientWidth = 6150
  KeyPreview = -1 'True'
  LinkTopic = "Form1"
  MaxButton = 0 'False'
  MinButton = 0 'False'
  ScaleHeight = 5040
  ScaleWidth = 6150
  ShowInTaskbar = 0 'False'
  StartUpPosition = 1 'オーナーフォームの中央'
  Tag = "オプション"
Begin VB.CommandButton cmdOK
  Caption = "OK"
  Height = 375
  Left = 2490
  TabIndex = 1
  Tag = "OK"
  Top = 4455
  Width = 1095
End
Begin VB.CommandButton cmdCancel
  Cancel = -1 'True'
  Caption = "キャンセル"
  Height = 375
  Left = 3720
  TabIndex = 3
  Tag = "キャンセル"
  Top = 4455
  Width = 1095
End
Begin VB.CommandButton cmdApply
  Caption = "適用(&A)"
  Height = 375
  Left = 4920
  TabIndex = 5
  Tag = "適用(&A)"
  Top = 4455
  Width = 1095
End
Begin VB.PictureBox picOptions
  BorderStyle = 0 'なし'
  Height = 3780
  Index = 3
  Left = -20000
  ScaleHeight = 3840.968
  ScaleMode = 0 'ユーザー'
  ScaleWidth = 5745.64
  TabIndex = 7
  TabStop = 0 'False'
  Top = 480
  Width = 5685
Begin VB.Frame fraSample4
  Caption = "サンプル 4"
  Height = 2022
  Left = 505

```

```

    TabIndex      = 11
    Tag           = "サンプル 4"
    Top          = 502
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 2
    Left          = -20000
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 9
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685
Begin VB.Frame fraSample3
    Caption       = "サンプル 3"
    Height        = 2022
    Left         = 406
    TabIndex      = 10
    Tag           = "サンプル 3"
    Top          = 403
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 1
    Left          = -20000
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 6
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685
Begin VB.Frame fraSample2
    Caption       = "サンプル 2"
    Height        = 2022
    Left         = 307
    TabIndex      = 8
    Tag           = "サンプル 2"
    Top          = 305
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 0
    Left          = 210
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 2
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685

```

```

    TabIndex      = 11
    Tag           = "サンプル 4"
    Top          = 502
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 2
    Left          = -20000
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 9
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685
Begin VB.Frame fraSample3
    Caption       = "サンプル 3"
    Height        = 2022
    Left         = 406
    TabIndex      = 10
    Tag           = "サンプル 3"
    Top          = 403
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 1
    Left          = -20000
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 6
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685
Begin VB.Frame fraSample2
    Caption       = "サンプル 2"
    Height        = 2022
    Left         = 307
    TabIndex      = 8
    Tag           = "サンプル 2"
    Top          = 305
    Width        = 2033
End
End
Begin VB.PictureBox picOptions
    BorderStyle   = 0 'なし
    Height        = 3780
    Index         = 0
    Left          = 210
    ScaleHeight   = 3840.968
    ScaleMode     = 0 'ユーザー
    ScaleWidth    = 5745.64
    TabIndex      = 2
    TabStop       = 0 'False
    Top           = 480
    Width         = 5685

```

```

Begin VB.Frame fraSample1
    Caption       = "サンプル 1"
    Height       = 2022
    Left        = 208
    TabIndex    = 4
    Tag         = "サンプル 1"
    Top         = 207
    Width       = 2033
End
End
Begin MSComctlLib.TabStrip tbsOptions
    Height       = 4245
    Left        = 105
    TabIndex    = 0
    Top         = 120
    Width       = 5895
    _ExtentX   = 10398
    _ExtentY   = 7488
    _Version   = 393216
    BeginProperty Tabs {1EFB6598-857C-11D1-B16A-00C0F0283628}
        NumTabs = 4
        BeginProperty Tab1 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 1"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab2 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 2"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab3 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 3"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab4 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 4"
            ImageVarType = 2
        EndProperty
    EndProperty
End
End
Attribute VB_Name = "frmOptions"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub cmdApply_Click()
    'ユーザーコード：'cmdApply_Click'用のコードを追加します。
    MsgBox "ダイアログを閉じないで、オプションの設定を適用する処理をここに記述します！"
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    'ユーザーコード：'cmdOK_Click'用のコードを追加します。
    MsgBox "オプションを設定後、ダイアログを閉じる処理をここに記述します！"
    Unload Me
End Sub

```

```

Begin VB.Frame fraSample1
    Caption       = "サンプル 1"
    Height       = 2022
    Left        = 208
    TabIndex    = 4
    Tag         = "サンプル 1"
    Top         = 207
    Width       = 2033
End
End
Begin MSComctlLib.TabStrip tbsOptions
    Height       = 4245
    Left        = 105
    TabIndex    = 0
    Top         = 120
    Width       = 5895
    _ExtentX   = 10398
    _ExtentY   = 7488
    _Version   = 393216
    BeginProperty Tabs {1EFB6598-857C-11D1-B16A-00C0F0283628}
        NumTabs = 4
        BeginProperty Tab1 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 1"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab2 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 2"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab3 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 3"
            ImageVarType = 2
        EndProperty
        BeginProperty Tab4 {1EFB659A-857C-11D1-B16A-00C0F0283628}
            Caption       = "グループ 4"
            ImageVarType = 2
        EndProperty
    EndProperty
End
End
Attribute VB_Name = "frmOptions"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub cmdApply_Click()
    'ユーザーコード：'cmdApply_Click'用のコードを追加します。
    MsgBox "ダイアログを閉じないで、オプションの設定を適用する処理をここに記述します！"
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    'ユーザーコード：'cmdOK_Click'用のコードを追加します。
    MsgBox "オプションを設定後、ダイアログを閉じる処理をここに記述します！"
    Unload Me
End Sub

```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim i As Integer
    i = tbsOptions.SelectedItem.Index
    ' 次のタブに移動するために、ctrl+tab を処理します。
    If (Shift And 3) = 2 And KeyCode = vbKeyTab Then
        If i = tbsOptions.Tabs.Count Then
            ' 最終タブの場合は先頭のタブに移動します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(1)
        Else
            ' タブ位置を増加します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i + 1)
        End If
    ElseIf (Shift And 3) = 3 And KeyCode = vbKeyTab Then
        If i = 1 Then
            ' 最終タブの場合は先頭のタブに移動します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(tbsOptions.Tabs.Count)
        Else
            ' タブ位置を増加します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i - 1)
        End If
    End If
End Sub

```

```
Private Sub tbsOptions_Click()
```

```

    Dim i As Integer
    ' 選択されたタブ コントロールを表示し、有効にします。
    ' ほかのすべてタブ コントロールを非表示にし、無効にします。
    For i = 0 To tbsOptions.Tabs.Count - 1
        If i = tbsOptions.SelectedItem.Index - 1 Then
            picOptions(i).Left = 210
            picOptions(i).Enabled = True
        Else
            picOptions(i).Left = -20000
            picOptions(i).Enabled = False
        End If
    Next
End Sub

```

```
End Sub
```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim i As Integer
    i = tbsOptions.SelectedItem.Index
    ' 次のタブに移動するために、ctrl+tab を処理します。
    If (Shift And 3) = 2 And KeyCode = vbKeyTab Then
        If i = tbsOptions.Tabs.Count Then
            ' 最終タブの場合は先頭のタブに移動します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(1)
        Else
            ' タブ位置を増加します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i + 1)
        End If
    ElseIf (Shift And 3) = 3 And KeyCode = vbKeyTab Then
        If i = 1 Then
            ' 最終タブの場合は先頭のタブに移動します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(tbsOptions.Tabs.Count)
        Else
            ' タブ位置を増加します。
            Set tbsOptions.SelectedItem = tbsOptions.Tabs(i - 1)
        End If
    End If
End Sub

```

```
Private Sub tbsOptions_Click()
```

```

    Dim i As Integer
    ' 選択されたタブ コントロールを表示し、有効にします。
    ' ほかのすべてタブ コントロールを非表示にし、無効にします。
    For i = 0 To tbsOptions.Tabs.Count - 1
        If i = tbsOptions.SelectedItem.Index - 1 Then
            picOptions(i).Left = 210
            picOptions(i).Enabled = True
        Else
            picOptions(i).Left = -20000
            picOptions(i).Enabled = False
        End If
    Next
End Sub

```

```
End Sub
```

VERSION 5.00

Begin VB. Form frmProperty

```

Caption           = "設定"
ClientHeight     = 1650
ClientLeft       = 60
ClientTop        = 345
ClientWidth      = 2565
Icon             = "frmProperty. frx":0000
LinkTopic        = "Form1"
MaxButton        = 0 'False
MinButton        = 0 'False
ScaleHeight      = 1650
ScaleWidth       = 2565
StartPosition    = 1 'オーナー フォームの中央

```

VERSION 5.00

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "comdlg32.ocx"

Begin VB. Form frmProperty

```

Caption           = "設定"
ClientHeight     = 7680
ClientLeft       = 60
ClientTop        = 345
ClientWidth      = 10350
Icon             = "frmProperty. frx":0000
LinkTopic        = "Form1"
MaxButton        = 0 'False
MinButton        = 0 'False
ScaleHeight      = 7680
ScaleWidth       = 10350
StartPosition    = 1 'オーナー フォームの中央

```

Begin VB. Frame Frame9

```

Caption           = "測点No."
Height           = 1035
Left             = 5700
TabIndex        = 99
Top             = 2760
Width           = 3375

```

Begin VB. TextBox txtPointNo

```

Height           = 270
Left            = 780
TabIndex        = 100
Text            = "Text2"
Top            = 360
Width          = 735

```

End

Begin VB. Label Label23

```

Caption           = "(0-9、0は番号なし) 0→200509. c s v      1→1_200509. c s v"
Height           = 615
Left            = 1680
TabIndex        = 102
Top            = 360
Width          = 1575

```

End

Begin VB. Label Label22

```

Caption           = "測点"
Height           = 255
Left            = 180
TabIndex        = 101
Top            = 420
Width          = 675

```

End

End

Begin VB. Frame Frame8

```

Caption           = "グラフ表示ピッチ"
Height           = 1035
Left            = 2580
TabIndex        = 92
Top            = 2760
Width          = 2895

```

Begin VB. OptionButton Option2

```

Caption           = "10分"
Enabled          = 0 'False
Height           = 180
Left            = 300
TabIndex        = 94
Top            = 600
Width          = 1275

```

```
End
Begin VB.OptionButton Option1
    Caption      = "1分"
    Height       = 195
    Left         = 300
    TabIndex     = 93
    Top          = 300
    Value        = -1 ' True
    Width        = 1995
End
End
Begin VB.Frame Frame7
    Caption      = "時間率の種類"
    Height       = 975
    Left         = 2580
    TabIndex     = 75
    Top          = 1620
    Width        = 2895
    Begin VB.OptionButton Op_Lv10
        Caption    = "時間率振動レベル Lv 10[dB]"
        Height     = 180
        Left       = 240
        TabIndex   = 77
        Top        = 600
        Width      = 2535
    End
    Begin VB.OptionButton Op_Lvmax
        Caption    = "時間率振動レベル Lv max[dB]"
        Enabled    = 0 ' False
        Height     = 180
        Left       = 240
        TabIndex   = 76
        Top        = 240
        Width      = 2535
    End
End
End
Begin MSComDlg.CommonDialog CommonDialog1
    Left        = 9540
    Top         = 6600
    _ExtentX    = 847
    _ExtentY    = 847
    _Version    = 393216
End
Begin VB.CommandButton Command1
    Caption     = "参照"
    Height      = 255
    Left        = 8400
    TabIndex    = 73
    Top         = 6900
    Width       = 615
End
Begin VB.TextBox Text1
    Height      = 270
    Left        = 2520
    TabIndex    = 72
    Text        = "Text1"
    Top         = 6900
    Width       = 5775
End
Begin VB.CheckBox Check1
    Caption     = "Maxデータ転送ディレクトリ"
    Height      = 255
```

```
Left = 60
TabIndex = 71
Top = 6900
Width = 2415
End
Begin VB.Frame Frame6
Caption = "生データ収録情報"
Height = 1995
Left = 120
TabIndex = 63
Top = 2520
Width = 2355
Begin VB.TextBox txtRecDataSec
Height = 270
Left = 1620
TabIndex = 69
Text = "Text1"
Top = 1080
Width = 615
End
Begin VB.TextBox txtRecPreDataSec
Height = 270
Left = 1620
TabIndex = 68
Text = "Text1"
Top = 660
Width = 615
End
Begin VB.TextBox txtSampSec
Height = 270
Left = 1620
TabIndex = 67
Text = "Text1"
Top = 300
Width = 615
End
Begin VB.Label Label21
Caption = "最大データ個数は100000まで"
Height = 435
Left = 180
TabIndex = 70
Top = 1500
Width = 1455
End
Begin VB.Label Label20
Caption = "収録秒数 [s] (プリトリガ含む)"
Height = 435
Left = 180
TabIndex = 66
Top = 960
Width = 1155
End
Begin VB.Label Label19
Caption = "プリトリガ秒数[s]"
Height = 195
Left = 180
TabIndex = 65
Top = 660
Width = 1335
End
Begin VB.Label Label18
Caption = "サンプリングタイム[s]"
```

```
        Height      = 255
        Left        = 180
        TabIndex    = 64
        Top         = 360
        Width       = 1275
    End
End
Begin VB.Frame Frame5
Caption      = "グラフ表示"
Height      = 2415
Left        = 5640
TabIndex    = 50
Top         = 120
Width       = 4275
    Begin VB.TextBox txtChName
        Height      = 270
        Index       = 3
        Left        = 840
        TabIndex    = 98
        Text        = "CH4"
        Top         = 1440
        Width       = 1695
    End
    Begin VB.TextBox txtChName
        Height      = 270
        Index       = 2
        Left        = 840
        TabIndex    = 97
        Text        = "CH3"
        Top         = 1080
        Width       = 1695
    End
    Begin VB.TextBox txtChName
        Height      = 270
        Index       = 1
        Left        = 840
        TabIndex    = 96
        Text        = "CH2"
        Top         = 720
        Width       = 1695
    End
    Begin VB.TextBox txtChName
        Height      = 270
        Index       = 0
        Left        = 840
        TabIndex    = 95
        Text        = "CH1"
        Top         = 360
        Width       = 1695
    End
    Begin VB.CheckBox OverOff
Caption      = "管理値オーバーは非表示"
Height      = 195
Left        = 1860
TabIndex    = 89
Top         = 1860
Width       = 2295
    End
    Begin VB.OptionButton Op_OverWrite2
Caption      = "0時でクリア"
Height      = 180
Left        = 300
```

```

        TabIndex      = 79
        Top           = 2100
        Width        = 1335
    End
Begin VB.OptionButton Op_OverWrite1
    Caption          = "上書きプロット"
    Height          = 195
    Left            = 300
    TabIndex        = 78
    Top             = 1860
    Width          = 1455
End
Begin VB.TextBox txtGraphYScale
    Alignment        = 1 ' 右揃え
    Height          = 270
    Index           = 3
    Left            = 2640
    TabIndex        = 62
    Text            = "60"
    Top             = 1440
    Width          = 855
End
Begin VB.TextBox txtGraphYScale
    Alignment        = 1 ' 右揃え
    Height          = 270
    Index           = 2
    Left            = 2640
    TabIndex        = 61
    Text            = "60"
    Top             = 1080
    Width          = 855
End
Begin VB.TextBox txtGraphYScale
    Alignment        = 1 ' 右揃え
    Height          = 270
    Index           = 1
    Left            = 2640
    TabIndex        = 60
    Text            = "60"
    Top             = 720
    Width          = 855
End
Begin VB.TextBox txtGraphYScale
    Alignment        = 1 ' 右揃え
    Height          = 270
    Index           = 0
    Left            = 2640
    TabIndex        = 59
    Text            = "60"
    Top             = 360
    Width          = 855
End
Begin VB.Label Label17
    Caption          = "dB"
    Height          = 255
    Index           = 7
    Left            = 3720
    TabIndex        = 58
    Top             = 1440
    Width          = 375
End
Begin VB.Label Label17

```

```
Caption      = "dB"  
Height      = 255  
Index       = 6  
Left        = 3720  
TabIndex    = 57  
Top         = 1080  
Width       = 375  
End  
Begin VB.Label Label17  
Caption      = "dB"  
Height      = 255  
Index       = 5  
Left        = 3720  
TabIndex    = 56  
Top         = 720  
Width       = 375  
End  
Begin VB.Label Label17  
Caption      = "dB"  
Height      = 255  
Index       = 4  
Left        = 3720  
TabIndex    = 55  
Top         = 360  
Width       = 375  
End  
Begin VB.Label Label17  
Caption      = "测点 4"  
Height      = 255  
Index       = 3  
Left        = 240  
TabIndex    = 54  
Top         = 1440  
Width       = 615  
End  
Begin VB.Label Label17  
Caption      = "测点 3"  
Height      = 255  
Index       = 2  
Left        = 240  
TabIndex    = 53  
Top         = 1080  
Width       = 615  
End  
Begin VB.Label Label17  
Caption      = "测点 2"  
Height      = 255  
Index       = 1  
Left        = 240  
TabIndex    = 52  
Top         = 720  
Width       = 615  
End  
Begin VB.Label Label17  
Caption      = "测点 1"  
Height      = 255  
Index       = 0  
Left        = 240  
TabIndex    = 51  
Top         = 360  
Width       = 615  
End
```

```
End
Begin VB.Frame Frame4
Caption      = "警報管理値"
Height      = 2655
Left        = 2580
TabIndex    = 32
Top         = 3840
Width       = 5715
Begin VB.CheckBox ChkKeihoEMail
Caption      = "警報Eメールをする"
Height      = 255
Left        = 2400
TabIndex    = 91
Top         = 2280
Width       = 2535
End
Begin VB.CheckBox Chk3Intv
Caption      = "3連続管理値オーバーで警報"
Height      = 195
Left        = 2400
TabIndex    = 90
Top         = 2040
Width       = 2655
End
Begin VB.TextBox txtKeiho14
Alignment    = 1 ' 右揃え
Height      = 270
Left        = 2340
TabIndex    = 84
Text        = "999"
Top         = 1680
Width       = 615
End
Begin VB.TextBox txtKeiho13
Alignment    = 1 ' 右揃え
Height      = 255
Left        = 2340
TabIndex    = 83
Text        = "999"
Top         = 1320
Width       = 615
End
Begin VB.TextBox txtKeiho12
Alignment    = 1 ' 右揃え
Height      = 270
Left        = 2340
TabIndex    = 82
Text        = "999"
Top         = 960
Width       = 615
End
Begin VB.TextBox txtKeiho11
Alignment    = 1 ' 右揃え
Height      = 270
Left        = 2340
TabIndex    = 81
Text        = "999"
Top         = 600
Width       = 615
End
Begin VB.CheckBox chkKeiho
Caption      = "警報を表示させない"
```

```
Height = 255
Left = 120
TabIndex = 49
Top = 2040
Width = 1995
End
Begin VB.TextBox txtKeiho4
Alignment = 1 '右揃え
Height = 270
Left = 840
TabIndex = 48
Text = "999"
Top = 1680
Width = 615
End
Begin VB.TextBox txtKeiho3
Alignment = 1 '右揃え
Height = 270
Left = 840
TabIndex = 47
Text = "999"
Top = 1320
Width = 615
End
Begin VB.TextBox txtKeiho2
Alignment = 1 '右揃え
Height = 270
Left = 840
TabIndex = 46
Text = "999"
Top = 960
Width = 615
End
Begin VB.TextBox txtKeiho1
Alignment = 1 '右揃え
Height = 270
Left = 840
TabIndex = 45
Text = "999"
Top = 600
Width = 615
End
Begin VB.Label Label16
Caption = "dB"
Height = 255
Index = 7
Left = 3000
TabIndex = 88
Top = 1740
Width = 315
End
Begin VB.Label Label16
Caption = "dB"
Height = 255
Index = 6
Left = 3000
TabIndex = 87
Top = 1380
Width = 315
End
Begin VB.Label Label16
Caption = "dB"
```

```

        Height      = 255
        Index       = 5
        Left        = 3000
        TabIndex    = 86
        Top         = 1020
        Width       = 315
    End
Begin VB.Label Label16
    Caption        = "dB"
    Height         = 255
    Index         = 4
    Left          = 3000
    TabIndex      = 85
    Top           = 660
    Width         = 315
End
Begin VB.Label Label15
    Caption        = "2次管理値（赤）          1次管理値（黄）"
    Height         = 255
    Index         = 4
    Left          = 180
    TabIndex      = 80
    Top           = 300
    Width         = 5175
End
Begin VB.Label Label16
    Caption        = "dB"
    Height         = 255
    Index         = 3
    Left          = 1500
    TabIndex      = 44
    Top           = 1680
    Width         = 315
End
Begin VB.Label Label16
    Caption        = "dB"
    Height         = 255
    Index         = 2
    Left          = 1500
    TabIndex      = 43
    Top           = 1380
    Width         = 255
End
Begin VB.Label Label16
    Caption        = "dB"
    Height         = 255
    Index         = 1
    Left          = 1500
    TabIndex      = 42
    Top           = 1020
    Width         = 315
End
Begin VB.Label Label16
    Caption        = "dB"
    Height         = 255
    Index         = 0
    Left          = 1500
    TabIndex      = 41
    Top           = 660
    Width         = 315
End
Begin VB.Label Label15

```

```

Caption      = "測点 4"
Height      = 255
Index       = 3
Left        = 120
TabIndex    = 40
Top         = 1680
Width       = 495
End
Begin VB.Label Label15
Caption     = "測点 3"
Height     = 255
Index      = 2
Left       = 120
TabIndex   = 39
Top        = 1320
Width      = 495
End
Begin VB.Label Label15
Caption     = "測点 2"
Height     = 255
Index      = 1
Left       = 120
TabIndex   = 38
Top        = 960
Width      = 495
End
Begin VB.Label Label15
Caption     = "測点 1"
Height     = 255
Index      = 0
Left       = 120
TabIndex   = 37
Top        = 600
Width      = 495
End
End
Begin VB.Frame Frame3
Caption     = "時間率評価時間"
Height     = 1455
Left       = 2580
TabIndex   = 28
Top        = 60
Width      = 2895
Begin VB.OptionButton maxGalIntvPitch3
Caption     = "1時間間隔"
Enabled     = 0
False
Height     = 375
Left       = 240
TabIndex   = 31
Top        = 960
Width      = 1695
End
End
Begin VB.OptionButton maxGalIntvPitch2
Caption     = "10分間隔"
Height     = 375
Left       = 240
TabIndex   = 30
Top        = 600
Width      = 1575
End
End
Begin VB.OptionButton maxGalIntvPitch1
Caption     = "1分間隔"

```

```

        Enabled      = 0 ' False
        Height       = 375
        Left         = 240
        TabIndex     = 29
        Top          = 240
        Width        = 1695
    End
End
Begin VB.Frame Frame2
Caption       = "変換レンジ"
Height       = 1935
Left         = 120
TabIndex     = 19
Top          = 4620
Width        = 2355
Begin VB.TextBox txtGalVRange4
Alignment    = 1 ' 右揃え
Height       = 270
Left         = 840
TabIndex     = 23
Text         = "100"
Top          = 1440
Width        = 855
End
Begin VB.TextBox txtGalVRange3
Alignment    = 1 ' 右揃え
Height       = 270
Left         = 840
TabIndex     = 22
Text         = "100"
Top          = 1080
Width        = 855
End
Begin VB.TextBox txtGalVRange2
Alignment    = 1 ' 右揃え
Height       = 270
Left         = 840
TabIndex     = 21
Text         = "100"
Top          = 660
Width        = 855
End
Begin VB.TextBox txtGalVRange1
Alignment    = 1 ' 右揃え
Height       = 270
Left         = 840
TabIndex     = 20
Text         = "100"
Top          = 240
Width        = 855
End
Begin VB.Label Label14
Caption      = "測点 4"
Height      = 255
Index       = 3
Left        = 120
TabIndex    = 36
Top         = 1440
Width       = 615
End
Begin VB.Label Label14
Caption      = "測点 3"

```

```
Height = 255
Index = 2
Left = 120
TabIndex = 35
Top = 1080
Width = 615
End
Begin VB.Label Label14
Caption = "测点 2"
Height = 255
Index = 1
Left = 120
TabIndex = 34
Top = 720
Width = 615
End
Begin VB.Label Label14
Caption = "测点 1"
Height = 255
Index = 0
Left = 120
TabIndex = 33
Top = 360
Width = 615
End
Begin VB.Label Label13
Caption = "dB"
Height = 255
Index = 3
Left = 1920
TabIndex = 27
Top = 1440
Width = 255
End
Begin VB.Label Label13
Caption = "dB"
Height = 255
Index = 2
Left = 1920
TabIndex = 26
Top = 1080
Width = 315
End
Begin VB.Label Label13
Caption = "dB"
Height = 255
Index = 1
Left = 1920
TabIndex = 25
Top = 660
Width = 315
End
Begin VB.Label Label13
Caption = "dB"
Height = 255
Index = 0
Left = 1920
TabIndex = 24
Top = 240
Width = 255
End
```

```
End
End
```

```

Begin VB.Frame Frame1
Caption      = "トリガーレベル"
Height      = 990
Left        = 60
TabIndex    = 2
Top         = 90
Width       = 2400

```

```

Begin VB.Frame Frame1
Caption      = "トリガーレベル"
Height      = 2310
Left        = 60
TabIndex    = 2
Top         = 90
Width       = 2400

```

```

Begin VB.OptionButton TeijiKansoku
Caption      = "定時観測 (毎1時間)"
Height      = 255
Left        = 240
TabIndex    = 74
Top         = 1800
Width       = 1935

```

```

End
Begin VB.TextBox txtTrgLevel4
Alignment    = 1 '右揃え
Height      = 270
Left        = 840
TabIndex    = 18
Text        = "10.0"
Top         = 1440
Width       = 495

```

```

End
Begin VB.TextBox txtTrgLevel3
Alignment    = 1 '右揃え
Height      = 270
Left        = 840
TabIndex    = 17
Text        = "10.0"
Top         = 1080
Width       = 495

```

```

End
Begin VB.TextBox txtTrgLevel2
Alignment    = 1 '右揃え
Height      = 270
Left        = 825
TabIndex    = 6
Text        = "10.0"
Top         = 615
Width       = 495

```

```

End
Begin VB.TextBox txtTrgLevel1
Alignment    = 1 '右揃え
Height      = 270
Left        = 825
TabIndex    = 3
Text        = "10.0"
Top         = 255
Width       = 495

```

```

End
Begin VB.Label Label12
Caption      = "(dB)"
Height      = 255
Left        = 1620
TabIndex    = 16
Top         = 1440
Width       = 495

```

```

End
Begin VB.Label Label11
Caption      = "(dB)"
Height      = 255

```

```

Begin VB.TextBox txtTrgLevel2
Alignment    = 1 '右揃え
Height      = 270
Left        = 825
TabIndex    = 6
Text        = "10.0"
Top         = 615
Width       = 495

```

```

End
Begin VB.TextBox txtTrgLevel1
Alignment    = 1 '右揃え
Height      = 270
Left        = 825
TabIndex    = 3
Text        = "10.0"
Top         = 255
Width       = 495

```

```

End

```

```

Begin VB.Label Label4
    AutoSize      = -1 ' True
    Caption       = "测点 2"
    Height        = 180
    Left          = 210
    TabIndex      = 10
    Top           = 675
    Width         = 480
End
Begin VB.Label Label3
    AutoSize      = -1 ' True
    Caption       = "测点 1"
    Height        = 180
    Left          = 210
    TabIndex      = 9
    Top           = 315
    Width         = 480
End
Begin VB.Label Label2
    AutoSize      = -1 ' True
    Caption       = "√"
    Height        = 180
    Left          = 1410
    TabIndex      = 8
    Top           = 675

```

```

    Left          = 1620
    TabIndex      = 15
    Top           = 1080
    Width         = 495
End
Begin VB.Label Label10
    Caption       = "√"
    Height        = 255
    Left          = 1410
    TabIndex      = 14
    Top           = 1440
    Width         = 255
End
Begin VB.Label Label8
    Caption       = "√"
    Height        = 255
    Left          = 1410
    TabIndex      = 13
    Top           = 1080
    Width         = 255
End
Begin VB.Label Label7
    Caption       = "测点 4"
    Height        = 255
    Left          = 240
    TabIndex      = 12
    Top           = 1440
    Width         = 615
End
Begin VB.Label Label5
    Caption       = "测点 3"
    Height        = 255
    Left          = 240
    TabIndex      = 11
    Top           = 1080
    Width         = 615
End
Begin VB.Label Label4
    AutoSize      = -1 ' True
    Caption       = "测点 2"
    Height        = 180
    Left          = 210
    TabIndex      = 10
    Top           = 675
    Width         = 480
End
Begin VB.Label Label3
    AutoSize      = -1 ' True
    Caption       = "测点 1"
    Height        = 180
    Left          = 210
    TabIndex      = 9
    Top           = 315
    Width         = 480
End
Begin VB.Label Label2
    AutoSize      = -1 ' True
    Caption       = "√"
    Height        = 180
    Left          = 1410
    TabIndex      = 8
    Top           = 675

```

```

Width = 135
End
Begin VB.Label Label1
AutoSize = -1 ' True
Caption = "(kine)"
Height = 180
Left = 1620
TabIndex = 7
Top = 675
Width = 435

```

```

End
Begin VB.Label Label6
AutoSize = -1 ' True
Caption = "√"
Height = 180
Left = 1410
TabIndex = 5
Top = 315
Width = 135

```

```

End
Begin VB.Label Label9
AutoSize = -1 ' True
Caption = "(kine)"
Height = 180
Left = 1620
TabIndex = 4
Top = 315
Width = 435

```

```

End
End
Begin VB.CommandButton cmdOK
Caption = "OK"
BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 ' False
Italic = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height = 375
Left = 195
TabIndex = 1
Top = 1185
Width = 1095

```

```

End
Begin VB.CommandButton cmdCancel
Caption = "キャンセル"
BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 ' False
Italic = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height = 375
Left = 1365
TabIndex = 0
Top = 1185

```

```

Width = 135
End
Begin VB.Label Label1
AutoSize = -1 ' True
Caption = "(dB)"
Height = 180
Left = 1620
TabIndex = 7
Top = 675
Width = 345

```

```

End
Begin VB.Label Label6
AutoSize = -1 ' True
Caption = "√"
Height = 180
Left = 1410
TabIndex = 5
Top = 315
Width = 135

```

```

End
Begin VB.Label Label9
AutoSize = -1 ' True
Caption = "(dB)"
Height = 180
Left = 1620
TabIndex = 4
Top = 315
Width = 345

```

```

End
End
Begin VB.CommandButton cmdOK
Caption = "OK"
BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 ' False
Italic = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height = 375
Left = 240
TabIndex = 1
Top = 7260
Width = 1095

```

```

End
Begin VB.CommandButton cmdCancel
Caption = "キャンセル"
BeginProperty Font
Name = "MS Pゴシック"
Size = 9.75
Charset = 128
Weight = 400
Underline = 0 ' False
Italic = 0 ' False
Strikethrough = 0 ' False
EndProperty
Height = 375
Left = 1500
TabIndex = 0
Top = 7260

```

```

Width = 1095
End
End
Attribute VB_Name = "frmProperty"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

```

```

Private Sub CmdCancel_Click()
Unload Me
End Sub
Private Sub cmdOK_Click()

```

```

MstData.TrgLevel1 = txtTrgLevel1
MstData.TrgLevel2 = txtTrgLevel2

```

```

Width = 1095
End
End
Attribute VB_Name = "frmProperty"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

```

```

Private tempSampSec0 As Single
Private tempSampSec As Single
Private Sub CmdCancel_Click()
Unload Me
End Sub
Private Sub cmdOK_Click()

```

```

MstData.TrgLevel1 = txtTrgLevel1
MstData.TrgLevel2 = txtTrgLevel2
MstData.TrgLevel3 = txtTrgLevel3
MstData.TrgLevel4 = txtTrgLevel4

```

```

MstData.GalVRange1 = txtGalVRange1
MstData.GalVRange2 = txtGalVRange2
MstData.GalVRange3 = txtGalVRange3
MstData.GalVRange4 = txtGalVRange4

```

```

MstData.Keiho1 = txtKeiho1
MstData.Keiho2 = txtKeiho2
MstData.Keiho3 = txtKeiho3
MstData.Keiho4 = txtKeiho4
MstData.Keiho11 = txtKeiho11
MstData.Keiho12 = txtKeiho12
MstData.Keiho13 = txtKeiho13
MstData.Keiho14 = txtKeiho14

```

```

ChName(0) = txtChName(0)
ChName(1) = txtChName(1)
ChName(2) = txtChName(2)
ChName(3) = txtChName(3)

```

```

If (Int(Val(txtGraphYScale(0).Text)) >= 1 And _
Int(Val(txtGraphYScale(1).Text)) >= 1 And _
Int(Val(txtGraphYScale(2).Text)) >= 1 And _
Int(Val(txtGraphYScale(3).Text)) >= 1) Then
GraphYScale(0) = Int(Val(txtGraphYScale(0).Text))
GraphYScale(1) = Int(Val(txtGraphYScale(1).Text))
GraphYScale(2) = Int(Val(txtGraphYScale(2).Text))
GraphYScale(3) = Int(Val(txtGraphYScale(3).Text))
GraphYScaleCount(0) = 1 + Left(GraphYScale(0), 1)
If GraphYScaleCount(0) = 2 Then GraphYScaleCount(0) = 11
GraphYScaleCount(1) = 1 + Left(GraphYScale(1), 1)
If GraphYScaleCount(1) = 2 Then GraphYScaleCount(1) = 11
GraphYScaleCount(2) = 1 + Left(GraphYScale(2), 1)
If GraphYScaleCount(2) = 2 Then GraphYScaleCount(2) = 11
GraphYScaleCount(3) = 1 + Left(GraphYScale(3), 1)
If GraphYScaleCount(3) = 2 Then GraphYScaleCount(3) = 11
If 0.001 <= txtSampSec <= 0.1 Then MstData.SampSec = txtSampSec
If 1 <= txtRecPreDataSec <= 100 Then MstData.RecPreDataSec = txtRecPreDataSec
If 2 <= txtRecDataSec <= 200 And txtRecDataSec > txtRecPreDataSec Then _
MstData.RecDataSec = txtRecDataSec

```

```

intCopyFlg = Check1.Value
strMaxDataCopyDir = Text1.Text
frmMain.Label16.Caption = strMaxDataCopyDir
If intCopyFlg = 0 Then frmMain.Label16.Caption = ""

```

```

SaveSetting App.Title, "Settings", "GraphYScale1", GraphYScale(0)
SaveSetting App.Title, "Settings", "GraphYScale2", GraphYScale(1)
SaveSetting App.Title, "Settings", "GraphYScale3", GraphYScale(2)
SaveSetting App.Title, "Settings", "GraphYScale4", GraphYScale(3)

```

```

SaveSetting App.Title, "Settings", "GraphYScaleCount1", GraphYScaleCount(0)
SaveSetting App.Title, "Settings", "GraphYScaleCount2", GraphYScaleCount(1)
SaveSetting App.Title, "Settings", "GraphYScaleCount3", GraphYScaleCount(2)
SaveSetting App.Title, "Settings", "GraphYScaleCount4", GraphYScaleCount(3)

```

```
End If
```

```

SaveSetting App.Title, "Settings", "TrgLevel1", MstData.TrgLevel1
SaveSetting App.Title, "Settings", "TrgLevel2", MstData.TrgLevel2

```

```

SaveSetting App.Title, "Settings", "TrgLevel1", MstData.TrgLevel1
SaveSetting App.Title, "Settings", "TrgLevel2", MstData.TrgLevel2
SaveSetting App.Title, "Settings", "TrgLevel3", MstData.TrgLevel3
SaveSetting App.Title, "Settings", "TrgLevel4", MstData.TrgLevel4

```

```

SaveSetting App.Title, "Settings", "GalVRange1", MstData.GalVRange1
SaveSetting App.Title, "Settings", "GalVRange2", MstData.GalVRange2
SaveSetting App.Title, "Settings", "GalVRange3", MstData.GalVRange3
SaveSetting App.Title, "Settings", "GalVRange4", MstData.GalVRange4

```

```

SaveSetting App.Title, "Settings", "IntvPitch", IntvPitch
SaveSetting App.Title, "Settings", "Keiho", chkKeiho.Value
SaveSetting App.Title, "Settings", "KeihoOffDisp", OverOff.Value
SaveSetting App.Title, "Settings", "KeihoEmail", ChkKeihoEMail.Value
SaveSetting App.Title, "Settings", "Keiho3Intv", Chk3Intv.Value
SaveSetting App.Title, "Settings", "IntLvX", IntLvX

```

```

SaveSetting App.Title, "Settings", "Keiho1", MstData.Keiho1
SaveSetting App.Title, "Settings", "Keiho2", MstData.Keiho2
SaveSetting App.Title, "Settings", "keiho3", MstData.Keiho3
SaveSetting App.Title, "Settings", "keiho4", MstData.Keiho4
SaveSetting App.Title, "Settings", "Keiho11", MstData.Keiho11
SaveSetting App.Title, "Settings", "Keiho12", MstData.Keiho12
SaveSetting App.Title, "Settings", "keiho13", MstData.Keiho13
SaveSetting App.Title, "Settings", "keiho14", MstData.Keiho14

```

```

SaveSetting App.Title, "Settings", "SampSec", MstData.SampSec
SaveSetting App.Title, "Settings", "RecPreDataSec", MstData.RecPreDataSec
SaveSetting App.Title, "Settings", "RecDataSec", MstData.RecDataSec
SaveSetting App.Title, "Settings", "intCopyFlg", intCopyFlg
SaveSetting App.Title, "Settings", "strMaxDataCopyDir", strMaxDataCopyDir

```

```

SaveSetting App.Title, "Settings", "dBTeiji", dBTeiji
SaveSetting App.Title, "Settings", "GraphOverWrite", GraphOverWrite

```

```

SaveSetting App.Title, "Settings", "ChName1", ChName(0)
SaveSetting App.Title, "Settings", "ChName2", ChName(1)
SaveSetting App.Title, "Settings", "ChName3", ChName(2)
SaveSetting App.Title, "Settings", "ChName4", ChName(3)

```

```

KeihoCheck = chkKeiho.Value
KeihoOffDisp = OverOff.Value

```

```
Keiho3Intv = Chk3Intv.Value
KeihoEmail = ChkKeihoEMail.Value
```

```
frmMain.Label7.Caption = IntvPitch
If IntLvX = 0 Then
    frmMain.Label8.Caption = "時間率振動レベル Lv10[dB]"
ElseIf IntLvX = 1 Then
    frmMain.Label8.Caption = "時間率振動レベル Lvmax[dB]"
End If
tempSampSec = MstData.SampSec
If tempSampSec <> tempSampSec0 Then
    MsgBox "生データ収録情報を変更するときはプログラムの再起動を行って下さい。"
End If
If OverOff.Value = 1 Then
    frmMain.Label5(0).Visible = False
    frmMain.Label5(1).Visible = False
    frmMain.Label5(2).Visible = False
    frmMain.Label5(3).Visible = False
Else
    frmMain.Label5(0).Visible = True
    frmMain.Label5(1).Visible = True
    frmMain.Label5(2).Visible = True
    frmMain.Label5(3).Visible = True
End If
If txtPointNo.Text > 0 And txtPointNo.Text < 10 Then
    PointNo = txtPointNo.Text
    SaveSetting App.Title, "Settings", "PointNo", PointNo
End If
```

```
Unload Me
End Sub
```

```
Unload Me
End Sub
```

```
Private Sub Command1_Click() '参照
Dim a As String
Dim i As Integer
On Error GoTo Error1
CommonDialog1.InitDir = Text1.Text
CommonDialog1.ShowOpen
a = CommonDialog1.FileName
i = Len(a)
If i < 3 Then Exit Sub
Do While Mid(a, i, 1) <> "¥"
    a = Mid(a, 1, i - 1)
    i = i - 1
Loop
Text1.Text = Mid(a, 1, i - 1)
Exit Sub
Error1:
i = i
End Sub
```

```
Private Sub Form_Load()
```

```
Private Sub Form_Load()
TeijiKansoku.Value = dBTeiji

If TeijiKansoku.Value = True Then
    txtSampSec.Text = 0.1
    txtRecPreDataSec.Text = 0
    txtRecDataSec.Text = 3600 - 10
    tempSampSec0 = 0.1

MstData.SampSec = txtSampSec.Text
```

```

MstData.RecPreDataSec = txtRecPreDataSec.Text
MstData.RecDataSec = txtRecDataSec.Text
MstData.SampSec = tempSampSec0

```

```

txtTrgLevel1 = 10
txtTrgLevel2 = 10
txtTrgLevel3 = 10
txtTrgLevel4 = 10
MstData.TrgLevel1 = txtTrgLevel1
MstData.TrgLevel2 = txtTrgLevel2
MstData.TrgLevel3 = txtTrgLevel3
MstData.TrgLevel4 = txtTrgLevel4

```

```
End If
```

```

txtTrgLevel1 = MstData.TrgLevel1
txtTrgLevel2 = MstData.TrgLevel2

```

```

txtTrgLevel1 = MstData.TrgLevel1
txtTrgLevel2 = MstData.TrgLevel2
txtTrgLevel3 = MstData.TrgLevel3
txtTrgLevel4 = MstData.TrgLevel4

```

```

txtGalVRange1 = MstData.GalVRange1
txtGalVRange2 = MstData.GalVRange2
txtGalVRange3 = MstData.GalVRange3
txtGalVRange4 = MstData.GalVRange4

```

```

txtKeiho1 = MstData.Keiho1
txtKeiho2 = MstData.Keiho2
txtKeiho3 = MstData.Keiho3
txtKeiho4 = MstData.Keiho4
txtKeiho11 = MstData.Keiho11
txtKeiho12 = MstData.Keiho12
txtKeiho13 = MstData.Keiho13
txtKeiho14 = MstData.Keiho14

```

```

chkKeiho.Value = KeihoCheck
OverOff.Value = KeihoOffDisp
Chk3Intv.Value = Keiho3Intv
ChkKeihoEMail.Value = KeihoEmail

```

```

txtGraphYScale(0).Text = GraphYScale(0)
txtGraphYScale(1).Text = GraphYScale(1)
txtGraphYScale(2).Text = GraphYScale(2)
txtGraphYScale(3).Text = GraphYScale(3)
txtSampSec.Text = MstData.SampSec
txtRecPreDataSec.Text = MstData.RecPreDataSec
txtRecDataSec.Text = MstData.RecDataSec
tempSampSec0 = MstData.SampSec

```

```

If IntvPitch = 1 Then
    maxGalIntvPitch1.Value = True
    maxGalIntvPitch2.Value = False
    maxGalIntvPitch3.Value = False
ElseIf IntvPitch = 10 Then
    maxGalIntvPitch1.Value = False
    maxGalIntvPitch2.Value = True
    maxGalIntvPitch3.Value = False

```

```

ElseIf IntvPitch = 60 Then
    maxGalIntvPitch1.Value = False
    maxGalIntvPitch2.Value = False

```

```

        maxGalIntvPitch3.Value = True
Else
    maxGalIntvPitch1.Value = False
    maxGalIntvPitch2.Value = True
    maxGalIntvPitch3.Value = False
End If
If IntLvX = 0 Then
    Op_Lv10.Value = True
    Op_Lvmax.Value = False
ElseIf IntLvX = 1 Then
    Op_Lv10.Value = False
    Op_Lvmax.Value = True
End If
Check1.Value = intCopyFlg
Text1.Text = strMaxDataCopyDir
If GraphOverWrite = 0 Then
    Op_OverWrite1.Value = True
    Op_OverWrite2.Value = False
ElseIf GraphOverWrite = 1 Then
    Op_OverWrite1.Value = False
    Op_OverWrite2.Value = True
End If
If chkKeiho.Value = 1 Then
    Chk3Intv.Enabled = False
    ChkKeihoEMail.Enabled = False
Else
    Chk3Intv.Enabled = True
    ChkKeihoEMail.Enabled = True
End If
txtChName(0) = ChName(0)
txtChName(1) = ChName(1)
txtChName(2) = ChName(2)
txtChName(3) = ChName(3)

txtPointNo.Text = PointNo
End Sub

Private Sub maxGalIntvPitch1_Click() ' 最大加速度収録ピッチの選択
If maxGalIntvPitch1.Value = True Then
    IntvPitch = 1
End If
End Sub
Private Sub maxGalIntvPitch2_Click()
If maxGalIntvPitch2.Value = True Then
    IntvPitch = 10
End If
End Sub
Private Sub maxGalIntvPitch3_Click()
If maxGalIntvPitch3.Value = True Then
    IntvPitch = 60
End If
End Sub

Private Sub Op_Lv10_Click() ' Lvmax, Lv10切り替え
If Op_Lv10.Value = True Then
    IntLvX = 0
End If
End Sub
Private Sub Op_Lvmax_Click() ' Lvmax, Lv10切り替え

```

```
End Sub
If Op_Lvmax.Value = True Then
    IntLvX = 1
End If
End Sub
Private Sub Op_OverWrite1_Click() 'Lvmax, Lv10切り替え
If Op_OverWrite1.Value = True Then
    GraphOverWrite = 0
End If
End Sub
Private Sub Op_OverWrite2_Click() 'Lvmax, Lv10切り替え
If Op_OverWrite2.Value = True Then
    GraphOverWrite = 1
End If
End Sub
```

```

VERSION 5.00
Begin VB.Form frmSplash
    BorderStyle = 3 '固定ﾀｲﾌﾟ'
    ClientHeight = 2775
    ClientLeft = 45
    ClientTop = 45
    ClientWidth = 7275
    ControlBox = 0 'False'
    LinkTopic = "Form1"
    MaxButton = 0 'False'
    MinButton = 0 'False'
    ScaleHeight = 2775
    ScaleWidth = 7275
    ShowInTaskbar = 0 'False'
    StartUpPosition = 2 '画面の中央'
    Visible = 0 'False'
Begin VB.Frame fraMainFrame
    BeginProperty Font
        Name = "MS Pゴシック"
        Size = 9.75
        Charset = 128
        Weight = 400
        Underline = 0 'False'
        Italic = 0 'False'
        Strikethrough = 0 'False'
    EndProperty
    Height = 2730
    Left = 60
    TabIndex = 0
    Top = 0
    Width = 7200
Begin VB.PictureBox picIcon
    BackColor = &H00C0C0C0&
    ClipControls = 0 'False'
    Height = 1980
    Left = 120
    Picture = "frmSplash. frx":0000
    ScaleHeight = 1920
    ScaleMode = 0 'ユーザー'
    ScaleWidth = 1980
    TabIndex = 9
    TabStop = 0 'False'
    Top = 240
    Width = 2040
End
Begin VB.Label lblVersion
    Alignment = 1 '右揃え'
    AutoSize = -1 'True'
    Caption = "ﾊﾞｰｼﾞｮﾝ 1.00"
    BeginProperty Font
        Name = "MS Pゴシック"
        Size = 12
        Charset = 128
        Weight = 700
        Underline = 0 'False'
        Italic = 0 'False'
        Strikethrough = 0 'False'
    EndProperty
    Height = 240
    Left = 5460
    TabIndex = 8
    Tag = "ﾊﾞｰｼﾞｮﾝ"

```

```

VERSION 5.00
Begin VB.Form frmSplash
    BorderStyle = 3 '固定ﾀｲﾌﾟ'
    ClientHeight = 2775
    ClientLeft = 45
    ClientTop = 45
    ClientWidth = 7275
    ControlBox = 0 'False'
    LinkTopic = "Form1"
    MaxButton = 0 'False'
    MinButton = 0 'False'
    ScaleHeight = 2775
    ScaleWidth = 7275
    ShowInTaskbar = 0 'False'
    StartUpPosition = 2 '画面の中央'
    Visible = 0 'False'
Begin VB.Frame fraMainFrame
    BeginProperty Font
        Name = "MS Pゴシック"
        Size = 9.75
        Charset = 128
        Weight = 400
        Underline = 0 'False'
        Italic = 0 'False'
        Strikethrough = 0 'False'
    EndProperty
    Height = 2730
    Left = 60
    TabIndex = 0
    Top = 0
    Width = 7200
Begin VB.PictureBox picIcon
    BackColor = &H00C0C0C0&
    ClipControls = 0 'False'
    Height = 1980
    Left = 120
    Picture = "frmSplash. frx":0000
    ScaleHeight = 1920
    ScaleMode = 0 'ユーザー'
    ScaleWidth = 1980
    TabIndex = 9
    TabStop = 0 'False'
    Top = 240
    Width = 2040
End
Begin VB.Label lblVersion
    Alignment = 1 '右揃え'
    AutoSize = -1 'True'
    Caption = "ﾊﾞｰｼﾞｮﾝ 1.00"
    BeginProperty Font
        Name = "MS Pゴシック"
        Size = 12
        Charset = 128
        Weight = 700
        Underline = 0 'False'
        Italic = 0 'False'
        Strikethrough = 0 'False'
    EndProperty
    Height = 240
    Left = 5460
    TabIndex = 8
    Tag = "ﾊﾞｰｼﾞｮﾝ"

```

Top = 1500  
Width = 1470

End  
Begin VB.Label lblPlatform  
Alignment = 1 '右揃え  
AutoSize = -1 'True  
Caption = "for Windows NT/2000"

BeginProperty Font  
Name = "MS Pゴシック"  
Size = 15.75  
Charset = 128  
Weight = 700  
Underline = 0 'False  
Italic = 0 'False  
Strikethrough = 0 'False

EndProperty  
Height = 315  
Left = 3780  
TabIndex = 7  
Tag = "プラットフォーム"  
Top = 1200  
Width = 3165

End  
Begin VB.Label lblLicenseTo  
Alignment = 1 '右揃え  
Caption = "License to Nextforest"

BeginProperty Font  
Name = "MS 明朝"  
Size = 9.75  
Charset = 128  
Weight = 400  
Underline = 0 'False  
Italic = -1 'True  
Strikethrough = 0 'False

EndProperty  
Height = 255  
Left = 4320  
TabIndex = 1  
Tag = "ライセンス先"  
Top = 120  
Width = 2715

End  
Begin VB.Label lblProductName  
Caption = "製品名"

BeginProperty Font  
Name = "MS P明朝"  
Size = 21.75  
Charset = 128  
Weight = 700  
Underline = 0 'False  
Italic = 0 'False  
Strikethrough = 0 'False

EndProperty  
Height = 495  
Left = 2340  
TabIndex = 6  
Tag = "製品名"  
Top = 600  
Width = 4815

End  
Begin VB.Label lblCompanyProduct  
AutoSize = -1 'True

Top = 1500  
Width = 1470

End  
Begin VB.Label lblPlatform  
Alignment = 1 '右揃え  
AutoSize = -1 'True  
Caption = "for Windows NT/2000/XP"

BeginProperty Font  
Name = "MS Pゴシック"  
Size = 15.75  
Charset = 128  
Weight = 700  
Underline = 0 'False  
Italic = 0 'False  
Strikethrough = 0 'False

EndProperty  
Height = 315  
Left = 3180  
TabIndex = 7  
Tag = "プラットフォーム"  
Top = 1200  
Width = 3765

End  
Begin VB.Label lblLicenseTo  
Alignment = 1 '右揃え  
Caption = "License to (株)計測リサーチコンサルタント"

BeginProperty Font  
Name = "MS 明朝"  
Size = 9.75  
Charset = 128  
Weight = 400  
Underline = 0 'False  
Italic = -1 'True  
Strikethrough = 0 'False

EndProperty  
Height = 255  
Left = 3960  
TabIndex = 1  
Tag = "ライセンス先"  
Top = 120  
Width = 3075

End  
Begin VB.Label lblProductName  
Caption = "製品名"

BeginProperty Font  
Name = "MS P明朝"  
Size = 14.25  
Charset = 128  
Weight = 700  
Underline = 0 'False  
Italic = 0 'False  
Strikethrough = 0 'False

EndProperty  
Height = 495  
Left = 2340  
TabIndex = 6  
Tag = "製品名"  
Top = 720  
Width = 4815

End  
Begin VB.Label lblCompanyProduct  
AutoSize = -1 'True

```

Caption = "Nextforest"
BeginProperty Font
  Name = "MS P明朝"
  Size = 11.25
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = -1 'True'
  Strikethrough = 0 'False'
EndProperty
Height = 225
Left = 2280
TabIndex = 5
Tag = "製作会社名"
Top = 300
Width = 1335

```

```

Caption = "(株)計測リサーチコンサルタント"
BeginProperty Font
  Name = "MS P明朝"
  Size = 11.25
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = -1 'True'
  Strikethrough = 0 'False'
EndProperty
Height = 225
Left = 2280
TabIndex = 5
Tag = "製作会社名"
Top = 360
Width = 2160

```

```

End
Begin VB.Label lblWarning
Caption = "$"frmSplash.frx":429C
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 8.25
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = 0 'False'
  Strikethrough = 0 'False'
EndProperty
Height = 345
Left = 240
TabIndex = 2
Tag = "警告"
Top = 2340
Width = 6825

```

```

End
Begin VB.Label lblWarning
Caption = "$"frmSplash.frx":429C
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 8.25
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = 0 'False'
  Strikethrough = 0 'False'
EndProperty
Height = 345
Left = 240
TabIndex = 2
Tag = "警告"
Top = 2340
Width = 6825

```

```

End
Begin VB.Label lblCompany
AutoSize = -1 'True'
Caption = "1992-2002 Nextforest"
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 9.75
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = 0 'False'
  Strikethrough = 0 'False'
EndProperty
Height = 195
Left = 5040
TabIndex = 4
Tag = "会社名"
Top = 1860
Width = 2085

```

```

End
Begin VB.Label lblCompany
AutoSize = -1 'True'
Caption = "2004 KeisokuResearchConsultant"
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 9.75
  Charset = 128
  Weight = 400
  Underline = 0 'False'
  Italic = 0 'False'
  Strikethrough = 0 'False'
EndProperty
Height = 195
Left = 3720
TabIndex = 4
Tag = "会社名"
Top = 1920
Width = 2865

```

```

End
Begin VB.Label lblCopyright
AutoSize = -1 'True'
Caption = "Copyright (C)"
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 9.75
  Charset = 128
  Weight = 400

```

```

End
Begin VB.Label lblCopyright
AutoSize = -1 'True'
Caption = "Copyright (C)"
BeginProperty Font
  Name = "MS Pゴシック"
  Size = 9.75
  Charset = 128
  Weight = 400

```

```

Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```
EndProperty
```

```
Height = 195
```

```
Left = 3660
```

```
TabIndex = 3
```

```
Tag = "著作権"
```

```
Top = 1860
```

```
Width = 1305
```

```
End
```

```
End
```

```
End
```

```
Attribute VB_Name = "frmSplash"
```

```
Attribute VB_GlobalNameSpace = False
```

```
Attribute VB_Creatable = False
```

```
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
```

```
Option Explicit
```

```
Private Sub Form_Load()
```

```
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
```

```
    lblProductName.Caption = App.Title
```

```
End Sub
```

```

Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False

```

```
EndProperty
```

```
Height = 195
```

```
Left = 3720
```

```
TabIndex = 3
```

```
Tag = "著作権"
```

```
Top = 1680
```

```
Width = 1305
```

```
End
```

```
End
```

```
End
```

```
Attribute VB_Name = "frmSplash"
```

```
Attribute VB_GlobalNameSpace = False
```

```
Attribute VB_Creatable = False
```

```
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
```

```
Option Explicit
```

```
Private Sub Form_Load()
```

```
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
```

```
    lblProductName.Caption = App.Title
```

```
End Sub
```

functionSub.bas x 2

```

Attribute VB_Name = "functionSub"
Option Explicit

Public Function StrSel1013(vCommandStr As String)
Dim buf As String, pos As Integer
buf = vCommandStr
Do While InStr(buf, Chr(10)) <> 0
pos = InStr(buf, Chr(10))
buf = Left(buf, pos - 1) + Mid(buf, pos + 1)
Loop
Do While InStr(buf, Chr(13)) <> 0
pos = InStr(buf, Chr(13))
buf = Left(buf, pos - 1) + Mid(buf, pos + 1)
Loop
StrSel1013 = buf
End Function

```

```

Attribute VB_Name = "functionSub"
Option Explicit

Public Function StrSel1013(vCommandStr As String)
Dim buf As String, pos As Integer
buf = vCommandStr
Do While InStr(buf, Chr(10)) <> 0
pos = InStr(buf, Chr(10))
buf = Left(buf, pos - 1) + Mid(buf, pos + 1)
Loop
Do While InStr(buf, Chr(13)) <> 0
pos = InStr(buf, Chr(13))
buf = Left(buf, pos - 1) + Mid(buf, pos + 1)
Loop
StrSel1013 = buf
End Function

```

```

Public Function FileNameToTimeAndID(vStr As String)
Dim buf As String, buf2 As String
buf = vStr + " "
buf2 = buf2 + Mid(buf, 1, 4) + "年"
buf2 = buf2 + Mid(buf, 5, 2) + "月"
buf2 = buf2 + Mid(buf, 7, 2) + "日"
buf2 = buf2 + Mid(buf, 9, 2) + "時"
buf2 = buf2 + Mid(buf, 11, 2) + "分"
buf2 = buf2 + Mid(buf, 13, 2) + "秒"
If Mid(buf, 15, 1) = "A" Then
buf2 = buf2 + "端末①"
ElseIf Mid(buf, 15, 1) = "B" Then
buf2 = buf2 + "端末②"
ElseIf Mid(buf, 15, 1) = "C" Then
buf2 = buf2 + "端末③"
End If

```

```

Public Function FileNameToTimeAndID(vStr As String)
Dim buf As String, buf2 As String
buf = vStr + " "
If Left(buf, 2) = "0" And Left(buf, 3) = "0" Then
buf2 = buf2 + Mid(buf, 1, 4) + "年"
buf2 = buf2 + Mid(buf, 5, 2) + "月"
buf2 = buf2 + Mid(buf, 7, 2) + "日"
buf2 = buf2 + Mid(buf, 9, 2) + "時"
buf2 = buf2 + Mid(buf, 11, 2) + "分"
buf2 = buf2 + Mid(buf, 13, 2) + "秒"
Else
buf2 = buf2 + Mid(buf, 3, 4) + "年"
buf2 = buf2 + Mid(buf, 7, 2) + "月"
buf2 = buf2 + Mid(buf, 9, 2) + "日"
buf2 = buf2 + Mid(buf, 11, 2) + "時"
buf2 = buf2 + Mid(buf, 13, 2) + "分"
buf2 = buf2 + Mid(buf, 15, 2) + "秒"
End If

```

```

FileNameToTimeAndID = buf2
End Function

```

```

If Mid(buf, 15, 1) = "A" Then
buf2 = buf2 + "端末①"
ElseIf Mid(buf, 15, 1) = "B" Then
buf2 = buf2 + "端末②"
ElseIf Mid(buf, 15, 1) = "C" Then
buf2 = buf2 + "端末③"
End If
FileNameToTimeAndID = buf2
End Function

```

```

Public Function 最大値保存()
Dim FileNo As Integer
Dim MaxDataFilename As String
Dim MaxDataFilename0 As String

Dim NowDateTime As String ' yyyy/mm/dd/hh/nn
FileNo = FreeFile

On Error GoTo Error
MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
If PointNo > 0 Then
MaxDataFilename = MstData.DataDir & "¥" & PointNo & "_" & Format(Date, "yyyymm") & ".csv"
Else

```

```

MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
End If

NowDateTime = Format(Date, "yyyy/mm/dd ") & Format(Time, "hh:mm")
If NowDateTime <> BeforeDatetime Then
    Open MaxDataFilename For Append As FileNo
    frmMain.txtOpenFileNum.Caption = Str(FileNo)

    Print #FileNo, NowDateTime; ", "; IntvPitch; ", ";
    L10DataGal(0); ", "; L10DataGal(1); ", "; L10DataGal(2); ", "; L10DataGal(3); ", ";
    MaxDataGal(0); ", "; MaxDataGal(1); ", "; MaxDataGal(2); ", "; MaxDataGal(3)
Close #FileNo
MaxDataX(MaxDataCount) = DateValue(Date) + TimeValue(Time) ' グラフデータ取得
MaxDataY(MaxDataCount, 0) = MaxDataGal(0)
MaxDataY(MaxDataCount, 1) = MaxDataGal(1)
MaxDataY(MaxDataCount, 2) = MaxDataGal(2)
MaxDataY(MaxDataCount, 3) = MaxDataGal(3)
L10DataY(MaxDataCount, 0) = L10DataGal(0)
L10DataY(MaxDataCount, 1) = L10DataGal(1)
L10DataY(MaxDataCount, 2) = L10DataGal(2)
L10DataY(MaxDataCount, 3) = L10DataGal(3)

' 事務所パソコンに最新データ受け渡し saishindata.csv
FileNo = FreeFile
MaxDataFilename0 = MstData.DataDir & "¥" & "saishindata" & ".csv"
Open MaxDataFilename0 For Output As FileNo
frmMain.txtOpenFileNum.Caption = Str(FileNo)
Print #FileNo, NowDateTime; ", ";
Print #FileNo, IntvPitch; ", ";
Print #FileNo, L10DataGal(0); ", ";
Print #FileNo, L10DataGal(1); ", ";
Print #FileNo, L10DataGal(2); ", ";
Print #FileNo, L10DataGal(3); ", ";
Print #FileNo, MaxDataGal(0); ", ";
Print #FileNo, MaxDataGal(1); ", ";
Print #FileNo, MaxDataGal(2); ", ";
Print #FileNo, MaxDataGal(3)
Close #FileNo

FileCopy MaxDataFilename0, "¥¥192.168.125.3¥data¥saishindata_2.csv"
FileNo = FreeFile
MaxDataFilename0 = strMaxDataCopyDir & "¥" & "data" & ".flg"
Open MaxDataFilename0 For Output As FreeFile
Print #FileNo, NowDateTime
Close #FileNo

If Int(DateValue(BeforeDatetime)) <> Int(DateValue(NowDateTime)) Then
    MaxDataCount0 = MaxDataCount
End If
BeforeDatetime = NowDateTime
MaxDataCount = MaxDataCount + 1
End If

Exit Function
Error:
Close #FileNo
End Function
Public Function 最大値保存Copy()
Dim i As Integer
Dim FileNo As Integer
Dim MaxDataFilename As String
Dim MaxDataFilename1 As String

```

```

Dim NowDateTime As String ' yyyy/mm/dd/hh/nn
' FileNo = FreeFile

On Error GoTo Error
' MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
' MaxDataFilename1 = strMaxDataCopyDir & "¥" & Format(Date, "yyyymm") & ".csv"
If PointNo > 0 Then
    MaxDataFilename = MstData.DataDir & "¥" & PointNo & "_" & Format(Date, "yyyymm") & ".csv"
    MaxDataFilename1 = strMaxDataCopyDir & "¥" & PointNo & "_" & Format(Date, "yyyymm") & ".c
Else
    MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
    MaxDataFilename1 = strMaxDataCopyDir & "¥" & Format(Date, "yyyymm") & ".csv"
End If

NowDateTime = Format(Date, "yyyy/mm/dd ") & Format(Time, "hh:mm")
' If NowDateTime <> BeforeDatetime Then

FileCopy MaxDataFilename, MaxDataFilename1
' Open MaxDataFilename For Append As FileNo
' Print #FileNo, NowDateTime; ",,"; IntvPitch; ",,"; MaxDataGal(0); ",,"; MaxDataGal(1
' Close #FileNo
MaxDataX(MaxDataCount) = DateValue(Date) + TimeValue(Time) ' グラフデータ取得
MaxDataY(MaxDataCount, 0) = MaxDataGal(0)
MaxDataY(MaxDataCount, 1) = MaxDataGal(1)
MaxDataY(MaxDataCount, 2) = MaxDataGal(2)
MaxDataY(MaxDataCount, 3) = MaxDataGal(3)

' If Int(DateValue(BeforeDatetime)) <> Int(DateValue(NowDateTime)) Then
'     MaxDataCount0 = MaxDataCount
' End If
' BeforeDatetime = NowDateTime
MaxDataCount = MaxDataCount + 1
' End If

Exit Function
Error:
i = i
End Function

```

LogRoutine.bas x 2

```
Attribute VB_Name = "LogRoutine"
Option Explicit

Public Sub LogFileClear()
    Dim fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Output As #fno
    Close #fno
    fMainForm.IsvActionLog.ListItems.Clear
End Sub

Public Sub AppendLog(vActionStr As String)
    Dim fno As Integer
    Dim TimeStr As String, ActionStr As String
    fno = FreeFile()
    Open LogFilePathName For Append As #fno
    TimeStr = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
    ActionStr = vActionStr
    Write #fno, TimeStr, ActionStr
    Close #fno

    With fMainForm
        Dim itmX As ListItem
        Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
        itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        .IsvActionLog.Refresh
    End With
End Sub

Public Sub ReadLog()
    Dim TimeStr As String, ActionStr As String
    Dim itmX As ListItem, fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Append As #fno: Close #fno
    Open LogFilePathName For Input As #fno

    With fMainForm
        .IsvActionLog.ListItems.Clear
        Do While Not EOF(fno)
            Input #fno, TimeStr, ActionStr
            Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
            itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        Loop

        End With
    Close #fno
End Sub
```

```
Attribute VB_Name = "LogRoutine"
Option Explicit

Public Sub LogFileClear()
    Dim fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Output As #fno
    Close #fno
    fMainForm.IsvActionLog.ListItems.Clear
End Sub

Public Sub AppendLog(vActionStr As String)
    Dim fno As Integer
    Dim TimeStr As String, ActionStr As String
    fno = FreeFile()
    Open LogFilePathName For Append As #fno
    TimeStr = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
    ActionStr = vActionStr
    Write #fno, TimeStr, ActionStr
    Close #fno

    With fMainForm
        Dim itmX As ListItem
        Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
        itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        .IsvActionLog.Refresh
    End With
End Sub

Public Sub ReadLog()
    Dim TimeStr As String, ActionStr As String
    Dim itmX As ListItem, fno As Integer
    fno = FreeFile()
    Open LogFilePathName For Append As #fno: Close #fno
    Open LogFilePathName For Input As #fno

    With fMainForm
        .IsvActionLog.ListItems.Clear
        Do While Not EOF(fno)
            Input #fno, TimeStr, ActionStr
            Set itmX = .IsvActionLog.ListItems.Add(, , TimeStr)
            itmX.SubItems(GetListHardIndex(.IsvActionLog, "ACTION")) = ActionStr
        Loop

        End With
    Close #fno
End Sub
```

```
Attribute VB_Name = "MainModule"
Option Explicit
```

```
Public MainForm As frmMain
Public MstData As CMeasData
Public MoveFileN As Integer, MoveFile(100) As String
Public AppDir As String, LogFilePathName As String
Public gMaxChNo As Integer
```

```
Attribute VB_Name = "MainModule"
Option Explicit
```

```
Public MainForm As frmMain
Public MstData As CMeasData
Public MoveFileN As Integer, MoveFile(100) As String
Public AppDir As String, LogFilePathName As String
Public gMaxChNo As Integer
```

```
Public IntvPitch As Integer
Public IntCopyPitch As Integer ' 最大値コピーピッチ分
Public IntCopyMin As Integer ' 最大値コピー分
Public IntLvX As Integer ' 時間率振動レベル 0; Lv10 1; Lvmax
```

```
Public MaxDataGal(3) As Single
Public L10DataGal(3) As Single
Public LvDataGal(3) As Single
Public L10DataBox(10000, 3) As Long
Public DataRing(5999, 3) As Integer ' Data Ring 10分間
Public RingCount As Integer ' Data Ring 10分間
```

```
Public MaxDataY(45000, 3) As Single
Public L10DataY(45000, 3) As Single
Public MaxDataY0(45000, 3) As Single
Public L10DataY0(45000, 3) As Single ' 任意月のデータ
```

```
Public MaxDataX(45000) As Single
Public MaxDataX0(45000) As Single ' 任意月のデータ
```

```
Public MaxDataCount0 As Long ' nowdateの最初のカウンタ
Public MaxDataCount As Long ' 20041129変更 integer→long
Public KeihoCheck As Integer ' 最大値警報チェック0; 警報する 1; 警報しない
Public KeihoOffDisp As Integer ' 警報オーバーを非表示 0; 表示する 1; 表示しない
Public Keiho3Intv As Integer ' 3連続警報オーバーで警報 0; 1回 1; 3連続
Public Keiho3Count(3) As Long ' 3連続警報オーバーで警報 カウント値
Public Keiho3IntvFlg As Boolean ' 3連続警報オーバーで警報 フラグ
```

```
Public KeihoFlg1(3) As Boolean ' 警報フラグ(黄) 最大値保存で使用 重複防止に使用
Public KeihoFlg(3) As Boolean ' 警報フラグ(赤) 最大値保存で使用 重複防止に使用
```

```
Public KeihoStopFlg As Boolean ' 警報stopフラグ true; stop falseでそのまま
Public KeihoEmail As Integer ' 警報Emailチェック 0; メールしない 1; メールする
Public KeihoEmailFlg As Boolean ' 警報Emailフラグ true; on falseでそのまま
Public KeihoEmailFlg1 As Boolean ' 警報Emailフラグ1 true; 1回掛けたので再度掛けない falseで掛
```

```
Public BeforeDatetime As String ' 最大値保存で使用
Public intCopyFlg As Integer ' 最大値保存で使用
Public strMaxDataCopyDir As String ' 最大値保存で使用
```

```
Public GraphYScale(3) As Integer ' Graph表示で使用
Public GraphYScaleCount(3) As Integer ' Graph表示で使用
Public dBTeiji As Boolean ' トリガで使用
Public GraphOverWrite As Integer ' Graph表示で使用 0: 上書き 1:0; 時クリア
```

```
Public ChName(3) As String ' Ch Name
Public PointNo As Integer ' 測点No
```

```
' Public SampSec As Integer ' サンプル周波数
' Public RecDataNum As Long ' 生データ収録回数
' Public RecPreDataNum As Long ' 生データプリトリガ収録回数
```

```

Sub Main()
    Dim Start As Single: Start = Timer ' 中断の開始時刻を設定します。
    frmSplash.Show
    frmSplash.Refresh
    Set MainForm = New frmMain
    Set MstData = New CMeasData
    Initialize ' 初期化
    Do While Timer < Start + 2 Or Timer < Start - 100
        DoEvents ' 他のプロセスに制御を渡します。
    Loop

    Load MainForm
    Unload frmSplash
    ReadLog
    DataFileListUpdate
    AppendLog "プログラム起動"
    Grig_AppendStatus "プログラム起動"
    MainForm.Show
End Sub

```

```

Public Sub Initialize()
    Dim DataDirectory As String

    'パス (ファイル) 設定
    ChDrive App.Path: ChDir App.Path
    AppDir = App.Path
    DataDirectory = AppDir & "%data"

```

```

    If Dir(DataDirectory, vbDirectory) = "" Then
        Mkdir DataDirectory
    End If
    ChDrive DataDirectory
    ChDir DataDirectory

```

```

gMaxChNo = 5

```

```

LogFilePathName = App.Path & "%BombVibrate.log"
MoveFileN = 0
MstData.TrgLevel1 = GetSetting(App.Title, "Settings", "TrgLevel1", 10)
MstData.TrgLevel2 = GetSetting(App.Title, "Settings", "TrgLevel2", 10)

```

```

Sub Main()
    Dim Start As Single: Start = Timer ' 中断の開始時刻を設定します。
    frmSplash.Show
    frmSplash.Refresh
    Set MainForm = New frmMain
    Set MstData = New CMeasData
    Initialize ' 初期化
    Do While Timer < Start + 2 Or Timer < Start - 100
        DoEvents ' 他のプロセスに制御を渡します。
    Loop

    MaxDataLoad
    Load MainForm
    Unload frmSplash
    ReadLog
    DataFileListUpdate
    AppendLog "プログラム起動"
    Grig_AppendStatus "プログラム起動"
    MainForm.Show
End Sub

```

```

Public Sub Initialize()
    Dim DataDirectory As String

    'パス (ファイル) 設定
    ChDrive App.Path: ChDir App.Path
    AppDir = App.Path
    DataDirectory = AppDir & "%data"
    DataDirectory = "d:%data"

```

```

    If Dir(DataDirectory, vbDirectory) = "" Then
        Mkdir DataDirectory
    End If
    ChDrive DataDirectory
    ChDir DataDirectory

```

```

gMaxChNo = 5
gMaxChNo = 3
BeforeDatetime = Date & Format(Time, "hh:nn")

```

```

LogFilePathName = App.Path & "%BombVibrate.log"
MoveFileN = 0
MstData.TrgLevel1 = GetSetting(App.Title, "Settings", "TrgLevel1", 10)
MstData.TrgLevel2 = GetSetting(App.Title, "Settings", "TrgLevel2", 10)
MstData.TrgLevel3 = GetSetting(App.Title, "Settings", "TrgLevel3", 10)
MstData.TrgLevel4 = GetSetting(App.Title, "Settings", "TrgLevel4", 10)

MstData.GalVRange1 = GetSetting(App.Title, "Settings", "GalVRange1", 100)
MstData.GalVRange2 = GetSetting(App.Title, "Settings", "GalVRange2", 100)
MstData.GalVRange3 = GetSetting(App.Title, "Settings", "GalVRange3", 100)
MstData.GalVRange4 = GetSetting(App.Title, "Settings", "GalVRange4", 100)

MstData.Keiho1 = GetSetting(App.Title, "Settings", "Keiho1", 75)
MstData.Keiho2 = GetSetting(App.Title, "Settings", "Keiho2", 75)
MstData.Keiho3 = GetSetting(App.Title, "Settings", "Keiho3", 75)
MstData.Keiho4 = GetSetting(App.Title, "Settings", "Keiho4", 75)
MstData.Keiho11 = GetSetting(App.Title, "Settings", "Keiho11", 70)
MstData.Keiho12 = GetSetting(App.Title, "Settings", "Keiho12", 70)
MstData.Keiho13 = GetSetting(App.Title, "Settings", "Keiho13", 70)
MstData.Keiho14 = GetSetting(App.Title, "Settings", "Keiho14", 70)

KeihoCheck = GetSetting(App.Title, "Settings", "Keiho", 0)
KeihoOffDisp = GetSetting(App.Title, "Settings", "KeihoOffDisp", 0)

```

```

Keiho3Intv = GetSetting(App.Title, "Settings", "Keiho3Intv", 0)
KeihoEmail = GetSetting(App.Title, "Settings", "KeihoEMail", 0)

GraphYScale(0) = GetSetting(App.Title, "Settings", "GraphYScale1", 60)
GraphYScale(1) = GetSetting(App.Title, "Settings", "GraphYScale2", 60)
GraphYScale(2) = GetSetting(App.Title, "Settings", "GraphYScale3", 60)
GraphYScale(3) = GetSetting(App.Title, "Settings", "GraphYScale4", 60)

GraphYScaleCount(0) = GetSetting(App.Title, "Settings", "GraphYScaleCount1", 6)
GraphYScaleCount(1) = GetSetting(App.Title, "Settings", "GraphYScaleCount2", 6)
GraphYScaleCount(2) = GetSetting(App.Title, "Settings", "GraphYScaleCount3", 6)
GraphYScaleCount(3) = GetSetting(App.Title, "Settings", "GraphYScaleCount4", 6)

' IntvPitch = GetSetting(App.Title, "Settings", "IntvPitch", 10)
IntvPitch = 10 ' 10分で固定
' IntLvX = GetSetting(App.Title, "Settings", "IntLvX", 0)
IntLvX = 0 ' Lv10で固定
MstData.SampSec = GetSetting(App.Title, "Settings", "SampSec", 0.01)
MstData.RecPreDataSec = GetSetting(App.Title, "Settings", "RecPreDataSec", 10)
MstData.RecDataSec = GetSetting(App.Title, "Settings", "RecDataSec", 60)

```

```
MstData.DataDir = DataDirectory
```

```

MstData.DataDir = DataDirectory
intCopyFlg = GetSetting(App.Title, "Settings", "intCopyFlg", 0)
strMaxDataCopyDir = GetSetting(App.Title, "Settings", "strMaxDataCopyDir", "c:¥")
dBTeiji = GetSetting(App.Title, "Settings", "dBTeiji", 1)
GraphOverWrite = GetSetting(App.Title, "Settings", "GraphOverWrite", 1)

ChName(0) = GetSetting(App.Title, "Settings", "ChName1", "騒音 [dB]")
ChName(1) = GetSetting(App.Title, "Settings", "ChName2", "振動Z [dB]")
ChName(2) = GetSetting(App.Title, "Settings", "ChName3", "振動X [dB]")
ChName(3) = GetSetting(App.Title, "Settings", "ChName4", "振動Y [dB]")

PointNo = GetSetting(App.Title, "Settings", "PointNo", 0)

```

```
End Sub
```

```
End Sub
```

```

Public Sub MaxDataLoad() ' 1ヶ月データ読み込み
Dim FileNo As Integer
Dim MaxDataFilename As String
Dim tempString As String
FileNo = FreeFile
On Error GoTo Error1
' MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
If PointNo > 0 Then
MaxDataFilename = MstData.DataDir & "¥" & PointNo & "_" & Format(Date, "yyyymm") & ".csv"
Else
MaxDataFilename = MstData.DataDir & "¥" & Format(Date, "yyyymm") & ".csv"
End If

Open MaxDataFilename For Input As FileNo
MaxDataCount0 = 0
MaxDataCount = 0
Do While Not EOF(FileNo)
Input #FileNo, tempString
MaxDataX(MaxDataCount) = DateValue(tempString) + TimeValue(tempString)

Input #FileNo, tempString
Input #FileNo, L10DataY(MaxDataCount, 0)
Input #FileNo, L10DataY(MaxDataCount, 1)
Input #FileNo, L10DataY(MaxDataCount, 2)
Input #FileNo, L10DataY(MaxDataCount, 3)

```

```

        Input #FileNo, MaxDataY (MaxDataCount, 0)
        Input #FileNo, MaxDataY (MaxDataCount, 1)
        Input #FileNo, MaxDataY (MaxDataCount, 2)
        Input #FileNo, MaxDataY (MaxDataCount, 3)
        If Int (MaxDataX (MaxDataCount)) < Int (DateValue (Date)) Then
            MaxDataCount0 = MaxDataCount + 1
        End If
        MaxDataCount = MaxDataCount + 1
    Loop

    Close #FileNo

Exit Sub
Error1:
Close #FileNo
End Sub
Function MaxDataLoad0 (ByVal Date00 As String) As Long ' 任意の1ヶ月データ読み込み
    Dim FileNo As Integer
    Dim MaxDataFilename As String
    Dim tempString As String
    Dim tempDataCount00 As Long

    FileNo = FreeFile
    On Error GoTo Error1
    If PointNo > 0 Then
        MaxDataFilename = MstData.DataDir & "¥" & PointNo & "_" & Format (Date00, "yyyymm") & ".cs"
    Else
        MaxDataFilename = MstData.DataDir & "¥" & Format (Date00, "yyyymm") & ".csv"
    End If

    Open MaxDataFilename For Input As FileNo
        tempDataCount00 = 0
        Do While Not EOF (FileNo)
            Input #FileNo, tempString
            MaxDataX0 (tempDataCount00) = DateValue (tempString) + TimeValue (tempString)

            Input #FileNo, tempString
            Input #FileNo, L10DataY0 (tempDataCount00, 0)
            Input #FileNo, L10DataY0 (tempDataCount00, 1)
            Input #FileNo, L10DataY0 (tempDataCount00, 2)
            Input #FileNo, L10DataY0 (tempDataCount00, 3)

            Input #FileNo, MaxDataY0 (tempDataCount00, 0)
            Input #FileNo, MaxDataY0 (tempDataCount00, 1)
            Input #FileNo, MaxDataY0 (tempDataCount00, 2)
            Input #FileNo, MaxDataY0 (tempDataCount00, 3)
            If Int (MaxDataX (MaxDataCount)) < Int (DateValue (Date)) Then
                MaxDataCount0 = MaxDataCount + 1
            End If
            tempDataCount00 = tempDataCount00 + 1
        Loop

    Close #FileNo
    MaxDataLoad0 = tempDataCount00

Exit Function
Error1:
Close #FileNo
End Function

```

```
Attribute VB_Name = "PDQFunc"
'-----
' Program Name: PDQCOMM. BAS
' Description : PDQComm v3.2J 関数定義ファイル(VB 4.0/VB5.0用)
' Version      : v1.5
' Created by   BOC 1997-06-15
'-----
```

## Option Explicit

DefInt A-Z

```
' AutoProcess プロパティ
Global Const PDQ_AUTOPROCESS_NONE = 0
Global Const PDQ_AUTOPROCESS_SERIAL = 1
Global Const PDQ_AUTOPROCESS_KEY = 2
Global Const PDQ_AUTOPROCESS_BOTH = 3

' AutoScroll プロパティ
Global Const PDQ_AUTOSCROLL_NONE = 0
Global Const PDQ_AUTOSCROLL_VERTICAL = 1
Global Const PDQ_AUTOSCROLL_HORIZONTAL = 2
Global Const PDQ_AUTOSCROLL_BOTH = 3
Global Const PDQ_AUTOSCROLL_VERTKEY = 4

' BackSpace プロパティ
Global Const PDQ_BACKSPACE_DESTRUCTIVE = 0
Global Const PDQ_BACKSPACE_NON_DESTRUCTIVE = 1

' CaptureMode プロパティ
Global Const PDQ_CAPTURE_STANDARD = 0
Global Const PDQ_CAPTURE_BINARY = 1
Global Const PDQ_CAPTURE_VISIBLE = 2

' ColorFilter プロパティ
Global Const PDQ_COLOR_FULL = 0
Global Const PDQ_COLOR_GREY = 1
Global Const PDQ_COLOR_MONO = 2

' CommEvent プロパティ (OnComm イベント)
Global Const PDQ_EV_SEND = 1
Global Const PDQ_EV_RECEIVE = 2
Global Const PDQ_EV_CTS = 3
Global Const PDQ_EV_DSR = 4
Global Const PDQ_EV_CD = 5
Global Const PDQ_EV_RING = 6
Global Const PDQ_EV_EOF = 7
Global Const PDQ_EV_ZMODEM = 8
Global Const PDQ_EV_XFER = 100

' CommEvent プロパティ (OnComm エラー)
Global Const PDQ_ER_BREAK = 1001
Global Const PDQ_ER_CTSTO = 1002
Global Const PDQ_ER_DSRTO = 1003
Global Const PDQ_ER_FRAME = 1004
Global Const PDQ_ER_INT0 = 1005
Global Const PDQ_ER_OVERRUN = 1006
Global Const PDQ_ER_CDTO = 1007
Global Const PDQ_ER_RXOVER = 1008
Global Const PDQ_ER_RXPARITY = 1009
Global Const PDQ_ER_TXFULL = 1010

' CursorType プロパティ
```

```
Attribute VB_Name = "PDQFunc"
'-----
' Program Name: PDQCOMM. BAS
' Description : PDQComm v3.2J 関数定義ファイル(VB 4.0/VB5.0用)
' Version      : v1.5
' Created by   BOC 1997-06-15
'-----
```

## Option Explicit

DefInt A-Z

```
' AutoProcess プロパティ
Global Const PDQ_AUTOPROCESS_NONE = 0
Global Const PDQ_AUTOPROCESS_SERIAL = 1
Global Const PDQ_AUTOPROCESS_KEY = 2
Global Const PDQ_AUTOPROCESS_BOTH = 3

' AutoScroll プロパティ
Global Const PDQ_AUTOSCROLL_NONE = 0
Global Const PDQ_AUTOSCROLL_VERTICAL = 1
Global Const PDQ_AUTOSCROLL_HORIZONTAL = 2
Global Const PDQ_AUTOSCROLL_BOTH = 3
Global Const PDQ_AUTOSCROLL_VERTKEY = 4

' BackSpace プロパティ
Global Const PDQ_BACKSPACE_DESTRUCTIVE = 0
Global Const PDQ_BACKSPACE_NON_DESTRUCTIVE = 1

' CaptureMode プロパティ
Global Const PDQ_CAPTURE_STANDARD = 0
Global Const PDQ_CAPTURE_BINARY = 1
Global Const PDQ_CAPTURE_VISIBLE = 2

' ColorFilter プロパティ
Global Const PDQ_COLOR_FULL = 0
Global Const PDQ_COLOR_GREY = 1
Global Const PDQ_COLOR_MONO = 2

' CommEvent プロパティ (OnComm イベント)
Global Const PDQ_EV_SEND = 1
Global Const PDQ_EV_RECEIVE = 2
Global Const PDQ_EV_CTS = 3
Global Const PDQ_EV_DSR = 4
Global Const PDQ_EV_CD = 5
Global Const PDQ_EV_RING = 6
Global Const PDQ_EV_EOF = 7
Global Const PDQ_EV_ZMODEM = 8
Global Const PDQ_EV_XFER = 100

' CommEvent プロパティ (OnComm エラー)
Global Const PDQ_ER_BREAK = 1001
Global Const PDQ_ER_CTSTO = 1002
Global Const PDQ_ER_DSRTO = 1003
Global Const PDQ_ER_FRAME = 1004
Global Const PDQ_ER_INT0 = 1005
Global Const PDQ_ER_OVERRUN = 1006
Global Const PDQ_ER_CDTO = 1007
Global Const PDQ_ER_RXOVER = 1008
Global Const PDQ_ER_RXPARITY = 1009
Global Const PDQ_ER_TXFULL = 1010

' CursorType プロパティ
```

```

Global Const PDQ_CURSOR_VBAR = 0
Global Const PDQ_CURSOR_BLOCK = 0

'Emulation プロパティ
Global Const PDQ_EMULATION_NONE = 0
Global Const PDQ_EMULATION_TTY = 1
Global Const PDQ_EMULATION_ANSI = 2
Global Const PDQ_EMULATION_VT52 = 3
Global Const PDQ_EMULATION_VT100 = 4

'Handshaking プロパティ
Global Const PDQ_HANDSHAKING_NONE = 0
Global Const PDQ_HANDSHAKING_XON = 1
Global Const PDQ_HANDSHAKING_RTS = 2
Global Const PDQ_HANDSHAKING_BOTH = 3

'KeyTranslation プロパティ
Global Const PDQ_KEY_NONE = 0
Global Const PDQ_KEY_MANUAL = 1
Global Const PDQ_KEY_ANSI = 2
Global Const PDQ_KEY_VT52 = 3
Global Const PDQ_KEY_VT100 = 4

'XferProtocol プロパティ
Global Const PDQ_XMODEM_CHECKSUM = 0
Global Const PDQ_XMODEM_CRC = 1
Global Const PDQ_XMODEM_1K = 2
Global Const PDQ_YMODEM_BATCH = 3
Global Const PDQ_YMODEM_G_BATCH = 4
Global Const PDQ_ZMODEM = 5
Global Const PDQ_KERMIT = 6
Global Const PDQ_COMPUSERVE_BPLUS = 7
Global Const PDQ_NIFTYSERVE_BPLUS = 8

'XferStatus プロパティ
Global Const PDQ_XFER_TERM_ERROR = -1
Global Const PDQ_XFER_TERM_OK = 0
Global Const PDQ_XFER_WAITING = 1
Global Const PDQ_XFER_FILE_READY = 2
Global Const PDQ_XFER_FILE_START = 3
Global Const PDQ_XFER_XFERING = 4
Global Const PDQ_XFER_SKIP = 5
Global Const PDQ_XFER_ABORT = 6
Global Const PDQ_XFER_FINISHED = 7
Global Const PDQ_XFER_LOSTCARRIER = 8
Global Const PDQ_XFER_TIMEOUT = 9

'XferStatusDialog プロパティ
Global Const PDQ_XFERDIALOG_NONE = 0
Global Const PDQ_XFERDIALOG_MODELESS = 1
Global Const PDQ_XFERDIALOG_MODAL = 2

'Notification プロパティ
Global Const PDQ_NOTIFICATION_MANUAL = 0
Global Const PDQ_NOTIFICATION_DRIVER = 1

```

---

サンプル関数に関する定義

---

```
Global CancelFlag% ' Trueに設定すると関連の関数の動作を中断します。
```

```

Global Const PDQ_CURSOR_VBAR = 0
Global Const PDQ_CURSOR_BLOCK = 0

'Emulation プロパティ
Global Const PDQ_EMULATION_NONE = 0
Global Const PDQ_EMULATION_TTY = 1
Global Const PDQ_EMULATION_ANSI = 2
Global Const PDQ_EMULATION_VT52 = 3
Global Const PDQ_EMULATION_VT100 = 4

'Handshaking プロパティ
Global Const PDQ_HANDSHAKING_NONE = 0
Global Const PDQ_HANDSHAKING_XON = 1
Global Const PDQ_HANDSHAKING_RTS = 2
Global Const PDQ_HANDSHAKING_BOTH = 3

'KeyTranslation プロパティ
Global Const PDQ_KEY_NONE = 0
Global Const PDQ_KEY_MANUAL = 1
Global Const PDQ_KEY_ANSI = 2
Global Const PDQ_KEY_VT52 = 3
Global Const PDQ_KEY_VT100 = 4

'XferProtocol プロパティ
Global Const PDQ_XMODEM_CHECKSUM = 0
Global Const PDQ_XMODEM_CRC = 1
Global Const PDQ_XMODEM_1K = 2
Global Const PDQ_YMODEM_BATCH = 3
Global Const PDQ_YMODEM_G_BATCH = 4
Global Const PDQ_ZMODEM = 5
Global Const PDQ_KERMIT = 6
Global Const PDQ_COMPUSERVE_BPLUS = 7
Global Const PDQ_NIFTYSERVE_BPLUS = 8

'XferStatus プロパティ
Global Const PDQ_XFER_TERM_ERROR = -1
Global Const PDQ_XFER_TERM_OK = 0
Global Const PDQ_XFER_WAITING = 1
Global Const PDQ_XFER_FILE_READY = 2
Global Const PDQ_XFER_FILE_START = 3
Global Const PDQ_XFER_XFERING = 4
Global Const PDQ_XFER_SKIP = 5
Global Const PDQ_XFER_ABORT = 6
Global Const PDQ_XFER_FINISHED = 7
Global Const PDQ_XFER_LOSTCARRIER = 8
Global Const PDQ_XFER_TIMEOUT = 9

'XferStatusDialog プロパティ
Global Const PDQ_XFERDIALOG_NONE = 0
Global Const PDQ_XFERDIALOG_MODELESS = 1
Global Const PDQ_XFERDIALOG_MODAL = 2

'Notification プロパティ
Global Const PDQ_NOTIFICATION_MANUAL = 0
Global Const PDQ_NOTIFICATION_DRIVER = 1

```

---

サンプル関数に関する定義

---

```
Global CancelFlag% ' Trueに設定すると関連の関数の動作を中断します。
```

' 下記の変数はモデム操作のルーチンを呼出す時に常に使われております。

```
Global CR$      ' CR (Asc 13)
Global LF$      ' LF (Asc 10)
Global CRLF$    ' Cr+Lf(13+10)
Global ESC$     ' Esc
Global BS$      ' バックスペース
Global SP$      ' スペース
```

' サンプル関数エラーコード

```
Global Const PDQ_SUCCESS = 2000
Global Const PDQ_ER_CANCELED = 2001 ' 実行中にキャンセルコードが設定された。CancelFlag
Global Const PDQ_ER_HANGUP_ERR = 2002 ' モデム回線を切断することはできませんでした。
Global Const PDQ_ER_TIMEOUT = 2003 ' タイムアウト
Global Const PDQ_ER_PORTCLOSED = 2004 ' 指定したポートはクローズされています。
Global Const PDQ_ER_CTRLERROR = 2005 ' 関数実行時のエラー(指定ポート使用不可等)
Global Const PDQ_ER_NOCMD = 2006 ' 指定されたコマンドはありません。
Global Const PDQ_ER_LOST_CARRIER = 2007 ' 回線は切断されました。
Global Const PDQ_ER_OFFLINE = 2008 ' 回線は既に切断されています。
Global Const PDQ_ER_NOANSWER = 2009 ' 応答がありません。
Global Const PDQ_ER_FILE_NOTFOUND = 2010 ' ファイルが見つかりません。
Global Const PDQ_ER_FILEERROR = 2011 ' ファイル読み出しエラー。
Global Const PDQ_ER_MODEM_ERROR = 2012 ' モデムでエラーが発生した。
Global Const PDQ_BADCOMMAND = 2013 ' スクリプトコマンドが不正
```

-----  
'  
'  
' スクリプト情報  
'  
'-----

' 内部: スクリプト内容の保存用として使用

```
Type ScriptVariableType
  variable As String
  Value As String
End Type
Global ScriptVars() As ScriptVariableType
```

' 内部: スクリプトラベルの保存時に使用

```
Type ScriptLabelType
  label As String
  LineNum As Integer
End Type
Global ScriptLabel() As ScriptLabelType
```

Function AsciiReceive%(CommCtl As Control, FileName\$, CDFlag%, TimeOut%, FileMode%)

' 本Functionは、ASCIITextファイルの受信をします。

```
Dim Handle%, AutoP%, RT%, SinP%, InpMode%
Dim LastTime!
Dim NewChar$, NewData$
Dim Start!
Dim Flag24 As Boolean
```

' COMポートをチェック

```
AsciiReceive% = PortCheck%(CommCtl)
If AsciiReceive% <> PDQ_SUCCESS Then
  Exit Function
End If
```

AsciiReceive% = PDQ\_SUCCESS

' 下記の変数はモデム操作のルーチンを呼出す時に常に使われております。

```
Global CR$      ' CR (Asc 13)
Global LF$      ' LF (Asc 10)
Global CRLF$    ' Cr+Lf(13+10)
Global ESC$     ' Esc
Global BS$      ' バックスペース
Global SP$      ' スペース
```

' サンプル関数エラーコード

```
Global Const PDQ_SUCCESS = 2000
Global Const PDQ_ER_CANCELED = 2001 ' 実行中にキャンセルコードが設定された。CancelFlag
Global Const PDQ_ER_HANGUP_ERR = 2002 ' モデム回線を切断することはできませんでした。
Global Const PDQ_ER_TIMEOUT = 2003 ' タイムアウト
Global Const PDQ_ER_PORTCLOSED = 2004 ' 指定したポートはクローズされています。
Global Const PDQ_ER_CTRLERROR = 2005 ' 関数実行時のエラー(指定ポート使用不可等)
Global Const PDQ_ER_NOCMD = 2006 ' 指定されたコマンドはありません。
Global Const PDQ_ER_LOST_CARRIER = 2007 ' 回線は切断されました。
Global Const PDQ_ER_OFFLINE = 2008 ' 回線は既に切断されています。
Global Const PDQ_ER_NOANSWER = 2009 ' 応答がありません。
Global Const PDQ_ER_FILE_NOTFOUND = 2010 ' ファイルが見つかりません。
Global Const PDQ_ER_FILEERROR = 2011 ' ファイル読み出しエラー。
Global Const PDQ_ER_MODEM_ERROR = 2012 ' モデムでエラーが発生した。
Global Const PDQ_BADCOMMAND = 2013 ' スクリプトコマンドが不正
```

-----  
'  
'  
' スクリプト情報  
'  
'-----

' 内部: スクリプト内容の保存用として使用

```
Type ScriptVariableType
  variable As String
  Value As String
End Type
Global ScriptVars() As ScriptVariableType
```

' 内部: スクリプトラベルの保存時に使用

```
Type ScriptLabelType
  label As String
  LineNum As Integer
End Type
Global ScriptLabel() As ScriptLabelType
```

Function AsciiReceive%(CommCtl As Control, FileName\$, CDFlag%, TimeOut%, FileMode%)

' 本Functionは、ASCIITextファイルの受信をします。

```
Dim Handle%, AutoP%, RT%, SinP%, InpMode%
Dim LastTime!
Dim NewChar$, NewData$
Dim Start!
Dim Flag24 As Boolean
```

' COMポートをチェック

```
AsciiReceive% = PortCheck%(CommCtl)
If AsciiReceive% <> PDQ_SUCCESS Then
  Exit Function
End If
```

AsciiReceive% = PDQ\_SUCCESS

```

' ファイルをFileMode%の設定に従いオープンします
On Error Resume Next
Handle% = FreeFile
If FileMode% = 0 Then
    Open FileName$ For Input As Handle%
    If Err Then
        Open FileName$ For Output As Handle%
    Else
        Close Handle%
        AsciiReceive% = PDQ_ER_FILEERROR
        Exit Function
    End If
Else
    ' 上書きモード
    Open FileName$ For Output As Handle%
End If

If Err <> 0 Then
    AsciiReceive% = PDQ_ER_FILEERROR
    Exit Function
End If

' AutoProcess & RThreshold & InputMode の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If
InpMode% = CommCtl.InputMode
If InpMode% Then
    CommCtl.InputMode = 0
End If

' 受信バッファをクリア
CommCtl.InBufferCount = 0

CommCtl.InputLen = 0

' タイムアウト設定
Start! = Timer

If Timer + TimeOut > 86400 Then
    LastTime! = TimeOut - (86400 - Timer)
    Flag24 = True
Else
    LastTime! = Timer + TimeOut%
    Flag24 = False
End If

Do
    ' キャリアのチェック
    If CDFlag% Then
        If CommCtl.CD Holding = False Then
            AsciiReceive% = PDQ_ER_LOST_CARRIER
            Exit Do
        End If
    End If
End Do

' キャンセル処理

```

```

' ファイルをFileMode%の設定に従いオープンします
On Error Resume Next
Handle% = FreeFile
If FileMode% = 0 Then
    Open FileName$ For Input As Handle%
    If Err Then
        Open FileName$ For Output As Handle%
    Else
        Close Handle%
        AsciiReceive% = PDQ_ER_FILEERROR
        Exit Function
    End If
Else
    ' 上書きモード
    Open FileName$ For Output As Handle%
End If

If Err <> 0 Then
    AsciiReceive% = PDQ_ER_FILEERROR
    Exit Function
End If

' AutoProcess & RThreshold & InputMode の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If
InpMode% = CommCtl.InputMode
If InpMode% Then
    CommCtl.InputMode = 0
End If

' 受信バッファをクリア
CommCtl.InBufferCount = 0

CommCtl.InputLen = 0

' タイムアウト設定
Start! = Timer

If Timer + TimeOut > 86400 Then
    LastTime! = TimeOut - (86400 - Timer)
    Flag24 = True
Else
    LastTime! = Timer + TimeOut%
    Flag24 = False
End If

Do
    ' キャリアのチェック
    If CDFlag% Then
        If CommCtl.CD Holding = False Then
            AsciiReceive% = PDQ_ER_LOST_CARRIER
            Exit Do
        End If
    End If
End Do

' キャンセル処理

```

```

If CancelFlag% Then
  AsciiReceive% = PDQ_ER_CANCELED
  Exit Do
End If

' ---
If CommCtl.InBufferCount > 0 Then
  Start! = Timer

  If Timer + TimeOut > 86400 Then
    LastTime! = TimeOut - (86400 - Timer)
    Flag24 = True
  Else
    LastTime! = Timer + TimeOut%
    Flag24 = False
  End If
  NewData$ = CommCtl.Input
  SinP% = InStr(NewData$, Chr$(&H1A))
  If SinP% Then
    NewData$ = Left$(NewData$, SinP% - 1)
    Print #Handle, NewData$
    Exit Do
  Else
    Print #Handle, NewData$;
  End If

  'ファイルエラー
  If Err Then
    AsciiReceive% = PDQ_ER_FILEERROR
    Exit Do
  End If
End If

'タイムアウトチェック
If Flag24 = False Then
  If Timer >= LastTime! Then
    AsciiReceive% = PDQ_ER_TIMEOUT
    Exit Do
  End If
Else
  If Timer < Start! Then
    Flag24 = False
  End If
End If

DoEvents

Loop

'ファイルクローズ
Close Handle%

If Err Then
  AsciiReceive% = PDQ_ER_FILEERROR
End If

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
  CommCtl.AutoProcess = AutoP%
End If
If RT% Then
  CommCtl.RThreshold = RT%

```

```

If CancelFlag% Then
  AsciiReceive% = PDQ_ER_CANCELED
  Exit Do
End If

' ---
If CommCtl.InBufferCount > 0 Then
  Start! = Timer

  If Timer + TimeOut > 86400 Then
    LastTime! = TimeOut - (86400 - Timer)
    Flag24 = True
  Else
    LastTime! = Timer + TimeOut%
    Flag24 = False
  End If
  NewData$ = CommCtl.Input
  SinP% = InStr(NewData$, Chr$(&H1A))
  If SinP% Then
    NewData$ = Left$(NewData$, SinP% - 1)
    Print #Handle, NewData$
    Exit Do
  Else
    Print #Handle, NewData$;
  End If

  'ファイルエラー
  If Err Then
    AsciiReceive% = PDQ_ER_FILEERROR
    Exit Do
  End If
End If

'タイムアウトチェック
If Flag24 = False Then
  If Timer >= LastTime! Then
    AsciiReceive% = PDQ_ER_TIMEOUT
    Exit Do
  End If
Else
  If Timer < Start! Then
    Flag24 = False
  End If
End If

DoEvents

Loop

'ファイルクローズ
Close Handle%

If Err Then
  AsciiReceive% = PDQ_ER_FILEERROR
End If

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
  CommCtl.AutoProcess = AutoP%
End If
If RT% Then
  CommCtl.RThreshold = RT%

```

```

End If
If InpMode% Then
    CommCtl.InputMode = InpMode%
End If

```

```
End Function
```

```
Function AsciiSend%(CommCtl As Control, FileName$, CDFlag%)
```

```

'本Functionは、ASCIITextファイルの転送(送信)をします
Dim Handle%, AutoP%, RT%, Test%, NextLine$

```

```

'COMポートをチェック
AsciiSend% = PortCheck%(CommCtl)
If AsciiSend% <> PDQ_SUCCESS Then
    Exit Function
End If

```

```
AsciiSend% = PDQ_SUCCESS
```

```

'ファイルをオープンします
On Error Resume Next
Handle% = FreeFile
Open FileName$ For Input As Handle%
If Err Then
    AsciiSend% = PDQ_ER_FILE_NOTFOUND
    Exit Function
End If

```

```

'AutoProcess & RThreshold の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

```

```

Do
'キャリアの検出
If CDFlag% Then
    If CommCtl.CD Holding = False Then
        AsciiSend% = PDQ_ER_LOST_CARRIER
        Exit Do
    End If
End If

```

```

'キャンセル処理
If CancelFlag% Then
    AsciiSend% = PDQ_ER_CANCELED
    Exit Do
End If

```

```

'次行の読出
Line Input #Handle%, NextLine$

```

```

'すべてのデータを送信した場合、EOFを送信する
If Len(NextLine$) = 0 Then
    If EOF(Handle%) Then
        CommCtl.Output = Chr$(&H1A)
        Exit Do
    End If
End If

```

```

End If
If InpMode% Then
    CommCtl.InputMode = InpMode%
End If

```

```
End Function
```

```
Function AsciiSend%(CommCtl As Control, FileName$, CDFlag%)
```

```

'本Functionは、ASCIITextファイルの転送(送信)をします
Dim Handle%, AutoP%, RT%, Test%, NextLine$

```

```

'COMポートをチェック
AsciiSend% = PortCheck%(CommCtl)
If AsciiSend% <> PDQ_SUCCESS Then
    Exit Function
End If

```

```
AsciiSend% = PDQ_SUCCESS
```

```

'ファイルをオープンします
On Error Resume Next
Handle% = FreeFile
Open FileName$ For Input As Handle%
If Err Then
    AsciiSend% = PDQ_ER_FILE_NOTFOUND
    Exit Function
End If

```

```

'AutoProcess & RThreshold の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

```

```

Do
'キャリアの検出
If CDFlag% Then
    If CommCtl.CD Holding = False Then
        AsciiSend% = PDQ_ER_LOST_CARRIER
        Exit Do
    End If
End If

```

```

'キャンセル処理
If CancelFlag% Then
    AsciiSend% = PDQ_ER_CANCELED
    Exit Do
End If

```

```

'次行の読出
Line Input #Handle%, NextLine$

```

```

'すべてのデータを送信した場合、EOFを送信する
If Len(NextLine$) = 0 Then
    If EOF(Handle%) Then
        CommCtl.Output = Chr$(&H1A)
        Exit Do
    End If
End If

```

```

    End If
End If

'ファイルエラー
If Err Then
    AsciiSend% = PDQ_ER_FILEERROR
    Exit Do
End If

'ポートヘッダを出力
CommCtl.Output = NextLine$ & Chr$(13) & Chr$(10) 'CRLF$

'送信バッファが空になるまで待つ
Do
    DoEvents
Loop Until CommCtl.OutBufferCount = 0

'ファイルの末尾の場合、EOFを送信する
If EOF(Handle%) Then
    CommCtl.Output = Chr$(&H1A)
    Exit Do
End If

```

Loop

```

'ファイルのクローズ
Close Handle%

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function Download%(CommCtl As Control, Protocol%, FileName\$, DialogMode%)

```

'本Functionは、指定したプロトコルでファイルをダウンロードします
Dim Test%

'COMポートをチェック
Download% = PortCheck%(CommCtl)
If Download% <> PDQ_SUCCESS Then
    Exit Function
End If

On Error Resume Next

'プロトコルを設定します
CommCtl.XferProtocol = Protocol%

'ダイアログボックスの表示を設定します
CommCtl.XferStatusDialog = DialogMode%

If Err Then
    '設定が無効
    Download% = PDQ_ER_CTRLERROR
    Exit Function
End If

```

```

    End If
End If

'ファイルエラー
If Err Then
    AsciiSend% = PDQ_ER_FILEERROR
    Exit Do
End If

'ポートヘッダを出力
CommCtl.Output = NextLine$ & Chr$(13) & Chr$(10) 'CRLF$

'送信バッファが空になるまで待つ
Do
    DoEvents
Loop Until CommCtl.OutBufferCount = 0

'ファイルの末尾の場合、EOFを送信する
If EOF(Handle%) Then
    CommCtl.Output = Chr$(&H1A)
    Exit Do
End If

```

Loop

```

'ファイルのクローズ
Close Handle%

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function Download%(CommCtl As Control, Protocol%, FileName\$, DialogMode%)

```

'本Functionは、指定したプロトコルでファイルをダウンロードします
Dim Test%

'COMポートをチェック
Download% = PortCheck%(CommCtl)
If Download% <> PDQ_SUCCESS Then
    Exit Function
End If

On Error Resume Next

'プロトコルを設定します
CommCtl.XferProtocol = Protocol%

'ダイアログボックスの表示を設定します
CommCtl.XferStatusDialog = DialogMode%

If Err Then
    '設定が無効
    Download% = PDQ_ER_CTRLERROR
    Exit Function
End If

```

```

'受信を開始する
CommCtl.Download = FileName$
If Err Then
  'ポートのアクセスは不能です
  Download% = PDQ_ER_CTRLERROR
Else
  'Download 開始
  Download% = PDQ_SUCCESS
End If

```

End Function

Function Hangup%(CommCtl As Control, NumTries%, Methods%, DelayTime%)

```

'本Functionは、回線からモデムを切断します
Dim Ret%, AutoP%, RT%, Test%
Dim i%, NoGood%, CDFlag%, ExcludeFlag%
Dim Duration!, HangupCmd$, Result$

```

```

'COMポートをチェック
Hangup% = PortCheck%(CommCtl)
If Hangup% <> PDQ_SUCCESS Then
  Exit Function
End If

```

```

'回線状況の確認
If CommCtl.CDHolding = False Then
  Hangup% = PDQ_ER_OFFLINE
  Exit Function
End If

```

```

'AutoProcess & RThresholdのの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
  CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
  CommCtl.RThreshold = 0
End If

```

```

'切断処理
Select Case Methods%
  Case 0 ' AT コマンド

```

```

  If NumTries% = 0 Then NumTries% = 1

```

```

  For i% = 1 To NumTries%
    'エスケープシーケンスを送る
    Pause 2
    CommCtl.Output = "+++"
    Pause 2

```

```

    ' "OK"文字列の受信待ち
    CDFlag% = True
    ExcludeFlag% = True
    Ret% = WaitFor%(CommCtl, "OK", 7, CDFlag%, ExcludeFlag%, Result$)

```

```

    Select Case Ret%
      Case PDQ_ER_CANCELED
        Exit For

```

```

'受信を開始する
CommCtl.Download = FileName$
If Err Then
  'ポートのアクセスは不能です
  Download% = PDQ_ER_CTRLERROR
Else
  'Download 開始
  Download% = PDQ_SUCCESS
End If

```

End Function

Function Hangup%(CommCtl As Control, NumTries%, Methods%, DelayTime%)

```

'本Functionは、回線からモデムを切断します
Dim Ret%, AutoP%, RT%, Test%
Dim i%, NoGood%, CDFlag%, ExcludeFlag%
Dim Duration!, HangupCmd$, Result$

```

```

'COMポートをチェック
Hangup% = PortCheck%(CommCtl)
If Hangup% <> PDQ_SUCCESS Then
  Exit Function
End If

```

```

'回線状況の確認
If CommCtl.CDHolding = False Then
  Hangup% = PDQ_ER_OFFLINE
  Exit Function
End If

```

```

'AutoProcess & RThresholdのの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
  CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
  CommCtl.RThreshold = 0
End If

```

```

'切断処理
Select Case Methods%
  Case 0 ' AT コマンド

```

```

  If NumTries% = 0 Then NumTries% = 1

```

```

  For i% = 1 To NumTries%
    'エスケープシーケンスを送る
    Pause 2
    CommCtl.Output = "+++"
    Pause 2

```

```

    ' "OK"文字列の受信待ち
    CDFlag% = True
    ExcludeFlag% = True
    Ret% = WaitFor%(CommCtl, "OK", 7, CDFlag%, ExcludeFlag%, Result$)

```

```

    Select Case Ret%
      Case PDQ_ER_CANCELED
        Exit For

```

```

    Case PDQ_ER_LOST_CARRIER
        Exit For
    End Select
    ' 回線切断命令を送出し結果値を戻す
    HangupCmd$ = "ATH0" & Chr$(13)

    Ret% = SendModemCmd%(CommCtl, HangupCmd$, 1, 5, Result$)

    ' 結果値のチェック
    Select Case Ret%
        Case PDQ_ER_CANCELED
            Exit For
        Case PDQ_SUCCESS
            Exit For
        Case PDQ_ER_LOST_CARRIER
            Exit For
        Case Else
            If i% = NumTries% Then
                If Ret% <> PDQ_ER_TIMEOUT Then
                    Ret% = PDQ_ER_HANGUP_ERR
                End If
            End If
        End Select
    Next i%

```

Case 1 ' DTR による切断

```

    CommCtl.DTREnable = False
    Duration! = Timer + DelayTime% / 1000
    Do
        DoEvents
        ' 処理が中断されたかを確認する
        If CancelFlag% Then
            Ret% = PDQ_ER_CANCELED
            Exit Do
        End If
    Loop Until Timer > Duration!
    CommCtl.DTREnable = True
    If CancelFlag% Then
        Ret% = PDQ_ER_CANCELED
    Else
        ' 回線の切断を待つ
        Pause 3
        If CommCtl.CDHolding Then
            Ret% = PDQ_ER_HANGUP_ERR
        Else
            Ret% = PDQ_SUCCESS
        End If
    End If
End If

```

End Select

HUExit:

```

' AutoProcess & RThreshold の設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

```

    Case PDQ_ER_LOST_CARRIER
        Exit For
    End Select
    ' 回線切断命令を送出し結果値を戻す
    HangupCmd$ = "ATH0" & Chr$(13)

    Ret% = SendModemCmd%(CommCtl, HangupCmd$, 1, 5, Result$)

    ' 結果値のチェック
    Select Case Ret%
        Case PDQ_ER_CANCELED
            Exit For
        Case PDQ_SUCCESS
            Exit For
        Case PDQ_ER_LOST_CARRIER
            Exit For
        Case Else
            If i% = NumTries% Then
                If Ret% <> PDQ_ER_TIMEOUT Then
                    Ret% = PDQ_ER_HANGUP_ERR
                End If
            End If
        End Select
    Next i%

```

Case 1 ' DTR による切断

```

    CommCtl.DTREnable = False
    Duration! = Timer + DelayTime% / 1000
    Do
        DoEvents
        ' 処理が中断されたかを確認する
        If CancelFlag% Then
            Ret% = PDQ_ER_CANCELED
            Exit Do
        End If
    Loop Until Timer > Duration!
    CommCtl.DTREnable = True
    If CancelFlag% Then
        Ret% = PDQ_ER_CANCELED
    Else
        ' 回線の切断を待つ
        Pause 3
        If CommCtl.CDHolding Then
            Ret% = PDQ_ER_HANGUP_ERR
        Else
            Ret% = PDQ_SUCCESS
        End If
    End If

```

End Select

HUExit:

```

' AutoProcess & RThreshold の設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

```
Hangup% = Ret%
```

```
End Function
```

```
Sub Pause(NumSecs!)
```

```
' 本Subは、指定した時間次の処理を待ちます
Dim Start!, EndTime!
```

```
Start! = Timer
EndTime! = Start! + NumSecs!
```

```
Do
  DoEvents
  If Start! > Timer And Start! <= EndTime! Then
    EndTime! = EndTime! - 86400
  End If
Loop Until Timer >= EndTime!
```

```
End Sub
```

```
Function PlaceCall%(CommCtl As Control, PhoneNum$, Retries%, TimeOut%, Result$, ModemBoud&, Re
```

```
' 本Functionは、電話番号をダイヤルし結果を戻します。
Dim Ret%, AutoP%, RT%, CDFlag%, ExcludeFlag%, Test%, CrPos%, ChkPort%
Dim CARRIER_Pos%, CONNECT_Pos%, LenBaud%, L%, i%, AscVal%, ChkCR%, Baud_Pos%
Dim DialStr$, BaudStr$, Chk_Result$, SettingBoud$
```

```
' COMポートをチェック
PlaceCall% = PortCheck%(CommCtl)
If PlaceCall% <> PDQ_SUCCESS Then
  Exit Function
End If
```

```
' AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
  CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
  CommCtl.RThreshold = 0
End If
```

```
' 引数の初期化
RealBaud& = 0
ModemBoud& = 0
Mode$ = ""
Result$ = ""
```

```
' ダイヤル命令の設定
DialStr$ = UCase$(Trim$(PhoneNum$)) & Chr$(13)
Select Case Mid$(DialStr$, 1, 1)
  Case "P", "T"
    DialStr$ = "ATD" & DialStr$
  Case Else
    DialStr$ = "ATDT" & DialStr$
End Select
```

```
If Retries% = 0 Then Retries% = 1
```

```
Hangup% = Ret%
```

```
End Function
```

```
Sub Pause(NumSecs!)
```

```
' 本Subは、指定した時間次の処理を待ちます
Dim Start!, EndTime!
```

```
Start! = Timer
EndTime! = Start! + NumSecs!
```

```
Do
  DoEvents
  If Start! > Timer And Start! <= EndTime! Then
    EndTime! = EndTime! - 86400
  End If
Loop Until Timer >= EndTime!
```

```
End Sub
```

```
Function PlaceCall%(CommCtl As Control, PhoneNum$, Retries%, TimeOut%, Result$, ModemBoud&, Re
```

```
' 本Functionは、電話番号をダイヤルし結果を戻します。
Dim Ret%, AutoP%, RT%, CDFlag%, ExcludeFlag%, Test%, CrPos%, ChkPort%
Dim CARRIER_Pos%, CONNECT_Pos%, LenBaud%, L%, i%, AscVal%, ChkCR%, Baud_Pos%
Dim DialStr$, BaudStr$, Chk_Result$, SettingBoud$
```

```
' COMポートをチェック
PlaceCall% = PortCheck%(CommCtl)
If PlaceCall% <> PDQ_SUCCESS Then
  Exit Function
End If
```

```
' AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
  CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
  CommCtl.RThreshold = 0
End If
```

```
' 引数の初期化
RealBaud& = 0
ModemBoud& = 0
Mode$ = ""
Result$ = ""
```

```
' ダイヤル命令の設定
DialStr$ = UCase$(Trim$(PhoneNum$)) & Chr$(13)
Select Case Mid$(DialStr$, 1, 1)
  Case "P", "T"
    DialStr$ = "ATD" & DialStr$
  Case Else
    DialStr$ = "ATDT" & DialStr$
End Select
```

```
If Retries% = 0 Then Retries% = 1
```

```

For i% = 1 To Retries%
  '受信バッファをクリアし、ATコマンドを送る
  CommCtl.InBufferCount = 0
  CommCtl.Output = DialStr$

  'モデムの接続待ち処理
  Do

    CDFlag% = False
    ExcludeFlag% = False

    'CRコードの受信待ち
    Ret% = WaitFor%(CommCtl, Chr$(13), Timeout, CDFlag, ExcludeFlag, Result$)

    Select Case Ret%
      Case PDQ_ER_CANCELED
        '中断文字を送信
        CommCtl.Output = ""
        Exit For
      Case PDQ_ER_TIMEOUT
        '中断文字を送信
        CommCtl.Output = ""
        '---結果コードが戻るまで待つ
        If i% <> Retries% Then
          Ret% = WaitFor%(CommCtl, "NO CARRIER", 5, CDFlag%, ExcludeFlag%, Result$)
        End If
        Exit Do
      Case PDQ_SUCCESS
        'CONNECTコマンドの取り出し
        ChkCR% = InStr(Result$, Chr$(13))
        If ChkCR% Then
          Chk_Result$ = Left$(Result$, ChkCR% - 1)
        Else
          Chk_Result$ = Result$
        End If

        'モデムからデータを受信した場合
        If InStr(Chk_Result$, "OK") Then
          '続く
        ElseIf InStr(Chk_Result$, "CONNECT") Then
          'CONNECT 受信
          CONNECT_Pos% = InStr(Chk_Result$, "CONNECT")
          If Len(Chk_Result$) < CONNECT_Pos% + 8 Then
            RealBaud& = 300
          Else
            BaudStr$ = Mid$(Chk_Result$, CONNECT_Pos% + 8)

            L% = Len(BaudStr$)
            For LenBaud% = 1 To L%
              AscVal% = Asc(Mid$(BaudStr$, LenBaud%, 1))
              If AscVal% < 48 Or AscVal% > 57 Then
                Exit For
              End If
            Next
            'モデムの通信速度を取得
            If LenBaud% > L% Then
              RealBaud& = Val(BaudStr$)
            Else
              RealBaud& = Val(Left$(BaudStr$, LenBaud%))
            End If
          End If
        End If
      End Select
    End Do
  End For

```

```

For i% = 1 To Retries%
  '受信バッファをクリアし、ATコマンドを送る
  CommCtl.InBufferCount = 0
  CommCtl.Output = DialStr$

  'モデムの接続待ち処理
  Do

    CDFlag% = False
    ExcludeFlag% = False

    'CRコードの受信待ち
    Ret% = WaitFor%(CommCtl, Chr$(13), Timeout, CDFlag, ExcludeFlag, Result$)

    Select Case Ret%
      Case PDQ_ER_CANCELED
        '中断文字を送信
        CommCtl.Output = ""
        Exit For
      Case PDQ_ER_TIMEOUT
        '中断文字を送信
        CommCtl.Output = ""
        '---結果コードが戻るまで待つ
        If i% <> Retries% Then
          Ret% = WaitFor%(CommCtl, "NO CARRIER", 5, CDFlag%, ExcludeFlag%, Result$)
        End If
        Exit Do
      Case PDQ_SUCCESS
        'CONNECTコマンドの取り出し
        ChkCR% = InStr(Result$, Chr$(13))
        If ChkCR% Then
          Chk_Result$ = Left$(Result$, ChkCR% - 1)
        Else
          Chk_Result$ = Result$
        End If

        'モデムからデータを受信した場合
        If InStr(Chk_Result$, "OK") Then
          '続く
        ElseIf InStr(Chk_Result$, "CONNECT") Then
          'CONNECT 受信
          CONNECT_Pos% = InStr(Chk_Result$, "CONNECT")
          If Len(Chk_Result$) < CONNECT_Pos% + 8 Then
            RealBaud& = 300
          Else
            BaudStr$ = Mid$(Chk_Result$, CONNECT_Pos% + 8)

            L% = Len(BaudStr$)
            For LenBaud% = 1 To L%
              AscVal% = Asc(Mid$(BaudStr$, LenBaud%, 1))
              If AscVal% < 48 Or AscVal% > 57 Then
                Exit For
              End If
            Next
            'モデムの通信速度を取得
            If LenBaud% > L% Then
              RealBaud& = Val(BaudStr$)
            Else
              RealBaud& = Val(Left$(BaudStr$, LenBaud%))
            End If
          End If
        End If
      End Select
    End Do
  End For

```

```

' Mode$の残りを取得
Mode$ = Mid$(BaudStr$, LenBaud% + 1)
End If
End If
' モデム～PC間のボーレートの設定
If CONNECT_Pos% Then
  SettingBoud$ = CommCtl.Settings
  Baud_Pos% = InStr(SettingBoud$, ",")
  ModemBoud& = Val(Left(SettingBoud$, Baud_Pos%))
End If

' 2400ボーが設定されている場合は少々時間を置き、モデムとのハンドシェイク
' を完了させる。
If RealBaud& <= 2400 Then
  Pause 8
End If
Exit For
ElseIf InStr(Chk_Result$, "RING") Then
  ' RINGを検出
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "NO CARRIER") Then
  ' キャリアを検出できない
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "ERROR") Then
  ' コマンドにエラーがある
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "NO DIALTONE") Then
  ' ダイアルトーンを検出できない
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "BUSY") Then
  ' 通話中... リダイヤル
  Ret% = PDQ_ER_MODEM_ERROR
  Exit Do
ElseIf InStr(Chk_Result$, "NO ANSWER") Then
  ' 未知結果コード
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "TIMEOUT") Then
  ' 未知結果コード
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
End If

```

```
End Select
```

```
DoEvents
```

```
Loop
```

```
DoEvents
```

```
Next
```

```
PCExit:
```

```
' AutoProcess&RThresholdの設定を戻す
```

```
If AutoP% Then
```

```
  CommCtl.AutoProcess = AutoP%
```

```
End If
```

```
If RT% Then
```

```
  CommCtl.RThreshold = RT%
```

```

' Mode$の残りを取得
Mode$ = Mid$(BaudStr$, LenBaud% + 1)
End If
End If
' モデム～PC間のボーレートの設定
If CONNECT_Pos% Then
  SettingBoud$ = CommCtl.Settings
  Baud_Pos% = InStr(SettingBoud$, ",")
  ModemBoud& = Val(Left(SettingBoud$, Baud_Pos%))
End If

' 2400ボーが設定されている場合は少々時間を置き、モデムとのハンドシェイク
' を完了させる。
If RealBaud& <= 2400 Then
  Pause 8
End If
Exit For
ElseIf InStr(Chk_Result$, "RING") Then
  ' RINGを検出
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "NO CARRIER") Then
  ' キャリアを検出できない
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "ERROR") Then
  ' コマンドにエラーがある
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "NO DIALTONE") Then
  ' ダイアルトーンを検出できない
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "BUSY") Then
  ' 通話中... リダイヤル
  Ret% = PDQ_ER_MODEM_ERROR
  Exit Do
ElseIf InStr(Chk_Result$, "NO ANSWER") Then
  ' 未知結果コード
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
ElseIf InStr(Chk_Result$, "TIMEOUT") Then
  ' 未知結果コード
  Ret% = PDQ_ER_MODEM_ERROR
  Exit For
End If

```

```
End Select
```

```
DoEvents
```

```
Loop
```

```
DoEvents
```

```
Next
```

```
PCExit:
```

```
' AutoProcess&RThresholdの設定を戻す
```

```
If AutoP% Then
```

```
  CommCtl.AutoProcess = AutoP%
```

```
End If
```

```
If RT% Then
```

```
  CommCtl.RThreshold = RT%
```

```
End If
```

```
PlaceCall% = Ret%
```

```
End Function
```

```
Function PortCheck%(CommCtl As Control)
```

```
' 本Functionは、COMポートをチェックする内部ルーチンです
```

```
Dim Test%
```

```
' COMポートをチェック
```

```
On Error Resume Next
```

```
Test% = CommCtl.CommPort
```

```
If Err Then
```

```
' 非アクセス可能ポート
```

```
PortCheck% = PDQ_ER_CTRLERROR
```

```
Else
```

```
If CommCtl.PortOpen = False Then
```

```
' ポートがクローズされています
```

```
PortCheck% = PDQ_ER_PORTCLOSED
```

```
Else
```

```
PortCheck% = PDQ_SUCCESS
```

```
End If
```

```
End If
```

```
On Error GoTo 0
```

```
End Function
```

```
Function RemoveCaret$(CaretString$)
```

```
Dim Caret%, L%
```

```
Dim Char$, Temp$
```

```
Do
```

```
Caret% = InStr(CaretString$, "^")
```

```
L% = Len(CaretString$)
```

```
If Caret Then
```

```
If Caret% < L% Then
```

```
Char$ = Chr$(Asc(Mid$(CaretString$, Caret% + 1, 1)) - 64)
```

```
Temp$ = Left$(CaretString$, Caret% - 1) & Char$
```

```
If Caret% + 2 <= L% Then
```

```
Temp$ = Temp$ & Mid$(CaretString$, Caret% + 2)
```

```
End If
```

```
CaretString$ = Temp$
```

```
End If
```

```
Else
```

```
Exit Do
```

```
End If
```

```
Loop
```

```
RemoveCaret$ = CaretString$
```

```
End Function
```

```
Function ScriptPlay%(CommCtl As Control, FileName$, ErrorLineNum%, ErrorLine$)
```

```
' 本Functionは、スクリプトファイルの実行をします。
```

```
Dim Test%, FileNum%, PortNum%, WasOpen%, AutoP%, RT%
```

```
Dim SinP%, RetValue%, Ret%, TimeOut%, i%, ChkRem%
```

```
Dim CurLine%, LastLine%, TotalLines%, TotalLabels%
```

```
Dim ModemBaud&, RealBaud&
```

```
Dim NumSecs!
```

```
End If
```

```
PlaceCall% = Ret%
```

```
End Function
```

```
Function PortCheck%(CommCtl As Control)
```

```
' 本Functionは、COMポートをチェックする内部ルーチンです
```

```
Dim Test%
```

```
' COMポートをチェック
```

```
On Error Resume Next
```

```
Test% = CommCtl.CommPort
```

```
If Err Then
```

```
' 非アクセス可能ポート
```

```
PortCheck% = PDQ_ER_CTRLERROR
```

```
Else
```

```
If CommCtl.PortOpen = False Then
```

```
' ポートがクローズされています
```

```
PortCheck% = PDQ_ER_PORTCLOSED
```

```
Else
```

```
PortCheck% = PDQ_SUCCESS
```

```
End If
```

```
End If
```

```
On Error GoTo 0
```

```
End Function
```

```
Function RemoveCaret$(CaretString$)
```

```
Dim Caret%, L%
```

```
Dim Char$, Temp$
```

```
Do
```

```
Caret% = InStr(CaretString$, "^")
```

```
L% = Len(CaretString$)
```

```
If Caret Then
```

```
If Caret% < L% Then
```

```
Char$ = Chr$(Asc(Mid$(CaretString$, Caret% + 1, 1)) - 64)
```

```
Temp$ = Left$(CaretString$, Caret% - 1) & Char$
```

```
If Caret% + 2 <= L% Then
```

```
Temp$ = Temp$ & Mid$(CaretString$, Caret% + 2)
```

```
End If
```

```
CaretString$ = Temp$
```

```
End If
```

```
Else
```

```
Exit Do
```

```
End If
```

```
Loop
```

```
RemoveCaret$ = CaretString$
```

```
End Function
```

```
Function ScriptPlay%(CommCtl As Control, FileName$, ErrorLineNum%, ErrorLine$)
```

```
' 本Functionは、スクリプトファイルの実行をします。
```

```
Dim Test%, FileNum%, PortNum%, WasOpen%, AutoP%, RT%
```

```
Dim SinP%, RetValue%, Ret%, TimeOut%, i%, ChkRem%
```

```
Dim CurLine%, LastLine%, TotalLines%, TotalLabels%
```

```
Dim ModemBaud&, RealBaud&
```

```
Dim NumSecs!
```

```
Dim NextLine$, Cmd$, Temp$, Temp1$, TimeOutLabel$, Result$, Mode$
```

```
' Comm ポートが有効であることをチェック
```

```
On Error Resume Next
```

```
Test% = CommCtl.CommPort
```

```
If Err Then
```

```
    ' ポートにアクセス不能
```

```
    ScriptPlay% = PDQ_ER_CTRLERROR
```

```
    Exit Function
```

```
End If
```

```
On Error GoTo 0
```

```
' スクリプトファイルのオープン 有効性の確認
```

```
FileNum% = FreeFile
```

```
On Error Resume Next
```

```
Open FileName$ For Input As FileNum%
```

```
If Err Then
```

```
    ScriptPlay% = PDQ_ER_FILEERROR
```

```
    Exit Function
```

```
End If
```

```
On Error GoTo 0
```

```
' AutoProcess & RThresholdの設定を変更
```

```
AutoP% = CommCtl.AutoProcess
```

```
If AutoP% Then
```

```
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
```

```
End If
```

```
RT% = CommCtl.RThreshold
```

```
If RT% Then
```

```
    CommCtl.RThreshold = 0
```

```
End If
```

```
ReDim script$(1 To 1)
```

```
' スクリプト内容を配列に読み込む
```

```
RetVal% = PDQ_SUCCESS
```

```
CurLine% = 1
```

```
Do
```

```
    Line Input #FileNum%, Temp$
```

```
    If EOF(FileNum%) Then
```

```
        LastLine% = True
```

```
        If Len(Temp$) = 0 Then
```

```
            TotalLines% = CurLine% - 1
```

```
            Exit Do
```

```
        End If
```

```
    End If
```

```
    ReDim Preserve script$(1 To CurLine%)
```

```
    script$(CurLine%) = Trim$(Temp$)
```

```
If Left$(Temp$, 1) = ":" Then
```

```
    ' ラベルをラベル配列に追加
```

```
    On Error Resume Next
```

```
    TotalLabels% = UBound(ScriptLabel)
```

```
    TotalLabels% = TotalLabels% + 1
```

```
    On Error GoTo 0
```

```
    ReDim Preserve ScriptLabel(1 To TotalLabels) As ScriptLabelType
```

```
    ScriptLabel(TotalLabels%).label = Temp$
```

```
    ScriptLabel(TotalLabels%).LineNum = CurLine%
```

```
End If
```

```
If LastLine% Then
```

```
    TotalLines% = CurLine%
```

```
Dim NextLine$, Cmd$, Temp$, Temp1$, TimeOutLabel$, Result$, Mode$
```

```
' Comm ポートが有効であることをチェック
```

```
On Error Resume Next
```

```
Test% = CommCtl.CommPort
```

```
If Err Then
```

```
    ' ポートにアクセス不能
```

```
    ScriptPlay% = PDQ_ER_CTRLERROR
```

```
    Exit Function
```

```
End If
```

```
On Error GoTo 0
```

```
' スクリプトファイルのオープン 有効性の確認
```

```
FileNum% = FreeFile
```

```
On Error Resume Next
```

```
Open FileName$ For Input As FileNum%
```

```
If Err Then
```

```
    ScriptPlay% = PDQ_ER_FILEERROR
```

```
    Exit Function
```

```
End If
```

```
On Error GoTo 0
```

```
' AutoProcess & RThresholdの設定を変更
```

```
AutoP% = CommCtl.AutoProcess
```

```
If AutoP% Then
```

```
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
```

```
End If
```

```
RT% = CommCtl.RThreshold
```

```
If RT% Then
```

```
    CommCtl.RThreshold = 0
```

```
End If
```

```
ReDim script$(1 To 1)
```

```
' スクリプト内容を配列に読み込む
```

```
RetVal% = PDQ_SUCCESS
```

```
CurLine% = 1
```

```
Do
```

```
    Line Input #FileNum%, Temp$
```

```
    If EOF(FileNum%) Then
```

```
        LastLine% = True
```

```
        If Len(Temp$) = 0 Then
```

```
            TotalLines% = CurLine% - 1
```

```
            Exit Do
```

```
        End If
```

```
    End If
```

```
    ReDim Preserve script$(1 To CurLine%)
```

```
    script$(CurLine%) = Trim$(Temp$)
```

```
If Left$(Temp$, 1) = ":" Then
```

```
    ' ラベルをラベル配列に追加
```

```
    On Error Resume Next
```

```
    TotalLabels% = UBound(ScriptLabel)
```

```
    TotalLabels% = TotalLabels% + 1
```

```
    On Error GoTo 0
```

```
    ReDim Preserve ScriptLabel(1 To TotalLabels) As ScriptLabelType
```

```
    ScriptLabel(TotalLabels%).label = Temp$
```

```
    ScriptLabel(TotalLabels%).LineNum = CurLine%
```

```
End If
```

```
If LastLine% Then
```

```
    TotalLines% = CurLine%
```

```

Exit Do
Else
  CurLine% = CurLine% + 1
End If
DoEvents
Loop
Close FileNum%

'メインループ
For ErrorLineNum% = 1 To TotalLines%

  '次行を読み出す
  NextLine$ = script$(ErrorLineNum%)
  ErrorLine$ = NextLine$

  SinP% = InStr(NextLine$, ":")
  If SinP% > 1 Then
    NextLine$ = Left$(NextLine$, SinP% - 1)
  ElseIf SinP% = 1 Then
    GoTo PS_LoopHere
  End If
  SinP% = InStr(NextLine$, "'")
  If SinP% > 1 Then
    'コメント行のチェック
    ChkRem% = InStr(LTrim(NextLine$), "'")
    If ChkRem% = 1 Then
      GoTo PS_LoopHere
    Else
      NextLine$ = Left$(NextLine$, SinP% - 1)
    End If
  ElseIf SinP% = 1 Then
    GoTo PS_LoopHere
  End If

  If Len(NextLine$) = 0 Then GoTo PS_LoopHere
  If Left$(NextLine$, 1) = ":" Then GoTo PS_LoopHere

  'コマンドを整理する
  Temp$ = UCase$(NextLine$)
  SinP% = InStr(Temp$, "'")
  If SinP% Then
    Cmd$ = Left$(Temp$, SinP% - 1)
  Else
    SinP% = InStr(Temp$, Chr$(9))
    If SinP% > 1 Then
      Cmd$ = Left$(Temp$, SinP% - 1)
    Else
      If Len(NextLine$) Then
        Cmd$ = Temp$
      End If
    End If
  End If
End If

'コマンドを実行する
Select Case Cmd$
  Case "ON_TIMEOUT_GOTO"
    If Len(NextLine$) > 15 Then
      TimeOutLabel$ = Trim$(Mid$(NextLine$, 16))
      If Left$(TimeOutLabel$, 1) <> ":" Then
        TimeOutLabel$ = ":" & TimeOutLabel$
      End If
    End If
  End If
End Case

```

```

Exit Do
Else
  CurLine% = CurLine% + 1
End If
DoEvents
Loop
Close FileNum%

'メインループ
For ErrorLineNum% = 1 To TotalLines%

  '次行を読み出す
  NextLine$ = script$(ErrorLineNum%)
  ErrorLine$ = NextLine$

  SinP% = InStr(NextLine$, ":")
  If SinP% > 1 Then
    NextLine$ = Left$(NextLine$, SinP% - 1)
  ElseIf SinP% = 1 Then
    GoTo PS_LoopHere
  End If
  SinP% = InStr(NextLine$, "'")
  If SinP% > 1 Then
    'コメント行のチェック
    ChkRem% = InStr(LTrim(NextLine$), "'")
    If ChkRem% = 1 Then
      GoTo PS_LoopHere
    Else
      NextLine$ = Left$(NextLine$, SinP% - 1)
    End If
  ElseIf SinP% = 1 Then
    GoTo PS_LoopHere
  End If

  If Len(NextLine$) = 0 Then GoTo PS_LoopHere
  If Left$(NextLine$, 1) = ":" Then GoTo PS_LoopHere

  'コマンドを整理する
  Temp$ = UCase$(NextLine$)
  SinP% = InStr(Temp$, "'")
  If SinP% Then
    Cmd$ = Left$(Temp$, SinP% - 1)
  Else
    SinP% = InStr(Temp$, Chr$(9))
    If SinP% > 1 Then
      Cmd$ = Left$(Temp$, SinP% - 1)
    Else
      If Len(NextLine$) Then
        Cmd$ = Temp$
      End If
    End If
  End If
End If

'コマンドを実行する
Select Case Cmd$
  Case "ON_TIMEOUT_GOTO"
    If Len(NextLine$) > 15 Then
      TimeOutLabel$ = Trim$(Mid$(NextLine$, 16))
      If Left$(TimeOutLabel$, 1) <> ":" Then
        TimeOutLabel$ = ":" & TimeOutLabel$
      End If
    End If
  End Case
End Select

```

```

Case "END"
  Beep
  RetValue% = PDQ_SUCCESS
  Exit For
Case "CLOSEPORT"
  If CommCtl.PortOpen Then
    CommCtl.PortOpen = False
    If Err Then
      RetValue% = PDQ_ER_CTRLERROR
      Exit For
    End If
  End If
Case "OPENPORT"
  If Len(NextLine$) > 9 Then
    PortNum% = Val(Mid$(NextLine$, 10))
    'ポート指定の確認
    If PortNum% Then
      Test% = CommCtl.CommPort
      If PortNum% <> Test% Then
        If CommCtl.PortOpen = True Then
          WasOpen% = True
          CommCtl.PortOpen = False
        End If
        '新ポート番号を設定
        CommCtl.CommPort = PortNum%
        On Error Resume Next
        'ポートをオープン
        CommCtl.PortOpen = True
        '有効なポートであるかを確認する
        If Err Then
          '元のポートに戻す
          RetValue% = PDQ_ER_CTRLERROR
          CommCtl.CommPort = Test%
          '再度オープンをかける
          If WasOpen% Then
            CommCtl.PortOpen = True
          End If
        End If
        Exit For
      End If
      On Error GoTo 0
    Else
      If CommCtl.PortOpen = False Then
        On Error Resume Next
        CommCtl.PortOpen = True
        If Err Then
          RetValue% = PDQ_ER_CTRLERROR
          Exit For
        End If
      End If
    End If
  Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
  End If
End If
Case "SETTINGS"
  If Len(NextLine$) > 8 Then
    '設定内容を取得
    Temp$ = Mid$(NextLine$, 9)

```

```

Case "END"
  Beep
  RetValue% = PDQ_SUCCESS
  Exit For
Case "CLOSEPORT"
  If CommCtl.PortOpen Then
    CommCtl.PortOpen = False
    If Err Then
      RetValue% = PDQ_ER_CTRLERROR
      Exit For
    End If
  End If
Case "OPENPORT"
  If Len(NextLine$) > 9 Then
    PortNum% = Val(Mid$(NextLine$, 10))
    'ポート指定の確認
    If PortNum% Then
      Test% = CommCtl.CommPort
      If PortNum% <> Test% Then
        If CommCtl.PortOpen = True Then
          WasOpen% = True
          CommCtl.PortOpen = False
        End If
        '新ポート番号を設定
        CommCtl.CommPort = PortNum%
        On Error Resume Next
        'ポートをオープン
        CommCtl.PortOpen = True
        '有効なポートであるかを確認する
        If Err Then
          '元のポートに戻す
          RetValue% = PDQ_ER_CTRLERROR
          CommCtl.CommPort = Test%
          '再度オープンをかける
          If WasOpen% Then
            CommCtl.PortOpen = True
          End If
        End If
        Exit For
      End If
      On Error GoTo 0
    Else
      If CommCtl.PortOpen = False Then
        On Error Resume Next
        CommCtl.PortOpen = True
        If Err Then
          RetValue% = PDQ_ER_CTRLERROR
          Exit For
        End If
      End If
    End If
  Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
  End If
End If
Case "SETTINGS"
  If Len(NextLine$) > 8 Then
    '設定内容を取得
    Temp$ = Mid$(NextLine$, 9)

```

```

Temp$ = ScriptProcessData$(Temp$)
' カレント設定内容を取得
Temp1$ = CommCtl.Settings

If Temp1$ <> Temp$ Then
  On Error Resume Next
  CommCtl.Settings = Temp$
  If Err Then
    RetValue% = PDQ_ER_CTRLERROR
    CommCtl.Settings = Temp1$
    GoTo PSExitHere
  End If
  On Error GoTo 0
End If
End If

Case "CAPTURE"
If Len(NextLine$) > 8 Then
  Temp$ = ZTrim$(Mid$(NextLine$, 9))
  SinP% = InStr(Temp$, ",")
  If SinP% Then
    CommCtl.CaptureMode = Val(Mid$(Temp$, SinP% + 1))
    If Err Then
      RetValue% = PDQ_BADCOMMAND
      Exit For
    End If
    Temp$ = Left$(Temp$, SinP% - 1)
  End If

  Temp$ = ScriptProcessData$(Temp$)
  CommCtl.CaptureFilename = Temp$
  If Err Then
    RetValue% = PDQ_BADCOMMAND
    Exit For
  End If
End If

Case "CLOSECAPTURE"
CommCtl.CaptureFilename = ""
If Err Then
  RetValue% = PDQ_BADCOMMAND
  Exit For
End If

Case "INPUT"
If Len(NextLine$) > 5 Then
  Temp$ = Mid$(NextLine$, 6)
  SinP% = InStr(Temp$, ",")
  If SinP% Then
    If SinP% + 1 < Len(Temp$) Then
      ' 入力メッセージ
      Temp1$ = Trim$(Left$(Temp$, SinP% - 1))
      Temp1$ = ScriptProcessData$(Temp1$)
      ' 変数
      Temp$ = Trim$(Mid$(Temp$, SinP% + 1))
      Temp$ = ScriptProcessData$(Temp$)

      ' 変数が正常に保存された場合は"XXX"が戻されます。
      ' ScriptVars 内に変数名及び値が保存されています。
      If Left$(Temp$, 3) = "XXX" Then
        ' InputBox$を設定し値を設定させる

```

```

Temp$ = ScriptProcessData$(Temp$)
' カレント設定内容を取得
Temp1$ = CommCtl.Settings

If Temp1$ <> Temp$ Then
  On Error Resume Next
  CommCtl.Settings = Temp$
  If Err Then
    RetValue% = PDQ_ER_CTRLERROR
    CommCtl.Settings = Temp1$
    GoTo PSExitHere
  End If
  On Error GoTo 0
End If
End If

Case "CAPTURE"
If Len(NextLine$) > 8 Then
  Temp$ = ZTrim$(Mid$(NextLine$, 9))
  SinP% = InStr(Temp$, ",")
  If SinP% Then
    CommCtl.CaptureMode = Val(Mid$(Temp$, SinP% + 1))
    If Err Then
      RetValue% = PDQ_BADCOMMAND
      Exit For
    End If
    Temp$ = Left$(Temp$, SinP% - 1)
  End If

  Temp$ = ScriptProcessData$(Temp$)
  CommCtl.CaptureFilename = Temp$
  If Err Then
    RetValue% = PDQ_BADCOMMAND
    Exit For
  End If
End If

Case "CLOSECAPTURE"
CommCtl.CaptureFilename = ""
If Err Then
  RetValue% = PDQ_BADCOMMAND
  Exit For
End If

Case "INPUT"
If Len(NextLine$) > 5 Then
  Temp$ = Mid$(NextLine$, 6)
  SinP% = InStr(Temp$, ",")
  If SinP% Then
    If SinP% + 1 < Len(Temp$) Then
      ' 入力メッセージ
      Temp1$ = Trim$(Left$(Temp$, SinP% - 1))
      Temp1$ = ScriptProcessData$(Temp1$)
      ' 変数
      Temp$ = Trim$(Mid$(Temp$, SinP% + 1))
      Temp$ = ScriptProcessData$(Temp$)

      ' 変数が正常に保存された場合は"XXX"が戻されます。
      ' ScriptVars 内に変数名及び値が保存されています。
      If Left$(Temp$, 3) = "XXX" Then
        ' InputBox$を設定し値を設定させる

```

```

        Test% = Val(Mid$(Temp$, 4))
        ScriptVars(Test%).Value = UCase$(InputBox$(Temp1$, UCase$(File
    Else
        RetValue% = PDQ_BADCOMMAND
        Exit For
    End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Case "DIAL"
If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

If Len(NextLine$) > 4 Then
    Temp$ = ZTrim$(Mid$(NextLine$, 5))
    SinP% = InStr(Temp$, ",")
    If SinP% Then
        Test% = Val(Mid$(Temp$, SinP% + 1))
        Temp$ = Left$(Temp$, SinP% - 1)
    Else
        Test% = 1
    End If
    Temp$ = ScriptProcessData$(Temp$)

    RetValue% = PlaceCall%(CommCtl, Temp$, Test%, TimeOut%, Result$, ModemBaud

    Select Case RetValue%
        Case PDQ_ER_TIMEOUT
            GoSub ProcessTimeout
        Case Is <> PDQ_SUCCESS
            Exit For
    End Select
End If

Case "TIMEOUT"
If Len(NextLine$) > 7 Then
    TimeOut% = Val(Mid$(NextLine$, 8))
End If

Case "WAITFOR"
If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If
If Len(NextLine$) > 7 Then
    Temp$ = Mid$(NextLine$, 8)
    Temp$ = ScriptProcessData$(Temp$)
    If Len(Temp$) = 0 Then
        ' 指定文字列がありません

```

```

        Test% = Val(Mid$(Temp$, 4))
        ScriptVars(Test%).Value = UCase$(InputBox$(Temp1$, UCase$(File
    Else
        RetValue% = PDQ_BADCOMMAND
        Exit For
    End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Else
    RetValue% = PDQ_BADCOMMAND
    Exit For
End If
Case "DIAL"
If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

If Len(NextLine$) > 4 Then
    Temp$ = ZTrim$(Mid$(NextLine$, 5))
    SinP% = InStr(Temp$, ",")
    If SinP% Then
        Test% = Val(Mid$(Temp$, SinP% + 1))
        Temp$ = Left$(Temp$, SinP% - 1)
    Else
        Test% = 1
    End If
    Temp$ = ScriptProcessData$(Temp$)

    RetValue% = PlaceCall%(CommCtl, Temp$, Test%, TimeOut%, Result$, ModemBaud

    Select Case RetValue%
        Case PDQ_ER_TIMEOUT
            GoSub ProcessTimeout
        Case Is <> PDQ_SUCCESS
            Exit For
    End Select
End If

Case "TIMEOUT"
If Len(NextLine$) > 7 Then
    TimeOut% = Val(Mid$(NextLine$, 8))
End If

Case "WAITFOR"
If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If
If Len(NextLine$) > 7 Then
    Temp$ = Mid$(NextLine$, 8)
    Temp$ = ScriptProcessData$(Temp$)
    If Len(Temp$) = 0 Then
        ' 指定文字列がありません

```

```

    RetValue% = PDQ_BADCOMMAND
    Exit For
End If

RetValue% = WaitFor%(CommCtl, Temp$, TimeOut%, False, False, Result$)

Select Case RetValue%
    Case PDQ_ER_TIMEOUT
        GoSub ProcessTimeout
    Case Is <> PDQ_SUCCESS
        Exit For
End Select
End If

Case "SEND_MODEM_CMD"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

If Len(NextLine$) > 4 Then
    Temp$ = Mid$(NextLine$, 5)
    Temp$ = ScriptProcessData$(Temp$)
    RetValue% = SendModemCmd%(CommCtl, Temp$, 1, TimeOut%, Result$)
    Select Case RetValue%
        Case PDQ_ER_TIMEOUT
            GoSub ProcessTimeout
        Case Is <> PDQ_SUCCESS
            Exit For
    End Select
End If

Case "SEND"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If
If Len(NextLine$) > 4 Then
    Temp$ = Mid$(NextLine$, 5)
    Temp$ = ScriptProcessData$(Temp$)
    CommCtl.Output = Temp$
End If

Case "HANGUP"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

RetValue% = Hangup%(CommCtl, 2, 1, 2000)

If RetValue% <> PDQ_SUCCESS Then
    Exit For
End If

Case "PROTOCOL"

If Len(NextLine$) > 8 Then
    Temp$ = Mid$(NextLine$, 9)
    Temp$ = UCase$(Trim$(Temp$))
    Temp$ = ScriptProcessData$(Temp$)

```

```

    RetValue% = PDQ_BADCOMMAND
    Exit For
End If

RetValue% = WaitFor%(CommCtl, Temp$, TimeOut%, False, False, Result$)

Select Case RetValue%
    Case PDQ_ER_TIMEOUT
        GoSub ProcessTimeout
    Case Is <> PDQ_SUCCESS
        Exit For
End Select
End If

Case "SEND_MODEM_CMD"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

If Len(NextLine$) > 4 Then
    Temp$ = Mid$(NextLine$, 5)
    Temp$ = ScriptProcessData$(Temp$)
    RetValue% = SendModemCmd%(CommCtl, Temp$, 1, TimeOut%, Result$)
    Select Case RetValue%
        Case PDQ_ER_TIMEOUT
            GoSub ProcessTimeout
        Case Is <> PDQ_SUCCESS
            Exit For
    End Select
End If

Case "SEND"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If
If Len(NextLine$) > 4 Then
    Temp$ = Mid$(NextLine$, 5)
    Temp$ = ScriptProcessData$(Temp$)
    CommCtl.Output = Temp$
End If

Case "HANGUP"

If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
End If

RetValue% = Hangup%(CommCtl, 2, 1, 2000)

If RetValue% <> PDQ_SUCCESS Then
    Exit For
End If

Case "PROTOCOL"

If Len(NextLine$) > 8 Then
    Temp$ = Mid$(NextLine$, 9)
    Temp$ = UCase$(Trim$(Temp$))
    Temp$ = ScriptProcessData$(Temp$)

```

```

Select Case Temp$
  Case "XMODEM-CHECKSUM"
    CommCtl.XferProtocol = PDQ_XMODEM_CHECKSUM
  Case "XMODEM-CRC"
    CommCtl.XferProtocol = PDQ_XMODEM_CRC
  Case "XMODEM-1K"
    CommCtl.XferProtocol = PDQ_XMODEM_1K
  Case "YMODEM-G-BATCH"
    CommCtl.XferProtocol = PDQ_YMODEM_G_BATCH
  Case "YMODEM-BATCH"
    CommCtl.XferProtocol = PDQ_YMODEM_BATCH
  Case "ZMODEM"
    CommCtl.XferProtocol = PDQ_ZMODEM
  Case "KERMIT"
    CommCtl.XferProtocol = PDQ_KERMIT
  Case "COMPUSERVE"
    CommCtl.XferProtocol = PDQ_COMPUSERVE_BPLUS
  Case "NIFTYSERVE"
    CommCtl.XferProtocol = PDQ_NIFTYSERVE_BPLUS
  Case Else
    RetValue% = PDQ_BADCOMMAND
  Exit For
End Select
End If

Case "UPLOAD"
  If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
  End If

  If Len(NextLine$) > 6 Then
    Temp$ = Mid$(NextLine$, 7)
    Temp$ = ScriptProcessData$(Temp$)

    Select Case CommCtl.XferProtocol
      Case Is <> PDQ_COMPUSERVE_BPLUS, PDQ_KERMIT, PDQ_NIFTYSERVE_BPLUS
        On Error Resume Next
        FileNum% = FreeFile
        Open Temp$ For Input As FileNum%
        Test% = Err
        Close FileNum%
        If Test% Then
          RetValue% = PDQ_ER_FILEERROR
          Exit For
        End If
      End Select
    End If

    ' ダイアログボックスを使用する
    CommCtl.XferStatusDialog = PDQ_XFERDIALOG_MODELESS
    ' 送信
    CommCtl.Upload = Temp$

    ' 送信を開始するまで待つ
    Do
      DoEvents
      If CancelFlag% Then
        RetValue% = PDQ_ER_CANCELED
        Exit Do
      End If
    Loop Until CommCtl.XferStatus

```

```

Select Case Temp$
  Case "XMODEM-CHECKSUM"
    CommCtl.XferProtocol = PDQ_XMODEM_CHECKSUM
  Case "XMODEM-CRC"
    CommCtl.XferProtocol = PDQ_XMODEM_CRC
  Case "XMODEM-1K"
    CommCtl.XferProtocol = PDQ_XMODEM_1K
  Case "YMODEM-G-BATCH"
    CommCtl.XferProtocol = PDQ_YMODEM_G_BATCH
  Case "YMODEM-BATCH"
    CommCtl.XferProtocol = PDQ_YMODEM_BATCH
  Case "ZMODEM"
    CommCtl.XferProtocol = PDQ_ZMODEM
  Case "KERMIT"
    CommCtl.XferProtocol = PDQ_KERMIT
  Case "COMPUSERVE"
    CommCtl.XferProtocol = PDQ_COMPUSERVE_BPLUS
  Case "NIFTYSERVE"
    CommCtl.XferProtocol = PDQ_NIFTYSERVE_BPLUS
  Case Else
    RetValue% = PDQ_BADCOMMAND
  Exit For
End Select
End If

Case "UPLOAD"
  If CommCtl.PortOpen = False Then
    RetValue% = PDQ_ER_PORTCLOSED
    GoTo PSExitHere
  End If

  If Len(NextLine$) > 6 Then
    Temp$ = Mid$(NextLine$, 7)
    Temp$ = ScriptProcessData$(Temp$)

    Select Case CommCtl.XferProtocol
      Case Is <> PDQ_COMPUSERVE_BPLUS, PDQ_KERMIT, PDQ_NIFTYSERVE_BPLUS
        On Error Resume Next
        FileNum% = FreeFile
        Open Temp$ For Input As FileNum%
        Test% = Err
        Close FileNum%
        If Test% Then
          RetValue% = PDQ_ER_FILEERROR
          Exit For
        End If
      End Select
    End If

    ' ダイアログボックスを使用する
    CommCtl.XferStatusDialog = PDQ_XFERDIALOG_MODELESS
    ' 送信
    CommCtl.Upload = Temp$

    ' 送信を開始するまで待つ
    Do
      DoEvents
      If CancelFlag% Then
        RetValue% = PDQ_ER_CANCELED
        Exit Do
      End If
    Loop Until CommCtl.XferStatus

```

```

'送信が終了するまで待つ
Do
  Pause 2
  If CommCtl.XferStatus = PDQ_XFER_TERM_OK Then Exit Do
  If CommCtl.XferStatus = PDQ_XFER_TERM_ERROR Then Exit Do
Loop

Case "DOWNLOAD"
If CommCtl.PortOpen = False Then
  RetValue% = PDQ_ER_PORTCLOSED
  GoTo PSExitHere
End If
If Len(NextLine$) > 8 Then
  Temp$ = Mid$(NextLine$, 9)
  Temp$ = ScriptProcessData$(Temp$)
End If

Select Case CommCtl.XferProtocol
  Case PDQ_XMODEM_CHECKSUM, PDQ_XMODEM_CRC, PDQ_XMODEM_1K
    On Error Resume Next
    FileNum% = FreeFile
    Open Temp$ For Input As FileNum%
    Test% = Err
    Close FileNum%
    If Test% Then
      RetValue% = PDQ_ER_FILEERROR
      Exit For
    End If
  End Select

CommCtl.XferStatusDialog = PDQ_XFERDIALOG_MODELESS
CommCtl.Download = Temp$

'受信を開始するまで待つ
Do
  DoEvents
  If CancelFlag% Then
    RetValue% = PDQ_ER_CANCELED
    Exit Do
  End If
Loop Until CommCtl.XferStatus

'受信が終了するまで待つ
Do
  Pause 2
  If CommCtl.XferStatus = PDQ_XFER_TERM_OK Then Exit Do
  If CommCtl.XferStatus = PDQ_XFER_TERM_ERROR Then Exit Do
Loop

Case "PAUSE"
If Len(NextLine$) > 5 Then
  NumSecs! = Val(Mid$(NextLine$, 6))
  If NumSecs! > 0 Then
    Pause NumSecs!
  End If
End If
Case Else
  RetValue% = PDQ_ER_NOCMD
  Exit For
End Select

```

PS\_LoopHere:

```

'送信が終了するまで待つ
Do
  Pause 2
  If CommCtl.XferStatus = PDQ_XFER_TERM_OK Then Exit Do
  If CommCtl.XferStatus = PDQ_XFER_TERM_ERROR Then Exit Do
Loop

Case "DOWNLOAD"
If CommCtl.PortOpen = False Then
  RetValue% = PDQ_ER_PORTCLOSED
  GoTo PSExitHere
End If
If Len(NextLine$) > 8 Then
  Temp$ = Mid$(NextLine$, 9)
  Temp$ = ScriptProcessData$(Temp$)
End If

Select Case CommCtl.XferProtocol
  Case PDQ_XMODEM_CHECKSUM, PDQ_XMODEM_CRC, PDQ_XMODEM_1K
    On Error Resume Next
    FileNum% = FreeFile
    Open Temp$ For Input As FileNum%
    Test% = Err
    Close FileNum%
    If Test% Then
      RetValue% = PDQ_ER_FILEERROR
      Exit For
    End If
  End Select

CommCtl.XferStatusDialog = PDQ_XFERDIALOG_MODELESS
CommCtl.Download = Temp$

'受信を開始するまで待つ
Do
  DoEvents
  If CancelFlag% Then
    RetValue% = PDQ_ER_CANCELED
    Exit Do
  End If
Loop Until CommCtl.XferStatus

'受信が終了するまで待つ
Do
  Pause 2
  If CommCtl.XferStatus = PDQ_XFER_TERM_OK Then Exit Do
  If CommCtl.XferStatus = PDQ_XFER_TERM_ERROR Then Exit Do
Loop

Case "PAUSE"
If Len(NextLine$) > 5 Then
  NumSecs! = Val(Mid$(NextLine$, 6))
  If NumSecs! > 0 Then
    Pause NumSecs!
  End If
End If
Case Else
  RetValue% = PDQ_ER_NOCMD
  Exit For
End Select

```

PS\_LoopHere:

```

Next
If RetValue% = PDQ_SUCCESS Then
    ErrorLineNum = 0
    ErrorLine$ = ""
End If

```

PSExitHere:

```

'変数配列を解放する
Erase ScriptVars, ScriptLabel

Close FileNum%

'AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

ScriptPlay% = RetValue%
Exit Function

```

ProcessTimeout:

```

If Len(TimeOutLabel$) Then
    On Error Resume Next
    TotalLabels% = UBound(ScriptLabel)
    If TotalLabels% Then
        For i% = 1 To TotalLabels%
            If ScriptLabel(i%).label = TimeOutLabel$ Then
                ErrorLineNum% = ScriptLabel(i%).LineNum
                Exit For
            End If
        Next
    End If
End If
Return

```

End Function

```

Function ScriptProcessData$(ScriptData$)
'当ルーチンは内部関数でスクリプトファイルを実行する時に使用します。
'一行ずつファイルからデータを読み込みすべてのコマンド文句を検出し実際の値に
'変換し実行します。
Dim FirstQuote%, SecondQuote%, index%
Dim Temp$

'文字列内のスペースを抜き取る
ScriptData$ = ZTrim$(ScriptData$)

'ダブルコーテーションを検出
FirstQuote% = InStr(ScriptData$, Chr$(34))
If FirstQuote% Then
    If FirstQuote% < Len(ScriptData$) Then
        Temp$ = Mid$(ScriptData$, FirstQuote% + 1)
        SecondQuote% = InStr(Temp$, Chr$(34))
        If SecondQuote% Then
            ScriptData$ = Mid$(ScriptData$, FirstQuote% + 1, SecondQuote% - 1)

```

```

Next
If RetValue% = PDQ_SUCCESS Then
    ErrorLineNum = 0
    ErrorLine$ = ""
End If

```

PSExitHere:

```

'変数配列を解放する
Erase ScriptVars, ScriptLabel

Close FileNum%

'AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

ScriptPlay% = RetValue%
Exit Function

```

ProcessTimeout:

```

If Len(TimeOutLabel$) Then
    On Error Resume Next
    TotalLabels% = UBound(ScriptLabel)
    If TotalLabels% Then
        For i% = 1 To TotalLabels%
            If ScriptLabel(i%).label = TimeOutLabel$ Then
                ErrorLineNum% = ScriptLabel(i%).LineNum
                Exit For
            End If
        Next
    End If
End If
Return

```

End Function

```

Function ScriptProcessData$(ScriptData$)
'当ルーチンは内部関数でスクリプトファイルを実行する時に使用します。
'一行ずつファイルからデータを読み込みすべてのコマンド文句を検出し実際の値に
'変換し実行します。
Dim FirstQuote%, SecondQuote%, index%
Dim Temp$

'文字列内のスペースを抜き取る
ScriptData$ = ZTrim$(ScriptData$)

'ダブルコーテーションを検出
FirstQuote% = InStr(ScriptData$, Chr$(34))
If FirstQuote% Then
    If FirstQuote% < Len(ScriptData$) Then
        Temp$ = Mid$(ScriptData$, FirstQuote% + 1)
        SecondQuote% = InStr(Temp$, Chr$(34))
        If SecondQuote% Then
            ScriptData$ = Mid$(ScriptData$, FirstQuote% + 1, SecondQuote% - 1)

```

```

Else
    ScriptData$ = Mid$(ScriptData$, FirstQuote% + 1)
End If
End If
Else
    ' コマンド配列に該当する文字列であるかを確認する
    index% = ScriptVarIndex(ScriptData$)
    If Len(ScriptVars(index%).Value) Then
        ' 存在する場合リターン値を戻す
        ScriptData$ = ScriptVars(index%).Value
    Else
        ' 印をつけて新しいスクリプト文字列であることを確認するためにインデックスをつける
        ScriptData$ = "XXX" & LTrim$(Str$(index%))
    End If
End If

ScriptData$ = RemoveCaret$(ScriptData$)

ScriptProcessData$ = ScriptData$

```

End Function

```

Function ScriptVarIndex(variable$)
' 内部関数: スクリプト配列から同一内容のデータの存在を確認します。
' 確認されない場合は新要素として追加します。

```

```

Dim Elements%, i%

' 配列の大きさを取得します
On Error Resume Next
Elements% = UBound(ScriptVars)
On Error GoTo 0

' 同一の内容の配列があるかどうかをチェック
If Elements% Then
    For i% = 1 To Elements%
        If ScriptVars(i%).variable = variable$ Then
            ScriptVarIndex = i%
            Exit Function
        End If
    Next
End If

' 新しいデータで、新要素として追加する
Elements% = Elements% + 1
ReDim Preserve ScriptVars(1 To Elements%) As ScriptVariableType
ScriptVars(Elements%).variable = variable$
ScriptVarIndex = Elements%

```

End Function

```

Function SendModemCmd%(CommCtl As Control, Cmd$, NumTries%, TimeOut%, Result$)

```

```

' 本Functionは、命令を端末に送ります。ユーザ定義命令も送れます。
Dim Ret%, AutoP%, RT%, Test%, CDFlag%, ExcludeFlag%, Found%, i%, Chk%
Dim Attention$

' COMポートをチェック
SendModemCmd% = PortCheck%(CommCtl)
If SendModemCmd% <> PDQ_SUCCESS Then
    Exit Function
End If

```

```

Else
    ScriptData$ = Mid$(ScriptData$, FirstQuote% + 1)
End If
End If
Else
    ' コマンド配列に該当する文字列であるかを確認する
    index% = ScriptVarIndex(ScriptData$)
    If Len(ScriptVars(index%).Value) Then
        ' 存在する場合リターン値を戻す
        ScriptData$ = ScriptVars(index%).Value
    Else
        ' 印をつけて新しいスクリプト文字列であることを確認するためにインデックスをつける
        ScriptData$ = "XXX" & LTrim$(Str$(index%))
    End If
End If

ScriptData$ = RemoveCaret$(ScriptData$)

ScriptProcessData$ = ScriptData$

```

End Function

```

Function ScriptVarIndex(variable$)
' 内部関数: スクリプト配列から同一内容のデータの存在を確認します。
' 確認されない場合は新要素として追加します。

```

```

Dim Elements%, i%

' 配列の大きさを取得します
On Error Resume Next
Elements% = UBound(ScriptVars)
On Error GoTo 0

' 同一の内容の配列があるかどうかをチェック
If Elements% Then
    For i% = 1 To Elements%
        If ScriptVars(i%).variable = variable$ Then
            ScriptVarIndex = i%
            Exit Function
        End If
    Next
End If

' 新しいデータで、新要素として追加する
Elements% = Elements% + 1
ReDim Preserve ScriptVars(1 To Elements%) As ScriptVariableType
ScriptVars(Elements%).variable = variable$
ScriptVarIndex = Elements%

```

End Function

```

Function SendModemCmd%(CommCtl As Control, Cmd$, NumTries%, TimeOut%, Result$)

```

```

' 本Functionは、命令を端末に送ります。ユーザ定義命令も送れます。
Dim Ret%, AutoP%, RT%, Test%, CDFlag%, ExcludeFlag%, Found%, i%, Chk%
Dim Attention$

' COMポートをチェック
SendModemCmd% = PortCheck%(CommCtl)
If SendModemCmd% <> PDQ_SUCCESS Then
    Exit Function
End If

```

```

' 変数を初期化
Result$ = ""
Ret% = PDQ_SUCCESS

' WaitForA$用の配列を作成する
ReDim WaitArray$(1 To 9)
WaitArray$(1) = "OK"
WaitArray$(2) = "CONNECT"
WaitArray$(3) = "RING"
WaitArray$(4) = "NO CARRIER"
WaitArray$(5) = "ERROR"
WaitArray$(6) = "NO DIALTONE"
WaitArray$(7) = "BUSY"
WaitArray$(8) = "NO ANSWER"
WaitArray$(9) = "TIMEOUT"

' 命令が有効であるかどうかを確認
If Len(Cmd$) = 0 Then
    SendModemCmd% = PDQ_ER_NOCMD
    Exit Function
Else
    Attention$ = "AT"
    ' 命令の先頭に「AT」を追加する
    If UCase$(Left$(Cmd$, 2)) <> Attention$ Then
        Cmd$ = Attention$ & Cmd$
    End If
    ' 命令文字列の末尾文字がCRであることを確認
    If Right$(Cmd$, 1) <> Chr$(13) Then
        Cmd$ = Cmd$ & Chr$(13)
    End If
    ' ダブルコーテーションで開始するかどうかを確認
    If Left$(Cmd$, 1) = Chr$(34) Then
        Cmd$ = Mid$(Cmd$, 2)
    End If
End If

' AutoProcess & RThreshold の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

' 受信バッファをクリアする
CommCtl.InBufferCount = 0

If NumTries% = 0 Then NumTries% = 1

For i% = 1 To NumTries%
    ' 命令を送出する
    CommCtl.Output = Cmd$
    ' --結果コードが戻るまで待つ
    CDFlag% = False
    ExcludeFlag% = False
    Ret% = WaitForA%(CommCtl, WaitArray$(), Timeout%, CDFlag%, ExcludeFlag%, Found%, Result)
    Select Case Ret%
        Case PDQ_ER_TIMEOUT ' --再送処理
            ' 1秒間ホールドする

```

```

' 変数を初期化
Result$ = ""
Ret% = PDQ_SUCCESS

' WaitForA$用の配列を作成する
ReDim WaitArray$(1 To 9)
WaitArray$(1) = "OK"
WaitArray$(2) = "CONNECT"
WaitArray$(3) = "RING"
WaitArray$(4) = "NO CARRIER"
WaitArray$(5) = "ERROR"
WaitArray$(6) = "NO DIALTONE"
WaitArray$(7) = "BUSY"
WaitArray$(8) = "NO ANSWER"
WaitArray$(9) = "TIMEOUT"

' 命令が有効であるかどうかを確認
If Len(Cmd$) = 0 Then
    SendModemCmd% = PDQ_ER_NOCMD
    Exit Function
Else
    Attention$ = "AT"
    ' 命令の先頭に「AT」を追加する
    If UCase$(Left$(Cmd$, 2)) <> Attention$ Then
        Cmd$ = Attention$ & Cmd$
    End If
    ' 命令文字列の末尾文字がCRであることを確認
    If Right$(Cmd$, 1) <> Chr$(13) Then
        Cmd$ = Cmd$ & Chr$(13)
    End If
    ' ダブルコーテーションで開始するかどうかを確認
    If Left$(Cmd$, 1) = Chr$(34) Then
        Cmd$ = Mid$(Cmd$, 2)
    End If
End If

' AutoProcess & RThreshold の設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

' 受信バッファをクリアする
CommCtl.InBufferCount = 0

If NumTries% = 0 Then NumTries% = 1

For i% = 1 To NumTries%
    ' 命令を送出する
    CommCtl.Output = Cmd$
    ' --結果コードが戻るまで待つ
    CDFlag% = False
    ExcludeFlag% = False
    Ret% = WaitForA%(CommCtl, WaitArray$(), Timeout%, CDFlag%, ExcludeFlag%, Found%, Result)
    Select Case Ret%
        Case PDQ_ER_TIMEOUT ' --再送処理
            ' 1秒間ホールドする

```

```

    If i% <> NumTries% Then
        Pause 1
    End If
Case PDQ_ER_CANCELED
Exit For
Case PDQ_SUCCESS 'モデムエラーのチェック
    If Found% > 2 Then
        Select Case Found%
            Case 3 'RINGを検出(RING)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 4 'キャリアを検出できない(NO CARRIER)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 5 'コマンドにエラーがある(ERROR)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 6 'ダイヤルトーンを検出できない(NO DIALTONE)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 7 '通話中...リダイヤル(BUSY)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 8 '無音区間を検出出来ない(NO ANSWER)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 9 '未知結果コード(TIMEOUT)
                Ret% = PDQ_ER_MODEM_ERROR
        End Select
    End Select
Else
Exit For
End If
End Select
Next i%

```

```
' AutoProcess and RThresholdの設定を戻す
```

```

If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

```
SendModemCmd% = Ret%
```

```
End Function
```

```
Function Upload%(CommCtl As Control, Protocol%, FileName$, DialogMode%)
```

```
' 本Functionは、指定したプロトコルでファイルをアップロードします
```

```
Dim Test%
```

```
' COMポートをチェック
```

```

Upload% = PortCheck%(CommCtl)
If Upload% <> PDQ_SUCCESS Then
Exit Function
End If

```

```
' プロトコルを設定します
```

```
CommCtl.XferProtocol = Protocol%
```

```
' ダイアログボックスの表示を設定
```

```
CommCtl.XferStatusDialog = DialogMode%
```

```
If Err Then
```

```

' 設定が無効です
Upload% = PDQ_ER_CTRLERROR
Exit Function

```

```

    If i% <> NumTries% Then
        Pause 1
    End If
Case PDQ_ER_CANCELED
Exit For
Case PDQ_SUCCESS 'モデムエラーのチェック
    If Found% > 2 Then
        Select Case Found%
            Case 3 'RINGを検出(RING)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 4 'キャリアを検出できない(NO CARRIER)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 5 'コマンドにエラーがある(ERROR)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 6 'ダイヤルトーンを検出できない(NO DIALTONE)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 7 '通話中...リダイヤル(BUSY)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 8 '無音区間を検出出来ない(NO ANSWER)
                Ret% = PDQ_ER_MODEM_ERROR
            Case 9 '未知結果コード(TIMEOUT)
                Ret% = PDQ_ER_MODEM_ERROR
        End Select
    End Select
Else
Exit For
End If
End Select
Next i%

```

```
' AutoProcess and RThresholdの設定を戻す
```

```

If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

```
SendModemCmd% = Ret%
```

```
End Function
```

```
Function Upload%(CommCtl As Control, Protocol%, FileName$, DialogMode%)
```

```
' 本Functionは、指定したプロトコルでファイルをアップロードします
```

```
Dim Test%
```

```
' COMポートをチェック
```

```

Upload% = PortCheck%(CommCtl)
If Upload% <> PDQ_SUCCESS Then
Exit Function
End If

```

```
' プロトコルを設定します
```

```
CommCtl.XferProtocol = Protocol%
```

```
' ダイアログボックスの表示を設定
```

```
CommCtl.XferStatusDialog = DialogMode%
```

```
If Err Then
```

```

' 設定が無効です
Upload% = PDQ_ER_CTRLERROR
Exit Function

```

```

End If

'送信を開始する
CommCtl.Upload = FileName$
If Err Then
    'Uploadを開始出来ません
    Upload% = PDQ_ER_CTRLERROR
Else
    'Upload 開始
    Upload% = PDQ_SUCCESS
End If

End Function

Function WaitFor%(CommCtl As Control, Wait$, TimeOut%, CDFlag%, ExcludeFlag%, Received$)
'本Functionは、特定文字列の受信を待ち、WaitForを呼び出したときから受信した
'すべての文字（受信待ちの文字列を含む）を返します。
Dim Test%, AutoP%, RT%, OldLen%, There%
Dim Start!, EndTime!
Dim InString$, NewData$, WaitChk$
Dim Flag24 As Boolean

'COMポートをチェック
WaitFor% = PortCheck%(CommCtl)
If WaitFor% <> PDQ_SUCCESS Then
    Exit Function
End If

'AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

'InputLen を1に変更
OldLen% = CommCtl.InputLen
If OldLen% <> 1 Then
    CommCtl.InputLen = 1
End If

'変数の初期化
WaitFor% = PDQ_SUCCESS
Received$ = ""

Start! = Timer
If Timer + TimeOut% > 86400 Then
    EndTime! = TimeOut% - (86400 - Timer)
    Flag24 = True
Else
    EndTime! = Start! + TimeOut%
    Flag24 = False
End If

'受信待ち文字列をANISコードに変換
WaitChk$ = StrConv(Wait$, vbFromUnicode)

Do

```

```

End If

'送信を開始する
CommCtl.Upload = FileName$
If Err Then
    'Uploadを開始出来ません
    Upload% = PDQ_ER_CTRLERROR
Else
    'Upload 開始
    Upload% = PDQ_SUCCESS
End If

End Function

Function WaitFor%(CommCtl As Control, Wait$, TimeOut%, CDFlag%, ExcludeFlag%, Received$)
'本Functionは、特定文字列の受信を待ち、WaitForを呼び出したときから受信した
'すべての文字（受信待ちの文字列を含む）を返します。
Dim Test%, AutoP%, RT%, OldLen%, There%
Dim Start!, EndTime!
Dim InString$, NewData$, WaitChk$
Dim Flag24 As Boolean

'COMポートをチェック
WaitFor% = PortCheck%(CommCtl)
If WaitFor% <> PDQ_SUCCESS Then
    Exit Function
End If

'AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

'InputLen を1に変更
OldLen% = CommCtl.InputLen
If OldLen% <> 1 Then
    CommCtl.InputLen = 1
End If

'変数の初期化
WaitFor% = PDQ_SUCCESS
Received$ = ""

Start! = Timer
If Timer + TimeOut% > 86400 Then
    EndTime! = TimeOut% - (86400 - Timer)
    Flag24 = True
Else
    EndTime! = Start! + TimeOut%
    Flag24 = False
End If

'受信待ち文字列をANISコードに変換
WaitChk$ = StrConv(Wait$, vbFromUnicode)

Do

```

```

'データを受信したかを確認
If CommCtl.InBufferCount Then
  'データを読み取る
  NewData$ = CommCtl.InputB
  DoEvents
  'InString$に追加する
  InString$ = InString$ & NewData$

  '受信待ちの文字列が取得されたかを確認
  There% = InStrB(InString$, WaitChk$)
  If There% Then
    If ExcludeFlag% Then
      Received$ = LeftB(InString$, There% - 1)
    Else
      Received$ = InString$
    End If
    WaitFor% = PDQ_SUCCESS
    Exit Do
  End If

  'InString$ は大きいすぎるかどうかをチェックする
  If Len(InString$) > 32000 Then
    InString$ = Right$(InString$, 32000)
  End If
End If

'処理が中断されたかを確認する
If CancelFlag% Then
  Received$ = InString$
  WaitFor% = PDQ_ER_CANCELED
  Exit Do
End If

'Carrierを確認する
If CDFlag% Then
  If CommCtl.CD Holding = False Then
    Received$ = InString$
    WaitFor% = PDQ_ER_LOST_CARRIER
    Exit Do
  End If
End If

'タイムアウトの確認
If Flag24 = False Then
  If Timer >= EndTime! Then
    Received$ = InString$
    WaitFor% = PDQ_ER_TIMEOUT
    Exit Do
  End If
Else
  If Timer < Start! Then
    Flag24 = False
  End If
End If

DoEvents

```

Loop

```

'受信文字列をUnicodeに変換
Received$ = StrConv(Received$, vbUnicode)

```

```

'データを受信したかを確認
If CommCtl.InBufferCount Then
  'データを読み取る
  NewData$ = CommCtl.InputB
  DoEvents
  'InString$に追加する
  InString$ = InString$ & NewData$

  '受信待ちの文字列が取得されたかを確認
  There% = InStrB(InString$, WaitChk$)
  If There% Then
    If ExcludeFlag% Then
      Received$ = LeftB(InString$, There% - 1)
    Else
      Received$ = InString$
    End If
    WaitFor% = PDQ_SUCCESS
    Exit Do
  End If

  'InString$ は大きいすぎるかどうかをチェックする
  If Len(InString$) > 32000 Then
    InString$ = Right$(InString$, 32000)
  End If
End If

'処理が中断されたかを確認する
If CancelFlag% Then
  Received$ = InString$
  WaitFor% = PDQ_ER_CANCELED
  Exit Do
End If

'Carrierを確認する
If CDFlag% Then
  If CommCtl.CD Holding = False Then
    Received$ = InString$
    WaitFor% = PDQ_ER_LOST_CARRIER
    Exit Do
  End If
End If

'タイムアウトの確認
If Flag24 = False Then
  If Timer >= EndTime! Then
    Received$ = InString$
    WaitFor% = PDQ_ER_TIMEOUT
    Exit Do
  End If
Else
  If Timer < Start! Then
    Flag24 = False
  End If
End If

DoEvents

```

Loop

```

'受信文字列をUnicodeに変換
Received$ = StrConv(Received$, vbUnicode)

```

```
' InputLenの設定を戻す
If OldLen% <> 1 Then
    CommCtl.InputLen = OldLen%
End If
```

```
' AutoProcess and RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If
```

End Function

Function WaitForA%(CommCtl As Control, Wait\$(), TimeOut%, CDFlag%, ExcludeFlag%, Found%, Recei

' 本Functionは、WaitFor\$と同機能ですが受信文字列に配列を指定することができます。

```
Dim Test%, AutoP%, RT%, NumEls%, FirstEl%, OldLen%, There%, i%
Dim Start!, EndTime!
Dim InString$, NewData$, T$
Dim Flag24 As Boolean
```

```
' COMポートをチェック
WaitForA% = PortCheck%(CommCtl)
If WaitForA% <> PDQ_SUCCESS Then
    Exit Function
End If
```

' 文字列の数を取得する

```
On Error Resume Next
NumEls% = UBound(Wait$)
If Err Then
    Exit Function
End If
On Error Resume Next
T$ = Wait$(0)
If Err Then
    FirstEl% = 1
Else
    FirstEl% = 0
End If
```

```
ReDim WaitChk$(FirstEl% To FirstEl% + NumEls%)
For i% = FirstEl% To NumEls%
    WaitChk$(i%) = StrConv(Wait$(i%), vbFromUnicode)
Next i%
```

' AutoProcess & RThresholdの設定を変更

```
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If
```

```
' InputLen を1に設定
OldLen% = CommCtl.InputLen
If OldLen% <> 1 Then
```

```
' InputLenの設定を戻す
If OldLen% <> 1 Then
    CommCtl.InputLen = OldLen%
End If
```

```
' AutoProcess and RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If
```

End Function

Function WaitForA%(CommCtl As Control, Wait\$(), TimeOut%, CDFlag%, ExcludeFlag%, Found%, Recei

' 本Functionは、WaitFor\$と同機能ですが受信文字列に配列を指定することができます。

```
Dim Test%, AutoP%, RT%, NumEls%, FirstEl%, OldLen%, There%, i%
Dim Start!, EndTime!
Dim InString$, NewData$, T$
Dim Flag24 As Boolean
```

```
' COMポートをチェック
WaitForA% = PortCheck%(CommCtl)
If WaitForA% <> PDQ_SUCCESS Then
    Exit Function
End If
```

' 文字列の数を取得する

```
On Error Resume Next
NumEls% = UBound(Wait$)
If Err Then
    Exit Function
End If
On Error Resume Next
T$ = Wait$(0)
If Err Then
    FirstEl% = 1
Else
    FirstEl% = 0
End If
```

```
ReDim WaitChk$(FirstEl% To FirstEl% + NumEls%)
For i% = FirstEl% To NumEls%
    WaitChk$(i%) = StrConv(Wait$(i%), vbFromUnicode)
Next i%
```

' AutoProcess & RThresholdの設定を変更

```
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If
```

```
' InputLen を1に設定
OldLen% = CommCtl.InputLen
If OldLen% <> 1 Then
```

```

CommCtl.InputLen = 1
End If

' 変数の初期化
WaitForA% = PDQ_SUCCESS
Found% = -1

Start! = Timer

If Timer + TimeOut% > 86400 Then
    EndTime! = TimeOut% - (86400 - Timer)
    Flag24 = True
Else
    EndTime! = Start! + TimeOut%
    Flag24 = False
End If

Do
    ' データが受信されたかどうかの確認
    If CommCtl.InBufferCount Then
        ' 受信データを読み取る
        DoEvents
        NewData$ = CommCtl.InputB
        ' InString$文字列に追加する
        InString$ = InString$ & NewData$

        ' 受信待ちの文字列を取得したかを確認する
        For i% = FirstEl% To NumEl%
            There% = InStrB(InString$, WaitChk$(i%))
            If There% Then Exit For
        Next
        If There% Then
            If ExcludeFlag% Then
                Received$ = LeftB(InString$, There% - 1)
            Else
                Received$ = InString$
            End If
            Found% = i%
            WaitForA% = PDQ_SUCCESS
            Exit Do
        End If

        ' InString$ が大きすぎるかを確認
        If Len(InString$) > 32000 Then
            InString$ = Right$(InString$, 32000)
        End If
    End If

    ' 処理が中断されたかどうかを確認する
    If CancelFlag% Then
        WaitForA% = PDQ_ER_CANCELED
        Received$ = InString$
        Exit Do
    End If

    ' キャリアのチェック
    If CDFlag% Then
        If CommCtl.CD Holding = False Then
            WaitForA% = PDQ_ER_LOST_CARRIER
            Received$ = InString$
            Exit Do
        End If
    End If
End Do

```

```

CommCtl.InputLen = 1
End If

' 変数の初期化
WaitForA% = PDQ_SUCCESS
Found% = -1

Start! = Timer

If Timer + TimeOut% > 86400 Then
    EndTime! = TimeOut% - (86400 - Timer)
    Flag24 = True
Else
    EndTime! = Start! + TimeOut%
    Flag24 = False
End If

Do
    ' データが受信されたかどうかの確認
    If CommCtl.InBufferCount Then
        ' 受信データを読み取る
        DoEvents
        NewData$ = CommCtl.InputB
        ' InString$文字列に追加する
        InString$ = InString$ & NewData$

        ' 受信待ちの文字列を取得したかを確認する
        For i% = FirstEl% To NumEl%
            There% = InStrB(InString$, WaitChk$(i%))
            If There% Then Exit For
        Next
        If There% Then
            If ExcludeFlag% Then
                Received$ = LeftB(InString$, There% - 1)
            Else
                Received$ = InString$
            End If
            Found% = i%
            WaitForA% = PDQ_SUCCESS
            Exit Do
        End If

        ' InString$ が大きすぎるかを確認
        If Len(InString$) > 32000 Then
            InString$ = Right$(InString$, 32000)
        End If
    End If

    ' 処理が中断されたかどうかを確認する
    If CancelFlag% Then
        WaitForA% = PDQ_ER_CANCELED
        Received$ = InString$
        Exit Do
    End If

    ' キャリアのチェック
    If CDFlag% Then
        If CommCtl.CD Holding = False Then
            WaitForA% = PDQ_ER_LOST_CARRIER
            Received$ = InString$
            Exit Do
        End If
    End If
End Do

```

```

    End If
End If

If Flag24 = False Then
    If Timer >= EndTime! Then
        Received$ = InString$
        WaitForA% = PDQ_ER_TIMEOUT
        Exit Do
    End If
Else
    If Timer < Start! Then
        Flag24 = False
    End If
End If

DoEvents

```

Loop

```

' InputLenの設定をもどす
If OldLen% <> 1 Then
    CommCtl.InputLen = OldLen%
End If

' 受信文字列をUnicodeに変換
Received$ = StrConv(Received$, vbUnicode)

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function WaitForCall%(CommCtl As Control, NumRings%, TimeOut%, ModemBoud&, RealBaud&, Mode\$, RFunction WaitForCall%(CommCtl As Control, NumRings%, TimeOut%, ModemBoud&, RealBaud&, Mode\$, R

```

' 本Functionは、呼び出し信号を受信待ちしてからモデムを接続します
' Dim Ret%, Test%, AutoP%, RT%, TimeOut%, NumTries%
Dim Ret%, Test%, AutoP%, RT%, NumTries%, CmdTimeOut%
Dim CONNECT_Pos%, CARRIER_Pos%, L%, LenBaud%, AscVal%, Baud_Pos%

Dim CONNECT$, InitString$, Result$, RecvData$, BaudStr$, SettingBoud$

' COMポートをチェック
WaitForCall% = PortCheck%(CommCtl)
If WaitForCall% <> PDQ_SUCCESS Then
    Exit Function
End If

' 変数の初期化
RealBaud& = 0
ModemBoud& = 0
Mode$ = ""
WaitForCall% = PDQ_SUCCESS

If NumRings% = 0 Then
    NumRings% = 1
End If

```

```

    End If
End If

If Flag24 = False Then
    If Timer >= EndTime! Then
        Received$ = InString$
        WaitForA% = PDQ_ER_TIMEOUT
        Exit Do
    End If
Else
    If Timer < Start! Then
        Flag24 = False
    End If
End If

DoEvents

```

Loop

```

' InputLenの設定をもどす
If OldLen% <> 1 Then
    CommCtl.InputLen = OldLen%
End If

' 受信文字列をUnicodeに変換
Received$ = StrConv(Received$, vbUnicode)

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function WaitForCall%(CommCtl As Control, NumRings%, TimeOut%, ModemBoud&, RealBaud&, Mode\$, RFunction WaitForCall%(CommCtl As Control, NumRings%, TimeOut%, ModemBoud&, RealBaud&, Mode\$, R

```

' 本Functionは、呼び出し信号を受信待ちしてからモデムを接続します
' Dim Ret%, Test%, AutoP%, RT%, TimeOut%, NumTries%
Dim Ret%, Test%, AutoP%, RT%, NumTries%, CmdTimeOut%
Dim CONNECT_Pos%, CARRIER_Pos%, L%, LenBaud%, AscVal%, Baud_Pos%

Dim CONNECT$, InitString$, Result$, RecvData$, BaudStr$, SettingBoud$

' COMポートをチェック
WaitForCall% = PortCheck%(CommCtl)
If WaitForCall% <> PDQ_SUCCESS Then
    Exit Function
End If

' 変数の初期化
RealBaud& = 0
ModemBoud& = 0
Mode$ = ""
WaitForCall% = PDQ_SUCCESS

If NumRings% = 0 Then
    NumRings% = 1
End If

```

```

CONNECT$ = "CONNECT"

' AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

' モデムを初期化します
CmdTimeOut% = 5
NumTries% = 3
InitString$ = "ATSO=" & LTrim$(Str$(NumRings%)) & Chr$(13)

Ret% = SendModemCmd$(CommCtl, InitString$, NumTries%, CmdTimeOut%, Result$)

Select Case Ret%
' 中断されたかタイムアウトエラーが発生した
Case PDQ_ER_CANCELED
    WaitForCall% = PDQ_ER_CANCELED
    GoTo WFCExit
Case PDQ_ER_TIMEOUT, PDQ_ER_MODEM_ERROR
    WaitForCall% = PDQ_ER_MODEM_ERROR
    GoTo WFCExit
End Select

' 接続するまで待つ
Do

    Ret% = WaitFor$(CommCtl, Chr(13), TimeOut%, False, True, RecvData$)

    Select Case Ret%
    Case PDQ_ER_CANCELED
        ' ATSO=0を送出する
        Ret% = SendModemCmd$(CommCtl, "ATSO=0", 2, 10, Result$)
        WaitForCall% = PDQ_ER_CANCELED
        Exit Do
    Case PDQ_ER_TIMEOUT
        WaitForCall% = PDQ_ER_TIMEOUT
        Exit Do
    Case PDQ_SUCCESS
        ' モデムから受信データがあるかどうかの確認
        If InStr(RecvData$, "OK") Then
            ' 続く
        ElseIf InStr(RecvData$, "CONNECT") Then
            ' CONNECT 受信
            CONNECT_Pos% = InStr(RecvData$, "CONNECT")
            If Len(RecvData$) < CONNECT_Pos% + 8 Then
                RealBaud& = 300
            Else
                BaudStr$ = Mid$(RecvData$, CONNECT_Pos% + 8)

                L% = Len(BaudStr$)
                For LenBaud% = 1 To L%
                    AscVal% = Asc(Mid$(BaudStr$, LenBaud%, 1))
                    If AscVal% < 48 Or AscVal% > 57 Then
                        Exit For
                    End If
                Next
            End If
        End Select
    End Do
End Do

```

```

CONNECT$ = "CONNECT"

' AutoProcess & RThresholdの設定を変更
AutoP% = CommCtl.AutoProcess
If AutoP% Then
    CommCtl.AutoProcess = PDQ_AUTOPROCESS_NONE
End If
RT% = CommCtl.RThreshold
If RT% Then
    CommCtl.RThreshold = 0
End If

' モデムを初期化します
CmdTimeOut% = 5
NumTries% = 3
InitString$ = "ATSO=" & LTrim$(Str$(NumRings%)) & Chr$(13)

Ret% = SendModemCmd$(CommCtl, InitString$, NumTries%, CmdTimeOut%, Result$)

Select Case Ret%
' 中断されたかタイムアウトエラーが発生した
Case PDQ_ER_CANCELED
    WaitForCall% = PDQ_ER_CANCELED
    GoTo WFCExit
Case PDQ_ER_TIMEOUT, PDQ_ER_MODEM_ERROR
    WaitForCall% = PDQ_ER_MODEM_ERROR
    GoTo WFCExit
End Select

' 接続するまで待つ
Do

    Ret% = WaitFor$(CommCtl, Chr(13), TimeOut%, False, True, RecvData$)

    Select Case Ret%
    Case PDQ_ER_CANCELED
        ' ATSO=0を送出する
        Ret% = SendModemCmd$(CommCtl, "ATSO=0", 2, 10, Result$)
        WaitForCall% = PDQ_ER_CANCELED
        Exit Do
    Case PDQ_ER_TIMEOUT
        WaitForCall% = PDQ_ER_TIMEOUT
        Exit Do
    Case PDQ_SUCCESS
        ' モデムから受信データがあるかどうかの確認
        If InStr(RecvData$, "OK") Then
            ' 続く
        ElseIf InStr(RecvData$, "CONNECT") Then
            ' CONNECT 受信
            CONNECT_Pos% = InStr(RecvData$, "CONNECT")
            If Len(RecvData$) < CONNECT_Pos% + 8 Then
                RealBaud& = 300
            Else
                BaudStr$ = Mid$(RecvData$, CONNECT_Pos% + 8)

                L% = Len(BaudStr$)
                For LenBaud% = 1 To L%
                    AscVal% = Asc(Mid$(BaudStr$, LenBaud%, 1))
                    If AscVal% < 48 Or AscVal% > 57 Then
                        Exit For
                    End If
                Next
            End If
        End Select
    End Do
End Do

```

```

' モデムの通信速度を取得
If LenBaud% > L% Then
    RealBaud& = Val (BaudStr$)
Else
    RealBaud& = Val (Left$ (BaudStr$, LenBaud%))
    ' Mode$の残りを取得
    Mode$ = Mid$ (BaudStr$, LenBaud% + 1)
End If
End If
' モデム~PC間のポートの設定
If CONNECT_Pos% Then
    SettingBoud$ = CommCtl.Settings
    Baud_Pos% = InStr (SettingBoud$, ",")
    ModemBoud& = Val (Left (SettingBoud$, Baud_Pos%))
End If

' 2400ボーが設定されている場合は少々時間を置き、モデムとのハンドシェイク
' を完了させる。
If RealBaud& <= 2400 Then
    Pause 8
End If

```

```

WaitForCall% = PDQ_SUCCESS
Exit Do
ElseIf InStr (RecvData$, "RING") Then
    ' RING
    ' 電話がかかってきました。... 応答中です。
ElseIf InStr (RecvData$, "ERROR") Then
    WaitForCall% = PDQ_ER_MODEM_ERROR
Exit Do
ElseIf InStr (RecvData$, "BUSY") Then
    ' 通話中
    WaitForCall% = PDQ_ER_MODEM_ERROR
Exit Do
End If

```

```

End Select
Loop

```

WFCExit:

```

' 受信文字列を返す
Received$ = RecvData$

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function ZTrim\$ (St\$)

'ヌルコード、タブコード、リターンコード、改行コードを文字列から削除します。

```

Dim Z$, T$, CR$, LF$, S$, Tilde$
Dim L%, i%

```

```

Z$ = Chr$ (0)
T$ = Chr$ (9)
CR$ = Chr$ (13)

```

```

' モデムの通信速度を取得
If LenBaud% > L% Then
    RealBaud& = Val (BaudStr$)
Else
    RealBaud& = Val (Left$ (BaudStr$, LenBaud%))
    ' Mode$の残りを取得
    Mode$ = Mid$ (BaudStr$, LenBaud% + 1)
End If
End If
' モデム~PC間のポートの設定
If CONNECT_Pos% Then
    SettingBoud$ = CommCtl.Settings
    Baud_Pos% = InStr (SettingBoud$, ",")
    ModemBoud& = Val (Left (SettingBoud$, Baud_Pos%))
End If

' 2400ボーが設定されている場合は少々時間を置き、モデムとのハンドシェイク
' を完了させる。
If RealBaud& <= 2400 Then
    Pause 8
End If

```

```

WaitForCall% = PDQ_SUCCESS
Exit Do
ElseIf InStr (RecvData$, "RING") Then
    ' RING
    ' 電話がかかってきました。... 応答中です。
ElseIf InStr (RecvData$, "ERROR") Then
    WaitForCall% = PDQ_ER_MODEM_ERROR
Exit Do
ElseIf InStr (RecvData$, "BUSY") Then
    ' 通話中
    WaitForCall% = PDQ_ER_MODEM_ERROR
Exit Do
End If

```

```

End Select
Loop

```

WFCExit:

```

' 受信文字列を返す
Received$ = RecvData$

' AutoProcess & RThresholdの設定を戻す
If AutoP% Then
    CommCtl.AutoProcess = AutoP%
End If
If RT% Then
    CommCtl.RThreshold = RT%
End If

```

End Function

Function ZTrim\$ (St\$)

'ヌルコード、タブコード、リターンコード、改行コードを文字列から削除します。

```

Dim Z$, T$, CR$, LF$, S$, Tilde$
Dim L%, i%

```

```

Z$ = Chr$ (0)
T$ = Chr$ (9)
CR$ = Chr$ (13)

```

```

LF$ = Chr$(10)
S$ = Chr$(32)
Tilde$ = "~"

L% = Len(St$)
For i% = 1 To L%
  Select Case Right$(St$, 1)
  Case Z$, T$, CR$, LF$, S$
    If L% > 1 Then
      St$ = Left$(St$, L% - 1)
      L% = L% - 1
    Else
      St$ = ""
    End If
  Case Else
    Exit For
  End Select
Next

L% = Len(St$)
For i% = 1 To L%
  If Mid$(St$, i%, 1) = Z$ Then
    Mid$(St$, i%, 1) = S$
  End If
Next

L% = Len(St$)
For i% = 1 To L%
  If Mid$(St$, i%, 1) = Tilde$ Then
    Mid$(St$, i%, 1) = S$
  End If
Next

ZTrim$ = Trim$(St$)

End Function

```

```

LF$ = Chr$(10)
S$ = Chr$(32)
Tilde$ = "~"

L% = Len(St$)
For i% = 1 To L%
  Select Case Right$(St$, 1)
  Case Z$, T$, CR$, LF$, S$
    If L% > 1 Then
      St$ = Left$(St$, L% - 1)
      L% = L% - 1
    Else
      St$ = ""
    End If
  Case Else
    Exit For
  End Select
Next

L% = Len(St$)
For i% = 1 To L%
  If Mid$(St$, i%, 1) = Z$ Then
    Mid$(St$, i%, 1) = S$
  End If
Next

L% = Len(St$)
For i% = 1 To L%
  If Mid$(St$, i%, 1) = Tilde$ Then
    Mid$(St$, i%, 1) = S$
  End If
Next

ZTrim$ = Trim$(St$)

End Function

```

```

VERSION 5.00
Begin VB.Form PDQPort
    Caption = "ポートの設定"
    ClientHeight = 3270
    ClientLeft = 3285
    ClientTop = 1950
    ClientWidth = 3915
    Icon = "PDQPort. frx":0000
    LinkTopic = "Form1"
    PaletteMode = 1 'Zオーダー'
    ScaleHeight = 3270
    ScaleWidth = 3915
    Begin VB.CommandButton CmdSet
        Caption = "設定"
        Height = 390
        Left = 1575
        TabIndex = 14
        Top = 2775
        Width = 1065
    End
    Begin VB.CommandButton CmdCancel
        Caption = "キャンセル"
        Height = 390
        Left = 2775
        TabIndex = 13
        Top = 2775
        Width = 1065
    End
    Begin VB.Frame Frame1
        Height = 2565
        Left = 75
        TabIndex = 0
        Top = 75
        Width = 3765
        Begin VB.ComboBox CmbHandshaking
            Height = 300
            ItemData = "PDQPort. frx":0442
            Left = 1335
            List = "PDQPort. frx":0452
            Style = 2 'ドロップダウン リスト'
            TabIndex = 11
            Top = 2100
            Width = 2280
        End
        Begin VB.ComboBox CmbBaudRate
            Height = 300
            ItemData = "PDQPort. frx":0489
            Left = 1335
            List = "PDQPort. frx":04AB
            Style = 2 'ドロップダウン リスト'
            TabIndex = 5
            Top = 600
            Width = 2280
        End
        Begin VB.ComboBox CmbStopBits
            Height = 300
            ItemData = "PDQPort. frx":04EE
            Left = 1335
            List = "PDQPort. frx":04FB
            Style = 2 'ドロップダウン リスト'
            TabIndex = 4
            Top = 1725
        End
    End
End

```

```

VERSION 5.00
Begin VB.Form PDQPort
    Caption = "ポートの設定"
    ClientHeight = 3270
    ClientLeft = 3285
    ClientTop = 1950
    ClientWidth = 3915
    Icon = "PDQPort. frx":0000
    LinkTopic = "Form1"
    PaletteMode = 1 'Zオーダー'
    ScaleHeight = 3270
    ScaleWidth = 3915
    Begin VB.CommandButton CmdSet
        Caption = "設定"
        Height = 390
        Left = 1575
        TabIndex = 14
        Top = 2775
        Width = 1065
    End
    Begin VB.CommandButton CmdCancel
        Caption = "キャンセル"
        Height = 390
        Left = 2775
        TabIndex = 13
        Top = 2775
        Width = 1065
    End
    Begin VB.Frame Frame1
        Height = 2565
        Left = 75
        TabIndex = 0
        Top = 75
        Width = 3765
        Begin VB.ComboBox CmbHandshaking
            Height = 300
            ItemData = "PDQPort. frx":0442
            Left = 1335
            List = "PDQPort. frx":0452
            Style = 2 'ドロップダウン リスト'
            TabIndex = 11
            Top = 2100
            Width = 2280
        End
        Begin VB.ComboBox CmbBaudRate
            Height = 300
            ItemData = "PDQPort. frx":0489
            Left = 1335
            List = "PDQPort. frx":04AB
            Style = 2 'ドロップダウン リスト'
            TabIndex = 5
            Top = 600
            Width = 2280
        End
        Begin VB.ComboBox CmbStopBits
            Height = 300
            ItemData = "PDQPort. frx":04EE
            Left = 1335
            List = "PDQPort. frx":04FB
            Style = 2 'ドロップダウン リスト'
            TabIndex = 4
            Top = 1725
        End
    End
End

```

```

Width = 2280
End
Begin VB.ComboBox CmbParity
Height = 300
ItemData = "PDQPort.frx":050A
Left = 1335
List = "PDQPort.frx":051D
Style = 2 'ドロップダウンリスト
TabIndex = 3
Top = 975
Width = 2280
End
Begin VB.ComboBox CmbDataBits
Height = 300
ItemData = "PDQPort.frx":0571
Left = 1335
List = "PDQPort.frx":0584
Style = 2 'ドロップダウンリスト
TabIndex = 2
Top = 1350
Width = 2280
End
Begin VB.ComboBox CmbComPort
Height = 300
ItemData = "PDQPort.frx":0597
Left = 1335
List = "PDQPort.frx":05AD
Style = 2 'ドロップダウンリスト
TabIndex = 1
Top = 225
Width = 2280
End
Begin VB.Label Label1
Caption = "フロー制御："
Height = 210
Left = 225
TabIndex = 12
Top = 2205
Width = 840
End
Begin VB.Label LblStopBits
Caption = "ストップビット："
Height = 210
Left = 225
TabIndex = 10
Top = 1830
Width = 840
End
Begin VB.Label LblParity
Caption = "パリティ："
Height = 285
Left = 225
TabIndex = 9
Top = 1080
Width = 840
End
Begin VB.Label LblDataBits
Caption = "データビット："
Height = 210
Left = 225
TabIndex = 8
Top = 1455

```

```

Width = 2280
End
Begin VB.ComboBox CmbParity
Height = 300
ItemData = "PDQPort.frx":050A
Left = 1335
List = "PDQPort.frx":051D
Style = 2 'ドロップダウンリスト
TabIndex = 3
Top = 975
Width = 2280
End
Begin VB.ComboBox CmbDataBits
Height = 300
ItemData = "PDQPort.frx":0571
Left = 1335
List = "PDQPort.frx":0584
Style = 2 'ドロップダウンリスト
TabIndex = 2
Top = 1350
Width = 2280
End
Begin VB.ComboBox CmbComPort
Height = 300
ItemData = "PDQPort.frx":0597
Left = 1335
List = "PDQPort.frx":05AD
Style = 2 'ドロップダウンリスト
TabIndex = 1
Top = 225
Width = 2280
End
Begin VB.Label Label1
Caption = "フロー制御："
Height = 210
Left = 225
TabIndex = 12
Top = 2205
Width = 840
End
Begin VB.Label LblStopBits
Caption = "ストップビット："
Height = 210
Left = 225
TabIndex = 10
Top = 1830
Width = 840
End
Begin VB.Label LblParity
Caption = "パリティ："
Height = 285
Left = 225
TabIndex = 9
Top = 1080
Width = 840
End
Begin VB.Label LblDataBits
Caption = "データビット："
Height = 210
Left = 225
TabIndex = 8
Top = 1455

```

```

        Width          = 915
    End
    Begin VB.Label LblBaudRate
        Caption          = "ボーレート:"
        Height           = 210
        Left             = 225
        TabIndex         = 7
        Top              = 705
        Width            = 915
    End
    Begin VB.Label LblComPort
        Caption          = "ポート:"
        Height           = 210
        Left             = 225
        TabIndex         = 6
        Top              = 330
        Width            = 840
    End
End
Attribute VB_Name = "PDQPort"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
' PDQCommコントロール
Dim CommCtl As Object

' 変更前の設定値
Dim SavePort As Integer
Dim SaveSettings As String
Dim SaveHandshaking As Integer

Public Sub SetPort(PDQCtl As Control)

    ' PDQCommコントロールの取得
    Set CommCtl = PDQCtl

    ' 現在の設定値を保存
    SavePort = CommCtl.CommPort
    SaveSettings = CommCtl.Settings
    SaveHandshaking = CommCtl.Handshaking

    ' フォーム表示位置の調整
    PDQPort.Left = (Screen.Width - PDQPort.ScaleWidth) ¥ 2
    PDQPort.Top = (Screen.Height - PDQPort.ScaleHeight) ¥ 2

    ' フォームの表示
    PDQPort.Show

End Sub

Private Sub CmdCancel_Click()

    ' 通信ポートがクローズ状態の場合、変更前の設定に戻す
    If CommCtl.PortOpen = False Then
        ' 変更前の設定に戻す
        CommCtl.CommPort = SavePort
        CommCtl.Settings = SaveSettings
        CommCtl.Handshaking = SaveHandshaking
        ' 通信ポートを再オープン
        On Error Resume Next
    End If
End Sub

```

```

        Width          = 915
    End
    Begin VB.Label LblBaudRate
        Caption          = "ボーレート:"
        Height           = 210
        Left             = 225
        TabIndex         = 7
        Top              = 705
        Width            = 915
    End
    Begin VB.Label LblComPort
        Caption          = "ポート:"
        Height           = 210
        Left             = 225
        TabIndex         = 6
        Top              = 330
        Width            = 840
    End
End
Attribute VB_Name = "PDQPort"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
' PDQCommコントロール
Dim CommCtl As Object

' 変更前の設定値
Dim SavePort As Integer
Dim SaveSettings As String
Dim SaveHandshaking As Integer

Public Sub SetPort(PDQCtl As Control)

    ' PDQCommコントロールの取得
    Set CommCtl = PDQCtl

    ' 現在の設定値を保存
    SavePort = CommCtl.CommPort
    SaveSettings = CommCtl.Settings
    SaveHandshaking = CommCtl.Handshaking

    ' フォーム表示位置の調整
    PDQPort.Left = (Screen.Width - PDQPort.ScaleWidth) ¥ 2
    PDQPort.Top = (Screen.Height - PDQPort.ScaleHeight) ¥ 2

    ' フォームの表示
    PDQPort.Show

End Sub

Private Sub CmdCancel_Click()

    ' 通信ポートがクローズ状態の場合、変更前の設定に戻す
    If CommCtl.PortOpen = False Then
        ' 変更前の設定に戻す
        CommCtl.CommPort = SavePort
        CommCtl.Settings = SaveSettings
        CommCtl.Handshaking = SaveHandshaking
        ' 通信ポートを再オープン
        On Error Resume Next
    End If
End Sub

```

```

CommCtl.PortOpen = True
If Err Then
    MsgBox "ポートの設定が無効です再設定してください。"
End If
End If

```

```

' フォームを閉じる
Unload Me

```

```
End Sub
```

```
Private Sub CmdSet_Click()
```

```

' 通信ポートをクローズする
If CommCtl.PortOpen Then
    CommCtl.PortOpen = False
End If

```

```

' 通信ポートの設定
CommCtl.CommPort = CmbComPort.ListIndex + 1
' 通信パラメータの設定
CommCtl.Settings = CmbBaudRate.List(CmbBaudRate.ListIndex) & "," & Left(CmbParity.List(Cmb
' ハンドシェイクの設定
CommCtl.Handshaking = CmbHandshaking.ListIndex

```

```

' 通信ポートをオープンする
On Error Resume Next
CommCtl.PortOpen = True
If Err Then
    MsgBox "ポート番号が無効です再度指定してください"
Else
    SaveSetting App.Title, "SettingCom", "Port", CommCtl.CommPort
    SaveSetting App.Title, "SettingCom", "Speed", CmbBaudRate.List(CmbBaudRate.ListIndex)
    ' フォームを閉じる
    Unload Me
End If

```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim Boud As String, Prity As String, DataBit As String, StopBit As String
```

```

' 通信ポートの設定
CmbComPort.ListIndex = CommCtl.CommPort - 1
' ハンドシェイクの設定
CmbHandshaking.ListIndex = CommCtl.Handshaking

```

```

' 通信ポートの設定を解析
AnalyzeSetting CommCtl.Settings, Boud, Prity, DataBit, StopBit

```

```

' ボーレートの設定
For i = 0 To CmbBaudRate.ListCount
    If CmbBaudRate.List(i) = Boud Then
        CmbBaudRate.ListIndex = i
        Exit For
    End If
Next i

```

```

' パリティの設定
Select Case UCase(Prity)
    Case "E"

```

```

CommCtl.PortOpen = True
If Err Then
    MsgBox "ポートの設定が無効です再設定してください。"
End If
End If

```

```

' フォームを閉じる
Unload Me

```

```
End Sub
```

```
Private Sub CmdSet_Click()
```

```

' 通信ポートをクローズする
If CommCtl.PortOpen Then
    CommCtl.PortOpen = False
End If

```

```

' 通信ポートの設定
CommCtl.CommPort = CmbComPort.ListIndex + 1
' 通信パラメータの設定
CommCtl.Settings = CmbBaudRate.List(CmbBaudRate.ListIndex) & "," & Left(CmbParity.List(Cmb
' ハンドシェイクの設定
CommCtl.Handshaking = CmbHandshaking.ListIndex

```

```

' 通信ポートをオープンする
On Error Resume Next
CommCtl.PortOpen = True
If Err Then
    MsgBox "ポート番号が無効です再度指定してください"
Else
    SaveSetting App.Title, "SettingCom", "Port", CommCtl.CommPort
    SaveSetting App.Title, "SettingCom", "Speed", CmbBaudRate.List(CmbBaudRate.ListIndex)
    ' フォームを閉じる
    Unload Me
End If

```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim Boud As String, Prity As String, DataBit As String, StopBit As String
```

```

' 通信ポートの設定
CmbComPort.ListIndex = CommCtl.CommPort - 1
' ハンドシェイクの設定
CmbHandshaking.ListIndex = CommCtl.Handshaking

```

```

' 通信ポートの設定を解析
AnalyzeSetting CommCtl.Settings, Boud, Prity, DataBit, StopBit

```

```

' ボーレートの設定
For i = 0 To CmbBaudRate.ListCount
    If CmbBaudRate.List(i) = Boud Then
        CmbBaudRate.ListIndex = i
        Exit For
    End If
Next i

```

```

' パリティの設定
Select Case UCase(Prity)
    Case "E"

```

```

    CmbParity.ListIndex = 0
Case "N"
    CmbParity.ListIndex = 1
Case "O"
    CmbParity.ListIndex = 2
Case "S"
    CmbParity.ListIndex = 3
Case "M"
    CmbParity.ListIndex = 4
End Select

```

```

' データビット設定
CmbDataBits.ListIndex = Val(DataBit) - 4

```

```

' ストップビット設定
Select Case StopBit
Case "1"
    CmbStopBits.ListIndex = 0
Case "1.5"
    CmbStopBits.ListIndex = 1
Case "2"
    CmbStopBits.ListIndex = 2
End Select

```

End Sub

```

Private Sub AnalyzeSetting(PortSetting As String, Boud As String, Prity As String, DataBit As

```

```

    Dim CheckComma As Integer
    Dim SaveComma As Integer

```

```

' 通信速度取得
CheckComma = InStr(PortSetting, ",")
Boud = Left(PortSetting, CheckComma - 1)

```

```

' パリティ取得
SaveComma = CheckComma + 1
CheckComma = InStr(SaveComma, PortSetting, ",")
Prity = Mid(PortSetting, SaveComma, CheckComma - SaveComma)

```

```

' データビット
SaveComma = CheckComma + 1
CheckComma = InStr(SaveComma, PortSetting, ",")
DataBit = Mid(PortSetting, SaveComma, CheckComma - SaveComma)

```

```

' ストップビット
SaveComma = CheckComma + 1
StopBit = Mid(PortSetting, SaveComma, Len(PortSetting) - SaveComma + 1)

```

End Sub

```

    CmbParity.ListIndex = 0
Case "N"
    CmbParity.ListIndex = 1
Case "O"
    CmbParity.ListIndex = 2
Case "S"
    CmbParity.ListIndex = 3
Case "M"
    CmbParity.ListIndex = 4
End Select

```

```

' データビット設定
CmbDataBits.ListIndex = Val(DataBit) - 4

```

```

' ストップビット設定
Select Case StopBit
Case "1"
    CmbStopBits.ListIndex = 0
Case "1.5"
    CmbStopBits.ListIndex = 1
Case "2"
    CmbStopBits.ListIndex = 2
End Select

```

End Sub

```

Private Sub AnalyzeSetting(PortSetting As String, Boud As String, Prity As String, DataBit As

```

```

    Dim CheckComma As Integer
    Dim SaveComma As Integer

```

```

' 通信速度取得
CheckComma = InStr(PortSetting, ",")
Boud = Left(PortSetting, CheckComma - 1)

```

```

' パリティ取得
SaveComma = CheckComma + 1
CheckComma = InStr(SaveComma, PortSetting, ",")
Prity = Mid(PortSetting, SaveComma, CheckComma - SaveComma)

```

```

' データビット
SaveComma = CheckComma + 1
CheckComma = InStr(SaveComma, PortSetting, ",")
DataBit = Mid(PortSetting, SaveComma, CheckComma - SaveComma)

```

```

' ストップビット
SaveComma = CheckComma + 1
StopBit = Mid(PortSetting, SaveComma, Len(PortSetting) - SaveComma + 1)

```

End Sub

```
Attribute VB_Name = "ProcessSub01"
Option Explicit

Public Sub Grig_AppendStatus(vStr As String)
    Dim mRow As Long, mRowNo As Long
    With fMainForm.gridStatus
        .Rows = .Rows + 1
        mRowNo = .Rows - 1
        .TextMatrix(mRowNo, 0) = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
        .TextMatrix(mRowNo, 0) = Format(Date + Time, "hh:nn:ss")
        .TextMatrix(mRowNo, 1) = vStr
        If .Rows > 200 Then
            .RemoveItem 1
        End If
        If .Rows > 3 Then .TopRow = .Rows - 3
        .Refresh
    End With
End Sub
```

ProcessSub01.bas x 2

```
Attribute VB_Name = "ProcessSub01"
Option Explicit

Public Sub Grig_AppendStatus(vStr As String)
    Dim mRow As Long, mRowNo As Long
    With fMainForm.gridStatus
        .Rows = .Rows + 1
        mRowNo = .Rows - 1
        .TextMatrix(mRowNo, 0) = Format(Date + Time, "yyyy/mm/dd hh:nn:ss")
        .TextMatrix(mRowNo, 0) = Format(Date + Time, "hh:nn:ss")
        .TextMatrix(mRowNo, 1) = vStr
        If .Rows > 200 Then
            .RemoveItem 1
        End If
        If .Rows > 3 Then .TopRow = .Rows - 3
        .Refresh
    End With
End Sub
```

```
Attribute VB_Name = "TypeDef"  
Option Explicit
```

```
Public Type DataFileHead
```

```
観測開始_年 As Integer  
観測開始_月 As Byte  
観測開始_日 As Byte  
観測開始_時 As Byte  
観測開始_分 As Byte  
観測開始_秒 As Byte
```

```
End Type
```

```
Public Enum e動作Mode
```

```
停止  
監視中  
収録中
```

```
End Enum
```

```
Attribute VB_Name = "TypeDef"  
Option Explicit
```

```
Public Type DataFileHead
```

```
観測開始_年 As Integer  
観測開始_月 As Byte  
観測開始_日 As Byte  
観測開始_時 As Byte  
観測開始_分 As Byte  
観測開始_秒 As Byte  
CH数_Int As Integer  
サンプリングタイム_秒 As Single  
収録データ数_LONG As Long  
プリトリガデータ数_LONG As Long  
CH1レンジ_gal As Single  
CH2レンジ_gal As Single  
CH3レンジ_gal As Single  
CH4レンジ_gal As Single
```

```
End Type
```

```
Public Enum e動作Mode
```

```
停止  
監視中  
収録中
```

```
End Enum
```