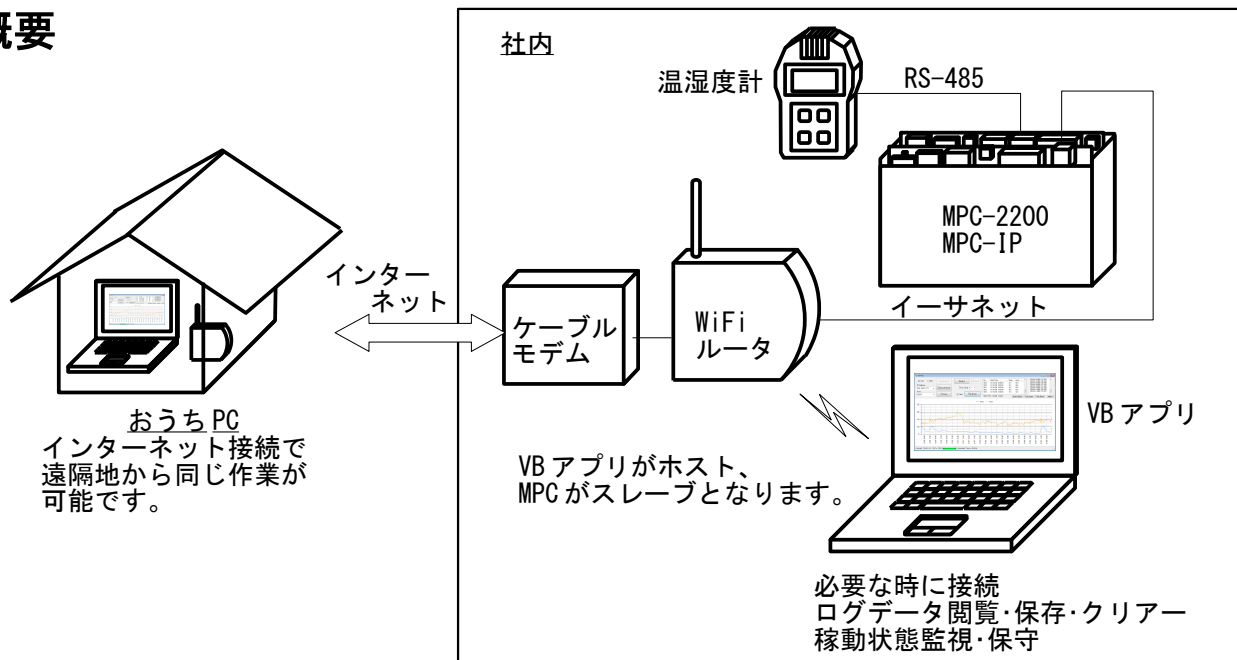


Application Note		Ref No: an2k-041	Last Modify 130122
テーマ	MPC-2200 を使った温湿度ロガーに TPC アクセスする		
使用機器	MPC-2200、MPC-IP、RS-485 温湿度計 他		

概要



- 5 分間隔で温湿度を計り MPC-2200 の点データ P(n) に蓄積していきます。各点の X~Z 成分に次のようにデータを格納します。

P(n) = X(n) : 年月日, Y(n) : 時分秒, U(n) : 温度, Z(n) : 湿度

記録例。2012 年 12 月 27 日 16:25:00 ~17:00:00 の温湿度。温度、湿度は 10 倍した数値です。

```

1 121227 162500 224 231
2 121227 163000 220 253
3 121227 163500 217 267
4 121227 164000 218 275
5 121227 164500 219 288
6 121227 165000 220 290
7 121227 165500 220 294
8 121227 170000 218 302

```

- MPC-2200 は下記のバージョンから点データが 32000 個になりました。5 分間隔でログすると約 111 日分が記録できます。

```

MPC-2200H BL/I 1.14_03 2012/12/12
All Rights reserved. ACCEL Corp.
PRG_1M PNT_32K DIM_40K .T32

```

- MPC-2200 を用いた理由
RS-485 通信が可能でリアルタイムクロック (RTC) がある、ということで MPC-2200 を用いました。タッチパネル等を使って時計を取得できれば MPC-1000 でも Ok、温湿度計との通信が RS-232 なら MPC-2000 でも Ok です。この事例では、MPC は RS-485 通信しか行っていませんが、もちろんメカ制御も Ok です。
- VB.net で作成した TCP アプリケーションでデータを読み込みグラフ化します。MPC はスタンドアロンで動作し TCP アクセスの影響を受けないので、いつでもアプリケーションの接続・切断が可能です。IP アドレス、ポート番号の変更で、同じアプリケーションを使って社内 LAN、インターネットやケーブル直結のアクセスができます。

VB アプリケーション

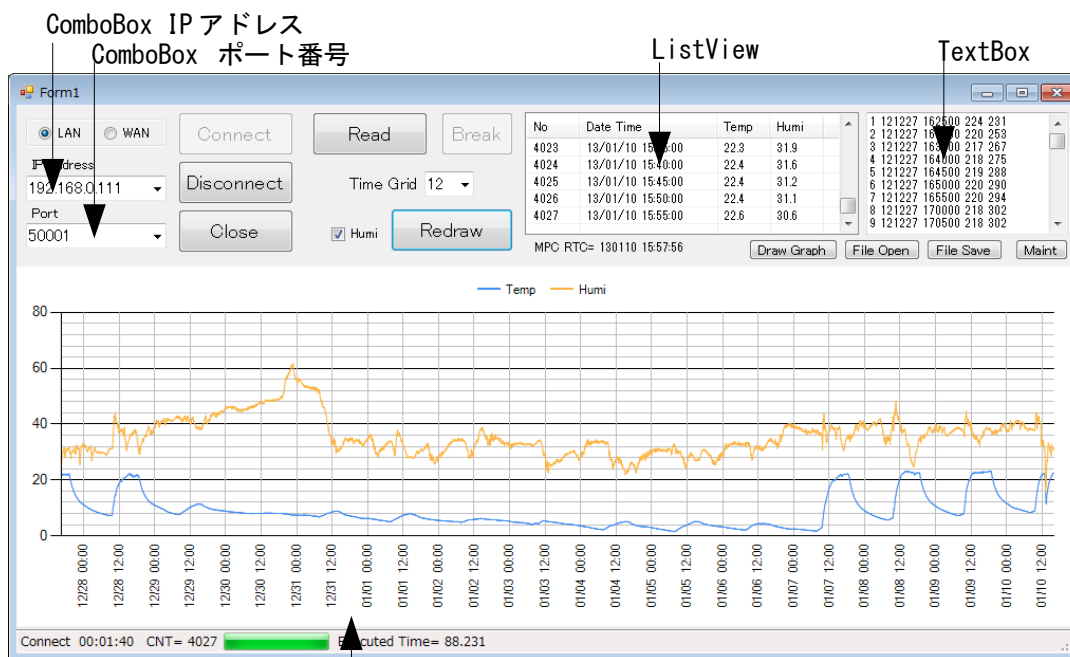
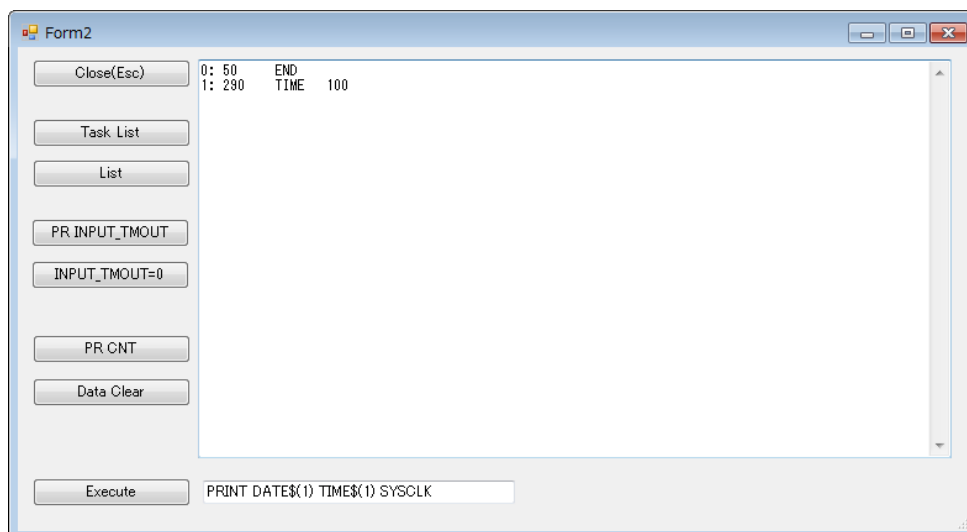


Chart X軸が月日、Y軸が温度（青：℃）と湿度（橙：%）。

VB2010 Expressで作成しました。上のグラフは2012年12月27日～2013年1月10日、つまり年末年始休業とその前後の社内の温湿度の変化です。（休み中と夜間は冷え切っています。12月30日に雨が降って湿度が高くなりましたが、それ以後は乾燥続きです。）

■ アプリケーションの機能

- IPアドレスとポート番号を設定してLAN、インターネットに接続します。このアプリではRadioButtonでLAN/WANの設定変更を行えるようにしました。
- MPCの点データをTextBoxに読み込み、文字列を加工してListViewに入れます。そしてListViewのItem.TextからChartを描画しています。TextBoxやListViewを介さずにChartにデータを入れることもできますが、今回はデータの可視化とListViewの試用も含めて作成しました。
- X軸のグリッド間隔を変更できます（1時間～24時間間隔）。
- TextBoxのファイル保存・読込、グラフ再描画ができます。
- メンテナンス機能として、MPCプログラムの実行中のリスト表示（下図）、全プログラムリスト表示、RS-485通信タイムアウトカウンタの表示/リセット、データ数の表示、データクリア、その他のコマンド実行ができます。



MPC プログラム

ネットワーク⇔MPC-IP⇔MPC-2200間の通信はバックグラウンドで行われるため、VBアプリとの通信に関するコードはMEWNET宣言しかありません。
VBアプリは点データ、変数、I/Oなどを直接読み書きできます。

```
CNFG# 2 RS485 "9600b7pes1NONE" /* 温湿度計 CH2 RS-485 モード
MEWNET 38400 1 /* MPC-IP と接続
S_MBK LONG_PRG

QUIT_FORK 1 *LOG

END

*LOG

DO
OFF -1
FILL P(1) 32000 0 /* NEWP コマンドはプログラム記述不可
CNT=0
PRINT "Data Clear"
DO
IF (TIME(0)&&HFFF==0) | (TIME(0)&&HFFF==&H500) THEN
ON 768
GOSUB *GET_DATA
CNT=CNT+1
PR CNT TIME(VOID) t h
' PRX TIME(0)
X(CNT)=DATE(VOID)
Y(CNT)=TIME(VOID)
U(CNT)=t
Z(CNT)=h
WAIT (TIME(0)&&HFFF<>0) & (TIME(0)&&HFFF<>&H500)
END_IF

IF SW(-1)==1 THEN /* VB から ON -1 されるとデータクリア
BREAK
END_IF
ON 768 /* 動作中を表す LED 点滅
TIME 100
OFF 768
TIME 100
LOOP

LOOP

*GET_DATA
PRINT# 2 CHR$(5) "01" CHR$(2) "RPV01" CHR$(3) "¥r¥n" /* 温湿度計にコマンド送信
INPUT# 2 TMOUT|3 a$ /* 温湿度計データ受信
IF rse ==1 THEN
INPUT_TMOUT=INPUT_TMOUT+1
PR "INPUT TMOUT" INPUT_TMOUT
INPUT# 2 CLR_BUF
GOTO *GET_DATA
END_IF

dummy=VAL(a$)
FOR I=1 to 9
dummy=VAL(0)
NEXT I
t=VAL(10) /* 温度値取得
dummy=VAL(0) /* 読み飛ばし
h=VAL(10) /* 湿度値取得

RETURN
```

VB プログラム

■ Form1.vb メイン画面

```
Imports System.Windows.Forms.DataVisualization.Charting
Imports System
Imports System.Net
Imports System.Net.Sockets
Imports System.Text
Imports System.Threading.Thread

Public Class Form1
    Private stopWatch As Stopwatch
    Private isRunning As Boolean = False
    Private Break_Reading As Boolean = False

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ComboBox1.Text = "192.168.0.111"
        ComboBox1.Items.Add("192.168.0.111")
        ComboBox1.Items.Add("xxx.xxx.xxx.xxx")

        ComboBox2.Text = "50001"
        ComboBox2.Items.Add("50001")
        ComboBox2.Items.Add("50002")

        ComboBox3.Text = "12"
        ComboBox3.Items.Add("1")
        ComboBox3.Items.Add("2")
        ComboBox3.Items.Add("3")
        ComboBox3.Items.Add("4")
        ComboBox3.Items.Add("6")
        ComboBox3.Items.Add("12")
        ComboBox3.Items.Add("24")

        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Button5.Enabled = False
        Button10.Enabled = False

        ToolStripStatusLabel1.Text = "00:00:00"
        ToolStripStatusLabel2.Text = ""
        ToolStripStatusLabel3.Text = ""
        ToolStripStatusLabel4.Text = ""
        ToolStripStatusLabel5.Text = ""

        Label2.Text = ""

        stopWatch = New Stopwatch()
    End Sub

    ''' <summary>
    ''' [Connect] Button: TCP接続
    ''' </summary>
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Button1.Click
        Dim ver As String = ""
        Dim res As Integer = 0
        Dim Stream_Name As String
        Dim PortNum As Integer

        Stream_Name = ComboBox1.Text
        PortNum = Val(ComboBox2.Text)

        TcpClientA = New TcpClient()

        Try
            TcpClientA.Connect(Stream_Name, PortNum)
            ToolStripStatusLabel2.Text = "Connect"
            MyStreamA = TcpClientA.GetStream()
            If MPCIP.set_stream_timeout(MyStreamA, 20000, 20000, res) < 1 Then Throw New
            Exception("set_stream_timeout")
            If MPCIP.set_stream_ID(MyStreamA, Stream_Name, res) < 1 Then Throw New
            Exception("set_stream_ID")
        End Try
    End Sub
End Class
```

```

        If MPCIP.set_mewnet_tmout(MyStreamA, 2000, res) < 1 Then Throw New
Exception("set_mewnet_tmout")

        Timer1.Interval = 1000
        Timer1.Enabled = True
        isRunning = True

        stopWatch.Reset()
        stopWatch.Start()

        Button1.Enabled = False
        Button2.Enabled = True
        Button3.Enabled = True
        Button4.Enabled = True
        Button5.Enabled = True

        Button3.Focus()

    Catch ex As Exception
        MsgBox("Error " & vbCrLf & ex.Message & " (" + res.ToString & ")")
    End Try

End Sub

''' <summary>
''' [Disconnect] Button: TCP切断
''' </summary>
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    isRunning = False

    Button1.Enabled = True
    Button2.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = False
    Button5.Enabled = False
    Button10.Enabled = False

    Button1.Focus()

    TcpClientA.Close()
    ToolStripStatusLabel2.Text = "Disconnect"

End Sub

''' <summary>
''' [Read] Button: 点データ読み込み→グラフ描画
''' </summary>
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    Dim res As Integer
    Dim cmd As String = ""
    Dim ret As String = ""
    Dim str_ary As String()

    Dim i As Integer
    Dim cnt As Integer

    Dim last_time As Long = DateTime.Now.Ticks

    Button3.Enabled = False
    Button5.Enabled = False
    Button10.Enabled = True
    Button10.Focus()

    Break_Reading = False
    ChartClear(Chart1)
    Try

        ToolStripStatusLabel4.Text = ""
        ' MPCの現在時刻の読み込み
        cmd = "PR DATE (VOID) TIMES(1)"
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        Label2.Text = "MPC RTC=" & ret

        ' データ数の読み込み
        cmd = "PR CNT"
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)

```

```

ToolStripStatusLabel5.Text = " CNT= " & ret
cnt = CInt(ret)

If cnt = 0 Then
    MsgBox("Data is empty")
    Exit Sub
End If

ToolStripProgressBar1.Maximum = cnt

TextBox1.Clear()
ListView1.Items.Clear()

Application.DoEvents()

' 点データを読み込んでTextBox1に入れる
For i = 1 To cnt
    cmd = "PR P(" + i.ToString + ")"
    If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
    TextBox1.SelectedText = i.ToString & " " & ret & vbCrLf

    str_ary = ret.Split(" ") ' データのチェック
    If str_ary.Count <> 4 Then Throw New Exception("Illegal Receive Count")

    ToolStripProgressBar1.Value = i
    Application.DoEvents()

    If Break_Reading = True Then
        Exit For
    End If
Next

set_ListView()
Draw_Graph()

Catch ex As Exception
    MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
End Try

Dim current_time As Long = DateTime.Now.Ticks
ToolStripStatusLabel4.Text = " Executed Time= " + ((current_time - last_time) /
1000000).ToString("F3")

Button3.Enabled = True
Button5.Enabled = True
Button10.Enabled = False
Button3.Focus()

End Sub

''' <summary>
''' アクセス経過時間表示
''' MPC-IPがアクセスタイムアウトにならないように定期的にアクセスする
''' </summary>
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Dim res As Integer
    Static cnt As Integer = 0

    Timer1.Enabled = False

    ToolStripStatusLabel1.Text = String.Format("{0:00} : {1:00} : {2:00}", _
        stopWatch.Elapsed.Hours, stopWatch.Elapsed.Minutes, stopWatch.Elapsed.Seconds)

    cnt = cnt + 1
    If cnt Mod 120 = 0 Then
        Try
            If MPCIP.reset_awdt(MyStreamA, res) < 1 Then Throw New Exception("reset_awdt ")
        Catch ex As Exception
            MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
            Exit Sub
        End Try
    End If

    If isRunning = False Then
        Exit Sub
    End If

```

```

    Timer1.Enabled = True
End Sub

''' <summary>
''' Chart を初期状態にする
''' </summary>
Private Sub ChartClear (ByVal cht As Chart)

    Dim customLabel1 As New CustomLabel

    With cht
        .Titles.Clear () ' タイトルの初期化
        .BackGradientStyle = GradientStyle.None ' 背景グラデーション
        .BorderSkin.PageColor = SystemColors.ButtonFace

        ' 外形をデフォルトに
        .BorderSkin.SkinStyle = BorderSkinStyle.None
        .Legends.Clear () ' 凡例の初期化
        .Legends.Add ("Legend1")
        .Legends ("Legend1").Alignment = StringAlignment.Center
        .Legends ("Legend1").Docking = Docking.Top

        .Series.Clear () ' 系列(データ関係)の初期化
        .ChartAreas.Clear () ' 軸メモリ・3D表示関係の初期化
        .ChartAreas.Add ("ChartArea1")
        .ChartAreas ("ChartArea1").Position = New ElementPosition (0, 10, 98, 90)

        .Annotations.Clear () ' グラフの注釈
        .DataSource = Nothing

        .Series.Add ("Temp")
        .Series.Add ("Humi")
        .Series ("Temp").ChartType = SeriesChartType.Line
        .Series ("Humi").ChartType = SeriesChartType.Line

        .ChartAreas (0).AxisX.CustomLabels.Add (customLabel1)
        .ChartAreas (0).AxisX.LabelAutoFitMaxFontSize = 9
        .ChartAreas (0).AxisX.LabelAutoFitMinFontSize = 9
        .ChartAreas (0).AxisX.MajorGrid.Enabled = True
        .ChartAreas (0).AxisX.MajorGrid.LineColor = Color.Silver
        .ChartAreas (0).AxisX.MajorTickMark.Enabled = False

        .ChartAreas (0).AxisY.MinorGrid.Enabled = True
        .ChartAreas (0).AxisY.MinorGrid.LineColor = Color.Silver

    End With

End Sub

Private Sub Button4_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click

    Form2.ShowDialog ()

End Sub

''' <summary>
''' [Redraw] Button: グラフ再描画
''' </summary>
Private Sub Button5_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click

    ChartClear (Chart1)
    Draw_Graph ()

End Sub

Private Sub Button6_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click

    Close ()

End Sub

''' <summary>
''' LAN/WAN選択 RadioButton
''' </summary>
Private Sub RadioButton1_CheckedChanged (ByVal sender As System.Object, ByVal e As System.EventArgs)

```

Handles RadioButton1.CheckedChanged, RadioButton2.CheckedChanged

```
If RadioButton1.Checked Then
    ComboBox1.Text = "192.168.0.111"
    ComboBox2.Text = "50001"
End If

If RadioButton2.Checked Then
    ComboBox1.Text = "xxx.xxx.xxx.xxx"
    ComboBox2.Text = "50002"
End If
```

End Sub

```
''' <summary>
''' [Draw Graph] Button: TextBox1.Textから再描画
''' </summary>
```

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

```
Dim ret As String = ""
```

```
If TextBox1.Lines.Count = 0 Then
    MsgBox("Data is empty")
Exit Sub
End If
```

```
ChartClear(Chart1)
ListView1.Items.Clear()
```

```
If TextBox1.SelectionLength > 0 Then
    ret = TextBox1.SelectedText
    TextBox1.Text = ret
End If
```

```
set_ListView()
Draw_Graph()
```

```
Button5.Enabled = True
```

End Sub

```
''' <summary>
''' TextBox1.textからListView1のリストを作成
''' </summary>
```

Private Sub set_ListView()
Dim ret As String = ""
Dim str_ary As String()

```
Dim i As Integer
Dim date_str As String
Dim time_str As String
Dim temp As Double
Dim humi As Double
```

```
ret = TextBox1.Text.Trim(Chr(10), Chr(13), " ")
TextBox1.Text = ret
```

```
ToolStripProgressBar1.Maximum = TextBox1.Lines.Count - 1
ToolStripStatusLabel5.Text = " CNT= " & (TextBox1.Lines.Count).ToString
Application.DoEvents()
```

```
For i = 0 To TextBox1.Lines.Count - 1
    ret = TextBox1.Lines(i)
    str_ary = ret.Split(" ")
    If str_ary.Count <> 5 Then
        MsgBox("Error" & vbCrLf & "Illegal Array Count")
Exit For
End If
```

```
date_str = CInt(str_ary(1)).ToString("00/00/00")
time_str = CInt(str_ary(2)).ToString("00:00:00")
```

```
temp = CInt(str_ary(3)) / 10
humi = CInt(str_ary(4)) / 10
```

```
ListView1.Items.Add(i)
ListView1.Items(i).SubItems(0).Text = (i + 1).ToString
ListView1.Items(i).SubItems.Add(date_str & " " & time_str)
```



```

        ListView1.Items(i).SubItems.Add(temp.ToString("F1"))
        ListView1.Items(i).SubItems.Add(humi.ToString("F1"))
        ToolStripProgressBar1.Value = i
        If (i Mod 100) = 0 Then
            ListView1.Items(ListView1.Items.Count - 1).EnsureVisible() 'ListView 最終行を表示
            Application.DoEvents()
        End If
    Next
    ListView1.Items(ListView1.Items.Count - 1).EnsureVisible() 'ListView 最終行を表示

End Sub

''' <summary>
''' ListView1のItemからグラフ描画
''' </summary>
Private Sub Draw_Graph()
    Dim ret As String = ""
    Dim str_ary As String()

    Dim i As Integer
    Dim date_str As String
    Dim time_str As String
    Dim temp As Double
    Dim humi As Double

    Dim Chart_Offset As Integer = -1
    Dim hh As Integer
    Dim x_label_span As Integer = CInt(ComboBox3.Text)

    Dim first_flg As Boolean = True

    For i = 0 To ListView1.Items.Count - 1

        str_ary = ListView1.Items(i).SubItems(1).Text.Split(" ")
        date_str = str_ary(0)
        time_str = str_ary(1)

        temp = ListView1.Items(i).SubItems(2).Text
        humi = ListView1.Items(i).SubItems(3).Text

        Chart1.Series("Temp").Points.AddY(temp)
        If CheckBox1.Checked Then
            Chart1.Series("Humi").Points.AddY(humi)
        End If

        hh = CInt(time_str.Substring(0, 2))

        If (time_str.Substring(3) = "00:00") Or first_flg Then
            If ((hh Mod x_label_span) = 0) Or first_flg Then

                Chart1.ChartAreas(0).AxisX.CustomLabels.Add((i + 1) * 2, 0, date_str.Substring(3) & "
                " & time_str.Substring(0, 5))
                If (Chart_Offset = -1) And (first_flg = False) Then
                    Chart_Offset = i + 1
                End If

                first_flg = False

            End If
        End If

    Next

    Chart1.ChartAreas(0).AxisX.MajorGrid.Interval = 12 * x_label_span ' Grid表示間隔
    Chart1.ChartAreas(0).AxisX.MajorGrid.IntervalOffset = Chart_Offset ' 最初のGridの位置

End Sub

''' <summary>
''' [File Save] Button: TextBox1.Textをファイル保存
''' </summary>
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
    SaveFileDialog1.Filter = "Text File|*.txt"
    If SaveFileDialog1.ShowDialog() = DialogResult.OK Then
        Dim sw As New System.IO.StreamWriter(SaveFileDialog1.FileName, _
        False, System.Text.Encoding.GetEncoding("shift_jis"))
        sw.Write(TextBox1.Text)
    End If
End Sub

```

```

        sw.Close()
    End If

End Sub

''' <summary>
''' [File Open] Button: ファイルからTextBox1に読み込み
''' </summary>
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
    OpenFileDialog1.FileName = ""
    OpenFileDialog1.Filter = "Text File|*.txt"
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        Dim sr As New IO.StreamReader(OpenFileDialog1.FileName)
        TextBox1.Text = sr.ReadToEnd()
        sr.Close()
    End If

End Sub

''' <summary>
''' [Break] Button: Read途中停止
''' </summary>
Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
    Break_Reading = True
    Button10.Enabled = False
End Sub

End Class

```

■ Form2.vb メンテナンス画面

```

Public Class Form2
    Dim close_flg As Boolean

    ''' <summary>
    ''' [Data Clear] Button: MPCのLOGデータクリア
    ''' </summary>
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Dim res As Integer
        Dim cmd As String = ""
        Dim ret As String = ""

        If MsgBox("Do you want to clear the data in the MPC?", vbYesNo) = vbNo Then
            Exit Sub
        End If

        Try
            cmd = "ON -1"
            If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        Catch ex As Exception
            MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
        End Try

    End Sub

    ''' <summary>
    ''' [PR INPUT_TMOUT] Button: MPCの変数 INPUT_TMOUT (RS-485タイムアウトカウント)取得
    ''' </summary>
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Dim res As Integer
        Dim cmd As String = ""
        Dim ret As String = ""

        Try
            cmd = "PR INPUT_TMOUT"
            If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
            TextBox1.Text = ret
        Catch ex As Exception
            MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
        End Try

    End Sub
End Class

```

```

''' <summary>
''' [INPUT_TMOU=0] Button: MPCの変数 INPUT_TMOU (RS-485タイムアウトカウンタ)リセット
''' </summary>
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    Dim res As Integer
    Dim cmd As String = ""
    Dim ret As String = ""

    If MsgBox("Do you want to clear the INPUT_TMOU?", vbYesNo) = vbNo Then
        Exit Sub
    End If

    Try
        cmd = "INPUT_TMOU=0"
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
    Catch ex As Exception
        MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
    End Try

End Sub

''' <summary>
''' [Task List] Button: 実行中のLIST取得
''' </summary>
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    Dim res As Integer
    Dim cmd As String = ""
    Dim ret As String = ""
    Dim str_ary0 As String()

    Button4.Enabled = False

    TextBox1.Clear()
    Try
        cmd = "PR MBK(7836~Lng) MBK(7838~Lng)"
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        str_ary0 = ret.Split(" ")
        If str_ary0.Count <> 2 Then Throw New Exception("Illegal array count(1)") ' 念のためのチェック

        cmd = "LIST " & str_ary0(0)
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        TextBox1.SelectedText = "0: " & ret & vbCrLf

        cmd = "LIST " & str_ary0(1)
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        TextBox1.SelectedText = "1: " & ret

    Catch ex As Exception
        MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
    End Try

    Button4.Enabled = True

End Sub

Private Sub Form2_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    close_flg = True

End Sub

Private Sub Form2_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles
Me.KeyDown
    If e.KeyCode = Keys.Escape Then
        Me.Close()
    End If

End Sub

Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Me.KeyPreview = True

End Sub

''' <summary>

```

```

''' [Close] Button
''' </summary>
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
    Me.Close()

End Sub

Private Sub Form2_Shown(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Shown
    Me.Top = Form1.Top

    TextBox1.Clear()
    TextBox2.Text = "PRINT DATE$(1) TIME$(1) SYSCLK"

    Button5.Focus()

End Sub

''' <summary>
''' [Execute] Button: TextBox2のコマンドを実行する
''' </summary>
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
    Dim res As Integer
    Dim cmd As String = ""
    Dim ret As String = ""

    TextBox1.Clear()

    Try
        cmd = TextBox2.Text
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        TextBox1.Text = ret
    Catch ex As Exception
        MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
    End Try

End Sub

''' <summary>
''' [List] Button: 全LIST取得
''' </summary>
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
    Dim res As Integer
    Dim cmd As String
    Dim ret As String = ""
    Dim ln1, ln2 As Integer
    Dim buf As String = ""

    Button7.Enabled = False

    close_flg = False
    TextBox1.Clear()
    Application.DoEvents()

    cmd = "LIST"
    Try
        If MPCIP.mpc_cmd(MyStreamA, "LIST 0", ret, res) < 0 Then Throw New Exception("LIST 0 ")
        ln1 = Val(ret)

        buf = ret + vbCrLf

        While True
            If MPCIP.mpc_cmd(MyStreamA, "LIST", ret, res) < 0 Then Throw New Exception("LIST ")
            ln2 = Val(ret)
            If ln1 > ln2 Then
                Exit While
            End If
            ln1 = ln2
            buf = buf + ret + vbCrLf

            TextBox1.Text = "Processing..." + ln2.ToString
            Application.DoEvents()

            If close_flg Then Exit While
        End While
    End Try

```

```

        End While

        Catch ex As Exception
            If res < 0 Then MsgBox("Error: " & ex.Message & " (res=" + res.ToString & ")") Else
MsgBox("Error: " & ex.Message)
            End Try

            TextBox1.Text = buf
            TextBox1.SelectionStart = 0

            Button7.Enabled = True

        End Sub

''' <summary>
''' [PR CNT] Button: MPCの変数 CNT (データ数)取得
''' </summary>
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
    Dim res As Integer
    Dim cmd As String = ""
    Dim ret As String = ""

    Try
        cmd = "PR CNT"
        If MPCIP.mpc_cmd(MyStreamA, cmd, ret, res) < 1 Then Throw New Exception(cmd)
        TextBox1.Text = ret
    Catch ex As Exception
        MsgBox("Error" & vbCrLf & ex.Message + " (" + res.ToString + ")")
    End Try

End Sub

End Class

```

■ Module1.vb

```

Imports System.Net
Imports System.Net.Sockets

Module Module1
    Public MPCIP As New ACClass.MPCIP
    Public TcpClientA As New TcpClient()
    Public MyStreamA As NetworkStream
    Public MyStreamA_Name As String
End Module

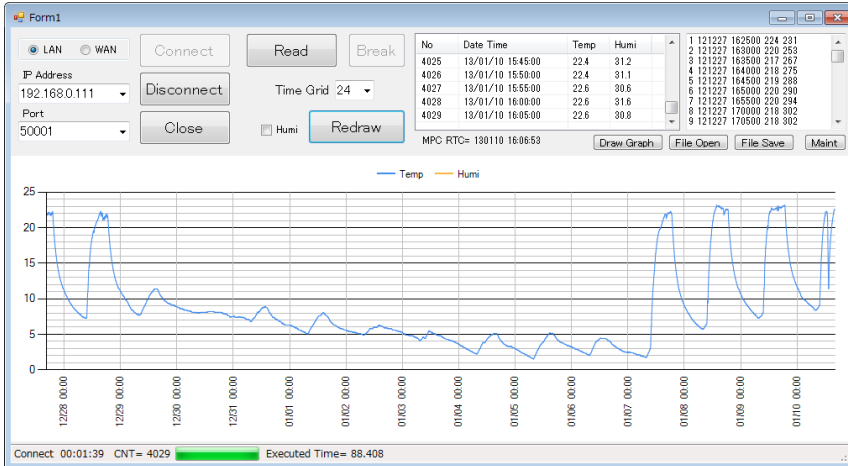
```

Appendix

VB アプリケーション スクリーンキャプチャ

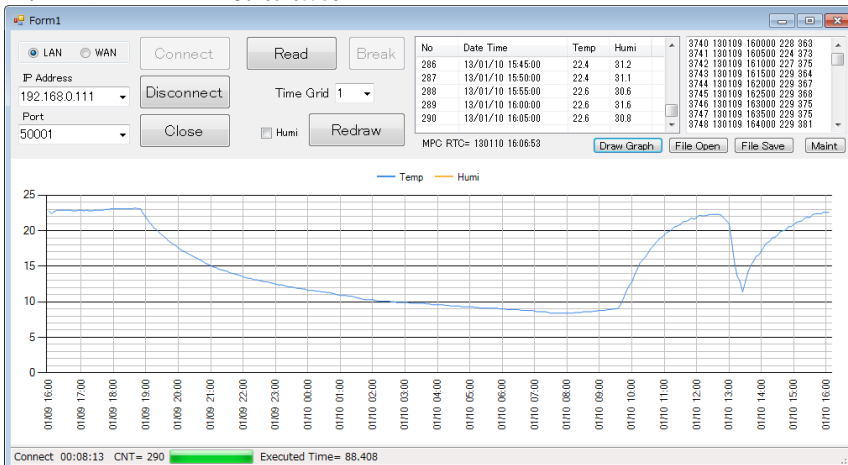
■ 温度だけ表示

X軸グリッドは24時間間隔。13/01/10の午後は掃除で窓を開けたので気温が下がっています。



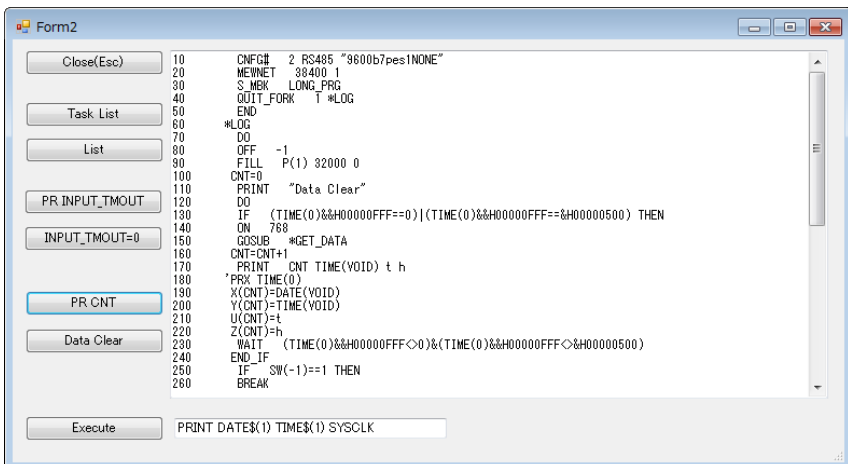
■ 温度部分表示(1日分)

X軸グリッドは1時間間隔。



■ メンテナンス画面

MPCプログラム全リスト表示。



--- End Of Doc ---