
如何利用 WinPAC-8xx7和 i-87018z 來量測溫度 並使用 C# .net 2008將所記錄的值畫成趨勢圖

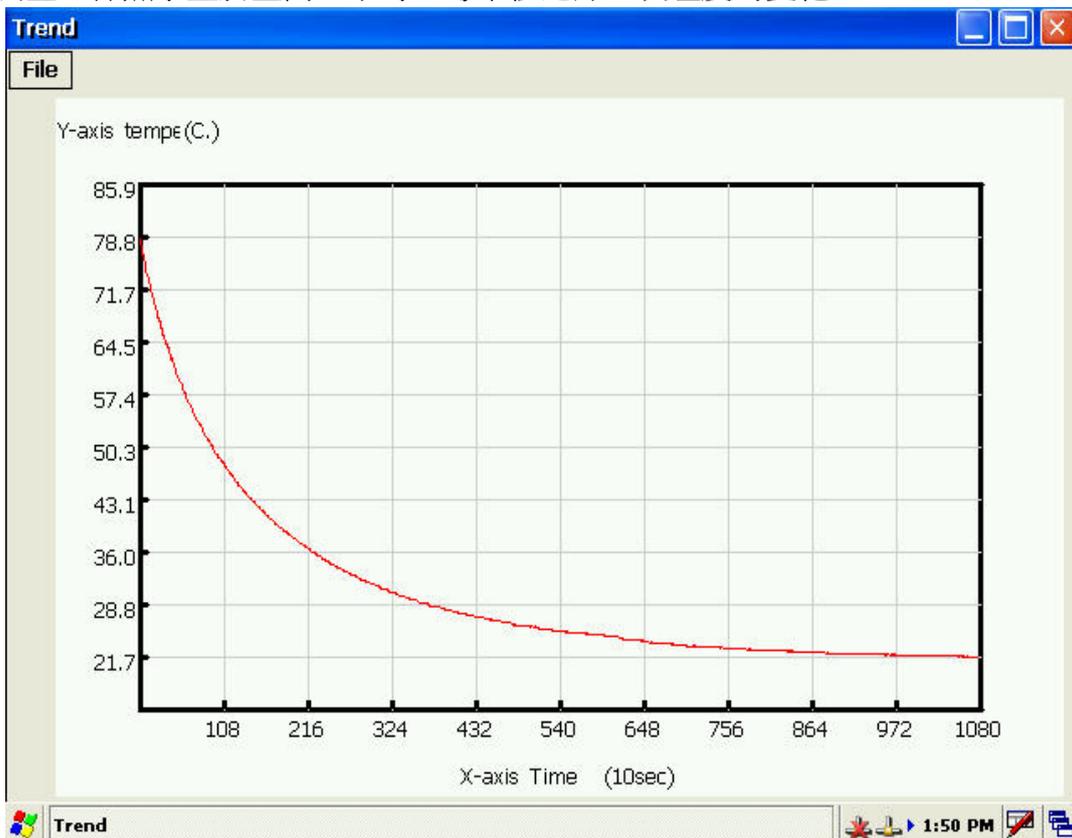
By joseph_dun@icpdas.com

本例將利用 WinPAC-8xx7和 I-87018z 和 ISaGRAF 做出一個量測溫度的應用

應用說明：

使用 ISaGRAF 程式再一段時間內，將 I-87018z 所測量到的溫度值記錄下來，並將記錄下來的資料寫成檔案放於 WinPAC-8xx7的記憶體內，之後利用一支 C# .net 的程式在 WinPAC-8xx7上將產生的檔案載入並將檔案中的資料畫成趨勢圖。這個應用可以使用於測量室內溫度，水溫...等等，然後運行該 C# .net 2008程式可以直接在 WinPAC-8xx7上就可以看出資料是否異常。

下圖是測量一杯熱水置於室內三小時，每十秒記錄一次溫度的變化



了解更多 WinPAC-8xx7與 I-87018z 的資訊可至下列網址中查詢

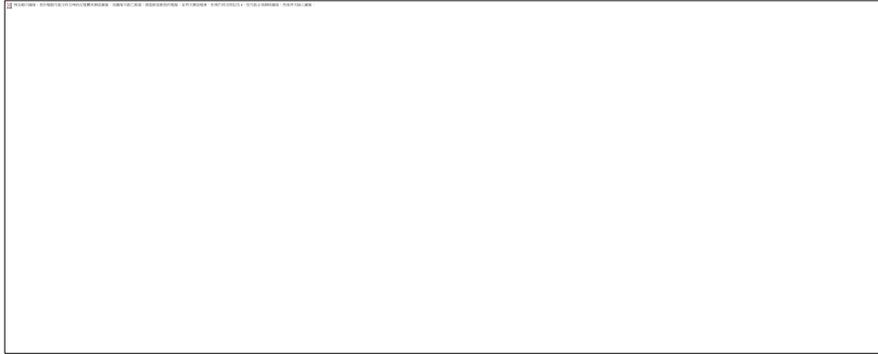
WinPAC-8xx7→ <http://www.icpdas.com/products/PAC/winpac/wp-8x47.html>

I-87018z→http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm

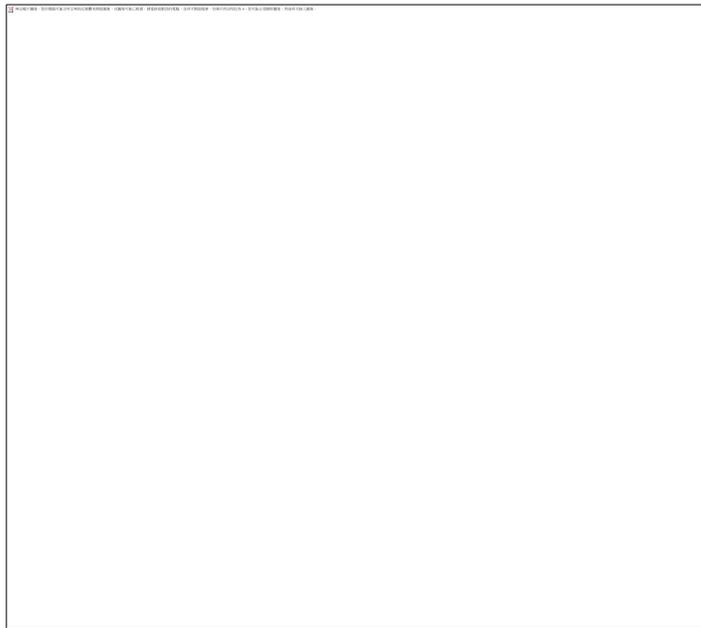
IsaGRAF 範例程式為 wpdmo108.pia 放於

www.icpdas.com→FAQ→Software→ISaGRAF Ver.3(Chinese)→108

請依照下列步驟可將範例程式(wpdmo108.pia)載入 ISaGRAF 中：

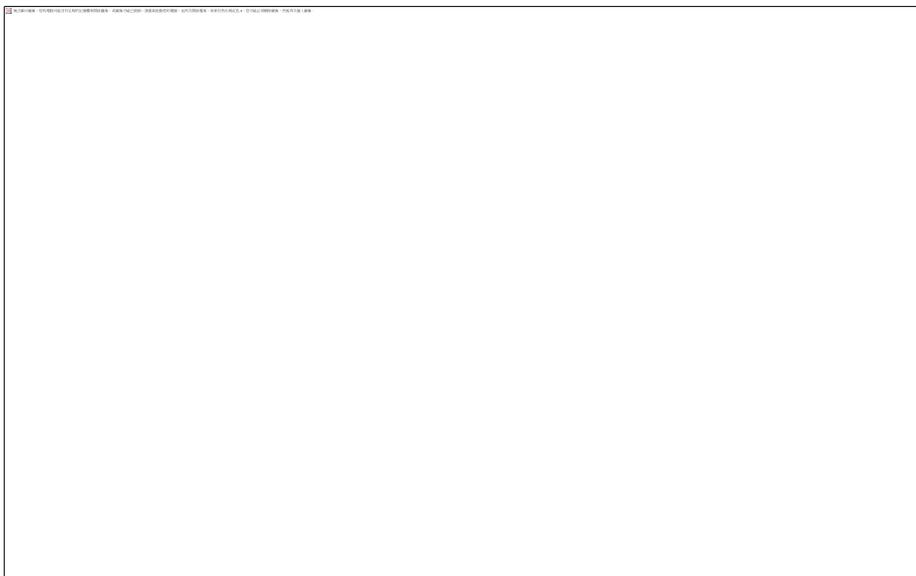


1.請點選 Tool→Archive→Projects

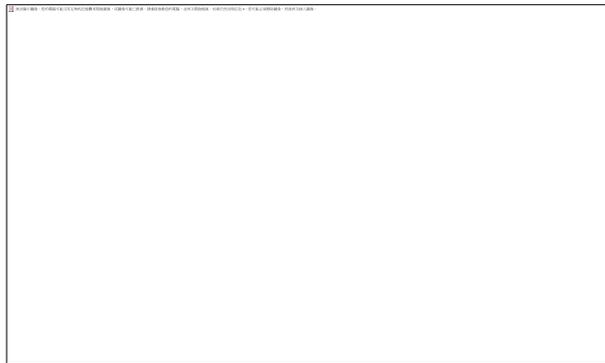
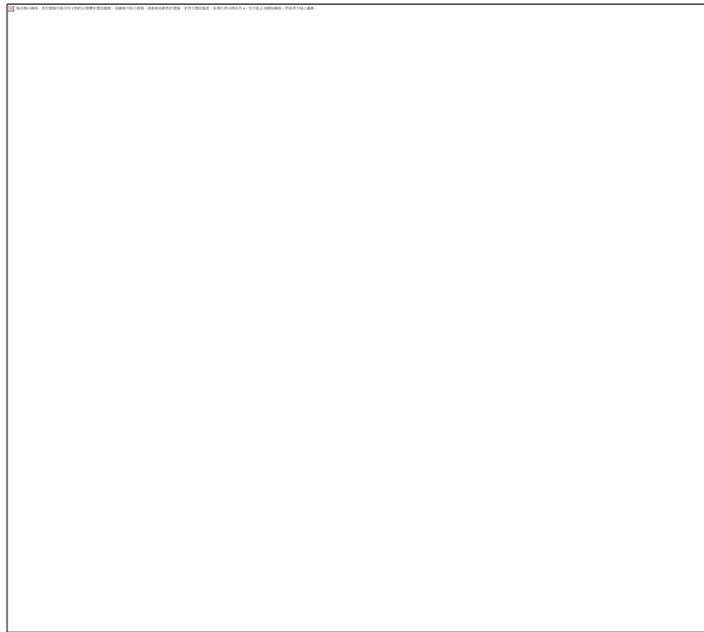


2.之後會開啟一個對話框按下 **Browse** 按鈕

3.選擇欲匯入專案檔所在資料夾，按下確定鍵



4.點選要載入的專案檔，按下 **restore** 鍵



5.之後就可以看到 ISaGRAF 的主畫面有剛剛匯入的專案

ISaGRAF 程式中有使用到變數陣列，詳細的設定可以參考 ISaGRAF 手冊的第2.6節或是可以在以下的位置找到相關說明

www.icpdas.com → FAQ → Software → IsaGARF Ver.3(Chinese) → 39

C# 範例程式為 Demo_2.exe 與原始碼放於

www.icpdas.com → FAQ → Software → ISaGRAF Ver.3(Chinese) → 108

可以直接執行範例程式 Demo_2.exe 於 WinPAC-8xx7上

可以利用 ftp 的方式將 Demo_2.exe 上傳到 WinPAC 的任何位置都可以，之後就可以在 WinPAC 上的 Demo_2.exe 點兩下左鍵即可執行該程式

若有需要直接修改原始碼的話

(請先確定您的 PC 上有 [Virtual Studio 2008 .net](#) 或其他相容於 C# 的開發平台，像 V.S 2005)

在原始碼的資料夾底下有一個檔案叫 Trend1(Demo_2).csproj 的檔案，在那個檔案上點兩下左鍵，系統便會幫開啟整個專案

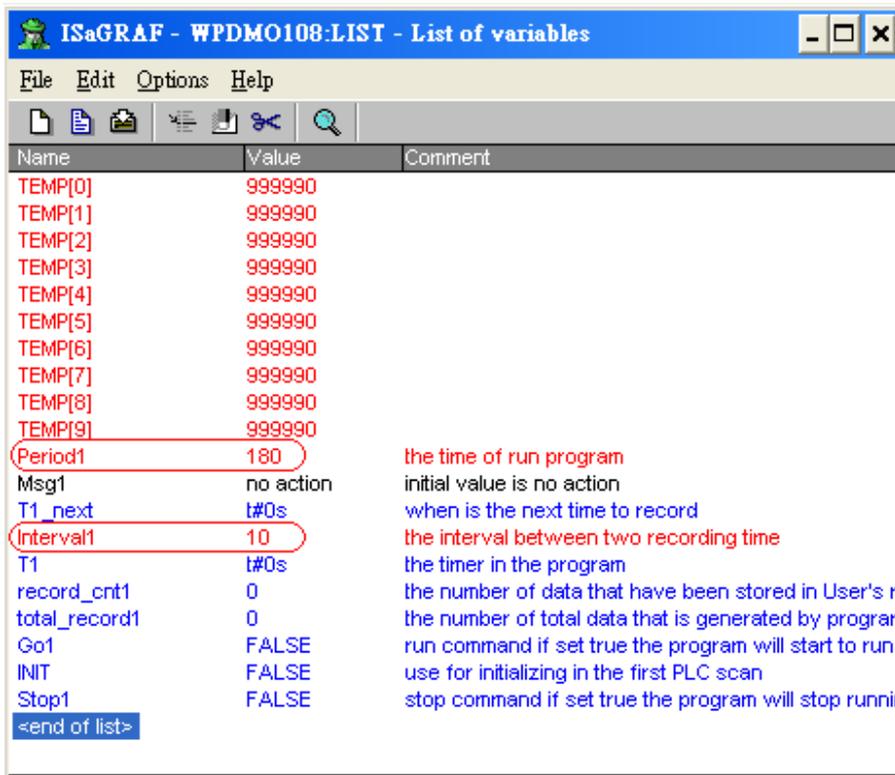
如何測試本範例?

1.請將 I-87018z 插在 WP-8xx7的 slot 2，並將 K-type 的訊號線接在 I-87018z 的第1個 channel 上，之後將 WinPAC 上電。準備一杯熱水，將 K-type 的訊號線放入水中。

2.將 ISaGRAF 中的範例程式 wpdmo108.pia 下載到 WP-8xx7中

3.在 list of variable 的視窗中修改一些初始值

請在”Interval1”中輸入一個適當的值。單位是1秒，如果給20指的是每20秒記錄一次。而 Period1指的是要記錄多久。單位為分鐘。輸入完後將 GO1設成 True 程式就會開始動作。程式會將 I-87018z 的第一個 channel 測到的溫度值按照剛剛設定的數值記錄下來



左圖為是設定 Interval1 為 10(10秒) Period1 為180 (180分鐘)

在記錄過程中，會看到 Record_cnt1 的值會一直增加，當等於 Total_record1 的值時，就表示記錄完畢，此時程式會開始將記錄下來的值存入檔案中。

範例程式會將產生出來的檔案存在/System_Disk/的資料夾中

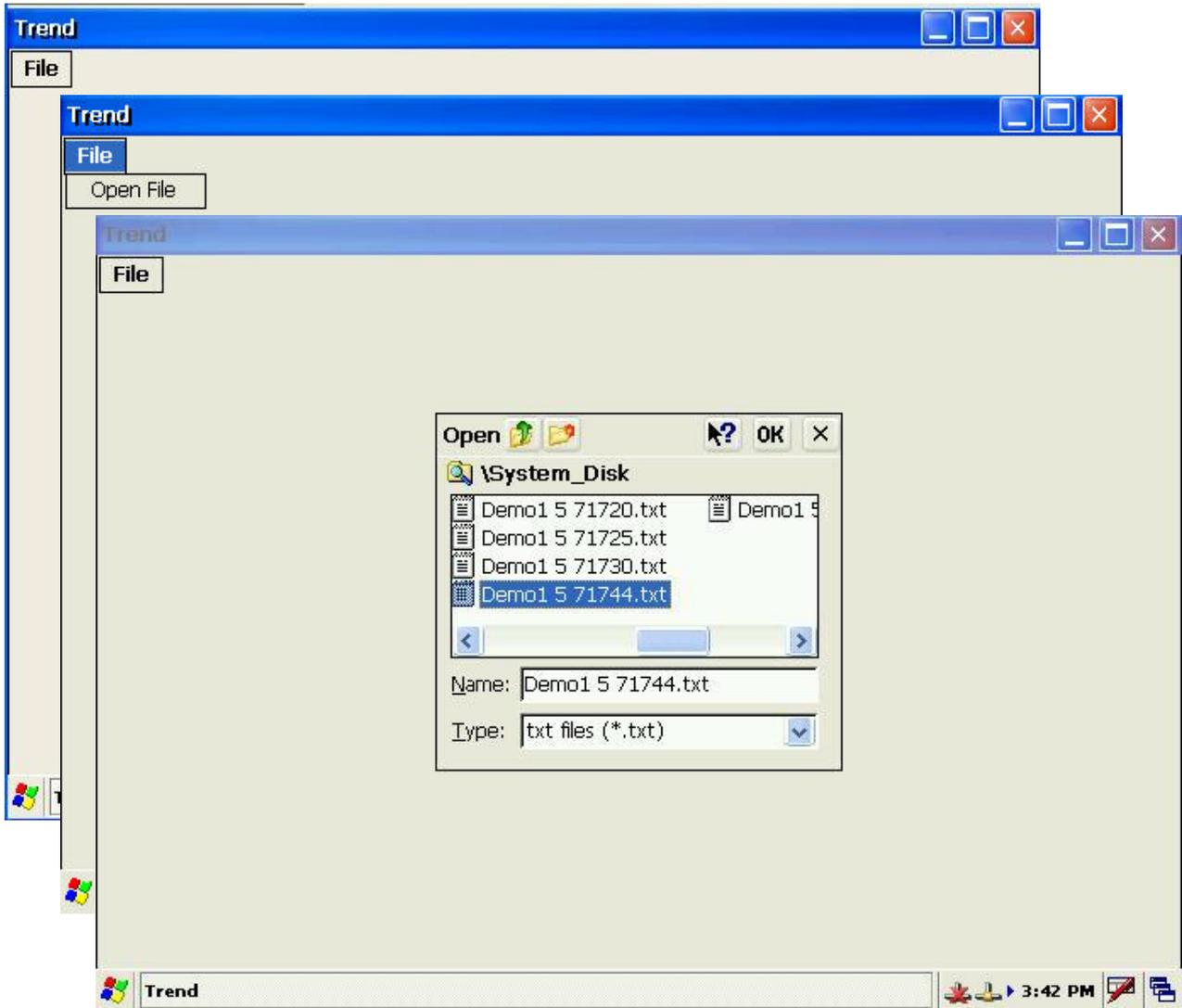
4.利用寫好的 C#程式在 WP-8xx7的/System_Disk/路徑中開啟剛剛記錄下的檔案，temperature.txt。就可以直接看到溫度的變化曲線圖(詳細操作如下一頁之說明)。

這裡說明一下 txt 檔的格式：



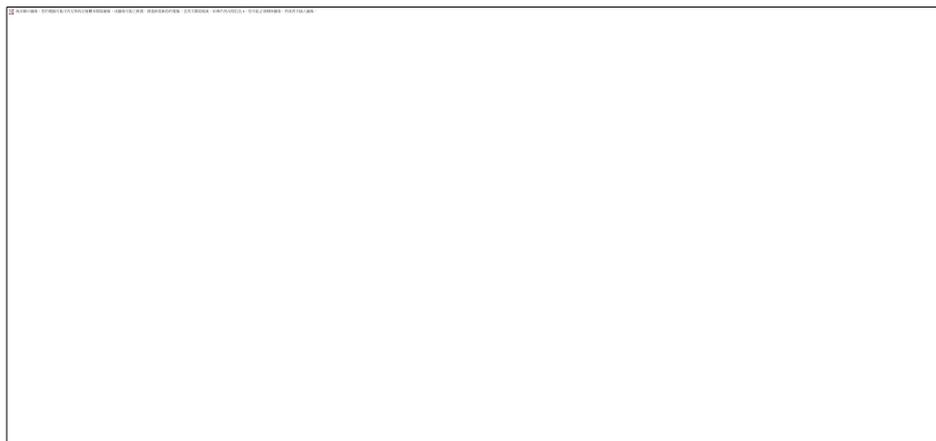
第一行：檔案中資料的個數
 第二行：X 軸的名稱
 第三行：X 軸的單位
 第四行：Y 軸的名稱
 第五行：Y 軸的單位
 之後每一行都只放一筆資料

操作步驟與結果如以下的圖示



ISaGRAF 範例程式：wpdmo108.pia

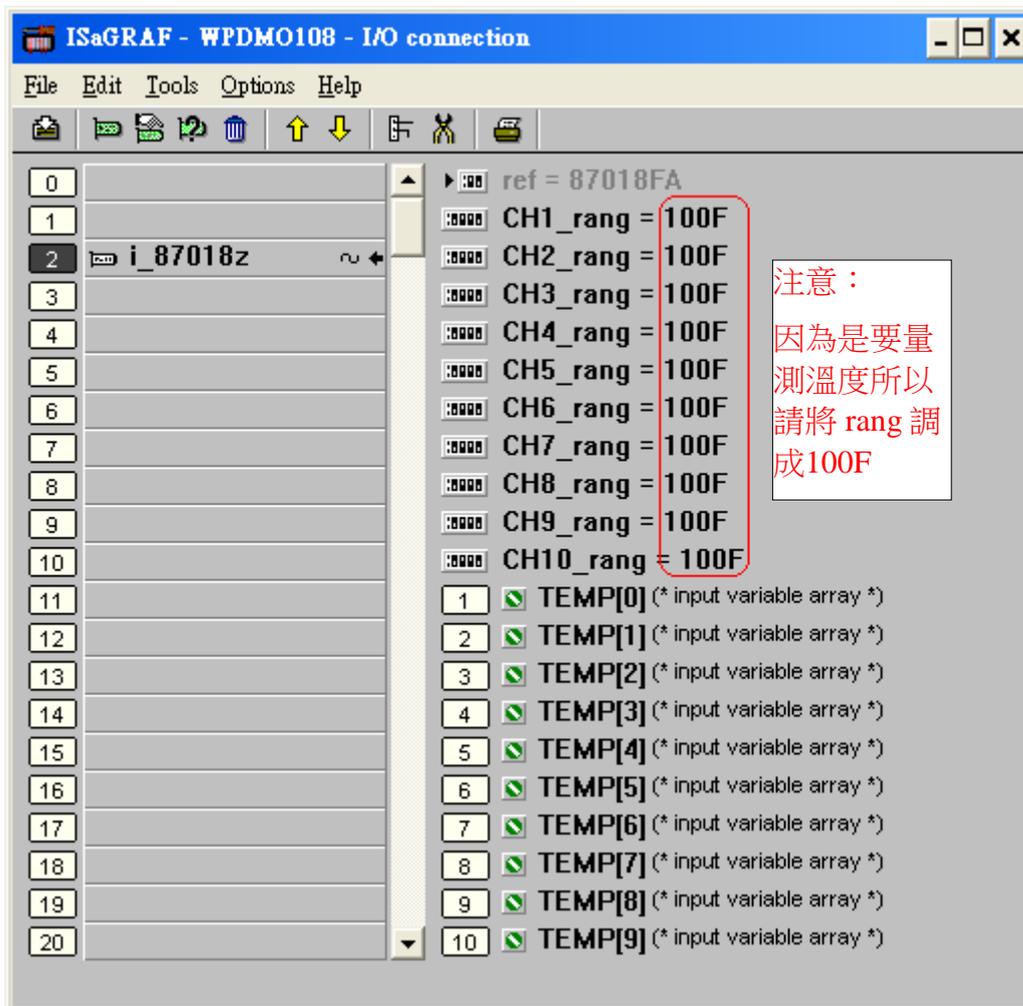
IsaGRAF 專案的架構：(包含一個 ST 程式：ST1)



變數定義：

Name	Type	Attribute	Description
Go1	Boolean	Internal	如果設為 True 時，程式將開始動作
INIT	Boolean	Internal	初值為 True
Save_file1	Boolean	Internal	程式會自動將此變數設為 True 並將開始把記錄完的值寫到檔案中
Stop1	Boolean	Internal	如果設為 True 程式將會停止
TMP	Boolean	Internal	給程式內部暫時使用的變數
TEMP[0...9]	Integer	Input	變數陣列 dim 設為8，需連接訊號至 I-87018z 的 CH.1-CH.10，這個 DEMO 只使用了 CH.3
TMP_v	Integer	Internal	程式內部暫時使用的變數
STEP1	Integer	Internal	目前記錄的狀態 0:無動作 1:紀錄中 2:記錄完成
Interval1	Integer	Internal	多久記錄一次資料，單位為1秒
Period1	Integer	Internal	本次記錄要記錄多久，單位為分鐘
ii	Integer	Internal	用在 for 迴圈中
File1	Integer	Internal	File ID
save_cnt1	Integer	Internal	目前已經寫入檔案中的資料個數
total_record1	Integer	Internal	本次總共有多少值要記錄? 這個值程式將會自動計算出來
record_cnt1	Integer	Internal	目前已經記錄完成的資料個數
T1	Timer	Internal	作為計時用的 Timer
T1_Interval	Timer	Internal	兩筆資料要多久記錄一次
T1_next	Timer	Internal	下一次記錄資料的時間
Msg1	Message	Internal	程式運作的狀態長度為255個字元初始值為”No Action”
File_name1	Message	Internal	檔案名稱長度64字元初始值為 \System_Disk\
str1	Message	Internal	長度為255字元，程式內部使用

IO 連結：



結構化文字程式-ST1

(* 下面這個區塊只有在程式的第一個 PLC scan 會動作 *)

```
if INIT then
  INIT := false;
  (* 設定檔案名稱 *)
  File_name1 := File_name1 + 'temperature.txt';
end_if;
```

(* 如果 Stop1被設成 True，程式將會停止計時與記錄資料 *)

```
if Stop1 then
  Stop1 := False;
  Step1 := 0; (* 0: 無動作 *)
  TStop(T1); (* 停止計時 *)
  T1 := T#0s; (* 將時間歸零 *)
  Msg1 := 'User stop recording!';
end_if;
```

(* 如果 Go1被設成 True 將準備開始記錄資料 *)

```
if Go1 then

  Go1 := False;
```

```

if STEP1 = 1 then
    (* 仍然在記錄中..... *)
    Msg1 := 'It is still recording now... Please wait';
else
    (* 確認 interval 的值是否有效，單位為1秒 *)
    (* 這裡假設1-60秒為有效 *)
    if (Interval1 < 1) or (Interval1 > 60) then
        Msg1 := 'Wrong Interval value, it should be in 1 to 60 second!';

        (* 確認 Period1 的值是否有效，單位為一分鐘 *)
        (* 這裡假設1-300分鐘為有效 *)
    elseif (Period1 < 1) or (Period1 > 300) then
        Msg1 := 'Wrong Period value,it should be in 1 to 1440 minute!';
    else
        (* 所有的參數檢查都是正確之後，開始準備記錄資料 *)
        (* 先計算 total_record1 的值 *)
        total_record1 := (Period1 * 60) / Interval1;
        record_cnt1 := 0; (* 重設 count_cnt1 為零 *)
        STEP1 := 1; (* 1: 記錄中 *)
        Msg1 := 'Recording now ... Please wait';

        (* 讓 T1 重新開始計時 *)
        T1 := T#0s;
        T1_Interval := TMR(Interval1 * 1000);
        T1_next := T1 + T1_Interval;
        TStart(T1); (* 開始計時 *)
        save_cnt1 := 0;

        (* 建立新的檔案 *)
        File1 := F_creat(File_name1);

        (* 如果建立檔案時發生錯誤 *)
        if File1 = 0 then
            Msg1 := 'Create File' + File_name1 + 'Error!!!';
        else
            (* 若沒有發生錯誤時 *)
            Msg1 := 'Create File' + File_name1 + 'Success!!!';

            (* 寫入一些必要的資料在檔案的一開始 *)
            (* 這樣方便我們之後畫資料的趨勢圖 *)
            str1 := "";
            (* 第一行為檔案中資料的個數 *)
            str1 := INT_str3(Total_record1,5) + '$0D$0A';
            (* 第二行 X 軸的名稱 *)
            str1 := str1 + 'Time' + '$0D$0A';
            (* 第三行為 X 軸的單位 *)
            str1 := str1 + '(' + INT_str3(Interval1, 2) + 'sec' + ')' + '$0D$0A';
            (* 第四行為 Y 軸的名稱 *)
            str1 := str1 + 'temperature' + '$0D$0A';
            (* 第五行為 Y 軸的單位 *)

```

```

        str1 := str1 + '(C.)' + '$0D$0A';
        TMP := F_writ_s(File1,str1);
        TMP := F_close(File1); (* 關閉檔案 *)
    end_if;
end_if;
end_if;
end_if;

(* 開始寫入資料到檔案中 *)
if STEP1 = 1 then

    (* 如果 T1 >= T1_next , 寫入當時溫度的資料到檔案中 *)
    if T1 >= T1_next then
        (* 更新下次要記錄時間 T1_next 的值 *)
        T1_next := T1_next + T1_Interval;

        File1 := F_wopen(File_name1);

        if File1 = 0 then
            Msg1 := 'Open File' + File_name1 + 'Error!!!';
            STEP1 := 0; (* 1: 無動作 *)
            TStop(T1); (* 停止計時 *);
            T1 := T#0s;
        else
            TMP := F_end(File1);
            TMP := FM_write(File1,Rea_Str2(REAL(TEMP[0]) / 100.0,1));
            TMP := F_close(File1);

            (* 當成功寫入檔案中時 , 將 record_cnt1加1 *)
            record_cnt1 := record_cnt1 + 1;

            (* 檢查所有的資料是不是已經記錄完畢 *)
            if (record_cnt1 >= total_record1) then
                (* 記錄結束 *)
                STEP1 := 0; (* 將 step1設成0 : 無動作 *)
                Tstop(T1); (* 停止計時 *)
            end_if;
        end_if;
    end_if;
end_if;
end_if;
end_if;

```

C# .net 程式

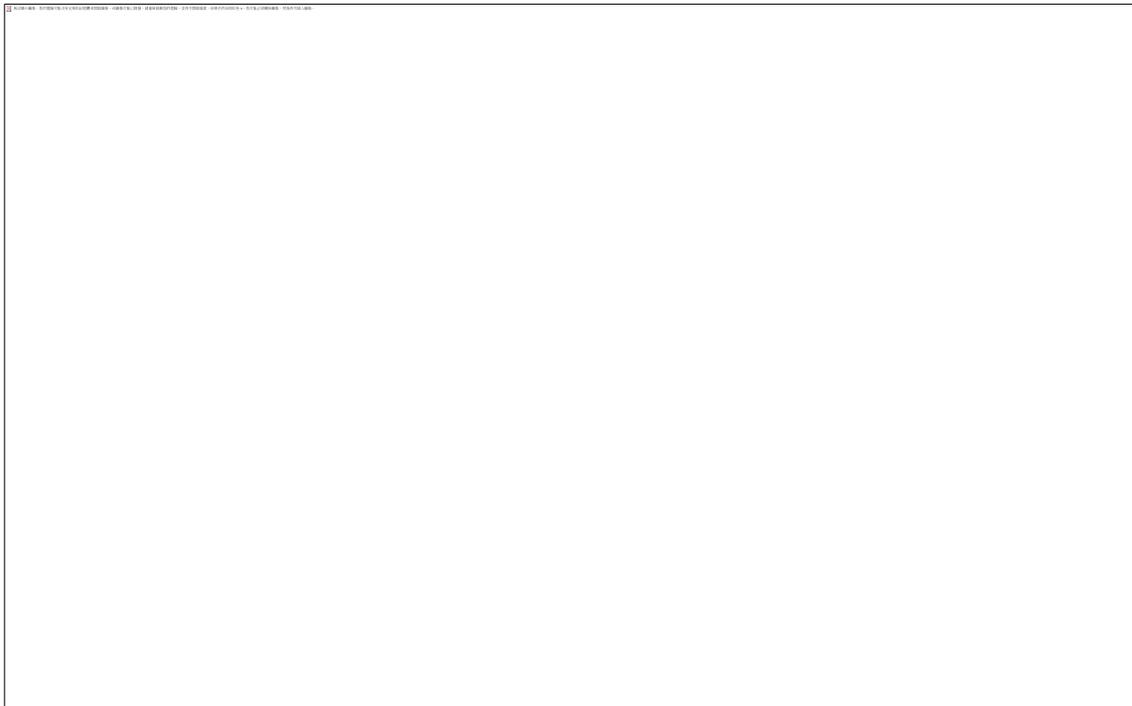
注意：這個程式是在 Virtual studio 2008 基於 .net framework 2.0 上開發的。因為 WP-8xx7 只支持 ISaGRAF, VS .net 2008/2005/2003 基於 .net framework 2.0 或 EVC++ 4.0

1. 開啟一個新的專案

步驟1：



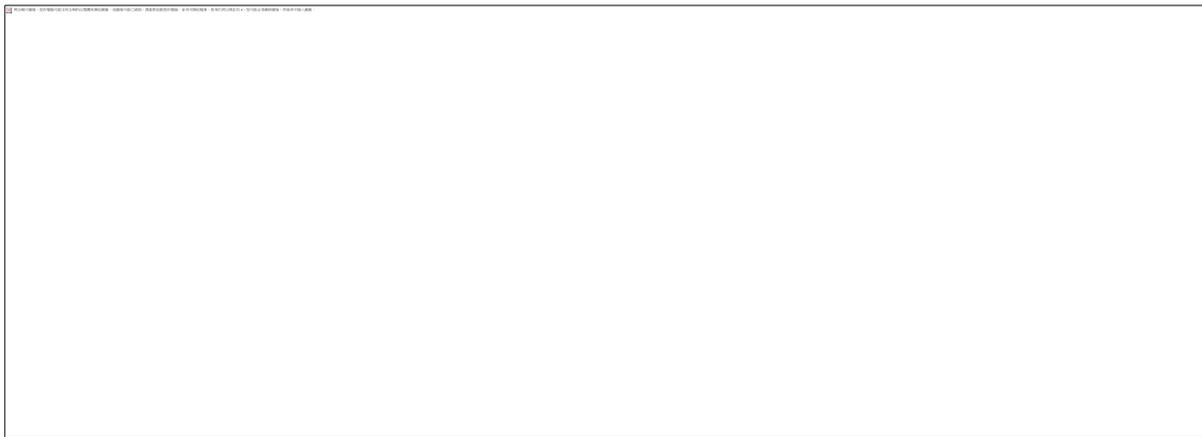
步驟2：選擇欲開發的專案 C# → SmartDevice → Smart Device Project



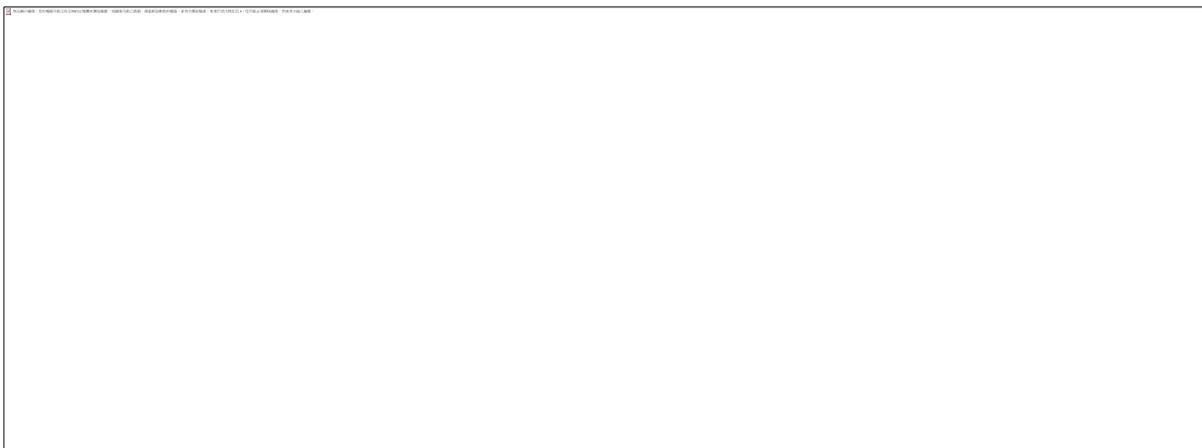
步驟3：選擇開發平台



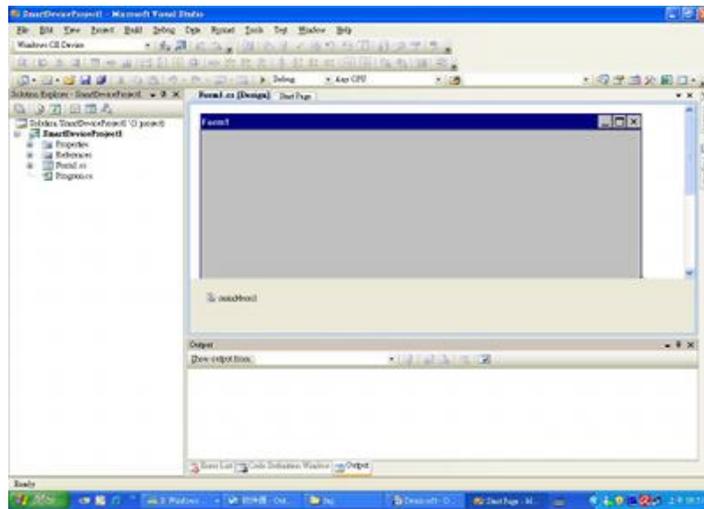
步驟4：選擇.net Compact Framework 版本(WP-8xx7 內建 Compact Framework 2.0)



步驟5：選擇範本



步驟6：點一下 OK 鍵然後你可以看到專案開啟的結果如下圖

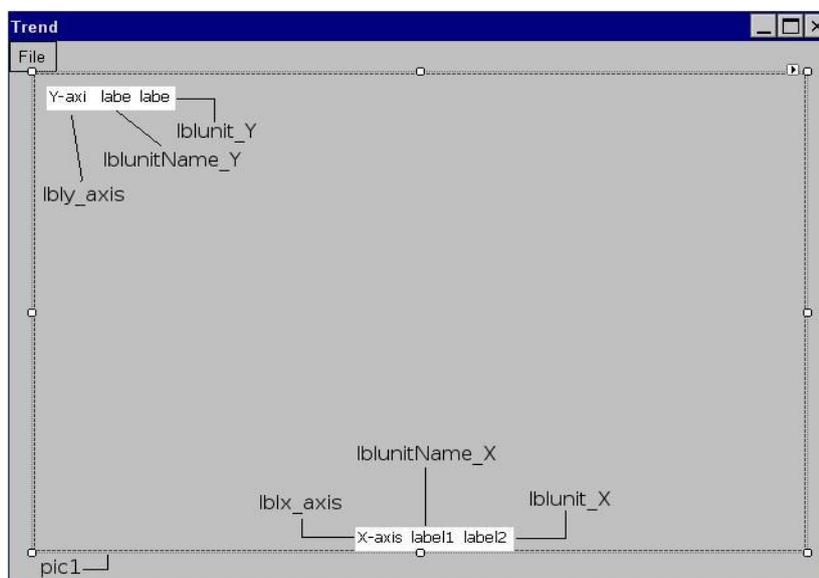


2.表單 FORM1控制項的屬性

控制項屬性表：

	Name	Text
label1	lbly_axis	Y-axis
label2	lblunitName_Y	-
label3	lblunit_Y	-
label4	lblx_axis	X-axis
label5	lblunitName_X	-
label6	lblunit_X	-
picturebox1	pic1	

控制項的位置示意圖：



新增子功能表於 file 主功能表下，如圖



新增一個檔案對話框，如圖



3.將以下的原始碼寫在 form1.cs 中

```
using System;

using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Trend1_Demo_
{
    public partial class form1 : Form
    {
        //設定全域的變數
        string filename;
        FileInfo fileInfo;
        FileStream fs;
        StreamReader sr;
        int[] list;

        //設定原點座標
        Point origin = new Point(50, 350);

        Bitmap img;
        Graphics g;

        //專案範本自動產生的原始碼
```

```

public form1()
{
    InitializeComponent();
}

//當子功能表open file被按下時所觸發的事件處理函式
private void menuItem2_Click(object sender, EventArgs e)
{
    //檢查檔案對話框的結果是不是OK
    if (openFileDialog1.ShowDialog() == DialogResult.OK)

        //如果是OK，將檔案對話框的成員FileName的值傳給全域變數filename
        filename = openFileDialog1.FileName;
    else
        ;//如果結果是取消、關閉和終止的話，不做任何回應

    //檢查檔案是否存在
    if (File.Exists(filename))
    {
        //檔案存在將執行下列的程式碼

        //宣告一個string的變數去接欲開啟檔案的完整路徑
        string fPath = Path.GetFullPath(filename);

        //建立一個fileinfo類別的實體 並將記憶體位置傳給全域變數fileInfo
        fileInfo = new FileInfo(fPath);

        //利用fileInfo的方法開啟檔案
        fs = fileInfo.Open(FileMode.Open);

        //建立一個檔案串流的實體並將記憶體位置傳給全域變數sr
        sr = new StreamReader(fs, System.Text.Encoding.Default);

        if (fs.Length > 0)
        {
            //讀入第一行得到檔案中資料的數量
            //這裡我們使用類別檔案串流的方法Readline()
            int Lenth = Int32.Parse(sr.ReadLine());

            //讀入第二行得到X軸的名字
            //傳給lblunitname_x的成員text
            //我們會在畫面中看到牠
            lblunitName_X.Text = sr.ReadLine();

            //讀入第三行得到X軸的單位
            lblunit_X.Text = sr.ReadLine();
        }
    }
}

```

```

//讀入第四行Y軸的名字
lblunitName_Y.Text = sr.ReadLine();

//讀入第五行Y軸的單位
lblunit_Y.Text = sr.ReadLine();

//建立一個整數的陣列用來儲存檔案中的資料
list = new int[Lenth];

//依序的將檔案中的資料寫入陣列中
int i=0;
do
{
    list[i]=(int)(float.Parse(sr.ReadLine()) * 10);
    i++;
} while (sr.Peek() != -1);
}

//讀完資料後，關閉檔案流
fs.Close();

//建立一個bitmap類的物件，大小跟物件pic一樣
img = new Bitmap(pic1.Size.Width, pic1.Size.Height);

//建立一個筆刷類的物件，顏色為黑色，寬度為3
Pen p_cod = new Pen(Color.Black, 3);

//在剛剛的img上建立一個畫布
g = Graphics.FromImage(img);

//用白色清除整個畫布
g.Clear(Color.White);

//把img的圖嵌入pic1
pic1.Image = img;

//在畫布上畫出座標

//畫上X軸
g.DrawLine(p_cod, origin.X - 1, origin.Y, origin.X + 500 + 2, origin.Y);

//畫上Y軸
g.DrawLine(p_cod, origin.X, origin.Y + 1, origin.X, origin.Y - 300 - 1);

//畫上座標上面的那一條線
g.DrawLine(p_cod, origin.X - 1, origin.Y - 300, origin.X + 500 + 2, origin.Y - 300);

//畫上座標右邊的那一條線

```

```

g.DrawLine(p_cod, origin.X+500, origin.Y + 1, origin.X + 500, origin.Y - 300 - 1);

//建立一個筆刷顏色是淺灰色，用來畫格子的
Pen pen_grid =new Pen(Color.LightGray,1);

//將座標畫上格線

//畫鉛直的線
for (int i = 0; i < 9; i++)
{
    g.DrawLine(pen_grid, origin.X + 50 * (i + 1), origin.Y, origin.X + 50 * (i + 1), origin.Y
- 300);
}

//畫水平的線
for (int i = 0; i < 9; i++)
{
    g.DrawLine(pen_grid, origin.X, origin.Y - 30 * (i + 1), origin.X + 500, origin.Y - 30 *
(i + 1));
}

//畫上X軸上的刻度
for (int i = 0; i < 9; i++)
{
    g.DrawLine(p_cod, origin.X + 50 * (i + 1), origin.Y, origin.X + 50 * (i + 1), origin.Y -
5);
}

//畫上Y軸上的刻度
for (int i = 0; i < 9; i++)
{
    g.DrawLine(p_cod, origin.X, origin.Y - 30 * (i + 1), origin.X + 5, origin.Y - 30 * (i +
1));
}

//利用pic的方法，重繪，這樣就可以把剛剛我們在畫布上畫的基本的座標顯示出來
pic1.Refresh();

//並且讓這些所有的標籤顯示
lblx_axis.Visible = true;
lbly_axis.Visible = true;
lblunitName_X.Visible = true;
lblunitName_Y.Visible = true;
lblunit_X.Visible = true;
lblunit_Y.Visible = true;

//得到資料中最大與最小值
int List_Max=get_max(list,list.Length);
int List_Min=get_min(list,list.Length);

//計算資料座標縮放的比例

```

```

double Scale_X = (double)list.Length / 500.0;
double Scale_Y = (double)(List_Max - List_Min) / 240;

//設定Y軸刻度的數值
set_label_Y(List_Max,List_Min);

//設定X軸刻度的數值
set_lable_X(list.Length);

//宣告一個Point的陣列來儲存轉換過後的資料
Point[] Data = new Point[list.Length];

//將每一筆資料轉換成座標並且依序儲存在剛剛建立的陣列中
for (int i = 0; i < list.Length; i++)
{
    Data[i] = new Point(origin.X + (int)((i + 1) / Scale_X), origin.Y - (int)((list[i] -
List_Min) / Scale_Y) - 30);
}

//開始將資料畫在座標中

//建立一個Pen的物件，紅色，寬度1
Pen line = new Pen(Color.Red, 1);

//將所有的點利用drawlines的方法畫在畫布上
g.DrawLine(line,Data);

//重繪pic1的物件
pic1.Refresh();
}
}

private void Form1_Load(object sender, EventArgs e)
{
}

//找出陣列中最大值的方法
private int get_max(int[] list,int count)
{
    int Max = list[0];
    for (int i = 1; i < count; i++)
        if (Max < list[i])
            Max = list[i];

    return Max;
}

//找出陣列中最小值的方法
private int get_min(int[] list, int count)
{
    int Min = list[0];

```

```

    for (int i = 1; i < count; i++)
        if (Min > list[i])
            Min = list[i];
    return Min;
}

//動態產生X軸上刻度的標籤
private void set_label_X(int number_X)
{
    //算出刻度間的級距
    int scale = number_X / 10;

    //建立一個Label的陣列
    Label[] label_X = new Label[10];

    //建立陣列中每一個label的物件並且初始化//建立陣列中每一個label的物件並且初始化
    for (int i = 0; i < label_X.Length; i++)
    {
        //建立Label的物件
        label_X[i] = new Label();

        //設定要擺放的位置
        label_X[i].Left = pic1.Location.X + origin.X + 50 * (i + 1) - 25;
        label_X[i].Top = pic1.Location.Y + origin.Y + 3;

        //設定Label的大小
        label_X[i].Size = new System.Drawing.Size(49, 20);

        //設定Label的背景色為白色
        label_X[i].BackColor = Color.White;

        //設定要顯示的文字
        label_X[i].Text = (scale * (i + 1)).ToString();

        //設定文字在Label中顯示的位置，我們設定為置中
        label_X[i].TextAlign = ContentAlignment.TopCenter;

        //設定是否為可見的
        label_X[i].Visible = true;

        //將這個控制項加到Form1中
        this.Controls.Add(label_X[i]);

        //利用bringtofront的方法來確保標籤是可見的
        label_X[i].BringToFront();
    }
}

//建立Y軸上標籤的方法

```

```

private void set_label_Y(int Max,int Min)
{
    //計算最大值與最小值之間的差距
    int range = Max-Min;

    //計算出刻度間間距
    double scale = (double)range / 8.0;

    //建立Label的陣列
    Label[] label_Y = new Label[10];

    //建立Label的物件並且初始化
    for (int i = 0; i < label_Y.Length; i++)
    {
        //建立Label的物件
        label_Y[i] = new Label();

        //設定Label的位置
        label_Y[i].Left = pic1.Location.X;
        label_Y[i].Top = pic1.Location.Y + origin.Y - 30 * (i + 1) - 5;

        //設定Label的大小
        label_Y[i].Size = new System.Drawing.Size(49, 20);

        //設定Label的背景色為白色
        label_Y[i].BackColor = Color.White;

        //設定標籤中的文字
        label_Y[i].Text = System.String.Format("{0:F1}", (double)(Min + scale * i) / 10.0);

        //設定標籤中文字顯示的位置，這裡是靠右
        label_Y[i].TextAlign = ContentAlignment.TopRight;

        //設定標籤是否可見
        label_Y[i].Visible = true;

        //將這個控制項加到Form1的表單中
        this.Controls.Add(label_Y[i]);

        //利用bringtofront的方法來確保標籤是可見的
        label_Y[i].BringToFront();
    }
}
}
}

```