如何利用 WinPAC-8xx7和 I-8084W 來量測頻率 或轉速輸入並使用 C# .net 2008將所記錄的值畫成趨勢圖

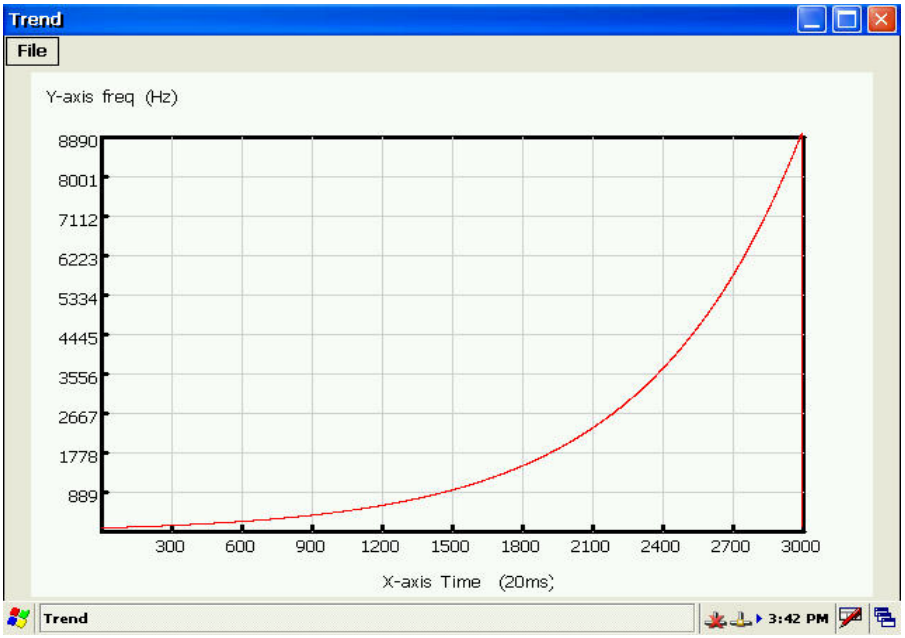
By joseph_dun@icpdas.com

本例將利用 WinPAC-8xx7和 I-8084W 和 ISaGRAF 作出一個量測頻率的應用

應用說明：

使用 ISaGRAF 程式將 I-8084W 在一段時間內記錄下訊號的頻率，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 WP-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馬達、引擎轉速...等，然後運行該 C# .net 2008程式可以直接在 WinPAC 上就可以看出資料是否異常。

下圖是利用安捷倫(Agilent 33220A)函數/任意波形產生器，在一分鐘的時間裡輸出5V 的方波使其頻率呈現 Log 的數值



關於在 ISaGRAF 與 WinPAC-8xx7中初次使用 I-8084W 的相關設定說明請參考
www.icpdas.com → FAQ → Software →ISaGRAF Ver.3(Chinese) → 100
(http://www.icpdas.com/faq/isagraf_c.htm → 100)

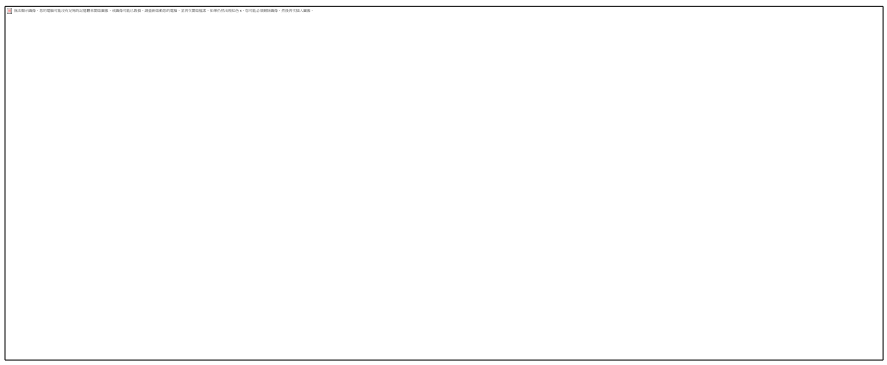
如需要了解更多 I-8084W 與 WinPAC-8xx7的資訊可至下列網址中查詢

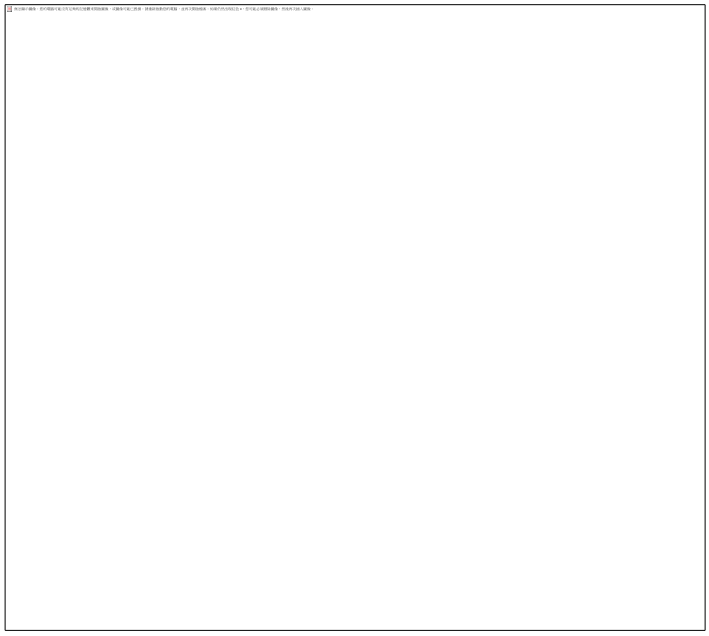
WinPAC-8xx7→ http://www.icpdas.com/products/PAC/winpac/wp-8x47.html
I-8084W→ http://www.icpdas.com/products/Remote_IO/i-8ke/i-8084w.htm

本 ISaGRAF 範例程式為 wpdmo106.pia 放於
www.icpdas.com → FAQ → Software → ISaGRAF Ver.3(Chinese) → 106

請依照下列步驟可將範例程式(wpdmo106.pia)載入 ISaGRAF 中：

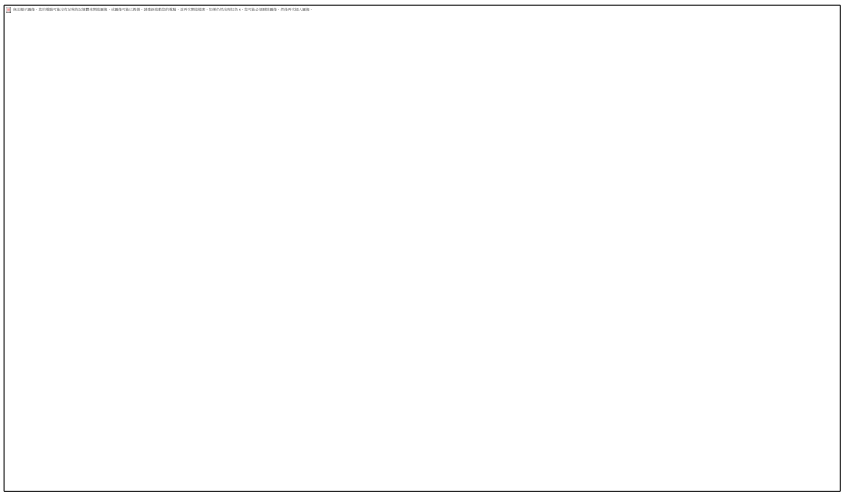
1.請點選 Tool→Archive→Projects



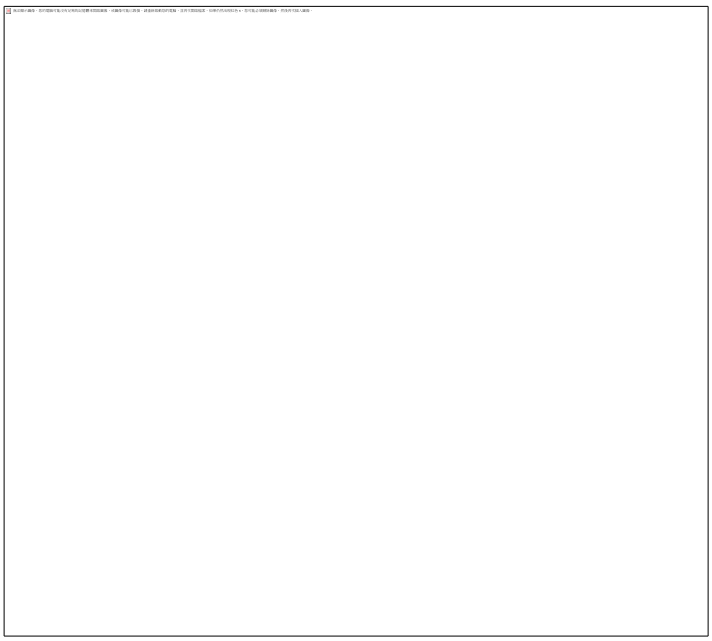


2.之後會開啟一個對話框按下 Browse 按鈕

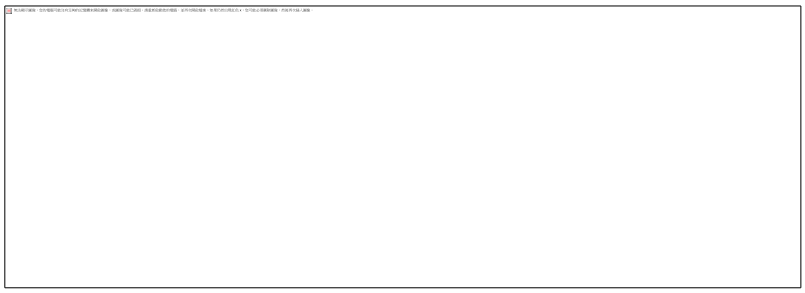
3.選擇欲匯入專案檔所在資料夾，按下確定鍵



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



在 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_1.exe 與原始碼放於
www.icpdas.com → FAQ → Software → ISaGRAF Ver.3(Chinese) → 106

可以直接執行範例程式 Demo_1.exe 於 WinPAC-8xx7上
可以利用 ftp 的方式將 Demo_1.exe 上傳到 WinPAC 的任何位置都可以，之後就可以在 WinPAC 上的 Demo_1.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).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

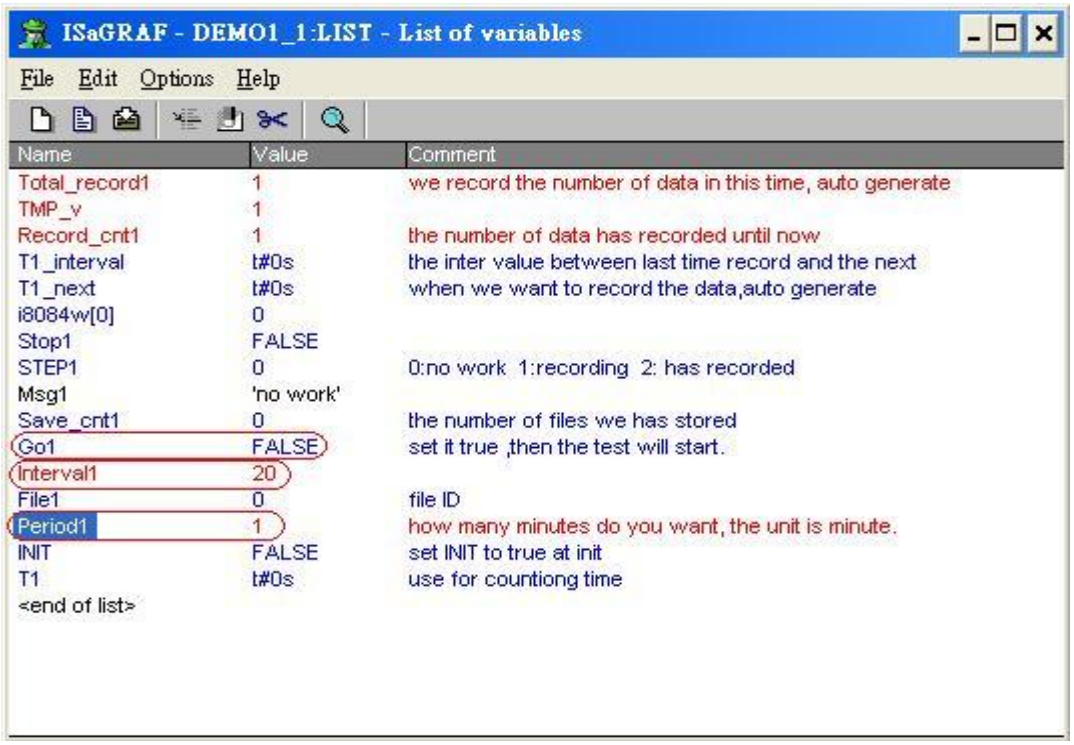
如何測試本範例?

1.請將 i-8084W 插在 WP-8xx7的 Slot 0，然後將你想要記錄的訊號接到 i-8084W 的 CH.1上，

這裡使用的訊號源為安捷倫(Agilent 33220A)的波形產生器，之後將 WinPAC 上電。

2.將 ISaGRAF 中的範例程式 wpdmo106下載到 WP-8xx7內

3:在 list of variables 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0.001秒，如果給20指的是每0.02秒記錄一次。而 "Period1"指的是要記錄多久。單位為分鐘，輸入好後將 GO1設成 True 就可以開始動作，程式會將輸入於 i-8084W 的 CH.1的訊號頻率值按照剛剛設定的數值記錄下來



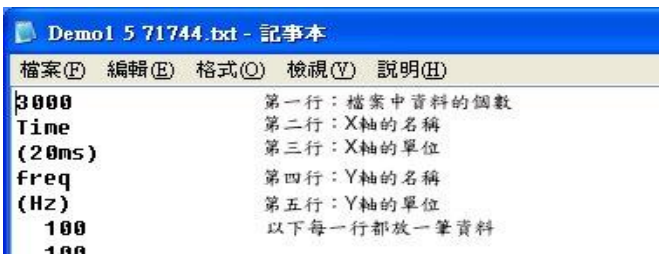
左圖為是設定 Interval1為 20(0.02秒) Period1為1 (1分鐘)

在記錄過程中，會看到"Record_cnt1"的值會一值增加，當等於"Total_record1"的值時，就表示記錄完畢。此時程式會將記錄下來的資料存入檔案中。

範例程式會將產生出來的檔案存在/System_Disk/的資料夾中

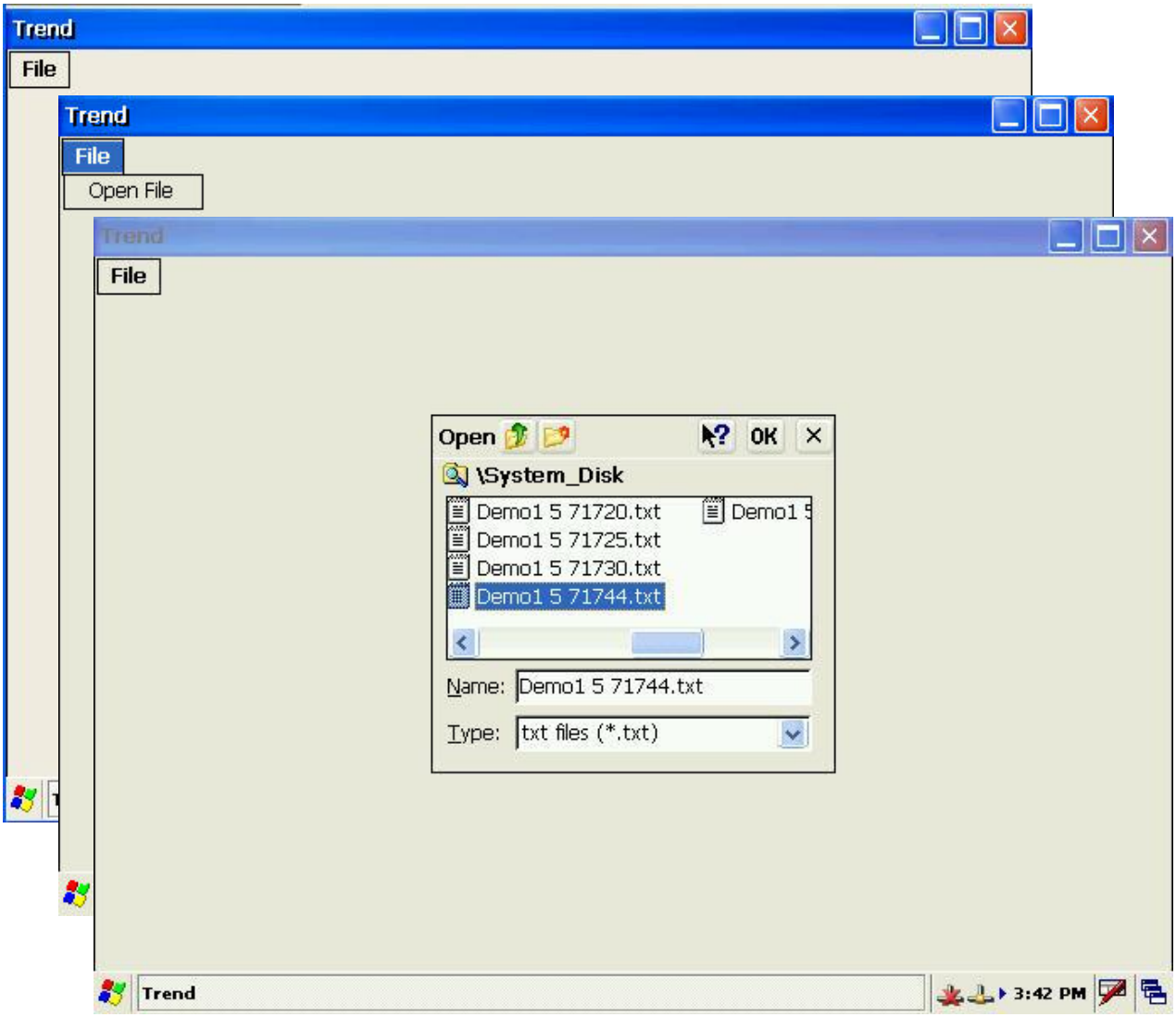
.4.之後利用寫好的 C#程式在 WP-8xx7的路徑/System_Disk/中開啟剛剛記錄下的檔案"Demo1*.txt"。就可以直接看到訊號的頻率趨勢圖(詳細操作如下一頁之說明)。

這裡說明一下 txt 檔的格式：



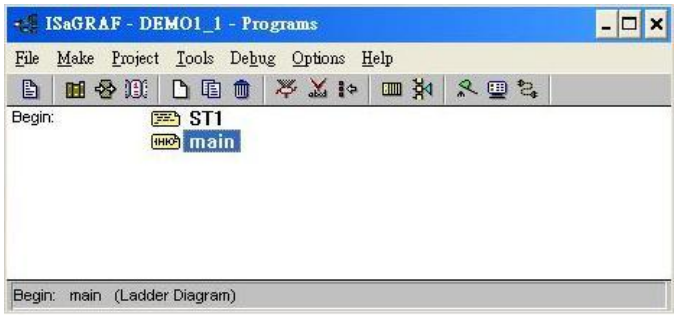
第一行：檔案中資料的個數
第二行：X 軸的名稱
第三行：X 軸的單位
第四行：Y 軸的名稱
第五行：Y 軸的單位
之後每一行都只放一筆資料

操作步驟與結果如以下的圖示



ISaGRAF 範例程式：wpdmo106.pia

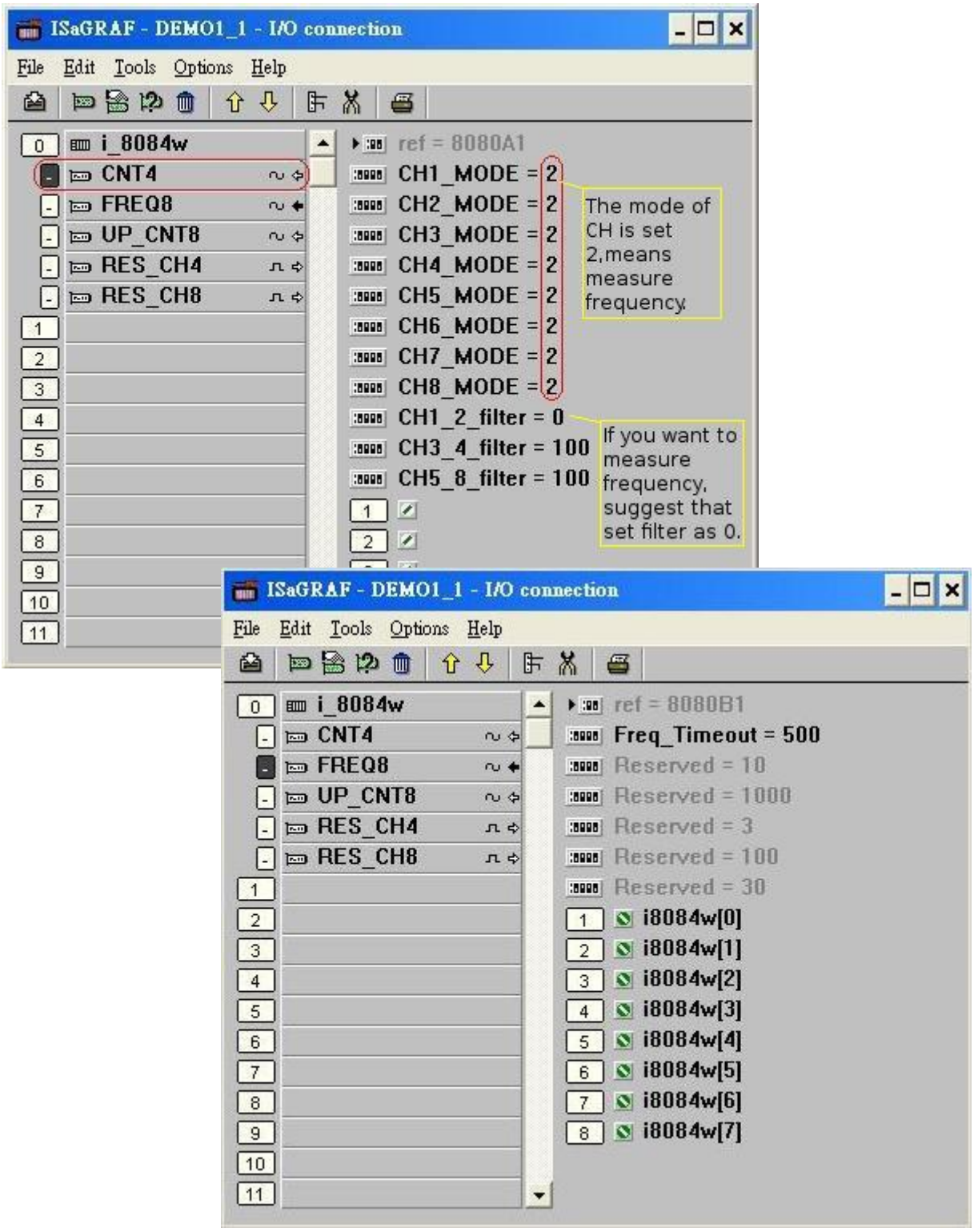
IsaGRAF 專案的架構：(包含一個 ST 程式：ST1 與階梯圖程式：main)



變數定義：

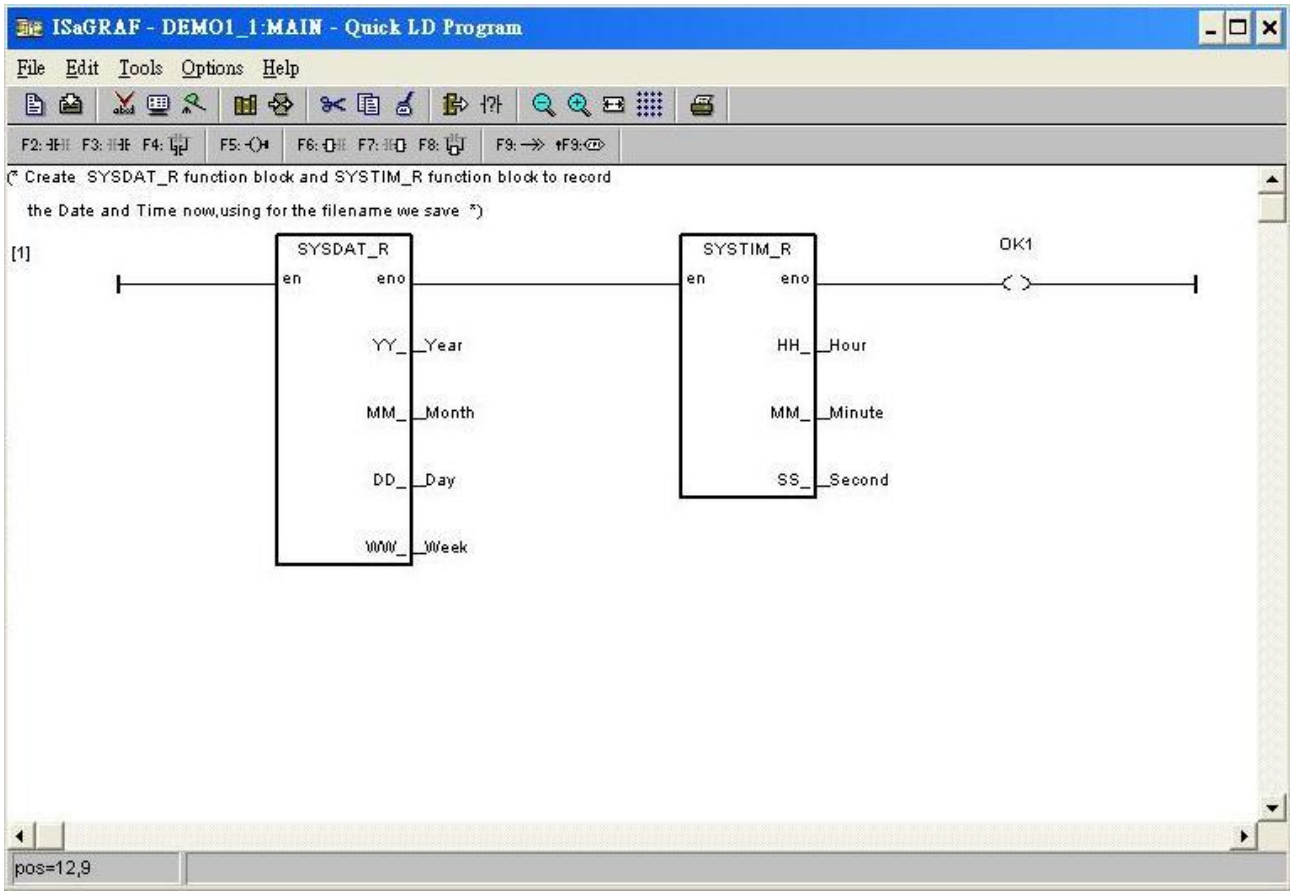
Name	Type	Attribute	Description
Go1	Boolean	**Internal**	如果設為 True 程式將開始動作
Stop1	Boolean	**Internal**	如果設為 True 程式將會停止
TMP	Boolean	**Internal**	給程式內部暫時使用的變數
INIT	Boolean	**Internal**	初始值為 True
Save_file1	Boolean	**Internal**	程式會自動將此變數設為 True 並將開始把記錄完的值寫到檔案中
File1	Integer	**Internal**	File ID
STEP1	Integer	**Internal**	目前記錄的狀態 0:無動作 1:紀錄中 2:記錄完成
Period1	Integer	**Internal**	本次記錄要記錄多久，單位為分鐘
Interval1	Integer	**Internal**	多久記錄一次資料，單位為0.001秒
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
ii2	Integer	**Internal**	用在 for 迴圈中
I8084W[0..7]	Integer	**Input**	變數陣列 dim 設為8，需連接訊號至 i8084w 的 CH.1-CH.8，這個 DEMO 只使用了 CH.1
Save_cnt1	Integer	**Internal**	目前已經寫入檔案中的資料個數
TMP_v	Integer	**Internal**	程式內部暫時使用的變數
T1	Timer	**Internal**	作為計時用的 Timer
T1_next	Timer	**Internal**	下一次記錄資料的時間
T1_interval	Timer	**Internal**	兩筆資料要多久記錄一次
File_name1	Message	**Internal**	檔案名稱長度64字元初始值為 \System_Disk\Demo1. 我們將加入一些數值像日期、時間來區分不同的記錄
File_name_t	Message	**Internal**	暫時儲存檔案名稱的變數
Msg1	Message	**Internal**	程式運作的狀態長度為255個字元初始值為”No Action now”.
Str1	Message	**Internal**	長度為255字元，程式內部使用

IO 連結：


ISaGRAF - DEMO1_1 - I/O connection
File Edit Tools Options Help
i_8084w
CNT4
FREQ8
UP_CNT8
RES_CH4
RES_CH8
ref = 8080A1
CH1_MODE = 2
CH2_MODE = 2
CH3_MODE = 2
CH4_MODE = 2
CH5_MODE = 2
CH6_MODE = 2
CH7_MODE = 2
CH8_MODE = 2
CH1_2_filter = 0
CH3_4_filter = 100
CH5_8_filter = 100
The mode of CH is set 2,means measure frequency.
If you want to measure frequency, suggest that set filter as 0.
ISaGRAF - DEMO1_1 - I/O connection
File Edit Tools Options Help
i_8084w
CNT4
FREQ8
UP_CNT8
RES_CH4
RES_CH8
ref = 8080B1
Freq_Timeout = 500
Reserved = 10
Reserved = 1000
Reserved = 3
Reserved = 100
Reserved = 30
i8084w[0]
i8084w[1]
i8084w[2]
i8084w[3]
i8084w[4]
i8084w[5]
i8084w[6]
i8084w[7]

階梯圖程式-main

這個階梯圖很簡單，只有在寫入檔案時用來當作識別檔案的檔案名稱用



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;

(*配置一塊可存放 30,000 整數 (或實數) 的空間 *)
(* 函數 ARcreate 只能在 ISaGRAF 程式中呼叫一次 如果呼叫第二次以上程式會出現錯誤 *)
(* 第一個參數一定為1，第二個參數為你需要的記憶體空間單位為4byte *)
        TMP_v := ARcreate(1,30000);

        if TMP_v <> 1 then (*如果回傳不是1代表失敗*)
                Msg1 := 'Parameter error or Can not allocate enough memory by ARcreate()
function!';
        end_if;

        (*設定 WinPAC 以最高速度執行，可能會影響非 ISaGRAF 的程式*)
        TMP := PLC_mode(-1);
end_if;

(* 如果 Stop1被設成 True 將停止計時 和 記錄資料*)
if Stop1 then
        Stop1 := False;
        STEP1 := 0;   (* 0: 無動作 *)
        TStop(T1);    (* 停止計時 *)
        T1 := T#0s; (* 回復計時時間為0 *)
        Msg1 := 'User stop recording!';
        save_cnt1 := 0;
end_if;

(* 如果 Go1被設成 True 將準備開始記錄 *)
if Go1 then

        Go1 := False;

        if STEP1=1 then (* 0:無動作, 1:記錄中, 2:記錄完成 *)
                (* 更新狀態顯示為，仍然在紀錄中..... *)
                Msg1 := 'It is still recording now... Please wait';
        else
                (* 檢查 interval 的值是否有效 *)
                (* 假設為10到10000之間單位為0.001秒 *)
                (* 如果平均的 PLC scan time 是比較長的,比如說將近10ms,
                   請將這個值設的比平均的 PLC scan time 還大
                   否則可能沒辦法正確的記錄到資料 *)
                if (Interval1 < 10) or (Interval1 > 10000) then
                        Msg1 := 'Wrong Interval value, it should be in 10 to 10000 milli-second!';
                (* 檢查 period1 的值是否有效 *)
                (* 在這個 demo 中假設有效值為1~10單位為分鐘 *)
                elsif (Period1 < 1) or (Period1 >10) then
                        Msg1 := 'Wrong Period value, it should be in 1 to 10 minute!';

                        else
                                (* 所有的參數都設定完成，即將開始記錄資料 *)
```

```
                    (* 計算本次需要記錄多少個資料 *)
                    total_record1 := (Period1 * 60000) / Interval1;
                    record_cnt1 := 0; (* 重設 record_cnt 為 0 *)
                    STEP1 := 1; (* 將 step1設成1代表狀態記錄中 *)
                    Msg1 := 'Recording now ... Please wait';

                    (*  啟動 T1從0開始計時 *)
                    T1 := T#0s;
                    T1_Interval := TMR(Interval1);
                    T1_next := T1 + T1_Interval;
                    TStart(T1);  (* 開始計時 *)
                    save_cnt1 := 0;
            end_if;
        end_if;
end_if;

(* 記錄中 *)
if STEP1 = 1 then

        (* 準備寫入一筆資料在記憶體中 *)
        if T1 >= T1_next then

                (* 當 T1>=T1_next 時 寫入一筆資料到記憶體中*)
                (* 更新下次要記錄的時間 T1_next *)
                T1_next := T1_next + T1_Interval;

                (* 準備將記錄下來的資料寫入記憶體中 *)
                (* 利用函式 ARwrite() 寫入記憶體 *)
                (* 第一個參數一定要是1 *)
                (* 第二個參數是要將資料放在記憶體中的位置 *)
                (* 第三個參數為要放進記憶體的資料 *)
                TMP_v := ARwrite(1,record_cnt1,i8084w[0]);

                (* 檢查函數 ARwrite() 的回傳值是否正常 *)
                if TMP_v <> 1 then
                        Msg1 := 'Can not operate ARwrite()!';
                        STEP1 := 0; (* 0:無動作 *)
                        TStop(T1);     (* 停止計時 *)
                        T1 := T#0s;
                end_if;

                (* 已記錄的資料數值加1*)
                record_cnt1 := record_cnt1 + 1;

                (* 檢查資料記錄是否完成 *)
                if(record_cnt1 >= total_record1) then
                        (* 資料記錄已經完成，準備利用幾次 PLC scan time，
                           將記錄下來的資料寫入 RAM Disk 檔案 *)
```

```
                    STEP1 := 0; (* 在寫入檔案前，將 step1設為0 *)
                    Tstop(T1); (* 停止計時 *)
                    T1 := T#0s;

                    (* 將目前的日期還有時間加到檔名中方便辨識不同的測試結果 *)
                    File_name_t := File_name1 + INT_str3(Month,2) +INT_str3(Day,2) +
                                   INT_str3(Hour,2) + INT_str3(Minute,2) + '.txt';

                    (* 開一個新的檔案 *)
                    File1 := F_creat(File_name_t);

                    (* 如果開檔失敗 *)
                    if File1 = 0 then
                        Msg1 := 'Create File' + 'File_nam1 Error!!!';
                    else
                        (* 因為寫入大量的資料會增加 PLC scan time，所以
                           不打算一次就將所有的資料一次寫入，將分幾次 scan
                           time 將資料寫入檔案 *)
                        Msg1 := 'Please wait... Saving data to file:' + File_name1 + '...';

                        (* 寫入一些必要的資訊在檔案的一開始 *)
                        (*  這些資訊方便在 C#程式中畫趨勢圖 *)

                        (* 第一行為總共記錄多少筆資料 *)
                        str1 := INT_str3(Total_record1,4) + '$0D$0A';

                        (* 第二行為 X 軸的名字 *)
                        str1 := str1 + 'Time' +'$0D$0A';

                        (* 第三行為 X 軸的單位 *)
                        str1 := str1 + '(' + INT_str3(Interval1,2) + 'ms)' + '$0D$0A';

                        (* 第四行為 Y 軸的名字 *)
                        str1 := str1 + 'freq' + '$0D$0A';

                        (* 第五行為 Y 軸的單位 *)
                        str1 := str1 + '(Hz)' + '$0D$0A';

                        (* 將上述的資料寫入檔案的一開始 *)
                        TMP := F_writ_s(File1,str1);

                        (* 將 save_file 的變數設成 True 準備開始 *)
                        save_file1 := True;
                        save_cnt1 :=0;  (* 將寫入檔案的資料個數清成0 *)
                    end_if;
                end_if;
            end_if;
        end_if;

(* 開始寫入資料到檔案中 *)
```

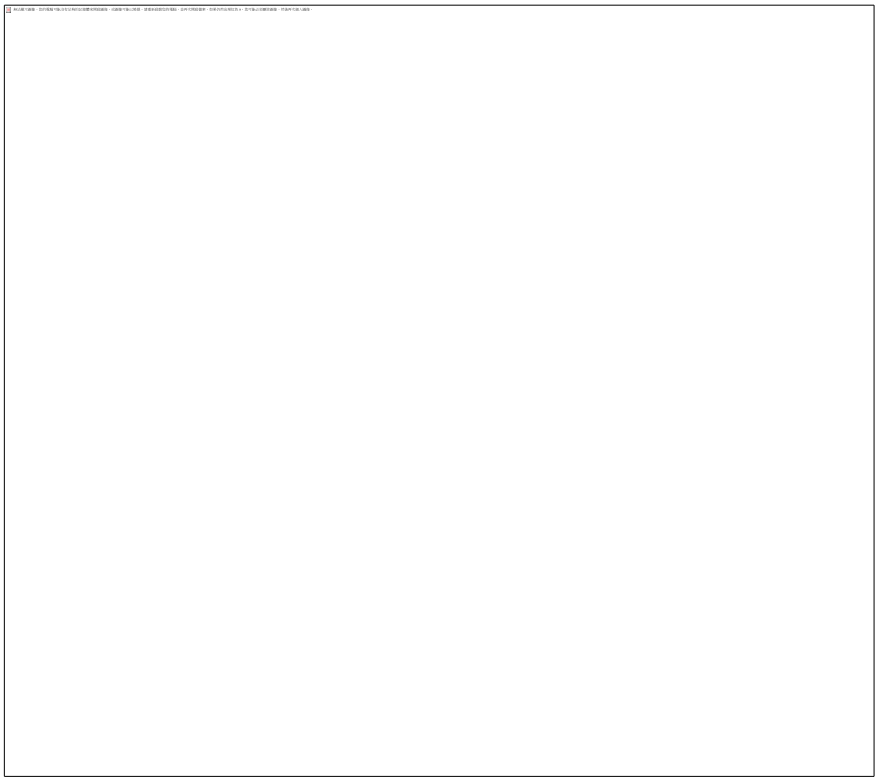
```
if save_file1 then

        for ii2 := 0 to 50 do (* 每次 scan time 只寫入50筆資料到檔案 *)

                if save_cnt1 < total_record1 then

                        str1 := '';

                        str1:= str1 + INT_str3(ARread(1,save_cnt1),5);

                        (* 加上 <CR> <LF> 字元在行尾 *)
                        str1 := str1 + '$0D$0A';

                        TMP := F_writ_s(File1,str1);
                        save_cnt1 := save_cnt1 + 1;
                else

                        (* 寫入檔案已經完成了 *)
                        save_file1 := False;
                        TMP := F_close(File1); (* 關閉檔案 *)
                        STEP1 := 2; (* 2: 記錄已經完成 *)
                        Msg1:= 'Record is finished! You may download the record file to your PC
now!';

                end_if;
        end_for;
end_if;
```

C# .net 程式

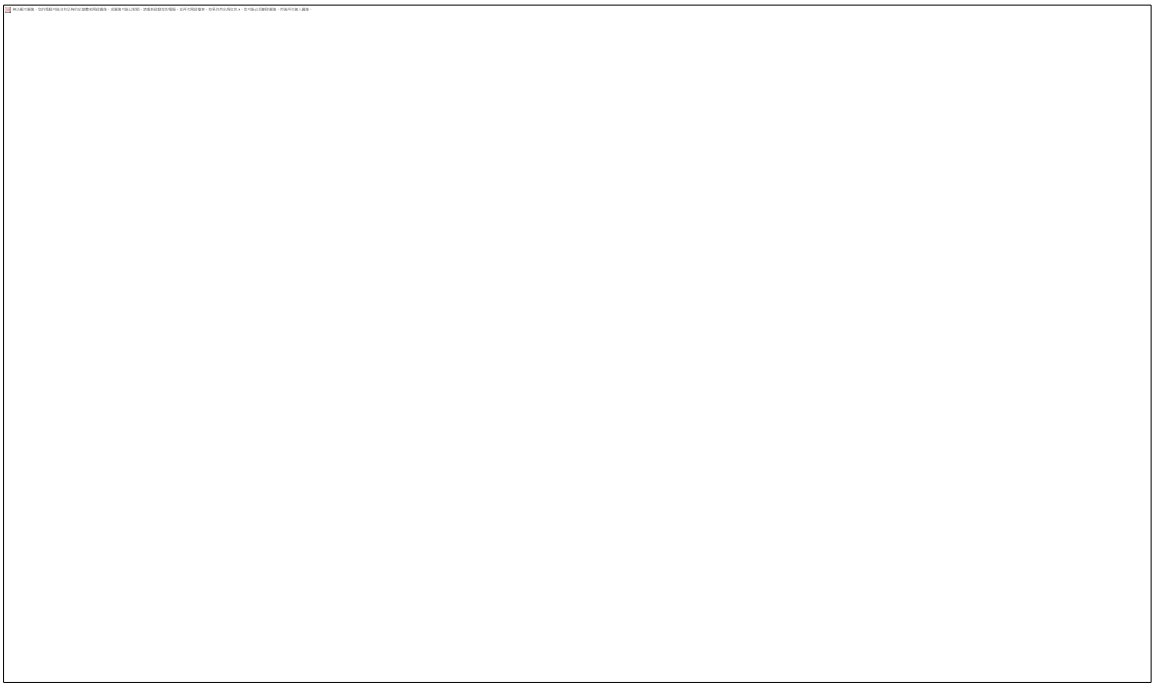
注意：這個程式是在 Virtual studio 2008 基於.net framework 2.0上開發的。因為 WP-8xx7只支持 ISaGRAF,VS .net 2008/2005/2003基於.net framework 2.0或 EVC++ 4.0

1.開啟一個新的專案

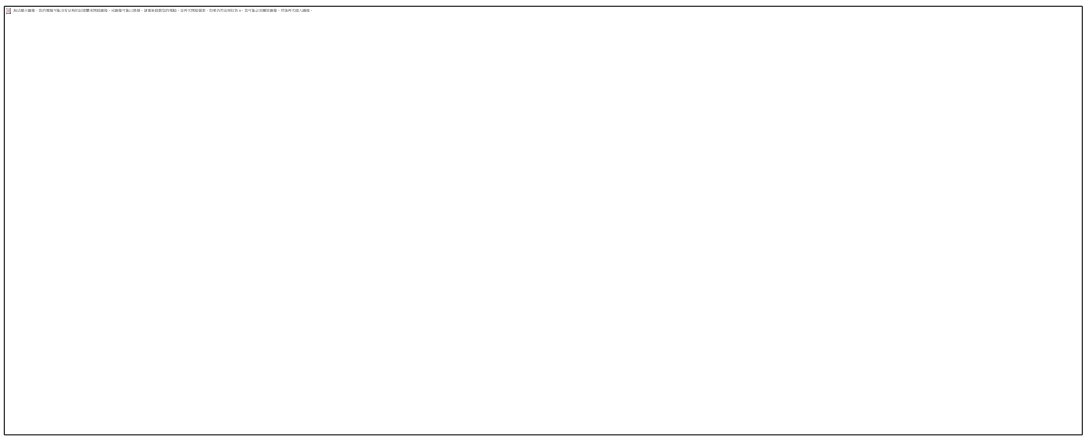
步驟1：



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



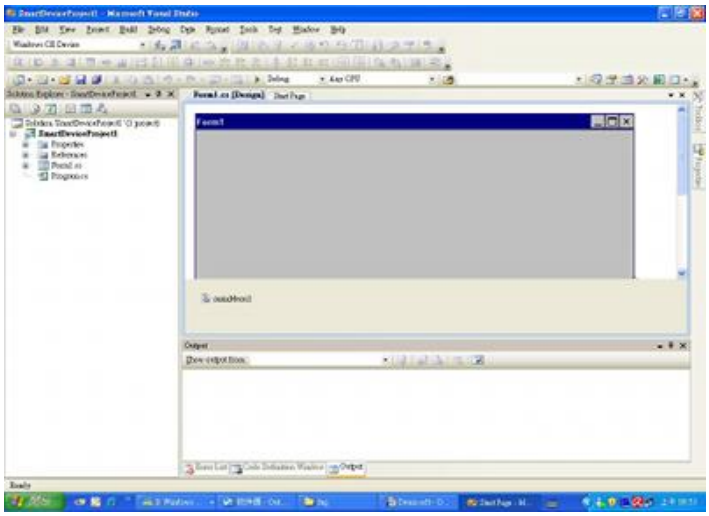
步驟3：選擇開發平台



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

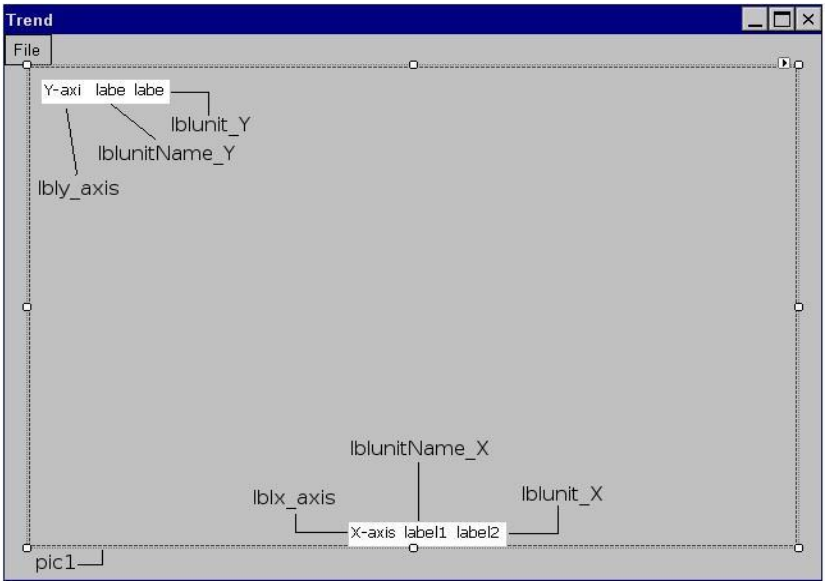


2.表單 FORM1控制項的屬性

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他
          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]=Int32.Parse(sr.ReadLine());
        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) / 300.0;

            //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((int)((double)origin.X + i / Scale_X), origin.Y - (int)((double)list[i]
/ Scale_Y));
            }

            //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.DrawLines(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_lable_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
   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;

   //計算出刻度間的間距
   int scale = range / 10;
```

```
            //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 = (scale * (i + 1)).ToString();

                //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();
            }
        }
    }
}
```