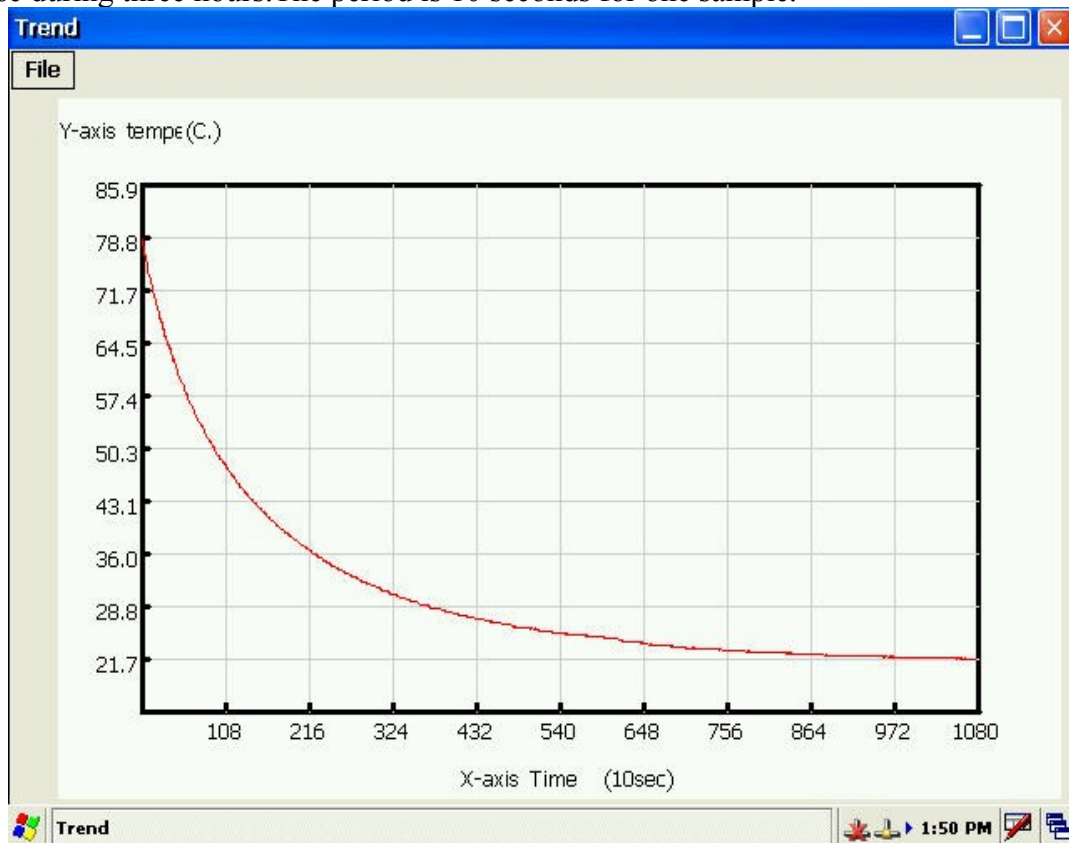

How to display the temperature trend curve by running ISaGRAF and C# .net 2008 program in the WinPAC-8xx7 plus i-87018z

By joseph_dun@icpdas.com

This paper shows you how to record the temperature input by ISaGRAF program in WinPAC-8xx7 + I-87018z, then draw a temperature trend curve with a C# .net program . This demo project measure and monitor the temperature of a glass of water in the office.

The below figure shows the temperature trend curve generated by measuring a glass of hot water in the house during three hours. The period is 10 seconds for one sample.

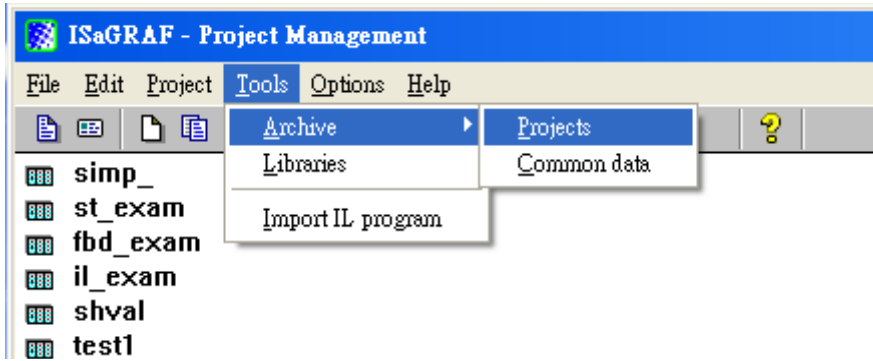


Please refer to the following website for more information about the WinPAC-8xx7 and I-87018z
WinPAC-8xx7→ <http://www.icpdas.com/products/PAC/winpac/wp-8x47.html>
I-87018z→http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm

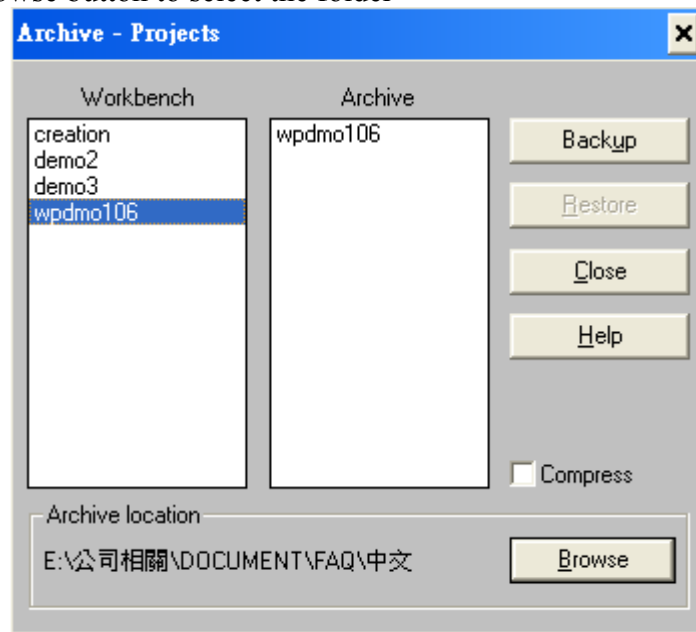
The ISaGRAF demo code “wpdmo108.pia” can be downloaded from
www.icpdas.com→FAQ→Software→ISaGRAF Ver.3→108

Please follow below steps to restore the “wpdmo108.pia” to your PC/ISaGRAF

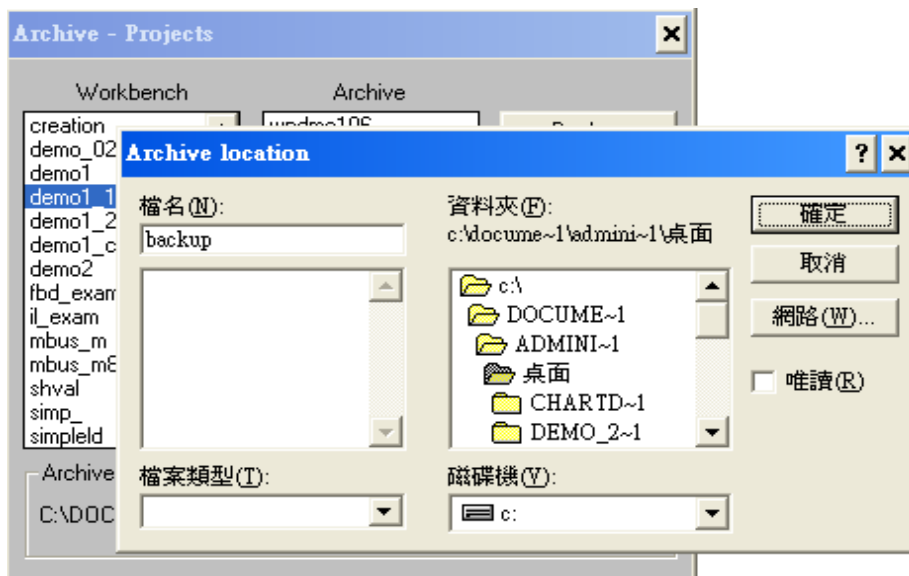
1.



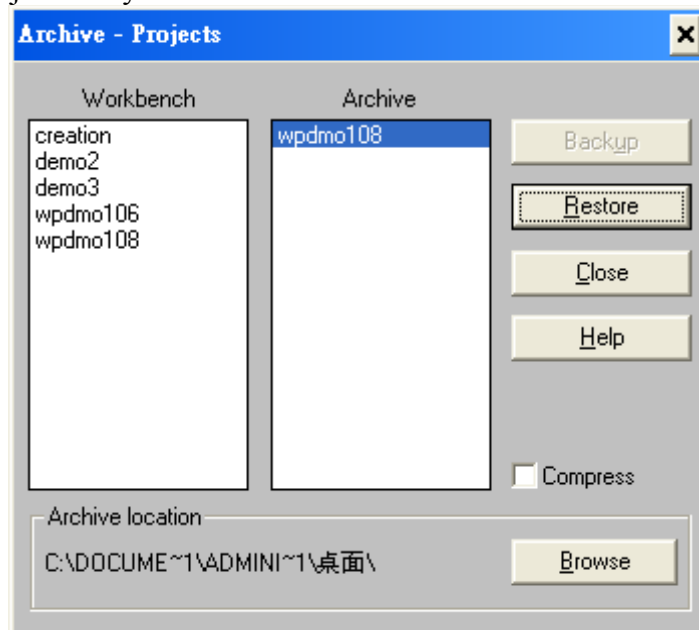
2. Click the Browse button to select the folder



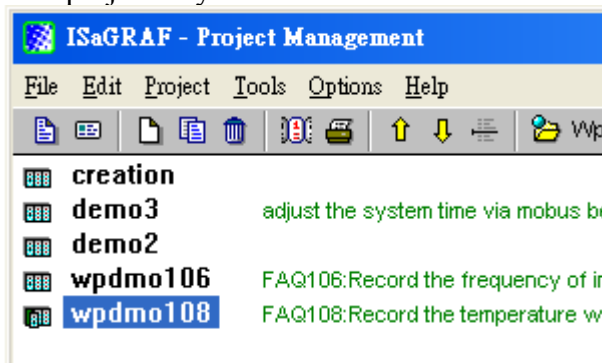
3. Select the folder that contain the “wpdmo108.pia” and click the OK button



4. Select the project that you want to restore



5. Then you will see the project in your ISaGRAF as below



We use “Variable array” in this demo program, please refer to Chapter 2.6 of the ISaGRAF user's manual or visit www.icpdas.com → FAQ → Software → IsaGARF Ver.3 → 39

The C# execution file, “Demo_2.exe”, and C# demo can be downloaded from www.icpdas.com → FAQ → Software → ISaGRAF Ver.3 → 108

Please download the “Demo_2.exe” to the WinPAC's “/System_Disk/” by ftp utility, and then double click on it to run it.

If you want to modify the C# demo code (make sure the Virtual Studio .net 2008 or 2005 has been installed in your PC.) You can double click on the file, “Trend(Demo_2).csproj”, in the folder, “C#source” to open it.

How to test this demo?

1. Please plug one I-87018z in the WP-8xx7's slot 2. Then connect the first channel of the I-87018z to the K-type thermocouple and put the other side of thermocouple in a hot water or what you want to measure.

2. Download the ISaGRAF project “wpdmo108.pia” to the WP-8xx7.

3. Modify the value of some variables in the below variable list.

Please enter proper “Interval1” value. The unit of Interval1 is 1 second. For example, if enter 20, it means to store one record per 20 seconds. The ”Period1” is the time of the period to record. The unit of Period1 is 1 minute. Then please change the value of GO1 to true to start the recording.

Name	Value	Comment
TEMP[0]	999990	
TEMP[1]	999990	
TEMP[2]	999990	
TEMP[3]	999990	
TEMP[4]	999990	
TEMP[5]	999990	
TEMP[6]	999990	
TEMP[7]	999990	
TEMP[8]	999990	
TEMP[9]	999990	
Period1	180	the time of run program
Msg1	no action	initial value is no action
T1_next	t#0s	when is the next time to record
Interval1	10	the interval between two recording time
T1	t#0s	the timer in the program
record_cnt1	0	the number of data that have been stored in User's r
total_record1	0	the number of total data that is generated by program
Go1	FALSE	run command if set true the program will start to run
INIT	FALSE	use for initializing in the first PLC scan
Stop1	FALSE	stop command if set true the program will stop runnin
<end of list>		

This figure shows “Period1” is 180 minutes,”Interval1” is 10 seconds.

The value of “Record_cnt1” will count up during the recording period. If it reaches the value of “Total_record1”, it means recording is finished. Then the ISaGRAF will store these records to a file at the “/System_Disk/” path automatically.

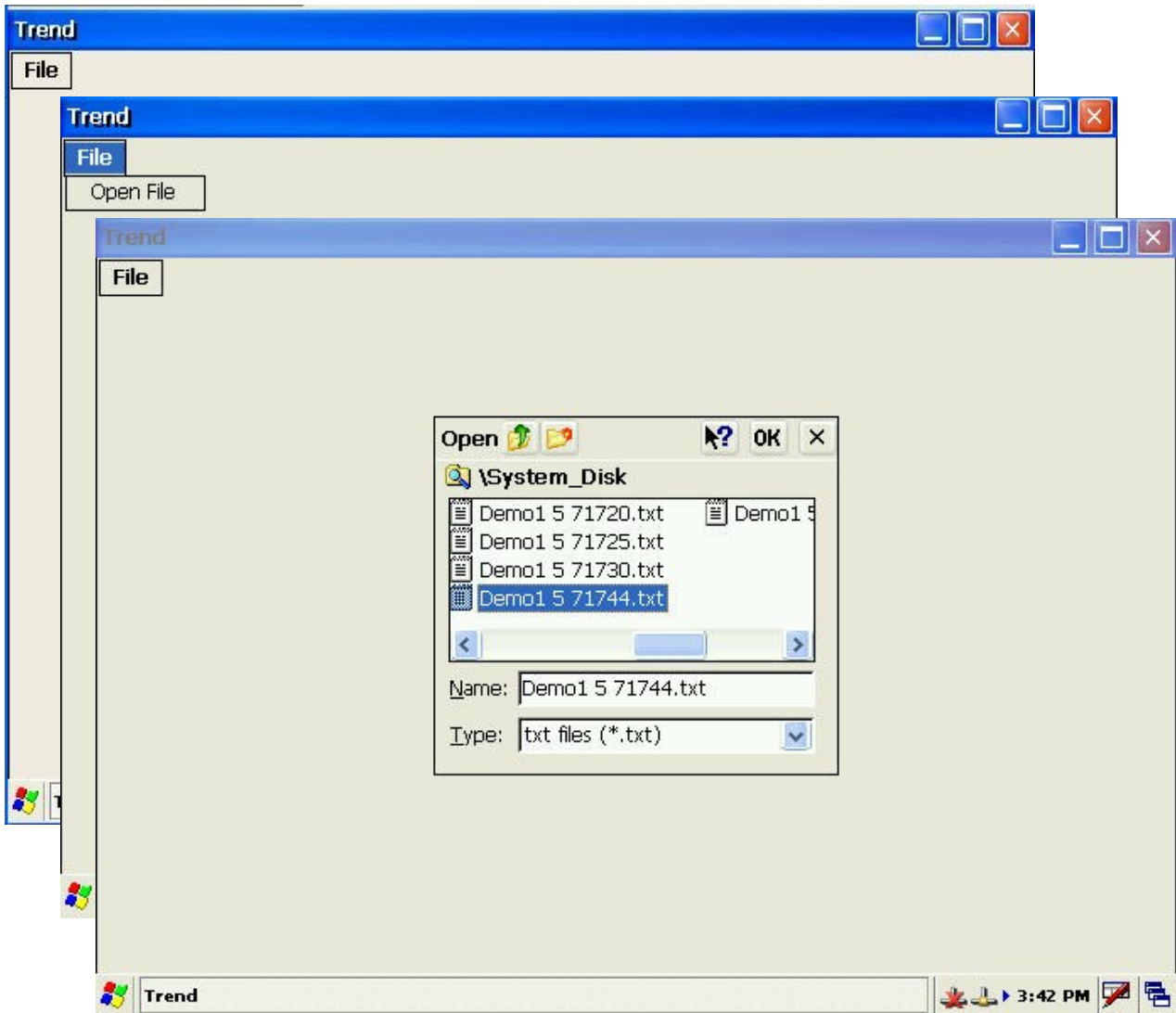
4. Open this record file,”temperature.txt” with C# program “Demo_2.exe” in the WP-8xx7. Then we will see a simple trend curve on the screen.

The format of “temperature.txt” :

Line	Content	Description
1	3000	第一行：檔案中資料的個數
2	Time	第二行：X軸的名稱
3	(20ms)	第三行：X軸的單位
4	freq	第四行：Y軸的名稱
5	(Hz)	第五行：Y軸的單位
6	100	以下每一行都放一筆資料
7	100	

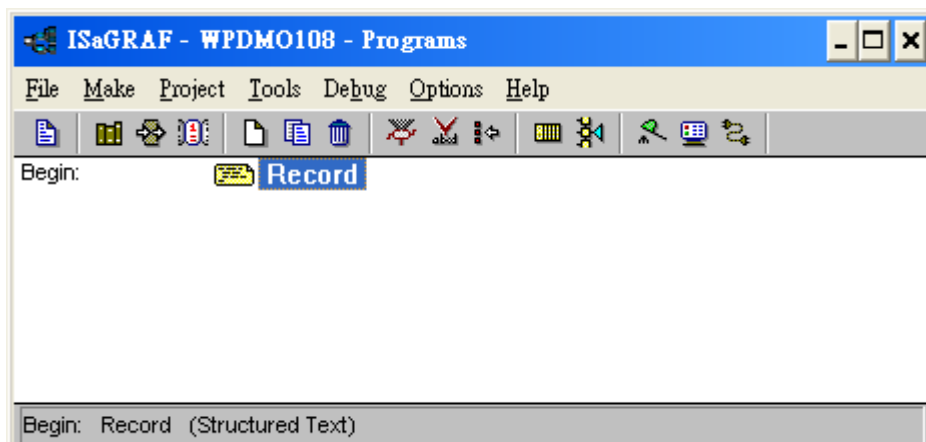
the first row : the number of data in the file
the second row : the name of X-axis
the third row : the unit of X-axis
the forth row : the name of Y-axis
the fifth row : the unit of Y-axis
others : one data per row

Please follows below steps:



The ISaGRAF program : wpdmo108.pia

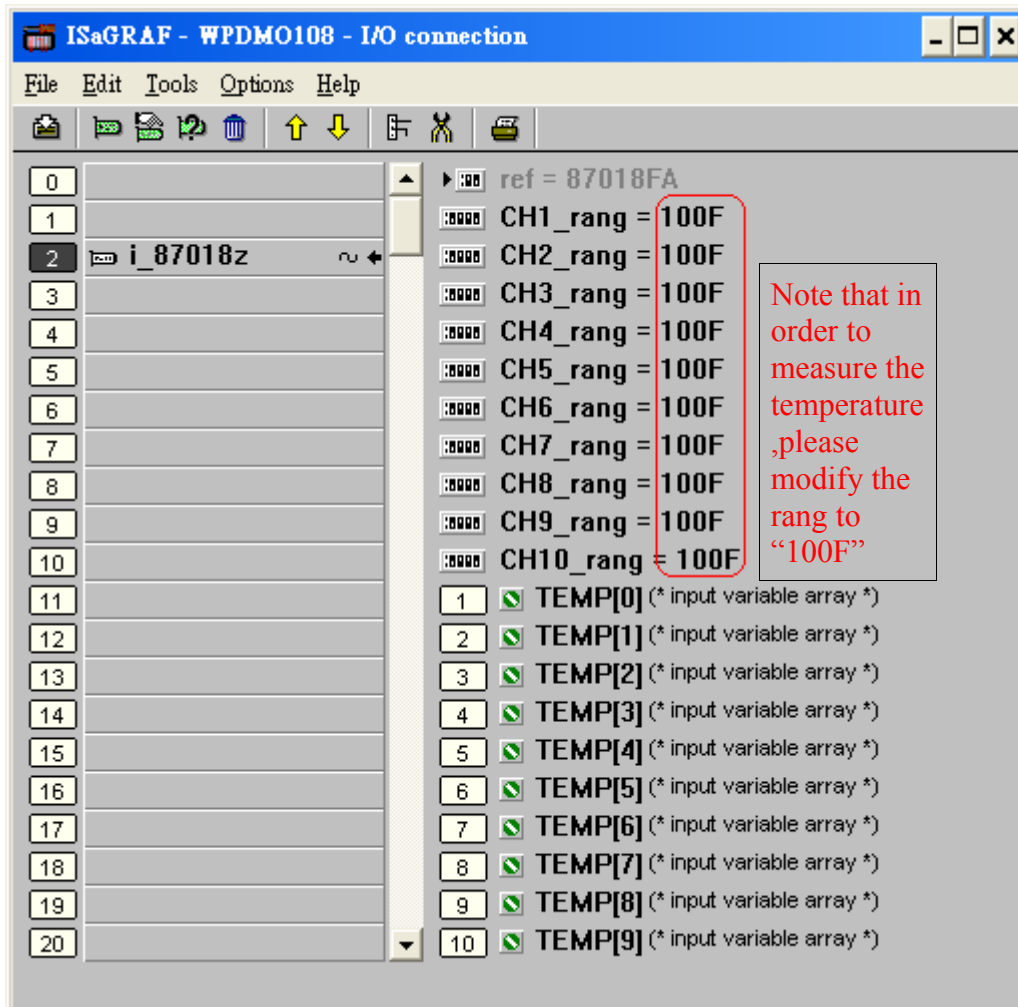
ISaGRAF project architecture : (include one ST program, ST1)



Variables :

Name	Type	Attribute	Description
Go1	Boolean	Internal	Set as True to start
INIT	Boolean	Internal	Init as true
Stop1	Boolean	Internal	IF set as True to stop
TMP	Boolean	Internal	Internal use
TEMP[0...9]	Integer	Input	Variable array,Dim as 10 to your signal
TMP_v	Integer	Internal	Internal use
STEP1	Integer	Internal	Recording state. 0:No action 1:recording, 2:finished
Interval1	Integer	Internal	How long to save a record? The unit is second.
Period1	Integer	Internal	How long to record? The unit is minute.
ii	Integer	Internal	Use in “for loop”
File1	Integer	Internal	File ID
total_record1	Integer	Internal	How many records in this recording action?This value is calculated by ISaGRAF program automatically.
record_cnt1	Integer	Internal	Current finished recored count.
T1	Timer	Internal	For counting time
T1_Interval	Timer	Internal	The interval time between two record
T1_next	Timer	Internal	The time to get and save next record
Msg1	Message	Internal	Operation state message,the lenth is 255,init as ”No Action now”.
File_name1	Message	Internal	File name, the lenth is 64 ,init as \System_Disk\.
str1	Message	Internal	The lenth is 255,internal use

IO connection :



ST program-Record

(* This example records the temperature with the wp-8xx7 and I-87018z to stored as string in a file.*)

(* the if block will only work at the first scan cycle *)

if INIT then

INIT := false;

(* set the filename *)

File_name1 := File_name1 + 'temperature.txt';

end_if;

(* If stop command is given, the following if block will be work *)

if Stop1 then

Stop1 := False;

Step1 := 0; (* 0:no action *)

TStop(T1); (* Stop T1 counting *)

T1 := T#0s; (* set timer to ZERO*)

Msg1 := 'User stop recording!';

```

end_if;

(* If Go1 is true,prepare to record the tmperature in the program*)
if Go1 then

    Go1 := False;
    if STEP1 = 1 then

        (* It is still recording *)
        Msg1 := 'It is still recording now... Please wait';
    else
        (* Check the value of interval,the unit is 1 second, is vaild or not *)
        (* We assume 1 to 60s is vaild in this example *)
        if (Intervall1 < 1) or (Intervall1 > 60) then
            Msg1 := 'Wrong Interval value, it should be in 1 to 60 second!';

            (*check Period1 value,the unit is one minute,is vaild*)
            (* We assume 1 to 300 minutes is valid in this example *)
        elseif (Period1 < 1) or (Period1 > 300) then
            Msg1 := 'Wrong Period value,it should be in 1 to 1440 minute!';
        else
            (* all of the parameter is correct, then start to record *)
            (* calculate the value of total_record1*)
            total_record1 := (Period1 * 60) / Intervall1;
            record_cnt1 := 0; (* reset current record count as 0 *)
            STEP1 := 1; (* 1:recording *)
            Msg1 := 'Recording now ... Please wait';

            (* start ticking T1 from 0 second *)
            T1 := T#0s;
            T1_Interval := TMR(Intervall1 * 1000);
            T1_next := T1 + T1_Interval;
            TStart(T1); (* ticking now *)

            (* create a new file *)
            File1 := F_creat(File_name1);

            (* creating file is faild *)
            if File1 = 0 then
                Msg1 := 'Create File' + File_name1 + 'Error!!!';
            else
                Msg1 := 'Create File' + File_name1 + 'Success!!!';

                (* write something to the begin of file *)
                (* this way is convenient to draw a map *)
                str1 := "";
                (* the fist line write the number of record in this file *)
                str1 := INT_str3(Total_record1,5) + '$0D$0A';
                (* the second line write the name of X-axis *)
                str1 := str1 + 'Time' + '$0D$0A';
            end if;
        end if;
    end if;
end if;

```



```

        (* the third line write the unit of X-axis *)
        str1 := str1 + '(' + INT_str3(Interval1, 2) + 'sec'+ ')' + '$0D$0A';
        (* the forth line write the name of Y-axis *)
        str1 := str1 + 'temperature' + '$0D$0A';
        (* the fifth line write the unit of Y-axis *)
        str1 := str1 + '(C.)' + '$0D$0A';
        TMP := F_writ_s(File1,str1);
        TMP := F_close(File1); (* close file *)
    end_if;
end_if;
end_if;
end_if;

(* in the recording state *)
if STEP1 = 1 then

    (* prepare to record one information,when T1 >= T1_next *)
    if T1 >= T1_next then
        (* renew the value of T1_next for next time record*)
        T1_next := T1_next + T1_Interval;

        (* Open the file we created before )
        File1 := F_wopen(File_name1);

        (* Check open file function is correct? *)
        if File1 = 0 then
            Msg1 := 'Open File' + File_name1 + 'Error!!!';
            STEP1 := 0; (* 1: no action *)
            TStop(T1); (* stop counting *);
            T1 := T#0s; (* set the t1 to zero *)
        else
            (* if open file is correct, do as follow *)
            (* seek the file point to the end of file *)
            TMP := F_end(File1);
            (* write a data in the end of file *)
            TMP := FM_write(File1,Rea_Str2(REAL(TEMP[0]) / 100.0,1));

            (*close file*)
            TMP := F_close(File1);

            (* record_cnt1 plus one when a record that have been recorded *)
            record_cnt1 := record_cnt1 + 1;
            (* check all of data have been record or not *)
            if (record_cnt1 >= total_record1) then
                STEP1 := 0; (* set step1 to 0 : no action *)
                Tstop(T1); (* stop counting T1 *)
            end_if;
        end_if;
    end_if;
end_if;
end_if;
end_if;

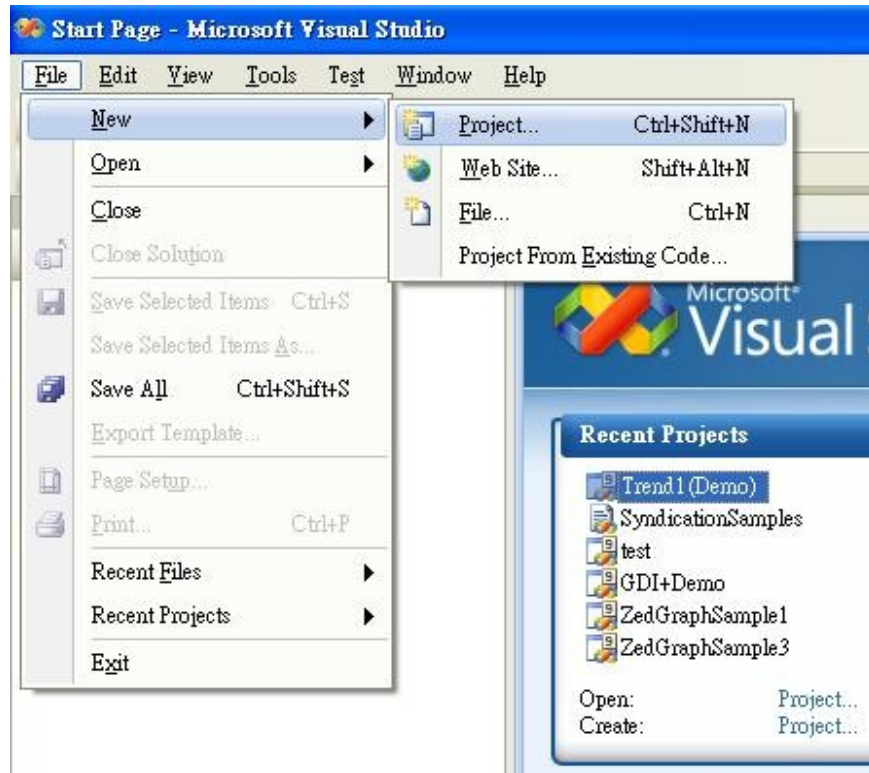
```

C# .net program

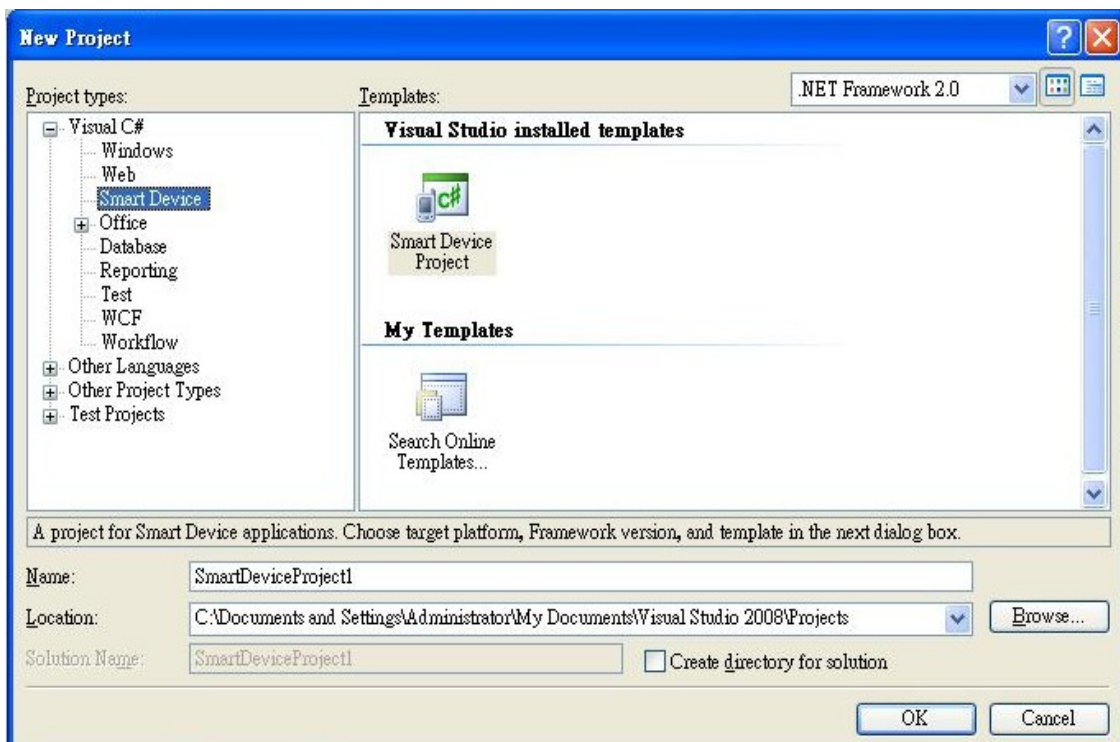
Note that we develop this C# .net program with Visual studio 2008 based on .net framework 2.0. So if you don't have V.S.2008,you can develop it on your IDE,like V.S.2005,or that support the library of .net framework 2.0.

1.Create a new project

Step1 :



STEP2 :



STEP3 : select platform



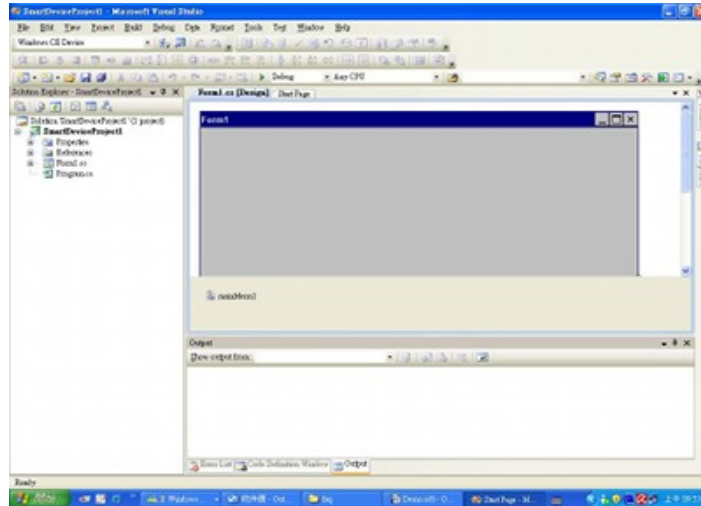
STEP4 : select .net Compact Framework version



STEP5 : select the type of Template



Step6 : Click OK button then you will see as follows

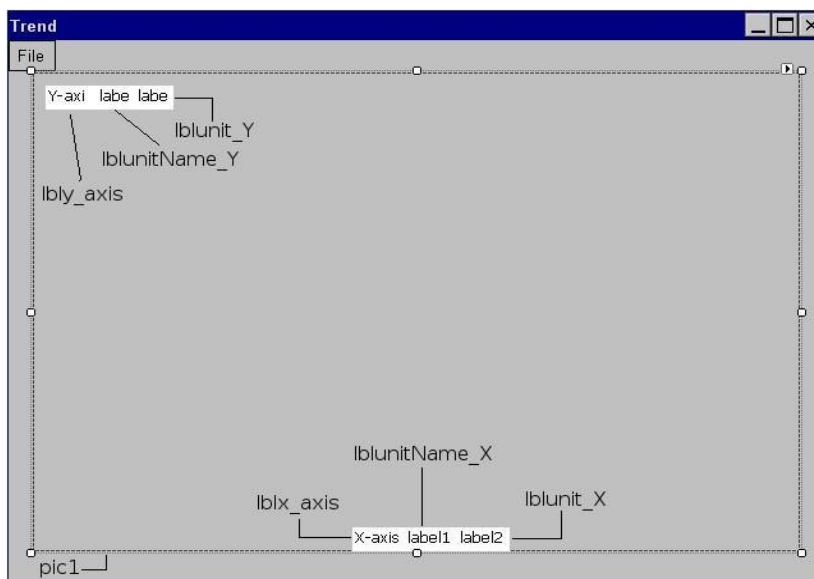


2.The controls of form1

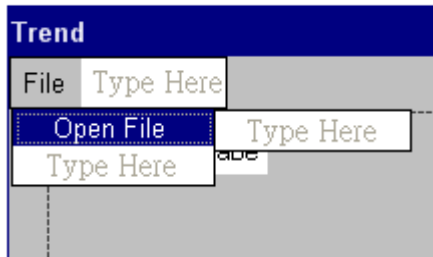
table of the properties of Controls:

	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	

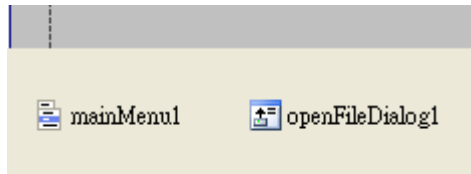
The position diagram of control item :



Create the child of menu,menuItem,its text is open_file.



Create a openFileDialog of the invisible control



3. Write codes in the file,form1.cs as follows

```
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
    {
        //set global variable
        string filename;
        FileInfo fileInfo;
        FileStream fs;
        StreamReader sr;
        int[] list;

        //set the point of origin
        Point origin = new Point(50, 350);

        Bitmap img;
```

Graphics g;

//form designer generated code

public form1()

{

InitializeComponent();

}

//the event handler when the menuItem2 of mainmenu is clicked

private void menuItem2_Click(object sender, EventArgs e)

{

//Check the result of openFileDialog is OK or not

if (openFileDialog1.ShowDialog() == DialogResult.OK)

 //if it is ok , then assign the FileName of openFileDialog to the filename of global variables

 filename = openFileDialog1.FileName;

else

 ; //if is cancel or close or abort ,do nothing because this project is simple

//check the file is exist

if (File.Exists(filename))

{

 //if check is exist then do as follow

 // declare a variable of string to get a full path of the file

 string fPath = Path.GetFullPath(filename);

 // create a instance of fileInfo and assign it to fileInfo of global variables

 fileInfo = new FileInfo(fPath);

 // use fileInfo to open the file and assign the file operator to the fs of global variables

 fs = fileInfo.Open(FileMode.Open);

 // create a instance of filestream and assigned it to the sr of global variables

 sr = new StreamReader(fs, System.Text.Encoding.Default);

 if (fs.Length > 0)

 {

 //read the first line to get the number of data in file ,here we use the method

 sr.ReadLine();

 int Lenth = Int32.Parse(sr.ReadLine());

 //read the second line to get the unit-name of X-axis, assign it to the lblunitname_x of

 label

```

//, we will see it on the map
lblunitName_X.Text = sr.ReadLine();

//read the third line to get the unit of X-axis
lblunit_X.Text = sr.ReadLine();

//read the forth line to get the name of unit of Y-axis
lblunitName_Y.Text = sr.ReadLine();

//read the fifth line to get the unit of Y-axis
lblunit_Y.Text = sr.ReadLine();

//create an array to store the data in file
list = new int[Lenth];

//store the data in order
int i=0;
do
{
    list[i]=(int)(float.Parse(sr.ReadLine()) * 10);
    i++;
} while (sr.Peek() != -1);
}

//close the file when finish reading data
fs.Close();

//Create a instance of Bitmap which size is the same as pic1
img = new Bitmap(pic1.Size.Width, pic1.Size.Height);

//create a instance of pen which color is black and width is 3
Pen p_cod = new Pen(Color.Black, 3);

//create a instance of graphics which image is bitmap that we created before
g = Graphics.FromImage(img);

//clear the graphics with white color
g.Clear(Color.White);

//let the image of pic1 to refer to img
pic1.Image = img;

//draw the coordinate as follow

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

```

```

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

//Draw the top line of coordinates
g.DrawLine(p_cod, origin.X - 1, origin.Y - 300, origin.X + 500 + 2, origin.Y - 300);

//Draw the right line of coordinates
g.DrawLine(p_cod, origin.X+500, origin.Y + 1, origin.X + 500, origin.Y - 300 - 1);

//create pen with LightGray color for drawing grid
Pen pen_grid =new Pen(Color.LightGray,1);

//Draw grid on the coordinates

//draw vertical line
for (int i = 0; i < 9; i++)
{
origin.Y - 300);
    g.DrawLine(pen_grid, origin.X + 50 * (i + 1), origin.Y, origin.X + 50 * (i + 1),
}

//draw horizontal line
for (int i = 0; i < 9; i++)
{
(i + 1));
    g.DrawLine(pen_grid, origin.X, origin.Y - 30 * (i + 1), origin.X + 500, origin.Y - 30 *
}

//Draw the scale of X-axis
for (int i = 0; i < 9; i++)
{
5);
    g.DrawLine(p_cod, origin.X + 50 * (i + 1), origin.Y, origin.X + 50 * (i + 1), origin.Y -
}

//Draw the scale of Y-axis
for (int i = 0; i < 9; i++)
{
1));
    g.DrawLine(p_cod, origin.X, origin.Y - 30 * (i + 1), origin.X + 5, origin.Y - 30 * (i +
}

//repaint the pic1
pic1.Refresh();

```



```

//set all of these label is visible
lblx_axis.Visible = true;
lbly_axis.Visible = true;
lblunitName_X.Visible = true;
lblunitName_Y.Visible = true;
lblunit_X.Visible = true;
lblunit_Y.Visible = true;

//get the Max and the Min in the Data
int List_Max=get_max(list,list.Length);
int List_Min=get_min(list,list.Length);

//Calculate the Scale of X-axis and Y-axis
double Scale_X = (double)list.Length / 500.0;
double Scale_Y = (double)(List_Max - List_Min) / 240;

//set the labels of Y-axis
set_label_Y(List_Max,List_Min);

//set the labels of X-axis
set_lable_X(list.Length);

//Declare the Data array of point to store the translated Data
Point[] Data = new Point[list.Length];

//translate the data to the coordinates and store it with an array
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);
}

//Draw the Data to the DrawMap

//create a instance of pen with Red color
Pen line = new Pen(Color.Red, 1);

//draw lines which connected point to point
g.DrawLine(line,Data);

//repaint the pic1
pic1.Refresh();
}
}

```

```

private void Form1_Load(object sender, EventArgs e)

```

```
{  
}
```

```
//the method to find the maximum in the int array
```

```
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;  
}
```

```
//the method to find the minimum in the int array
```

```
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;  
}
```

```
//the method to create labels of X-axis dynamiclly
```

```
private void set_lable_X(int number_X)
```

```
{  
    //set the scale  
    int scale = number_X / 10;  
  
    //create an array of reference  
    Label[] label_X = new Label[10];  
  
    //create an instance of label and initailize it  
    for (int i = 0; i < label_X.Length; i++)  
    {  
        //create an instance of label  
        label_X[i] = new Label();  
  
        //set the location  
        label_X[i].Left = pic1.Location.X + origin.X + 50 * (i + 1) - 25;  
        label_X[i].Top = pic1.Location.Y + origin.Y + 3;  
  
        //set the size of label
```

```

label_X[i].Size = new System.Drawing.Size(49, 20);

//set the bgcolor with black
label_X[i].BackColor = Color.White;

//set the text of label
label_X[i].Text = (scale * (i + 1)).ToString();

//set contentAligment
label_X[i].TextAlign = ContentAlignment.TopCenter;

//set the visible
label_X[i].Visible = true;

//add control to form1
this.Controls.Add(label_X[i]);

//use the method of bringtofront to ensure the label is visible
label_X[i].BringToFront();
}
}

//the method to create labels of Y-axis
private void set_label_Y(int Max,int Min)
{
//calculate the value of range
int range = Max-Min;

//calculate the value of scale
double scale = (double)range / 8.0;

//create an array of reference
Label[] label_Y = new Label[10];

//create an instance of label and initailize it
for (int i = 0; i < label_Y.Length; i++)
{
//create an instance of label
label_Y[i] = new Label();

//set the location of label
label_Y[i].Left = pic1.Location.X;
label_Y[i].Top = pic1.Location.Y + origin.Y - 30 * (i + 1) - 5;
}
}

```

```
//set the size of label
label_Y[i].Size = new System.Drawing.Size(49, 20);

//set bgcolor with white color
label_Y[i].BackColor = Color.White;

//set text of label
label_Y[i].Text = System.String.Format("{0:F1}", (double)(Min + scale * i) / 10.0);

//set alignment of label
label_Y[i].TextAlign = ContentAlignment.TopRight;

//set lable if is visible
label_Y[i].Visible = true;

//add a control to form1
this.Controls.Add(label_Y[i]);

//use the method of bringtofront to ensure the label is visible
label_Y[i].BringToFront();
    }
}
}
```