

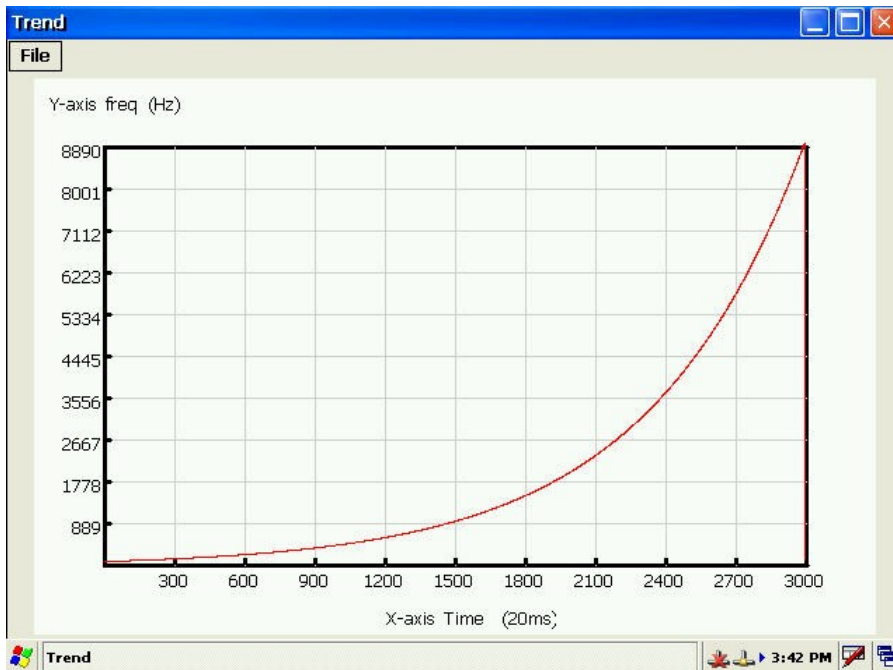
---

## How to display the frequency trend curve by running ISaGRAF and C# .net 2008 program in the WinPAC-8xx7 plus I-8084W?

---

By [joseph\\_dun@icpdas.com](mailto:joseph_dun@icpdas.com)

This paper shows you how to record the frequency input by running ISaGRAF program in WinPAC-8xx7 +I-8084W, then draw a frequency trend curve with a C# .net program. This application can measure and monitor the rotation speed of motors or engines.



This figure shows the frequency curve generated by one function generator (Agilent 33220A). The curve is a log curve with 5V square signal in one minute.

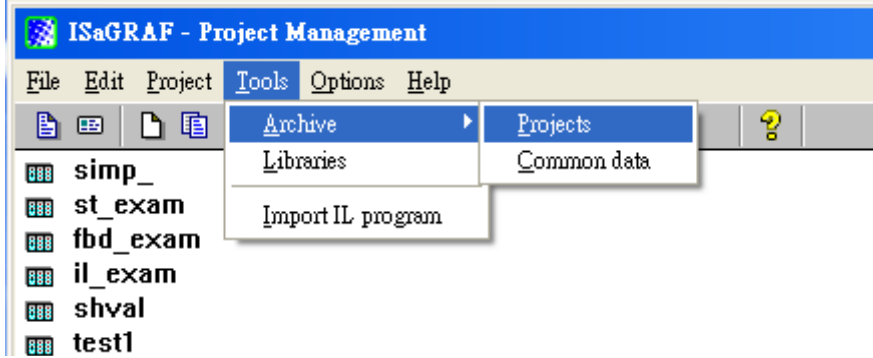
Please refer to the following website for more information about I-8084W.  
[www.icpdas.com](http://www.icpdas.com) → FAQ → Software → ISaGRAF Ver.3(English) → 100

More other information about I-8084W or WinPAC-8xx7  
WinPAC-8xx7 → [www.icpdas.com/products/PAC/winpac/wp-8x47.html](http://www.icpdas.com/products/PAC/winpac/wp-8x47.html)  
I-8084W → [http://www.icpdas.com/products/Remote\\_IO/i-8ke/i-8084w.htm](http://www.icpdas.com/products/Remote_IO/i-8ke/i-8084w.htm)

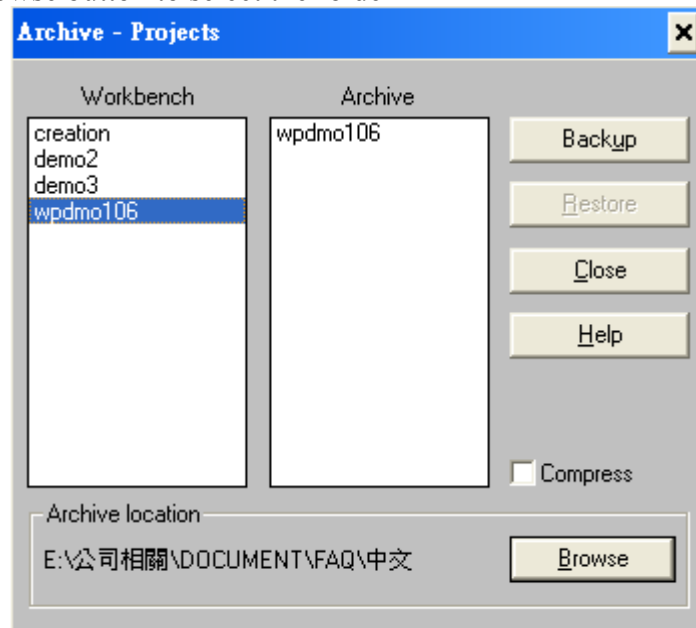
The demo code “wpmo106.pia” can be downloaded from  
[www.icpdas.com](http://www.icpdas.com) → FAQ → Software → ISaGRAF Ver.3(English) → 106

Please follow below steps to restore the “wpdmo106.pia” to yourPC/ ISaGRAF

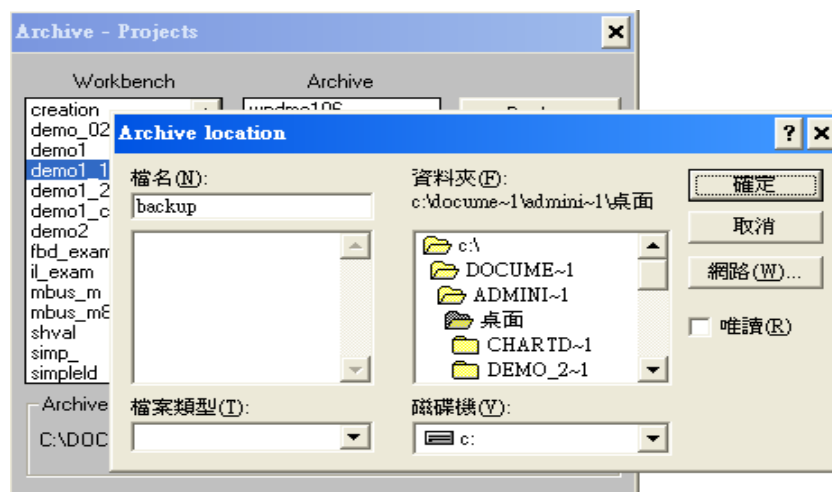
1.



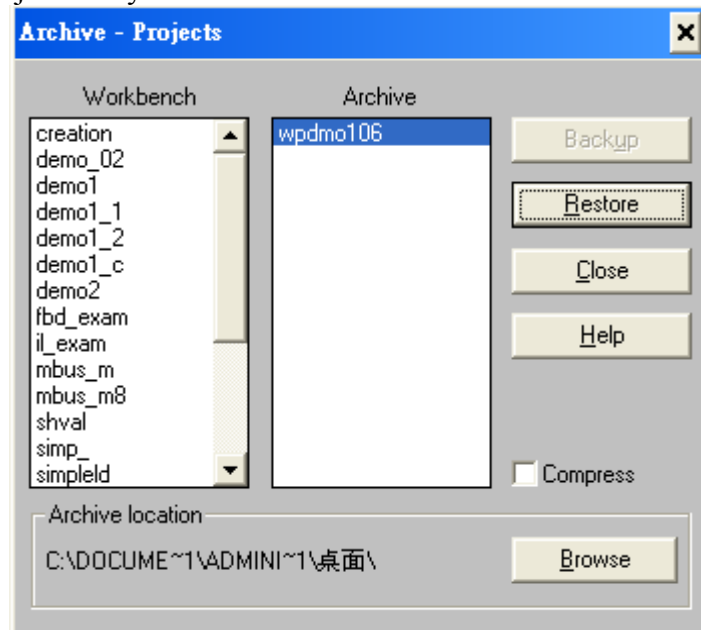
2. Click the Browse button to select the folder



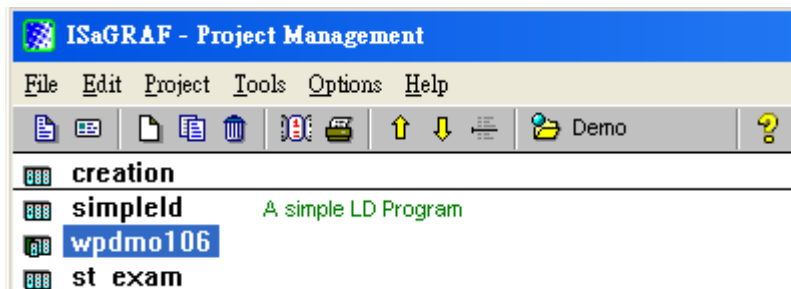
3. select the folder that contain the “wpdmo106.pia” and click the OK button



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](http://www.icpdas.com) → FAQ → Software → IsaGARF Ver.3(English) → 39

The C# execution file, ”Demo\_1.exe”, and C# demo code can be downloaded from [www.icpdas.com](http://www.icpdas.com) → FAQ → Software → ISaGRAF Ver.3(English) → 106

Please download the “Demo\_1.exe” to the WinPAC's “/System\_Disk/” by ftp utility, and then double click on it to run.

If you want to modify the C# demo code (make sure the Visual Studio .net 2008 or 2005 has been installed in your PC). You can double click on the file, ”Trend1(Demo).csproj”, in the folder, ”C#source” to open it.

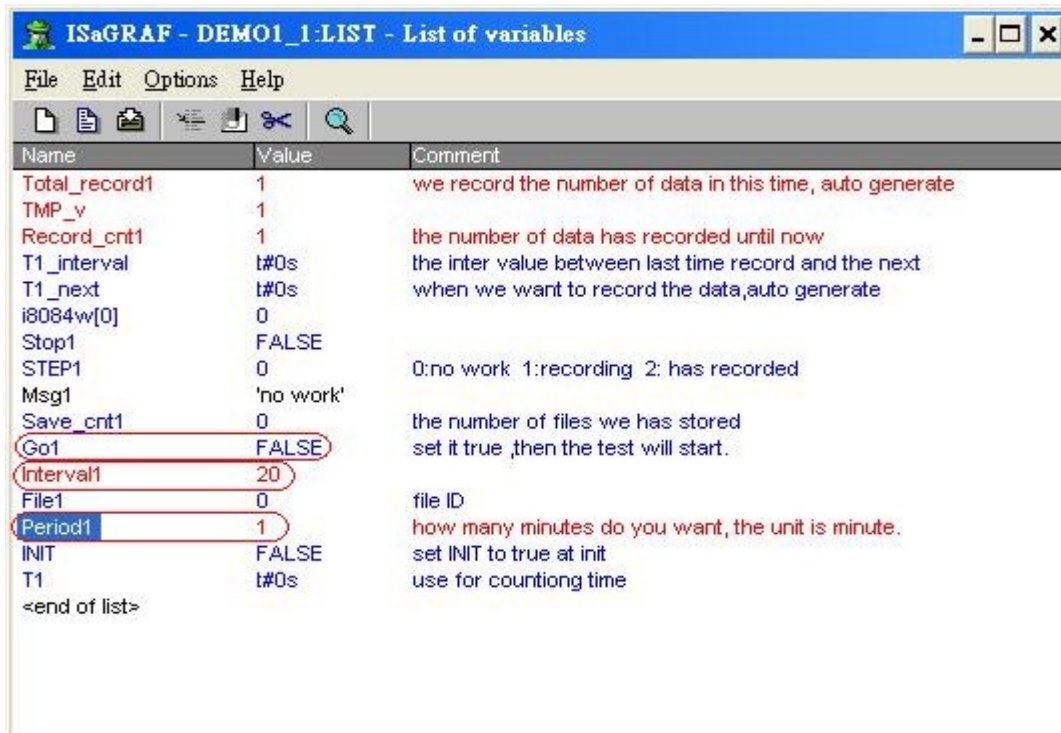
## How to test this demo?

1:Please plug one I-8084W in the WP-8XX7's slot 0.Then connect the CH.1 frequency input of the I-8084W to your frequency generation equipment(like function generator 、 motor) ,here we use the Agilent 33220A function waveform generator.

2:Download the ISaGRAF project “wpdmo106” to the WinPAC-8xx7.

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

Please enter proper “Interval1” value. The unit of Interval1 is 0.001 second(1 millisecond).For example, if enter 20,it means to store one record per 20ms. The “period1” is the time period to record. The unit of Period1 is minute. Then please change the value of GO1 to true to start the recording.



This figure shows “Period1” is 1 minute, “Interval” is 20ms.

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.

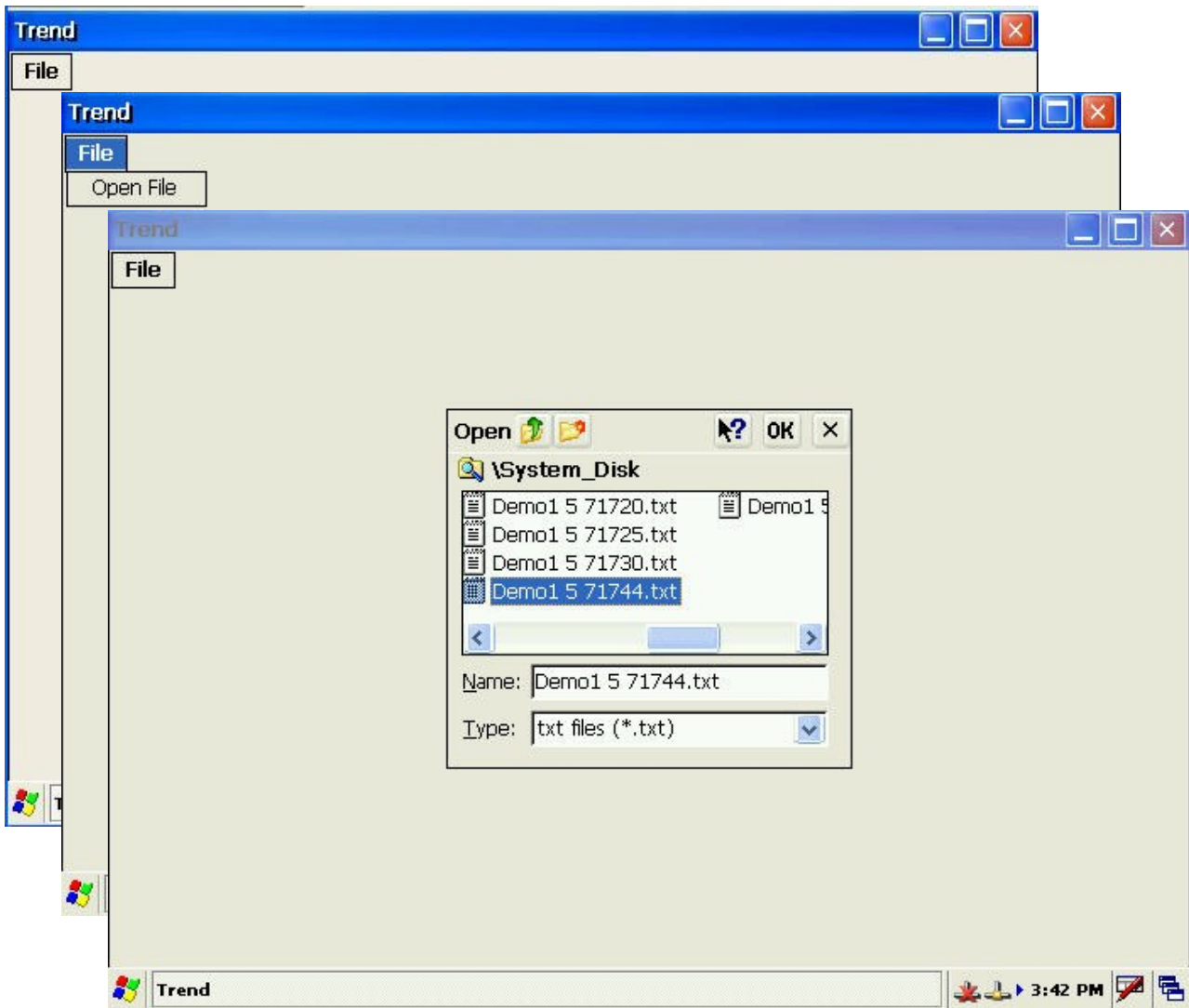
4.Open this record file,”demo1\*.txt” with C# program “Demo\_1.exe”in the WP-8447.Then we will see a simple trend curve on the screen.

The format of “Demo1\*.txt”:



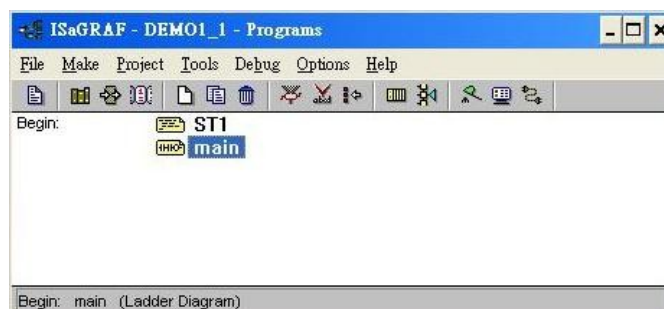
the first row : the number of data in the file  
the second row: the name of X-axis  
the third row : the unit of X-axis  
the forth row : the name of Y-axis  
the fifth row : the unit of Y-axis  
others: one data per row

Please follows below steps :



## The ISaGRAF program: wpdmo106.pia

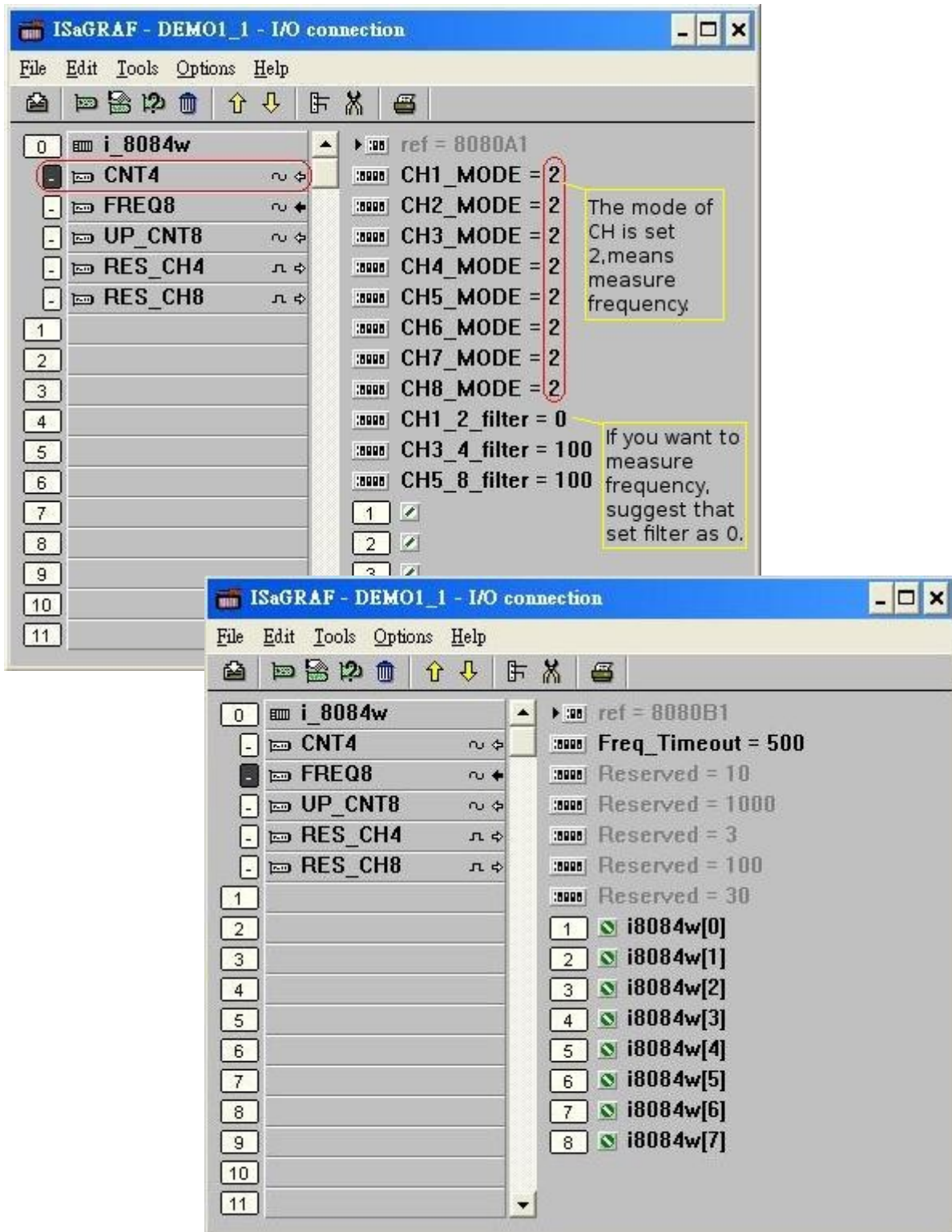
IsaGRAF project architecture:(include one ST program,ST1,and QLD program,main.)



Variables:

Name	Type	Attribute	Description
Go1	Boolean	<b>Internal</b>	Set as True to start
Stop1	Boolean	<b>Internal</b>	Set as True to stop
TMP	Boolean	<b>Internal</b>	Internal use
INIT	Boolean	<b>Internal</b>	Init as true
Save_file1	Boolean	<b>Internal</b>	The IsaGRAF program will set this value to True to store records to a RAM Disk File
File1	Integer	<b>Internal</b>	File ID
STEP1	Integer	<b>Internal</b>	Recording state. 0:No action,1:recording,2:finished
Period1	Integer	<b>Internal</b>	How long to record? Unit is minute.
Intervall1	Integer	<b>Internal</b>	How long to save a record? Unit is million-second
Total_record1	Integer	<b>Internal</b>	How many records in this recording action? This value is calculated by the ISaGRAF program automatically.
Record_cnt1	Integer	<b>Internal</b>	Current finished record count.
ii2	Integer	<b>Internal</b>	Use in “for” loop.
I8084W[0..7]	Integer	<b>Input</b>	Variable array,Dim as 8. link Ch.1 to you signal
Save_cnt1	Integer	<b>Internal</b>	Current saving record amount in the RAM disk File
TMP_v	Integer	<b>Internal</b>	Internal use
T1	Timer	<b>Internal</b>	For counting time
T1_next	Timer	<b>Internal</b>	The time to get and save next record
T1_interval	Timer	<b>Internal</b>	The interval time between two record
File_name1	Message	<b>Internal</b>	File name, Length is 64,init as \System_Disk\Demo1. We will add some meaning numbers,like date and time,to the file_name1. It is helpful to identify different testing.
File_name_t	Message	<b>Internal</b>	The temp of file name that generated by program
Msg1	Message	<b>Internal</b>	Operation state message, the length is 255,init as “No Action now”.
Str1	Message	<b>Internal</b>	The Length is 255,internal use

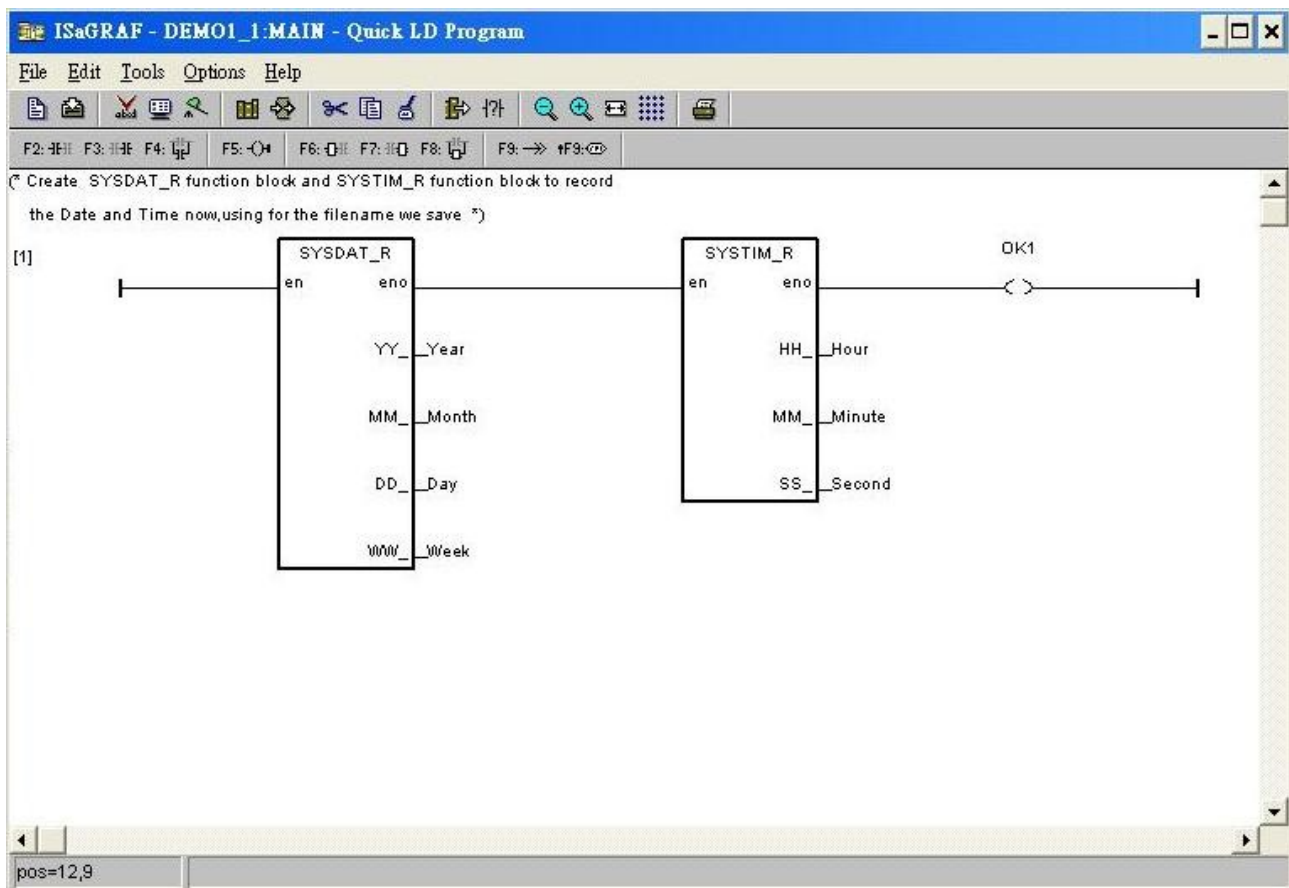
IO connection:



## Quick LD Program-main

---

This QLD program is very simple. It only use for the time when we save file.





## ST program-ST1

---

(\* This example records the frequency of signal that is generated by the function waveform generator,Agilent 33220A to a User created memory by “Arcreate” function in Winpac and store these record as string in a Ram disk when it is finished.

File operation in '\System\_disk\' is very slow. However Read / Write operation in user create memory is vary fast. So we can store the data in our created memory and copy the data to a file when finish recording ,if we want this demo can record the data in the high speed (ex: record a datum per 10 million-second).\*)

(\*WP-8447 can have max. speed of 100Hz to record data(minimum sample interval is 10ms)\*)

(\*1 minute will record  $50 \times 1 \times 60 = 3,000$  integer value\*)

(\* If period is set as 10 minutes,we will need  $3000 \times 10 = 30,000$  integer value,memory =  $30,000 \times 4 = 120,000$ bytes\*)

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

**if INIT then**

**INIT := false;**

(\*Allocate 30,000 integer (or 32-bit REAL) space to store records up to 10 minutes. total size =  $30,000 \times 4 = 120,000$  bytes \*)

(\* The first parameter of the function ONLY should be 1 \*)

(\* the ARcreate function can be called ONLY ONCE in the ISaGRAF program \*)

**TMP\_v := ARcreate(1,30000);**

**if TMP\_v <> 1 then**

**Msg1 := 'Parameter error or Can not allocate enough memory by ARcreate()**

**function!';**

**end\_if;**

**TMP := PLC\_mode(-1);** (\*Set WP-8447 ISaGRAF driver running at fastest mode \*)

**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 the timer\*)

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

**Msg1 := 'User stop recording!';**

**save\_cnt1 := 0;**

**end\_if;**

(\* If start command is given \*)

**if Go1 then**

**Go1 := False;**

```

if STEP1=1 then (* 0:no action, 1:recording, 2:recored *)
    (* it is still recording now *)
    Msg1 := 'It is still recording now... Please wait';
else
    (* Check the value of interval is valid or not *)
    (* we assume 10 to 10000 ms is valid in this example *)
    (* If your average PLC scan time is larger, for example,near 10ms,
    Please set the value of interval larger than 10ms.
    Or the record time won't be correct *)
    if (Interval1 < 10) or (Interval1 > 10000) then
        Msg1 := 'Wrong Interval value, it should be in 10 to 10000 milli-second!';
    (* Checked the value of period is valid or not *)
    (* we assume 1 to 10 minute is valid in this example *)
    elsif (Period1 < 1) or (Period1 >10) then
        Msg1 := 'Wrong Period value, it should be in 1 to 10 minute!';

    else
        (* all of the parameter is correct,then start to record *)

        (* calculate total record number *)
        total_record1 := (Period1 * 60000) / Interval1;
        record_cnt1 := 0; (* reset current record count as 0 *)
        STEP1 := 1; (* set step as 1: reacording *)
        Msg1 := 'Recording now ... Please wait';

        (* start ticking T1 from 0 second *)
        T1 := T#0s;
        T1_Interval := TMR(Interval1);
        T1_next := T1 + T1_Interval;
        TStart(T1); (* ticking now *)
        save_cnt1 := 0;

    end_if;
end_if;
end_if;

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

    (* recording one datum *)
    if T1 >= T1_next then

        (* prepare to record the information when T1 > T1_next *)
        (* renew T1_next for next time to record*)
        T1_next := T1_next + T1_Interval;

        (* prepare to write the information to User's RAM *)
        (* store it by ARwrite() function *)
        (* the first parameter of the function ONLY should be 1 *)
        (* the second parameter of the function is the position
        in the array that you store the datum *)

```

(\* the third parameter of the function is the datum that you want to store \*)

**TMP\_v := ARwrite(1,record\_cnt1,i8084w[0]);**

(\* check if ARwrite() is correct or not \*)

**if TMP\_v <> 1 then**

**Msg1 := 'Can not operate ARwrite(!';**

**STEP1 := 0; (\* 0:no action \*)**

**TStop(T1); (\* stop counting to T1 \*)**

**T1 := T#0s;**

**end\_if;**

(\*Check if record number reach the end\*)

(\* 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**

(\* record is finished, prepare to save records to a RAM disk file  
in several separate PLC scans \*)

**STEP1 := 0; (\* set step1 to 0: no action before save \*)**

**Tstop(T1); (\* stop counting T1\*)**

**T1 := T#0s;**

(\*set the filename attached Date and Time to distinguish different testing \*)

**File\_name\_t := File\_name1 + INT\_str3(Month,2) + INT\_str3(Day,2) +  
INT\_str3(Hour,2) + INT\_str3(Minute,2) + '.txt';**

(\* create a new file \*)

**File1 := F\_creat(File\_name\_t);**

(\* creating file is failed \*)

**if File1 = 0 then**

**Msg1 := 'Create File' + 'File\_nam1 Error!!!';**

**else**

(\* Because saving lots of data to file would take lots of PLC scan  
time , so we are not going to save all data in a single PLC time.  
We will save it in several separate PLC time \*)

**Msg1 := 'Please wait... Saving data to file:' + File\_name1 + '...';**

(\* write something to the begin of file \*)

(\* this way is convenient to draw a map \*)

(\* the first line write the number of record in this file \*)

**str1 := INT\_str3(Total\_record1,4) + '\$0D\$0A';**

(\* the second line write the name of X-axis \*)

**str1 := str1 + 'Time' + '\$0D\$0A';**

(\* the third line write the unit of the interval \*)

**str1 := str1 + '(' + INT\_str3(Interval1,2) + 'ms)' + '\$0D\$0A';**

```

(* the forth line write the name of Y-axis *)
str1 := str1 + 'freq' + '$0D$0A';

(* the fifth line write the unit of frequency *)
str1 := str1 + '(Hz)' + '$0D$0A';
TMP := F_writ_s(File1,str1);

save_file1 := True; (* set as True to start saving Ram disk file *)
save_cnt1 :=0; (* from 0 to total_record1 - 1 *)
end_if;
end_if;
end_if;
end_if;

(* start to copy the data to RAM disk file *)
if save_file1 then

for ii2 := 0 to 50 do (* We ONLY copy 50 records in one PLC scan time *)

if save_cnt1 < total_record1 then

str1 := "";

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

(* add <CR> <LF> at the end of row *)
str1 := str1 + '$0D$0A';

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

(* saving is finished *)
save_file1 := False;
TMP := F_close(File1); (*close file*)
STEP1 := 2; (* 2: record finished *)
Msg1:= 'Record is finished! You may download the record file to your PC
now!';

end_if;
end_for;
end_if;

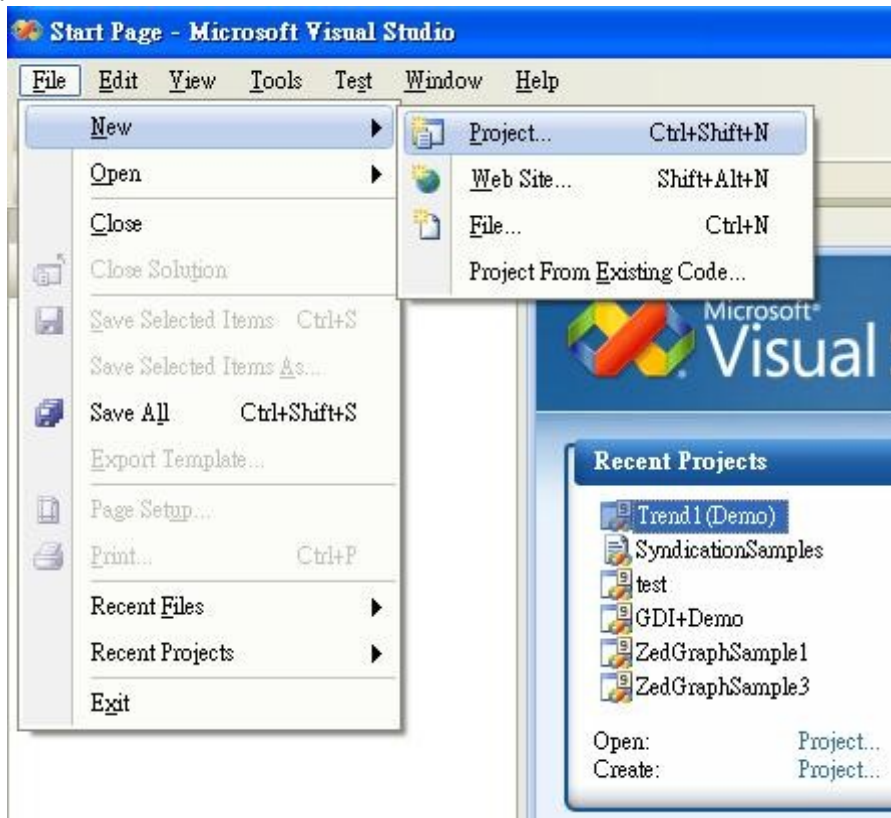
```

## 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:select platform



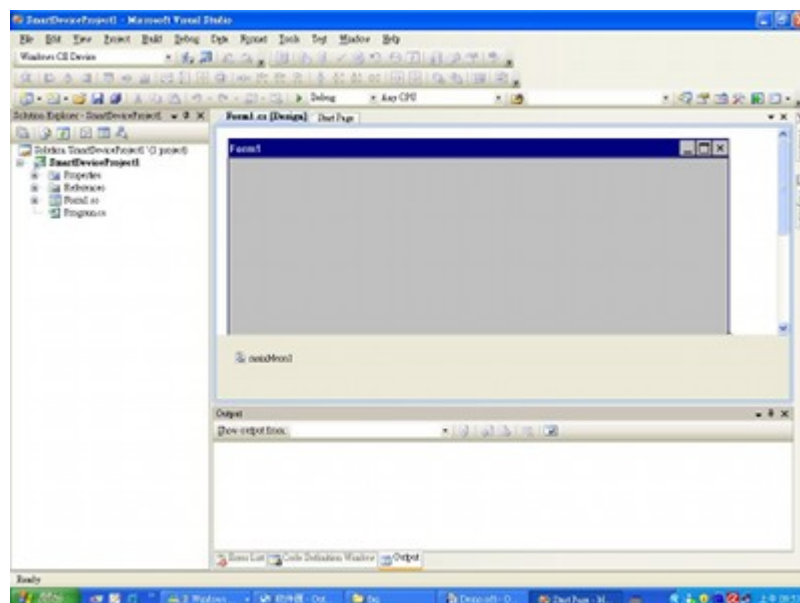
STEP3:select .net Compact Framework version



STEP4:select the type of Template



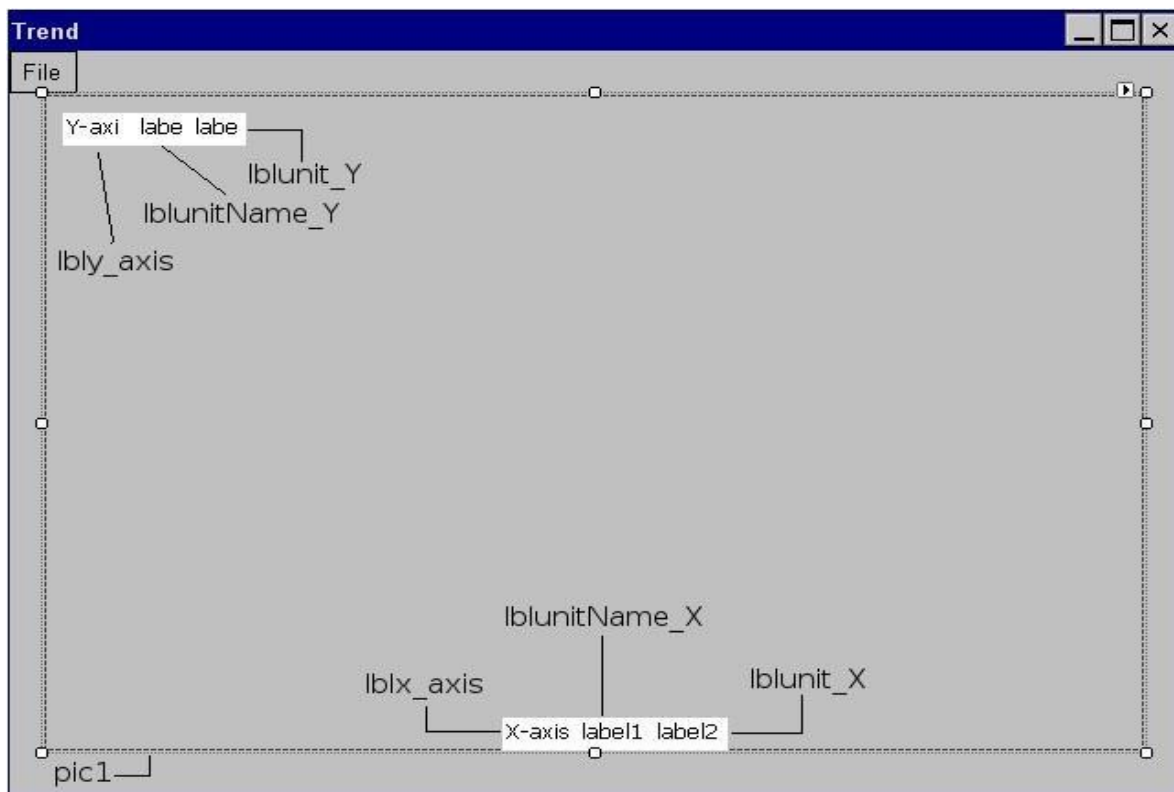
STEP5:Click OK button then you will see as follows



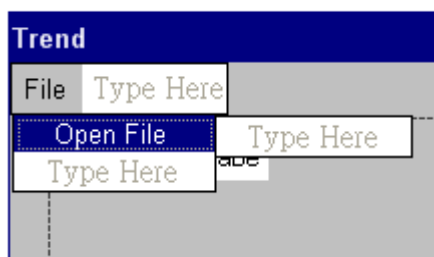
## 2.The controls of form1

the properties of Controls table:

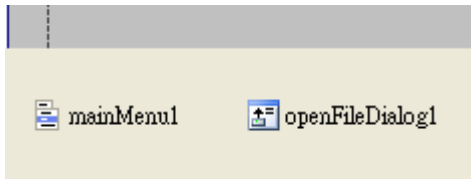
	Name	Text
<b>label1</b>	lbly_axis	Y-axis
<b>label2</b>	lblunitName_Y	-
<b>label3</b>	lblunit_Y	-
<b>label4</b>	lblx_axis	X-axis
<b>label5</b>	lblunitName_X	-
<b>label6</b>	lblunit_X	-
<b>picturebox1</b>	pic1	



Create the child of menu,menuItem2,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 disigner 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 follows

// 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 ,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 fourth 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]=Int32.Parse(sr.ReadLine());
        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++)
{
    g.DrawLine(pen_grid, origin.X + 50 * (i + 1), origin.Y, origin.X + 50 * (i + 1), origin.Y
- 300);
}

//draw horizontal line

```

```

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));
}

//Draw the scale of X-axis
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);
}

//Draw the scale of Y-axis
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));
}

//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) / 300.0;

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

```

```

//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 initialize 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 contentAlignment
    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
    int scale = range / 10;

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

    //create an instance of label and initialize 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 = (scale * (i + 1)).ToString();

    //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();
}
}
}
}
}

```