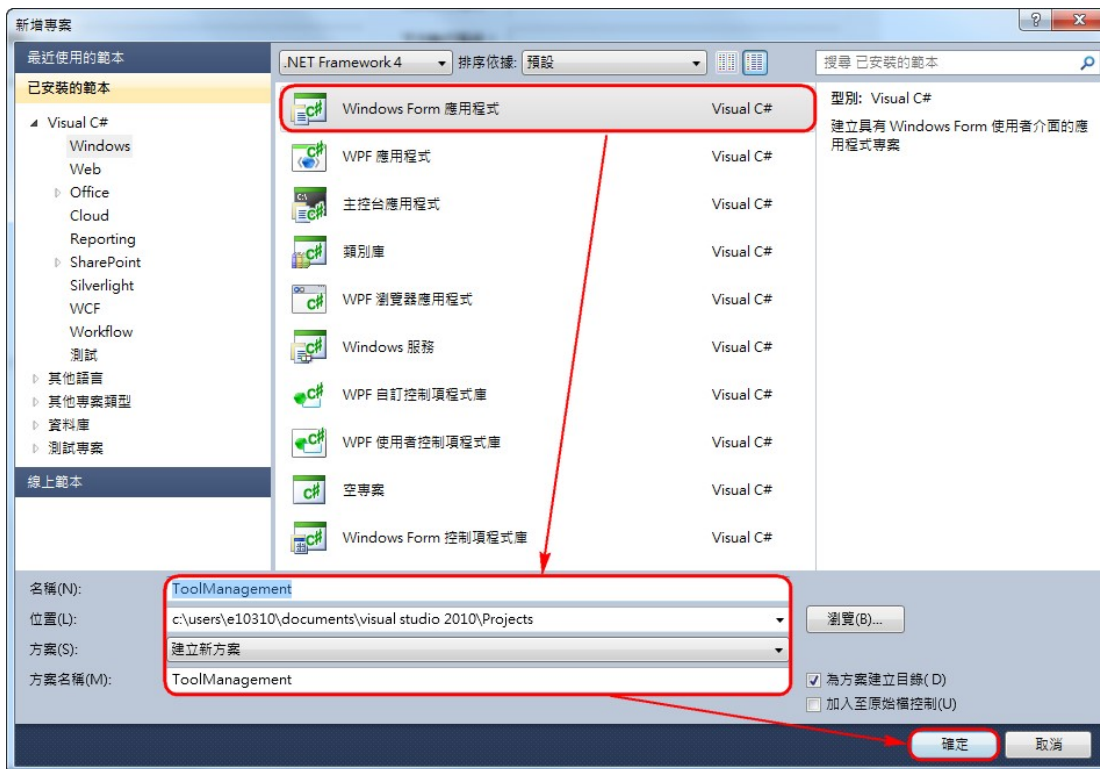


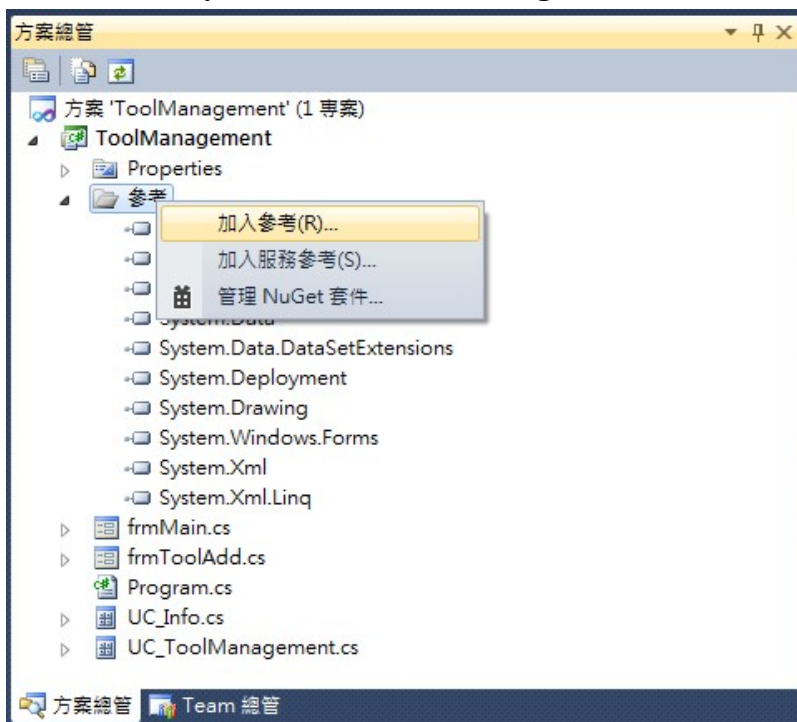
基本資訊功能

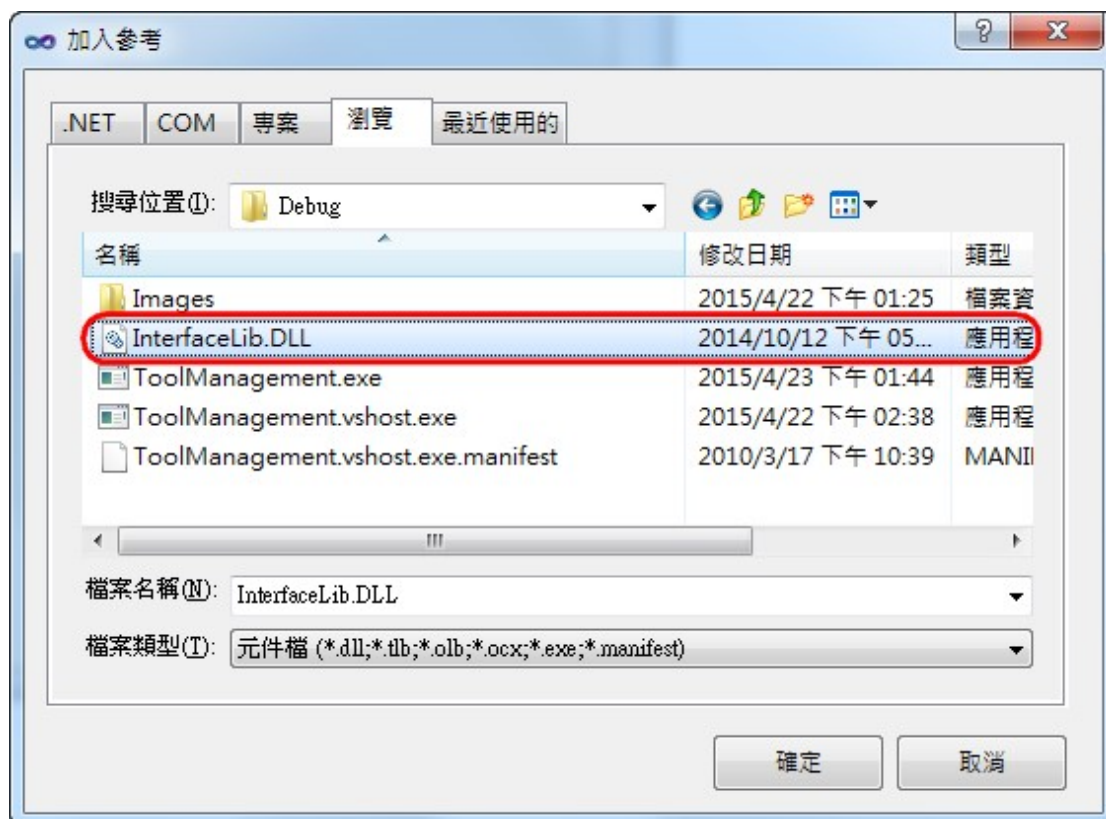
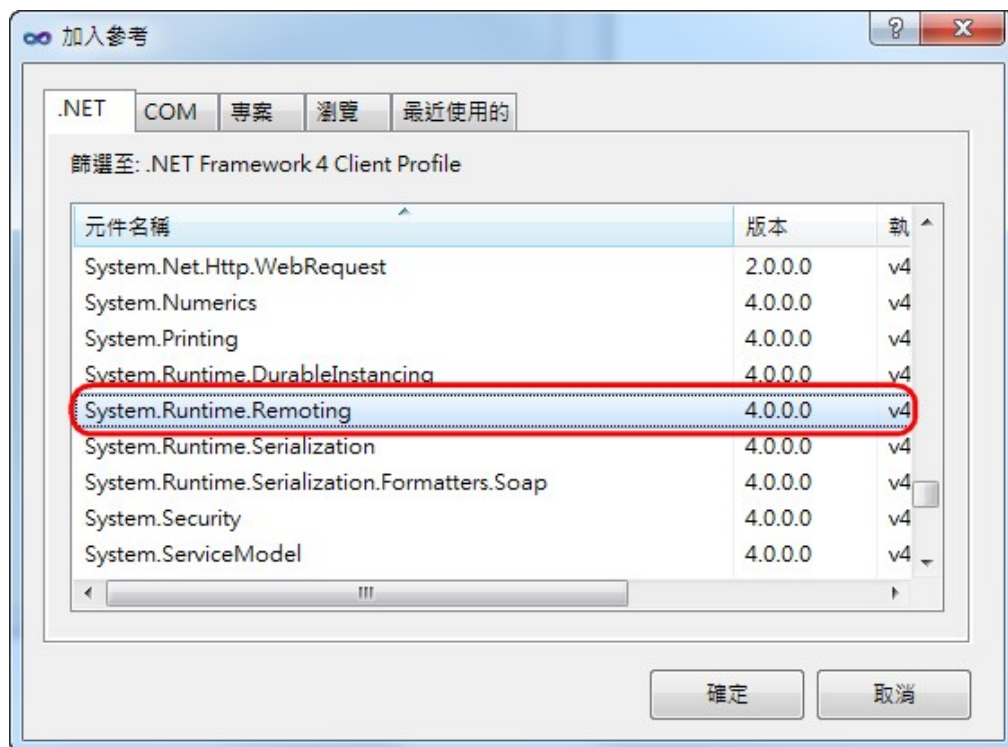
透過 SkyMars 進行實際開發練習

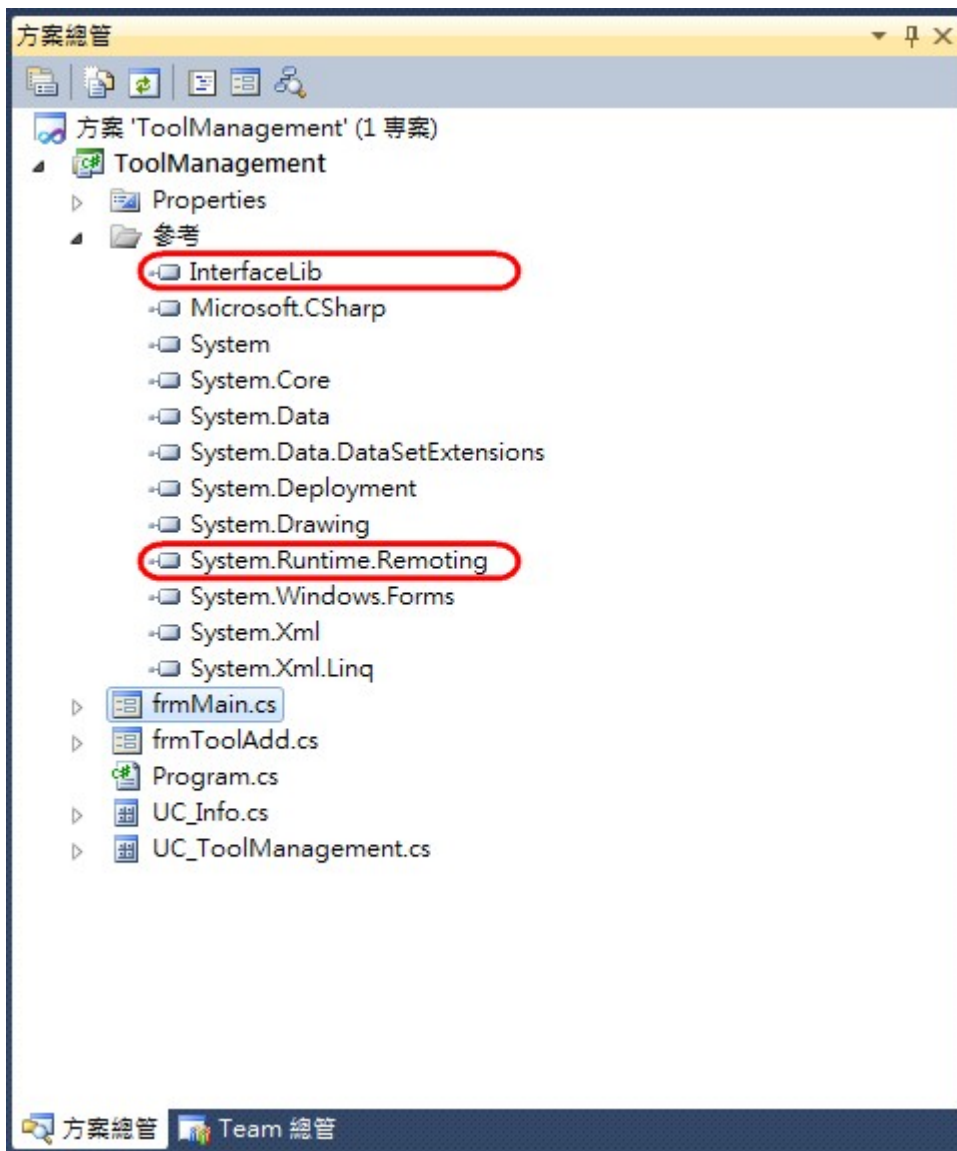
1、建立專案



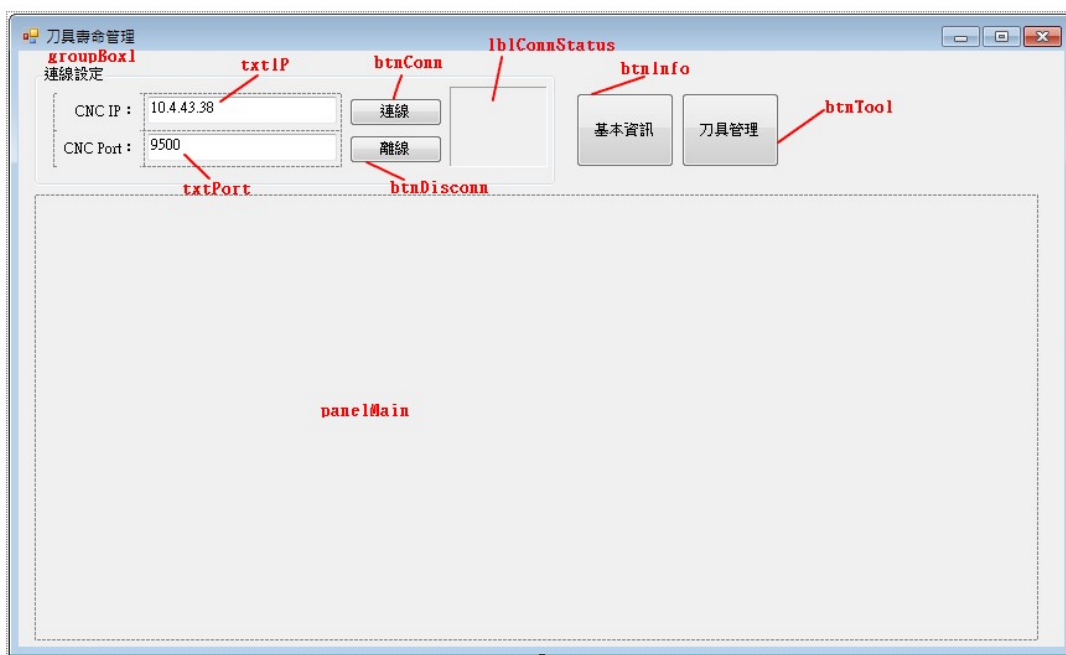
2、加入參考 System.Runtime.Remoting 與 InterfaceLib.dll







3、主畫面設計



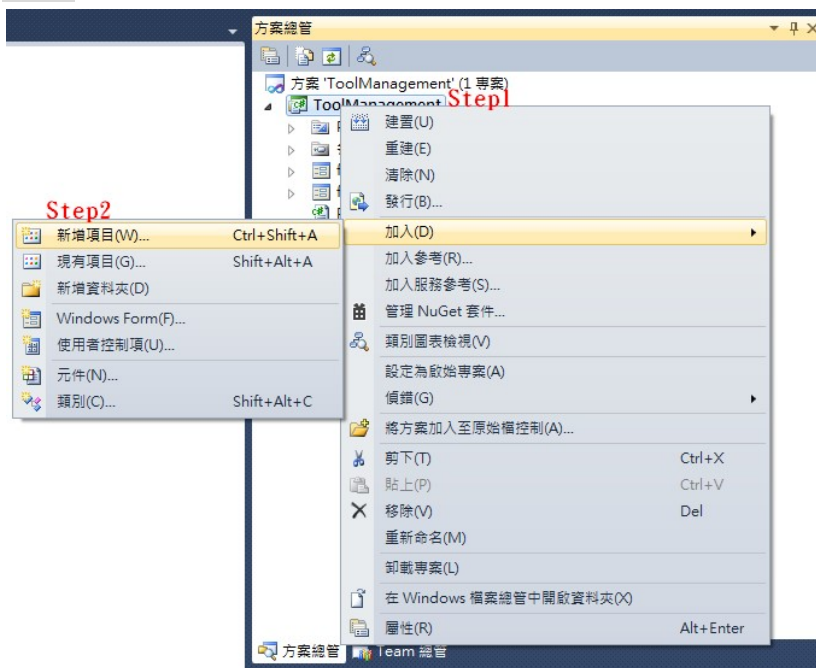
元件名稱	Form
Name	frmMain
Size	860,519
Text	刀具壽命管理

元件名稱	Timer
Name	timer1

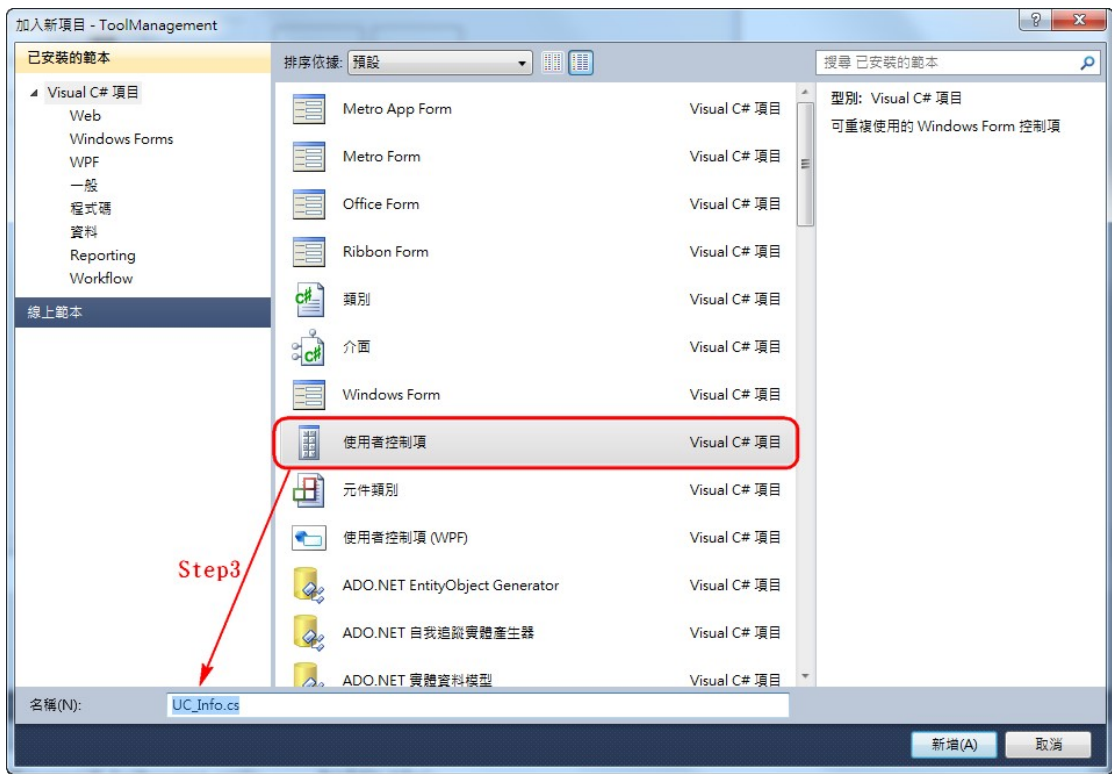
4、新增使用者控制項 UC_Info

Step1 點選方案，按滑鼠右鍵

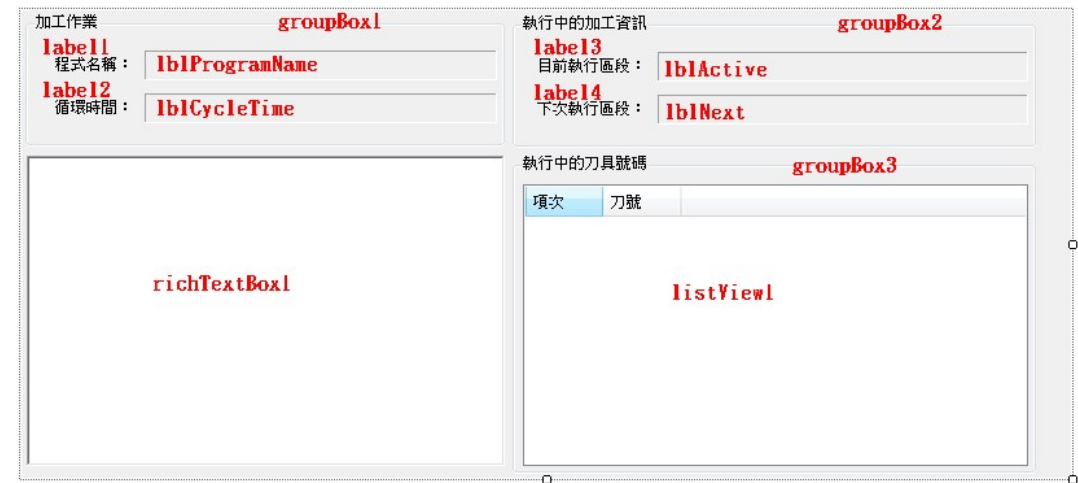
Step2 選擇加入→新增項目



Step3 新增使用者控制項，並且輸入名稱為 **UC_Info.cs**



5、使用者控制項 UC_Info 畫面設計




元件名稱	UserControl
Name	UC_Info
Size	810,360

元件名稱	GroupBox
Name	groupBox1
Text	加工作業
Name	groupBox2
Text	執行中的加工資訊
Name	groupBox3
Text	執行中的刀具號碼

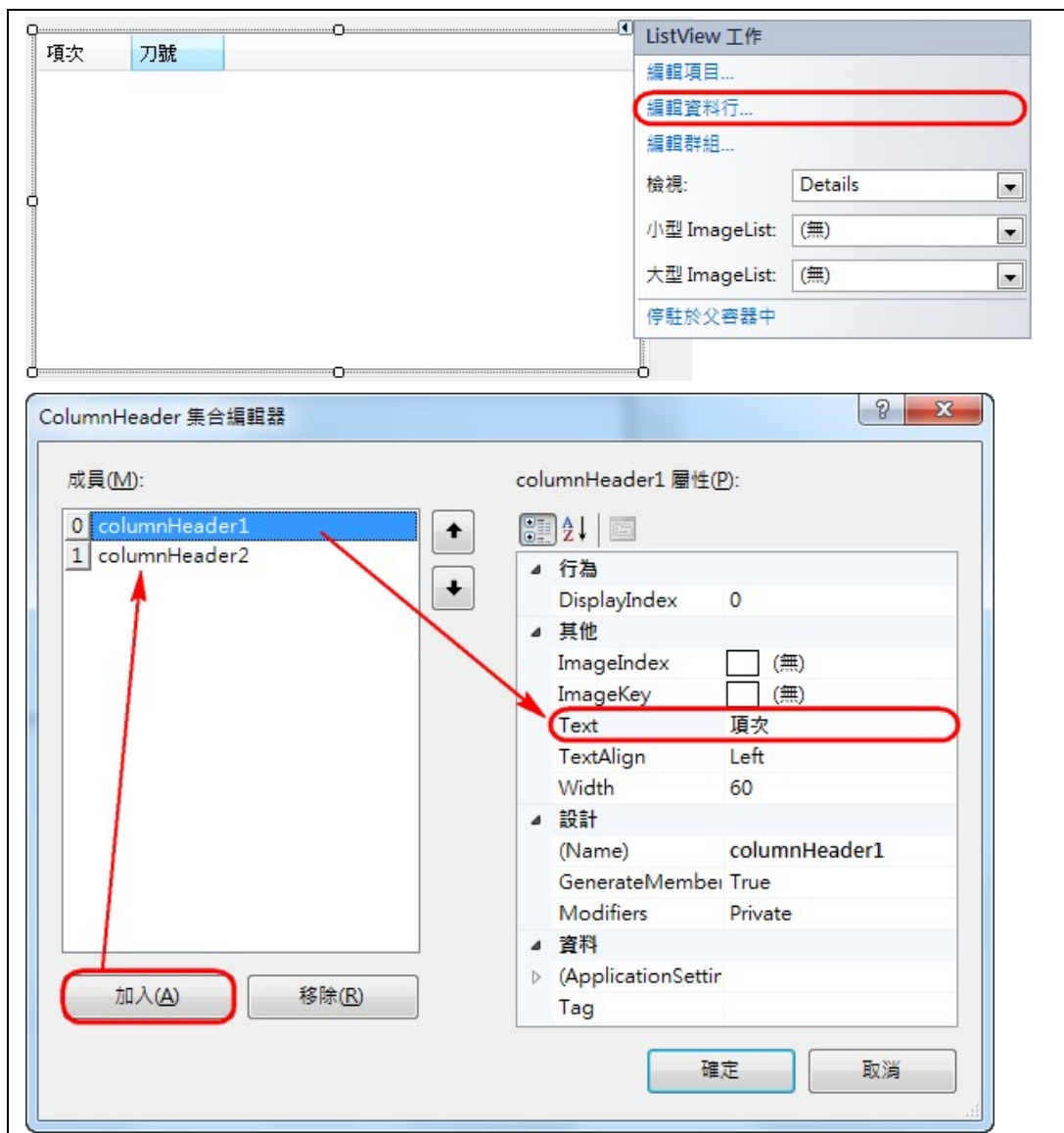
元件名稱	Label
------	-------

Name	label1
AutoSize	False
Text	程式名稱：
TextAlign	
Name	label2
Text	循環時間：
Name	label3
Text	目前執行區段：
Name	label4
Text	下次執行區段：

元件名稱	Label
Name	lblProgramName
BorderStyle	Fixed3D
Text	
TextAlign	
Name	lblCycleTime
Name	lblActive
Name	lblNext

元件名稱	RichTextBox
Name	richTextBox1
HideSelection	False

元件名稱	ListView	集合編輯屬性	集合編輯內容
Name	listView1		
View	Details		
columnsHeader	columnsHeader1	Text	項次
	columnsHeader2	Text	刀號



6、基本資料功能程式碼

UC_Info.cs

命名空間

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

程式碼

```
namespace ToolManagement
{
    public partial class UC_Info : UserControl
    {
```

```
public UC_Info()
{
    InitializeComponent();
}

public void showToProgram(string richProgram, string activeProg, string nextProg)
{
    lblActive.Text = activeProg;
    lblNext.Text = nextProg;

    richTextBox1.Text = "";
    richTextBox1.Text = richProgram;
    richTextBox1.Select(0, richTextBox1.GetFirstCharIndexFromLine(1) - 1);
}

List<string> info = new List<string>();
string oldProgName = "";
public void showToToolNumber(string strTool, string newProgName)
{
    if (oldProgName == "")
    {
        oldProgName = newProgName;
    }

    if (oldProgName != newProgName)
    {
        info.Clear();
        oldProgName = newProgName;
    }

    if (!info.Contains(strTool))
    {
        info.Add(strTool);

        listView1.Items.Clear();
        int i = 0;
        foreach (string item in info)
        {
            i = i + 1;

            ListViewItem toolItem = new ListViewItem(i.ToString());
            toolItem.SubItems.Add("T" + item);
            listView1.Items.Add(toolItem);
        }
    }
}
```



```

        }
    }
}

public void showCycleTime(int[] timeData, string mainProg)
{
    lblCycleTime.Text = string.Format("{0:00}", timeData[0]) + "H " +
        string.Format("{0:00}", timeData[1]) + "M " +
        string.Format("{0:00}", timeData[2]) + "S";

    lblProgramName.Text = mainProg;
}
}
}

```

frmMain.cs

命名空間

```

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

using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using InterfaceLib;

```

程式碼

```

namespace ToolManagement
{
    public partial class frmMain : Form
    {
        InterfaceLib.IMsg iRemoting;
        StructMsg.Pwd _Pwd;
        StructMsg.SkyConn_status _SkyConn_status;
        StructMsg.SkyNc_filename _SkyNc_filename;
        StructMsg.nc_current_block _nc_current_block;
        StructMsg.time _time;
        StructMsg.othercode _othercode;
    }
}

```

```
UC_Info uc_info = null;

string currentStatus = "";
string currentProgName = "";
string currentProg = "";
string currentTCode = "";
int[] cycleTime = new int[3];

public frmMain()
{
    InitializeComponent();
    lblConnStatus.Image = Image.FromFile(Application.StartupPath + "\\images\\offline.ico");
}

private void btnConn_Click(object sender, EventArgs e)
{
    if (ChannelServices.RegisteredChannels.Length == 0)
        ChannelServices.RegisterChannel(new TcpChannel(), false);

    string IP = txtIP.Text.Trim();
    string Port = "9501";
    iRemoting = (IMsg)Activator.GetObject(typeof(IMsg), "tcp://"+ IP + ":" + Port + "/RemoteObjectURI" + Port);
    _Pwd.ConnectionKey = "pmc";
    _Pwd.WritePwd = "pmc";

    short ret = iRemoting.SKY_conn_status(_Pwd, ref _SkyConn_status);
    if (ret == 0)
    {
        MessageBox.Show("連線成功!");
        lblConnStatus.Image = Image.FromFile(Application.StartupPath + "\\images\\online.ico");
        timer1.Enabled = true;
    }
}

private void btnDisconn_Click(object sender, EventArgs e)
{
    lblConnStatus.Image = Image.FromFile(Application.StartupPath + "\\images\\offline.ico");
    timer1.Enabled = false;
}

private void btnInfo_Click(object sender, EventArgs e)
```

```

{
    panelMain.Controls.Clear();

    uc_info = new UC_Info();
    uc_info.Location = new Point(0, 0);
    uc_info.Name = "UC_Info";
    uc_info.MaximumSize = new System.Drawing.Size(0, 0);
    uc_info.MinimumSize = new System.Drawing.Size(0, 0);
    uc_info.Dock = DockStyle.Fill;
    panelMain.Controls.Add(uc_info);
}

private void timer1_Tick(object sender, EventArgs e)
{
    getCNCInfo();

    if (uc_info != null)
    {
        if (currentStatus == "RUN")
        {
            uc_info.showToProgram(currentProg, _nc_current_block.Block[0], _nc_current_block.Block[1]);
            uc_info.showToToolNumber(currentTCode, currentProgName);
            uc_info.showCycleTime(cycleTime, currentProgName);
        }
    }
}

private void getCNCInfo()
{
    //取得状態
    string eqpStatus = "";
    short ret = iRemoting.SKY_conn_status(_Pwd, ref _SkyConn_status);
    if (ret == 0)
    {
        switch (_SkyConn_status.Status[0])
        {
            case 0:
                eqpStatus = "OFF";
                break;
            case 1:
                eqpStatus = "RUN";
                break;

```

```
                case 2:
                    eqpStatus = "IDLE";
                    break;
                case 3:
                    eqpStatus = "Alarm";
                    break;
            }
        }
        currentStatus = eqpStatus;

        //取得程式名稱
        ret = iRemoting.SKY_nc_filename(_Pwd, ref _SkyNc_filename);
        if (ret == 0)
        {
            currentProgName = _SkyNc_filename.MainProg[0];
        }

        //取得程式內容
        ret = iRemoting.GET_nc_current_block(_Pwd, ref _nc_current_block);
        if (ret == 0)
        {
            currentProg = "";
            for (int i = 0; i <= _nc_current_block.Block.Length - 1; i++)
            {
                currentProg += _nc_current_block.Block[i] + "\r\n";
            }
        }

        //取得循環時間
        ret = iRemoting.GET_time(_Pwd, ref _time);
        if (ret == 0)
        {
            cycleTime = _time.Cycle;
        }

        //取得刀號
        ret = iRemoting.GET_othercode(_Pwd, ref _othercode);
        if (ret == 0)
        {
            currentTCode = _othercode.TCode.ToString();
        }
    }
}
```

}

}

7、基本資料功能完成畫面

刀具壽命管理

連線設定

CNC IP : 10.4.43.38

連線

CNC Port : 9500

離線

基本資訊

刀具管理

加工作業

程式名稱 : O0098

循環時間 : 00H 00M 23S

執行中的加工資訊

目前執行區段 : X-39.377

下次執行區段 : Y-54.8

X-39.377

Y-54.8

X99.813

X-2.884Y6.852

G17G02X-.238Y.777R4.

G01X-2.648Y13.418

Y12.615

X1.215Y8.075

G02X.083Y.407R4.

G01X.573Y2.213

G02X.092Y.306R4.

G01X2.919Y8.438

G02X.373Y.788R4.

G01X.561Y.911

Y-60.482

執行中的刀具號碼

項次	刀號
1	T1
2	T3