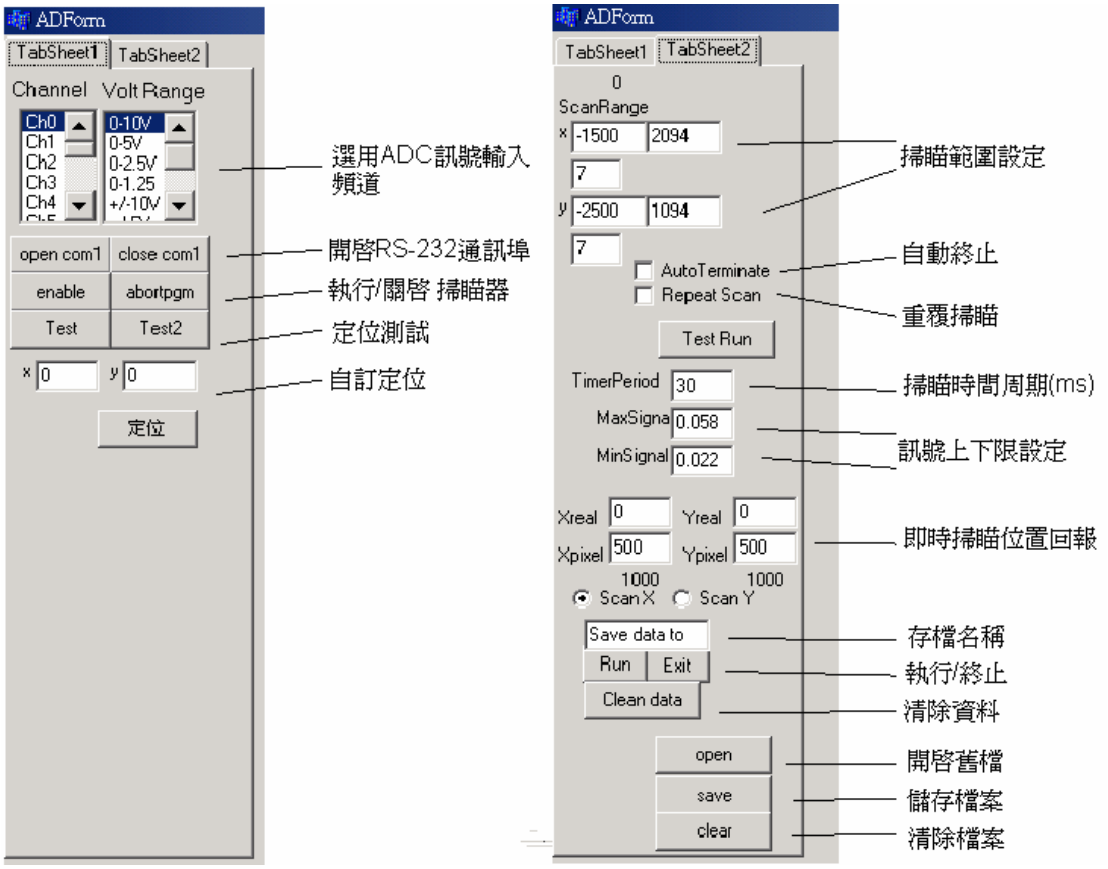
附錄一

以 C++ Builder 撰寫 RS-232 界面對掃瞄器控制程式與即時成像程式

操控界面:



伺服步進馬達控制程式碼:

```
int x1,y1,x2,y2,time;

//掃瞄範圍參數輸入

x1=StrToInt(Edit2->Text);
y1=StrToInt(Edit3->Text);
x2=StrToInt(Edit6->Text);
y2=StrToInt(Edit10->Text);
time=StrToInt(Edit1->Text);
```

```
//指令輸入前預先刪除多餘位數

if(x1<0)
Edit4->Text=IntToHex(x1,4).Delete(1,4);
else
Edit4->Text=IntToHex(x1,4);
if(y1<0)
Edit5->Text=IntToHex(y1,4).Delete(1,4);
else
Edit5->Text=IntToHex(y1,4);
Edit11->Text=IntToHex(time,4);
byte SetPos[7];
DWORD lrc;

//指令格式轉換

SetPos[0]=0x06;
SetPos[1]=StrToInt("0x"+Edit4->Text.Delete(3,2));
SetPos[2]=StrToInt("0x"+Edit4->Text.Delete(1,2));
SetPos[3]=StrToInt("0x"+Edit5->Text.Delete(3,2));
SetPos[4]=StrToInt("0x"+Edit5->Text.Delete(1,2));
SetPos[5]=StrToInt("0x"+Edit11->Text.Delete(3,2));
SetPos[6]=StrToInt("0x"+Edit11->Text.Delete(1,2));
WriteFile(hcomm,&SetPos,7,&lrc,NULL);

//即使成像設定

    //灰階定義

float MaxSignal=StrToFloat(Edit22->Text);
float MinSignal=StrToFloat(Edit25->Text);
float multiplier=765/(MaxSignal-MinSignal);
int valRGB=multiplier*(voltage-MinSignal);
int tmp=valRGB/255;
    if(valRGB<=0)
        Image1->Canvas->Pen->Color=RGB(0,0,0);
    else
         switch(tmp){
                          case                    0                    :
Image1->Canvas->Pen->Color=RGB(valRGB,0,0);break;
```

```
case 1 :
Image1->Canvas->Pen->Color=RGB(255,valRGB-255,0);break;
case 2 :
Image1->Canvas->Pen->Color=RGB(255,255,valRGB-510);break;
default:
Image1->Canvas->Pen->Color=RGB(255,255,255); break;
                          }

  //資料在畫布上之定位

int
Xinterval=Xpixel*StrToInt(Edit7->Text)/(StrToInt(Edit6->Text)-StrToInt(Ed
it12->Text));
int
Yinterval=Ypixel*StrToInt(Edit8->Text)/(StrToInt(Edit10->Text)-StrToInt(E
dit13->Text));

int Xposition=StrToInt(Label1->Caption);
int Yposition=StrToInt(Label2->Caption);
Image1->Canvas->Pen->Width=Yinterval;

//控制掃瞄方向

switch(RadioButton1->Checked)
  {
   case true:
   Image1->Canvas->MoveTo(Xposition-Xinterval,Yposition-Yinterval);
   Image1->Canvas->LineTo(Xposition,Yposition-Yinterval);
    if(x1<x2)
      {
       //scan
        x1+=StrToInt(Edit7->Text);

        Edit2->Text=IntToStr(x1);        //改變 Edit2(即 x)的值

        //image

        Xposition-=Xinterval;              //像素位置改變

        Label1->Caption=IntToStr(Xposition);
       }
      else
```

```
    {
     if(y1<y2)
        {
         //掃瞄位置暫存
         y1+=StrToInt(Edit8->Text);
         Edit3->Text=IntToStr(y1);
         Edit2->Text=Edit12->Text;
         //影像位置暫存
         Yposition-=Yinterval;
         Label2->Caption=IntToStr(Yposition);
         Label1->Caption=StrToInt(Edit20->Text);
         }
        else
         {
            if(CheckBox1->Checked)//repeat
               {    Edit2->Text=Edit12->Text;
                    Edit3->Text=Edit13->Text;
               }
             else    //terminate
                {TimerADGet->Enabled=false;
                //abortpgm atomatically
                DWORD lrc;
                byte AbortData[1];
                AbortData[0]=0x20;
                WriteFile(hcomm,&AbortData,1,&lrc,NULL);
                 }
           }
       }
       break;
case false:
Image1->Canvas->MoveTo(Xposition-Xinterval,Yposition-Yinterval);
Image1->Canvas->LineTo(Xposition-Xinterval,Yposition);
      if(y1<y2)
  {
    //掃瞄位置暫存
```

```
   y1+=StrToInt(Edit8->Text);
   Edit3->Text=IntToStr(y1);        //改變 Edit2(即 x)的值

   //影像處理

   Yposition-=Yinterval;              //像素位置改變
   Label2->Caption=IntToStr(Yposition);
  }
else
  {
   if(x1<x2)
      {
       //掃瞄位置暫存
       x1+=StrToInt(Edit7->Text);
       Edit2->Text=IntToStr(x1);
       Edit3->Text=Edit13->Text;
       //影像定位
       Xposition-=Xinterval;
       Label1->Caption=IntToStr(Xposition);
       Label2->Caption=StrToInt(Edit21->Text);
       }
      else
       {
           if(CheckBox1->Checked)//repeat
          {    Edit2->Text=Edit12->Text;
               Edit3->Text=Edit13->Text;
          }
          else   //終止上述程式
          {TimerADGet->Enabled=false;
           //自動關閉程式
           DWORD lrc;
           byte AbortData[1];
           AbortData[0]=0x20;
```

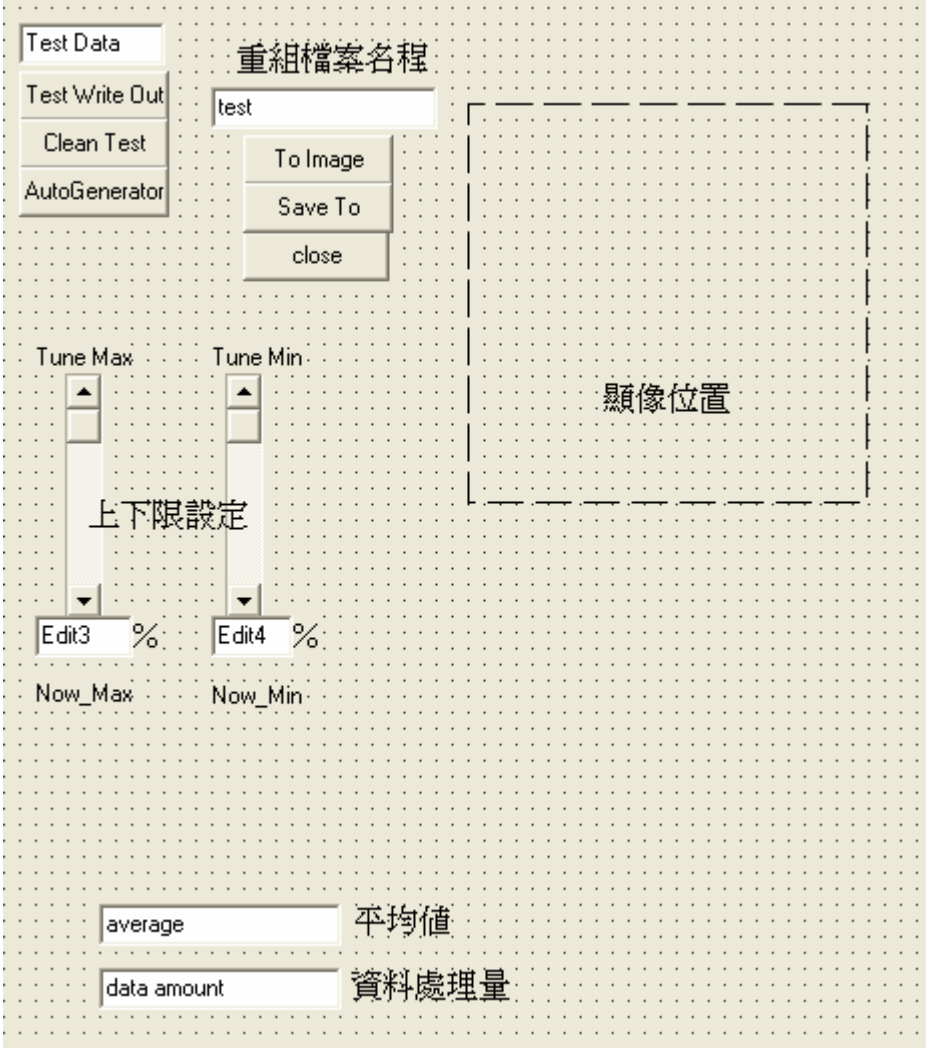
```
                    WriteFile(hcomm,&AbortData,1,&lrc,NULL);
                    }
                }
            }
            break;

    }
}
```

附錄二

影像重組及歸一化程式

操控界面:



重組程式碼:

```
void __fastcall TForm1::Button3Click(TObject *Sender)
{

//取值放入未定數量陣列 storage[]

String ss="";
char ch;
int i=0,j=0;
String storage[160000];
```

```
//開啟文件讀取通道

ifstream fin;
String Opened_File="D:"+ Edit5->Text +".txt";
fin.open(Opened_File.c_str(), ios::in);
while(fin.get(ch))
   {
   if(ch=='\n'){j=0;i++;}    //i++;
   else
      if(j<8){storage[i]+=ch;j++;}    //storage[i]+=ch;
   }

fin.close();   //關閉文件讀取通道


//資料重組

float max=0,min=10,sum=0,average=0;
for(int k=0;k<160000;k++)
     {
     max=StrToFloat(storage[k])>max? StrToFloat(storage[k]):max;
     min=StrToFloat(storage[k])<min? StrToFloat(storage[k]):min;
     sum+=StrToFloat(storage[k]);
     }

//平均值

average=sum/160000;

//上下限設定

max-=((max-min)*scBar2->Position)/100;          //tuned
min+=((max-min)*(scBar1->Position))/100;

//測試碼

Now_Max->Caption=FloatToStr(max);
Now_Min->Caption=FloatToStr(min);

//重組定位

int index=0;
float color=0;
for(int y_position=5;y_position<405;y_position++)
```

```
        {
            for(int x_position=5;x_position<405;x_position++)
           {
            if(StrToFloat(storage[index])>max)
                color=255;
            else
                color=StrToFloat(storage[index])>min?
(StrToFloat(storage[index])-min)*255/(max-min):0;
             Image1->Canvas->Pen->Color=RGB(color,color,color);
                Image1->Canvas->MoveTo(x_position,y_position);
                Image1->Canvas->LineTo(x_position+1,y_position);
                index++;
            }
         }
                Edit7->Text=IntToStr(index);
}
```