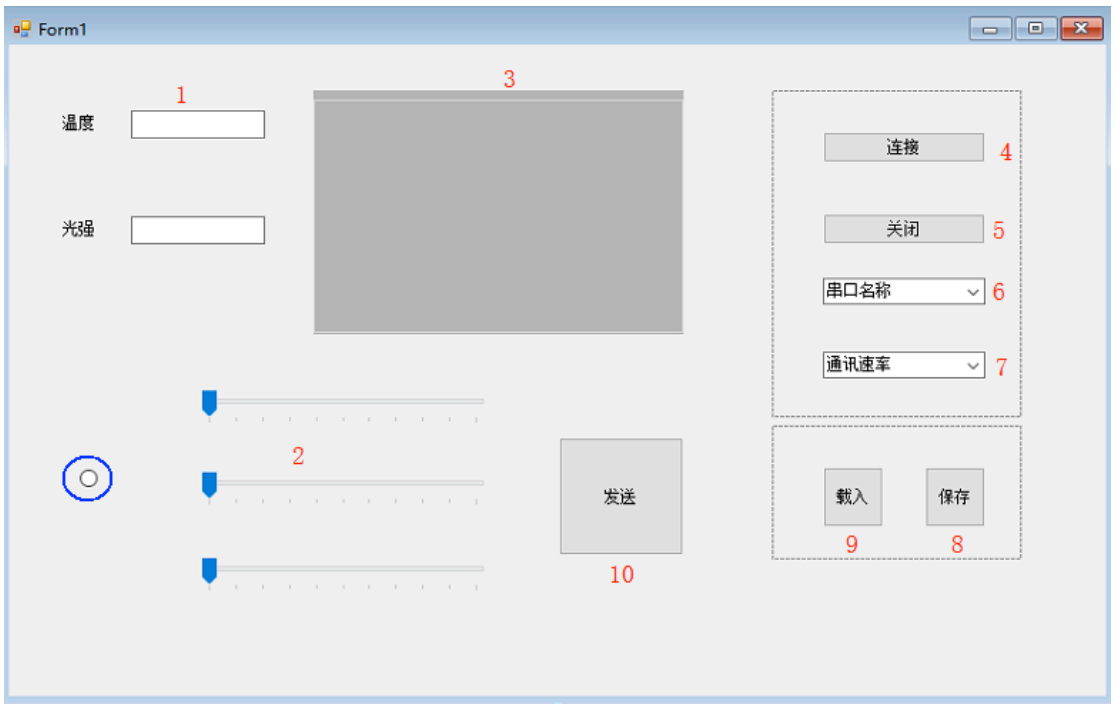
目录

1. 实验要求

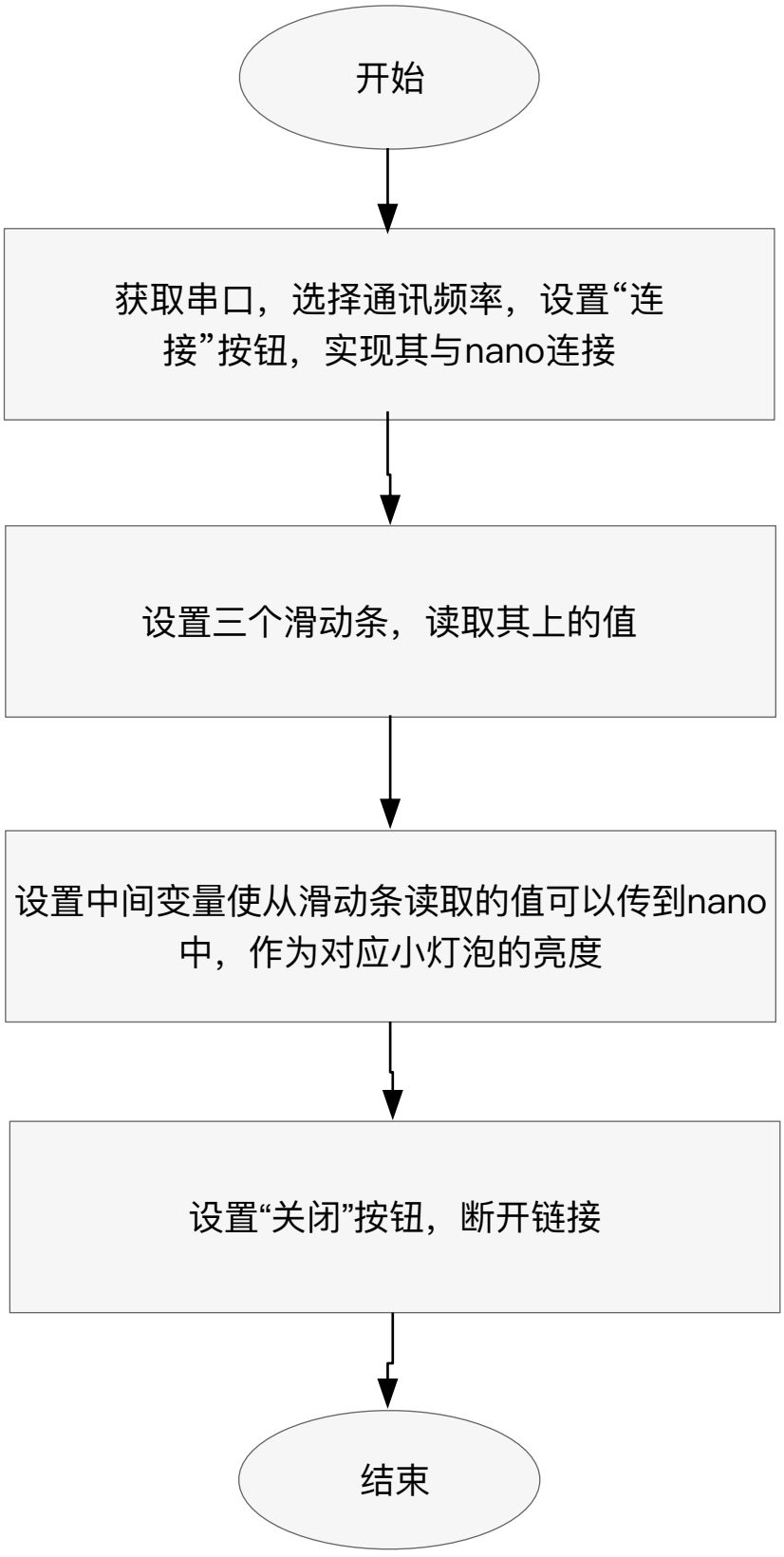
利用pc机和arduino开发板，编写一个windows程序（可以使用WPF或者Windows Form）：程序实现 pc 和 arduino 之间的数据传输、控制、显示。

2. 实验内容



①利用按钮控制开发板上三种灯的开启与关闭、利用滑动按钮(slider)调节三种灯的明暗程度（0-255），点击“发送”按钮后三种灯亮起。
③4 处点击“连接”按钮连接到开发板，5 处点击“关闭”按钮断开连接。
④6 处 combobox 显示所有 pc 机上的串口名，7 处 combobox 显示设定的传输波特率速率(BPS)（9600、19200、38400、57600）,且该速率应当与 arduino 开发板一致。

3. 详细设计


开始
获取串口，选择通讯频率，设置“连接”按钮，实现其与nano连接
设置三个滑动条，读取其上的值
设置中间变量使从滑动条读取的值可以传到nano中，作为对应小灯泡的亮度
设置“关闭”按钮，断开链接
结束

4. 上机实验步骤

cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        private SerialPort port = null;
        private int num1, num2, num3;
        private string[] str=SerialPort.GetPortNames();

        /// <summary>
        /// 初始化串口实例
        /// </summary>
        private void InitialSerialPort()
        {
            try
            {

                string portName = this.comboBox2.SelectedItem.ToString();
                string hz = this.comboBox1.SelectedItem.ToString();
                port = new SerialPort(portName,int.Parse(hz));
                port.Encoding = Encoding.ASCII;
                port.DataReceived += new
SerialDataReceivedEventHandler(port_DataReceived);
                port.Open();
                this.ChangeArduinoSendStatus(true);
            }
            catch (Exception ex)
            {
                MessageBox.Show("初始化串口发生错误：" + ex.Message, "提
示信息", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
            }
        }
        /// <summary>
        ///  关闭并销毁串口实例
        /// </summary>
        private void DisposeSerialPort()
        {
            if (port != null)
            {
                try
                {
                    this.ChangeArduinoSendStatus(false);
                    if (port.IsOpen)
                    {
                        port.Close();
                    }
                    port.Dispose();
                }
                catch (Exception ex)
                {
                    MessageBox.Show("关闭串口发生错误：" + ex.Message, "
提示信息", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
            }
        }

        /// <summary>
        ///  改变Arduino串口的发送状态
        /// </summary>
        /// <param name="allowSend">是否允许发送数据</param>
        private void port_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            try
            {
                string s = port.ReadExisting(); // sp是串口控件名
            }catch
            {

            }


        }
```

```
private void ChangeArduinoSendStatus(bool allowSend)
{
    if (port != null && port.IsOpen)
    {
        if (allowSend)
        {
            port.WriteLine("serial start");
        }
        else
        {
            port.WriteLine("serial stop");
        }
    }
}
public Form1()
{
    InitializeComponent();
    foreach (string x in str)
    {
        comboBox2.Items.Add(x);
    }
    comboBox1.Text = "通讯速率";
    comboBox2.Text = "串行名称";
    trackBar1.Maximum = 255;
    trackBar2.Maximum = 255;
    trackBar3.Maximum = 255;

    comboBox1.Items.Add(9600);
    comboBox1.Items.Add(19200);
    comboBox1.Items.Add(38400);
    comboBox1.Items.Add(57600);
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    //RefreshInfoTextBox1();
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
   // RefreshInfoTextBox2();
}

private void trackBar3_Scroll(object sender, EventArgs e)
```

```
        {
            num3 = trackBar3.Value;//黄
            label6.Text = num3.ToString();
        }

        private void trackBar2_Scroll(object sender, EventArgs e)
        {
            num2 = trackBar2.Value;//红
            label7.Text = num2.ToString();
        }

        private void trackBar1_Scroll(object sender, EventArgs e)
        {
            num1 = trackBar1.Value;//蓝
            label8.Text = num1.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if(port != null&&port.IsOpen)
                {
                    port.WriteLine("," + num1.ToString() + "," +
num2.ToString() + "," + num3.ToString() + ",");
                }
                else
                {
                    MessageBox.Show("发送失败，请检查是否连接");
                }
            }
            catch(Exception ce)
            {

            }

        }

        private void button3_Click(object sender, EventArgs e)
        {
            InitialSerialPort();//连接按钮
        }

        private void button2_Click(object sender, EventArgs e)
```

```
        {
            if (port != null && port.IsOpen)
            {

                port.WriteLine("c");
            }
            DisposeSerialPort();//关闭按钮
        }

        private void comboBox2_SelectedIndexChanged(object sender, EventArgs
e)
        {
            //串行名称
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs
e)
        {
            //通讯数率
        }

        private void button6_Click(object sender, EventArgs e)
        {
            port.WriteLine("y" + num3.ToString());

        }

        private void button7_Click(object sender, EventArgs e)
        {
            port.WriteLine("r" + num2.ToString());

        }

        private void button8_Click(object sender, EventArgs e)
        {
            port.WriteLine("b" + num1.ToString());

        }

        private void label6_Click(object sender, EventArgs e)
        {

        }
```

```
        private void label7_Click(object sender, EventArgs e)
        {

        }

        private void label8_Click(object sender, EventArgs e)
        {

        }

        private void button5_Click(object sender, EventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Arduino:

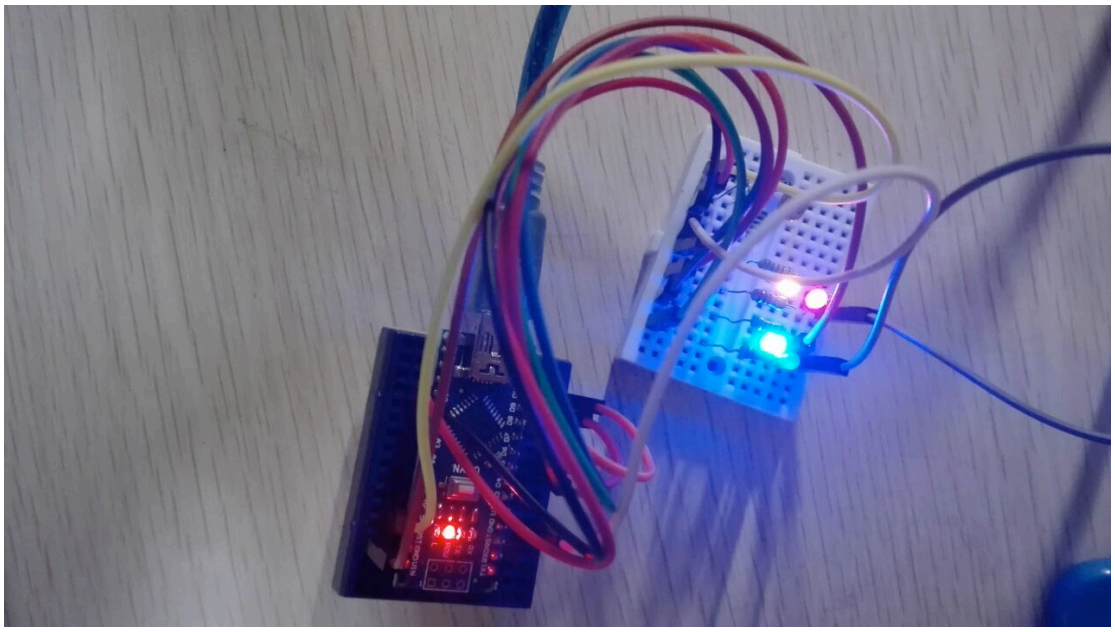
```
int pinLedb = 10;//定义连接 LED 的数字口，当允许通过串口发送数据时，点亮 LED，否则关闭 LED
int pinLedr = 9;//定义连接 LED 的数字口，当允许通过串口发送数据时，点亮 LED，否则关闭 LED
int pinLedy = 6;//定义连接 LED 的数字口，当允许通过串口发送数据时，点亮 LED，否则关闭 LED
int brightness10;
int brightness9;
int brightness6;
boolean sendFlag = false;//指示是否允许通过串口发送数据
boolean readCompleted = false;//指示是否完成读取串口数据
String serialString = "";//串口数据缓存字符串
String a[3];
void setup()
{
    pinMode(pinLedb,OUTPUT);
    pinMode(pinLedr,OUTPUT);
    pinMode(pinLedy,OUTPUT);

    Serial.begin(9600);
    serialString.reserve(200);//初始化字符串
```

```
}
void loop()
{
      if(readCompleted)//判断串口是否接收到数据并完成读取
      {
          if(serialString == "serial start")//当读取到的信息是"serial start"时，设置发送标志设置为 true
          {
              sendFlag = true;
              digitalWrite(pinLedb, HIGH);
              digitalWrite(pinLedr, HIGH);
              digitalWrite(pinLedy, HIGH);
              delay(500);
              digitalWrite(pinLedb, LOW);
              digitalWrite(pinLedr, LOW);
              digitalWrite(pinLedy, LOW);
              delay(500);
          }
          else if(serialString == "serial stop")//当读取到的信息是"serial stop"时，设置发送标志设置为 false
          {
              sendFlag = false;
          }
          else if(serialString =="c")
          {
              digitalWrite(pinLedb, LOW);
              digitalWrite(pinLedr, LOW);
              digitalWrite(pinLedy, LOW);
          }
         else if(serialString.charAt(0)==',')
         {
             stringCut(serialString);
             brightness10=a[0].toInt();
             brightness9=a[1].toInt();
             brightness6=a[2].toInt();
             analogWrite(10,brightness10);
             analogWrite(9,brightness9);
             analogWrite(6,brightness6);
         }
         else if(serialString.charAt(0)=='b')
        {
             brightness10=serialString.substring(1).toInt();
```

```
            analogWrite(10,brightness10);
        }
        else if(serialString.charAt(0)=='r')
        {

  brightness9=serialString.substring(1).toInt();
            analogWrite(9,brightness9);
        }
        else if(serialString.charAt(0)=='y')
        {
          brightness6=serialString.substring(1).toInt();
            analogWrite(6,brightness6);

        }
             serialString = "";
             readCompleted = false;
        }
     delay(2);//延时 20ms
}
void function2(){//PWM  功能改变亮度

   analogWrite(10,brightness10);
   analogWrite(9,brightness9);
   analogWrite(6,brightness6);
  digitalWrite(pinLedb, HIGH);
  digitalWrite(pinLedr, HIGH);
  digitalWrite(pinLedy, HIGH);
   delay(10000); //延时 1000ms

}
void serialEvent()//串口事件处理方法
{
      while(Serial.available())
  {
                  char inChar = (char)Serial.read();
                  if(inChar!='\n')
                  {
                     serialString+=inChar;

                  }
                  else
                  {
                     readCompleted=true;
                  }
```

```
  }
}
void stringCut(String s)
{
   int point[4];
   int j=0;
   for(int i=0;i<s.length();i++)
   {

      if(s.charAt(i)==',')
      {
         point[j]=i;
         j++;
      }
   }
   a[0]=s.substring(point[0]+1,point[1]);
   a[1]=s.substring(point[1]+1,point[2]);
   a[2]=s.substring(point[2]+1,point[3]);

}
```

5. 实验结果



6. 结论

实现了由窗体控制硬件功能，提高了 c#编程能力。