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

namespace 串口调试助手  
{  
public partial class Form1 : Form  
{  
bool m\_mac = false;  
Class1 mac = new Class1();  
public Form1()  
{  
InitializeComponent();  
if (mac.ProductVersion\_MacGet(m\_mac))  
{ }  
else  
{  
MessageBox.Show("No permission to open!", "Hint!", MessageBoxButtons.OK);  
this.Close();  
}  
}

string hms;  
SerialPort m\_Light\_;  
Boolean SerialPort\_status = false;

public Color AppWorkspace { get; private set; }

private void button2\_Click(object sender, EventArgs e)  
{  
try  
{  
if (m\_Light\_ != null && m\_Light\_.IsOpen == false)  
{  
m\_Light\_ = new SerialPort(comboBox1.Text, 19200, Parity.None, 8, StopBits.One);  
m\_Light\_.Open();  
button2.Text = "Stop";  
button2.BackColor = Color.Red;  
SerialPort\_status = true;  
Send\_button.Enabled = true;  
Send\_Data.Enabled = true;  
trackBar1.Enabled = true;  
trackBar2.Enabled = true;  
trackBar3.Enabled = true;  
trackBar4.Enabled = true;  
timer1.Enabled = true;

listBox1.Items.Add(hms + "Serial Port Opened!");  
}  
else  
{  
m\_Light\_.Close();  
button2.Text = "Open";  
button2.BackColor = AppWorkspace;  
SerialPort\_status = false;  
Send\_button.Enabled = false;  
Send\_Data.Enabled = false;  
trackBar1.Enabled = false;  
trackBar2.Enabled = false;  
trackBar3.Enabled = false;  
trackBar4.Enabled = false;  
timer1.Enabled = false;

listBox1.Items.Add(hms + "Serial Port Closed!");  
  
}  
}  
catch (Exception a)  
{  
listBox1.Items.Add(hms + " [" + a + "]");  
MessageBox.Show(a.Message,"Hint!",MessageBoxButtons.OK);  
}  
}

private void timer1\_Tick(object sender, EventArgs e)  
{  
hms = DateTime.Now.ToString("yyyy:MM:dd hh:mm:ss");  
}

string Value1;  
private void trackBar1\_Scroll(object sender, EventArgs e)  
{  
if (SerialPort\_status)  
{  
try  
{  
int trackBar = Convert.ToInt16((sender as TrackBar).Tag);  
switch (trackBar)  
{  
case 1:  
Value1 = "SA0" + trackBar1.Value.ToString("000") + "#";  
CH1\_label.Text = trackBar1.Value.ToString("000");  
break;  
case 2:  
Value1 = "SB0" + trackBar2.Value.ToString("000") + "#";  
CH2\_label.Text = trackBar2.Value.ToString("000");  
break;  
case 3:  
Value1 = "SC0" + trackBar3.Value.ToString("000") + "#";  
CH3\_label.Text = trackBar3.Value.ToString("000");  
break;  
case 4:  
Value1 = "SD0" + trackBar4.Value.ToString("000") + "#";  
CH4\_label.Text = trackBar4.Value.ToString("000");  
break;  
}  
m\_Light\_.Write(Value1);  
listBox1.Items.Add(hms + Value1);  
}  
catch (Exception a)  
{  
MessageBox.Show(a.Message, "Hint!", MessageBoxButtons.OK);  
}  
}  
else  
{  
MessageBox.Show("Serial Port Not Open!","Hint!",MessageBoxButtons.OK);  
}  
}

private void Send\_button\_Click(object sender, EventArgs e)  
{  
if (SerialPort\_status && Send\_Data.Text != string.Empty)  
{  
string m\_Send\_Data = Send\_Data.Text;  
m\_Light\_.Write(m\_Send\_Data);  
listBox1.Items.Add(hms + m\_Send\_Data);  
}  
else  
{  
MessageBox.Show("Not Data!!!","Hint!",MessageBoxButtons.OK);  
}  
}

private void Form1\_Load(object sender, EventArgs e)  
{  
Send\_button.Enabled = false;  
Send\_Data.Enabled = false;  
trackBar1.Enabled = false;  
trackBar2.Enabled = false;  
trackBar3.Enabled = false;  
trackBar4.Enabled = false;  
timer1.Enabled = false;

}

private void serialPort\_DataReceived()  
{  
string str = "";  
do  
{  
int count = m\_Light\_.BytesToRead;  
if (count <= 0)  
{  
break;  
}  
byte[] readBuffer = new byte[count];  
Application.DoEvents();  
m\_Light\_.Read(readBuffer, 0, count);  
str += System.Text.Encoding.Default.GetString(readBuffer);

}  
while (m\_Light\_.BytesToRead > 0);  
{  
listBox1.Items.Add(str);  
}  
}

}  
}