using System;

using System.IO.Ports;

using System.Windows;

using System.Windows.Documents;

using System.Diagnostics;

using System.Windows.Data;

using System.Collections.Generic;

namespace SignVAlpha

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

DataContext = this;

GetPorts();

ButtonsControl.IsEnabled = false;

Save.IsEnabled = false;

DisconnectButton.IsEnabled = false;

DisconnectedText.Visibility = Visibility.Visible;

ConnectedText.Visibility = Visibility.Collapsed;

}

//Serial

SerialPort serial = new SerialPort

{

//Sets up serial port

BaudRate = Convert.ToInt32(9600),

Handshake = System.IO.Ports.Handshake.None,

Parity = Parity.None,

DataBits = 8,

};

private void GetPorts()

{

string[] ports = SerialPort.GetPortNames();

List<string> ComList = new List<string>();

ComList.Add("Choose one");

foreach(string port in ports)

{

ComList.Add(port);

}

COMBox.ItemsSource = ComList;

}

private void Connect(object sender, RoutedEventArgs e)

{

try

{

if (!serial.IsOpen)

{

serial.Open();

DisconnectedText.Visibility = Visibility.Collapsed;

ConnectedText.Visibility = Visibility.Visible;

DisconnectButton.IsEnabled = true;

ConnectButton.IsEnabled = false;

ButtonsControl.IsEnabled = true;

Save.IsEnabled = true;

}

}

catch (Exception a)

{

Debug.Write("Not Working");

}

}

private void Disconnect(object sender, RoutedEventArgs e)

{

if (serial.IsOpen)

{

DisconnectedText.Visibility = Visibility.Visible;

ConnectedText.Visibility = Visibility.Collapsed;

serial.Close();

ConnectButton.IsEnabled = true;

DisconnectButton.IsEnabled = false;

ButtonsControl.IsEnabled = false;

Save.IsEnabled = false;

}

}

private void SaveStuffToArduino(object sender, RoutedEventArgs e)

{

if (serial.IsOpen)

{

// signal to send data to MCU

serial.Write("{");

// Send SingleLine data of each button

string SingleLine = b1SingleLine.IsChecked == true ? "1" : "0";

SingleLine += b2SingleLine.IsChecked == true ? "1" : "0";

SingleLine += b3SingleLine.IsChecked == true ? "1" : "0";

SingleLine += b4SingleLine.IsChecked == true ? "1" : "0";

SingleLine += b5SingleLine.IsChecked == true ? "1" : "0";

SingleLine += b6SingleLine.IsChecked == true ? "1" : "0";

serial.Write(SingleLine);

// Send AutoScroll data of each button

string AutoScroll = b1AutoScroll.IsChecked == true ? "1" : "0";

AutoScroll += b2AutoScroll.IsChecked == true ? "1" : "0";

AutoScroll += b3AutoScroll.IsChecked == true ? "1" : "0";

AutoScroll += b4AutoScroll.IsChecked == true ? "1" : "0";

AutoScroll += b5AutoScroll.IsChecked == true ? "1" : "0";

AutoScroll += b6AutoScroll.IsChecked == true ? "1" : "0";

serial.Write(AutoScroll);

// Send ScrollSpeed data of each button

string ScrollSpeed = b1ScrollSpeed.SelectedItem.ToString();

ScrollSpeed += b2ScrollSpeed.SelectedItem.ToString();

ScrollSpeed += b3ScrollSpeed.SelectedItem.ToString();

ScrollSpeed += b4ScrollSpeed.SelectedItem.ToString();

ScrollSpeed += b5ScrollSpeed.SelectedItem.ToString();

ScrollSpeed += b6ScrollSpeed.SelectedItem.ToString();

serial.Write(ScrollSpeed);

// Send DisplayTime data of each button

string DisplayTime = b1DisplayTime.Text;

DisplayTime += b2DisplayTime.Text;

DisplayTime += b3DisplayTime.Text;

DisplayTime += b4DisplayTime.Text;

DisplayTime += b5DisplayTime.Text;

DisplayTime += b6DisplayTime.Text;

// Send Line1 of each button

serial.Write(b1Line1.Text + "=");

serial.Write(b2Line1.Text + "=");

serial.Write(b3Line1.Text + "=");

serial.Write(b4Line1.Text + "=");

serial.Write(b5Line1.Text + "=");

serial.Write(b6Line1.Text + "=");

// Send Line2 of each button

if(b1SingleLine.IsChecked == false)

{

serial.Write(b1Line2.Text + "=");

}

if (b2SingleLine.IsChecked == false)

{

serial.Write(b2Line2.Text + "=");

}

if (b3SingleLine.IsChecked == false)

{

serial.Write(b3Line2.Text + "=");

}

if (b4SingleLine.IsChecked == false)

{

serial.Write(b4Line2.Text + "=");

}

if (b5SingleLine.IsChecked == false)

{

serial.Write(b5Line2.Text + "=");

}

if (b6SingleLine.IsChecked == false)

{

serial.Write(b6Line2.Text + "=");

}

serial.Write("}");

}

}

void ShutDown(object sender, RoutedEventArgs e)

{

if (serial.IsOpen)

{

serial.Close();

}

System.Windows.Application.Current.Shutdown();

}

private void b1SingleLine\_Click(object sender, RoutedEventArgs e)

{

b1Line2.IsEnabled = (b1SingleLine.IsChecked == true) ? false : true;

}

private void b2SingleLine\_Click(object sender, RoutedEventArgs e)

{

b2Line2.IsEnabled = (b2SingleLine.IsChecked == true) ? false : true;

}

private void b3SingleLine\_Click(object sender, RoutedEventArgs e)

{

b3Line2.IsEnabled = (b3SingleLine.IsChecked == true) ? false : true;

}

private void b4SingleLine\_Click(object sender, RoutedEventArgs e)

{

b4Line2.IsEnabled = (b4SingleLine.IsChecked == true) ? false : true;

}

private void b5SingleLine\_Click(object sender, RoutedEventArgs e)

{

b5Line2.IsEnabled = (b5SingleLine.IsChecked == true) ? false : true;

}

private void b6SingleLine\_Click(object sender, RoutedEventArgs e)

{

b6Line2.IsEnabled = (b6SingleLine.IsChecked == true) ? false : true;

}

private void COMBox\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)

{

if (COMBox.SelectedIndex != 0)

{

serial.PortName = COMBox.SelectedItem.ToString();

ConnectButton.IsEnabled = true;

if (serial.IsOpen)

{

serial.Close();

}

ButtonsControl.IsEnabled = false;

DisconnectButton.IsEnabled = false;

DisconnectedText.Visibility = Visibility.Visible;

ConnectedText.Visibility = Visibility.Collapsed;

Save.IsEnabled = false;

}

else

{

if (serial.IsOpen)

{

serial.Close();

}

ButtonsControl.IsEnabled = false;

ConnectButton.IsEnabled = false;

DisconnectButton.IsEnabled = false;

DisconnectedText.Visibility = Visibility.Visible;

ConnectedText.Visibility = Visibility.Collapsed;

Save.IsEnabled = false;

}

}

}

}