VIDHUSURAKSHA ARDUINO DOCUMENTARY

* .NET APPLICATION COMMANDS

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;

using System.Threading;

using System.IO;

namespace testingmcu

{

public partial class Form1 : Form

{

SerialPort port;

public Form1()

{

InitializeComponent();

try

{

UNOSERIAL.Open();

}

catch (IOException ex)

{

MessageBox.Show("UNO USB is not connected properly");

}

try

{

NODESERIAL.Open();

}

catch (IOException ex)

{

MessageBox.Show("NODE SERIAL is not connected properly");

}

SendCommandnode("6");

}

private void ButtonRedOn\_Click(object sender, EventArgs e)

{

SendCommanduno("2");

}

private void ButtonRedOff\_Click(object sender, EventArgs e)

{

SendCommanduno("1");

}

private void ButtonGreenOn\_Click(object sender, EventArgs e)

{

SendCommanduno("4");

}

private void ButtonGreenOff\_Click(object sender, EventArgs e)

{

SendCommanduno("3");

}

private void ButtonSoundOn\_Click(object sender, EventArgs e)

{

SendCommandnode("5");

}

private void ButtonSoundOff\_Click(object sender, EventArgs e)

{

SendCommandnode("6");

}

Mutex sendCommandMutex = new Mutex();

private void SendCommanduno(string command)

{

if (!UNOSERIAL.IsOpen)

{

return;

}

sendCommandMutex.WaitOne();

UNOSERIAL.WriteLine(command);

sendCommandMutex.ReleaseMutex();

}

private void SendCommandnode(string command)

{

if (!NODESERIAL.IsOpen)

{

return;

}

sendCommandMutex.WaitOne();

NODESERIAL.WriteLine(command);

sendCommandMutex.ReleaseMutex();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

NODE MCU CODE

* WHICH IS USED TO ON AND OFF THE SOUND AS THE COMMANDS FROM THE D.APP..
* USED TO REDUCE THE NOICE WHICH COMES FROM THE SPEAKER

int led2 = D1;

void setup()

{

Serial.begin(9600);

pinMode(led2,OUTPUT);

}

void loop()

{

if(Serial.available()>0)

{

int data = Serial.read();

switch(data)

{

case '1':

//digitalWrite(led,HIGH);

break;

case '2':

//digitalWrite(led,LOW);

break;

case '3':

//digitalWrite(led1,HIGH);

break;

case '4':

//digitalWrite(led1,LOW);

break;

case '5':

digitalWrite(led2,LOW);

break;

case '6':

digitalWrite(led2,HIGH);

break;

}

}

}

UNO CODE

* WHICH IS USED TO ON AND OFF THE LIGHTS (RED & GREES) AS IDENTIFIED AND NOT IDENTIFIED

int led = 9;

int led1 = 10;

void setup()

{

Serial.begin(9600);

pinMode(led,OUTPUT);

pinMode(led1,OUTPUT);

}

void loop()

{

if(Serial.available()>0)

{

int data = Serial.read();

switch(data)

{

case '1':

digitalWrite(led,LOW);

break;

case '2':

digitalWrite(led,HIGH);

break;

case '3':

digitalWrite(led1,LOW);

break;

case '4':

digitalWrite(led1,HIGH);

break;

case '5':

//digitalWrite(led2,LOW);

break;

case '6':

//digitalWrite(led2,HIGH);

break;

}

}

}