Interface AT2.2

30003673

Contents

[Pseudo Code 1](#_Toc12533459)

[Question 1 & 2 1](#_Toc12533460)

[Question 3 1](#_Toc12533461)

[Question 4 & 5 1](#_Toc12533462)

[Question 6 1](#_Toc12533463)

[Question 7 2](#_Toc12533464)

[Test Plan 3](#_Toc12533465)

[Write Code and Debug/Test 4](#_Toc12533466)

[Arduino: 4](#_Toc12533467)

[Microsoft Visual Studio (C#): 7](#_Toc12533468)

# Pseudo Code

## Question 1 & 2

float data = read(LM35sensor);

float outsidetemp = data \* 500 / 1023;

float insidetemp = read dht temp();

float humidity = read dht humidity();

print(outsidetemp);

print(insidetemp);

print(humidity);

delay(100);

## Question 3

Case 300: redLED(ON);

Case 301: redLED(OFF);

End Switch

## Question 4 & 5

If(read(lightSensor) < Too bright)

blueLED(ON);

delay(10000);

Else

blueLED(OFF);

delay(10000);

End If

## Question 6

Int tempmax = dht read temp();

Switch(tempmax)

Case 25: buzzer(1000, 50);

break;

Case 26: buzzer(3000, 100);

break;

End Switch

## Question 7

Int val = serial.parseInt();

Switch(val)  
 Case 101: RGBGreen(ON);  
 RGBRed(OFF);  
 RGBBlue(OFF);break;

Case 102: RGBRed(ON);

RGBBlue(OFF);

RGBGreen(OFF);break;

Case 103: RGBBlue(ON);

RGBRed(OFF);

RGBGreen(OFF);break;

Case 200: RGBGreen(OFF);  
 RGBRed(OFF);  
 RGBBlue(OFF);

# Test Plan

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Description** | **Test?** |
| # 1 | Read and display the current inside temperature and humidity. Use the Arduino Humidity sensor DHT11 | ✔ |
| # 2 | Read and display the current outside temperature. Use the single Temperature Sensor LM35 | ✔ |
| # 3 | Control the Red LED using a single toggle button, which clicks once for on and once for off. | ✔ |
| # 4 | The Blue LED will turn on when the Light A1 is covered (Dark) and turn off when under normal daylight. | ✔ |
| # 5 | Sound a warning when the temperature is above a set comfort maximum. | ✔ |
| # 6 | Control mood light by manually adjusting the variable dial (Rotation A0). This feature will use the RGB LED and the Rotation Sensor | ✔ |

# Write Code and Debug/Test

## Arduino:

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

const int sensorLM35 = A2; //The LM35 sensor is connected to A2 of the Arduino

int DHT11Pin = 4; //The DHT11 sensor is connected pin 4 of the Arduino

int buzzerPin = 5; //The buzzer is connected pin 5 of the Arduino

DHT dht(DHT11Pin, DHT11); //declare DHT sensor

double outsidetemp; // declare outsidetemp

double insidetemp; // declare insidetemp

double humidity; // declare humidity

int RGBRedPin = 9; //The red RGB is connected pin 9 of the Arduino.

int RGBGreenPin = 10; //The green RGB is connected pin 10 of the Arduino.

int RGBBluePin = 11; //The blue RGB is connected pin 11 of the Arduino.

int redLED = 12; //The red LED is connected pin 12 of the Arduino.

int blueLED = 13; //The blue LED is connected pin 13 of the Arduino.

int rotationPin = A0; //The rotation sensor is plugged into pin A0 of the Arduino.

int lightPin = A1;//The light sensor is plugged into pin A0 of the Arduino.

void setup() {

// put your setup code here, to run once:

Serial.begin(9600); //Serial monitor port

pinMode(sensorLM35, INPUT); // Setup LM35 sensor as input pin for reading data

dht.begin();

pinMode(RGBRedPin, OUTPUT); //Setup red RGB LED as an output pin.

pinMode(RGBGreenPin, OUTPUT); //Setup green RGB pin as an output pin.

pinMode(RGBBluePin, OUTPUT); //Setup blue RGB pin as an output pin.

pinMode(redLED, OUTPUT); //Setup red LED pin as an output pin.

pinMode(blueLED, OUTPUT); //Setup blue LED pin as an output pin.

}

void loop() {

// put your main code here, to run repeatedly:

//QUESTION 1 & 2

outsidetemp = analogRead(sensorLM35); //read temperature data from LM35

outsidetemp = (outsidetemp \* 500) / 1023; // convert it into celsius format

insidetemp = dht.readTemperature(); //read temperature data from dht sensor

humidity = dht.readHumidity(); //read humidity data data from dht sensor

Serial.print(outsidetemp);

Serial.print(',');

Serial.print(insidetemp);

Serial.print(',');

Serial.println(humidity);

delay(50);

//QUESTION 6

int tempmax = dht.readTemperature(); // declare max temperature

switch (tempmax) {

case 25: tone(buzzerPin, 1000, 50); break; //if 25 degree then buzzer

delay(1000);

case 26: tone(buzzerPin, 3000, 100); break;//if 26 degree then buzzer sound higher

}

//QUESTION 7

int val = Serial.parseInt(); //The Arduino sketch uses a Serial.parseInt method to read the string data from C# and convert into an Integer.

//this code is event driven and requires the user to click the

//button before any data is sent. The code sends a text values of 101,102,103, 200, 201,202 or 300,301 to the

//Arduino which is read and converted into and integer for the Arduino Switch/Case statement.

switch (val) {

case 101: analogWrite(RGBGreenPin, HIGH);

analogWrite(RGBRedPin, LOW);

analogWrite(RGBBluePin, LOW);

break;

case 102: analogWrite(RGBRedPin, HIGH);

analogWrite(RGBBluePin, LOW);

analogWrite(RGBGreenPin, LOW);

break;

case 103: analogWrite(RGBBluePin, HIGH);

analogWrite(RGBRedPin, LOW);

analogWrite(RGBGreenPin, LOW);

break;

case 200: analogWrite(RGBGreenPin, LOW); break;

case 201: analogWrite(RGBRedPin, LOW); break;

case 202: analogWrite(RGBBluePin, LOW); break;

// Question 3

case 300: digitalWrite(redLED, HIGH); break;

case 301: digitalWrite(redLED, LOW); break;

} // end of switch

//QUESTION 4 & 5

if (analogRead(lightPin) < 500) {//read data from light sensor if greater than 500(bright)

digitalWrite(blueLED, HIGH);

delay(100);

}

else {

digitalWrite(blueLED, LOW);

delay(10000);

}

}

## Microsoft Visual Studio (C#):

using System;

using System.IO.Ports;

using System.Windows.Forms;

namespace AT2\_Project

{

    public partial class Form1 : Form

    {

        //declare port variable

        public SerialPort myPort;

        public Form1()

        {

            InitializeComponent();

        }

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

        {

            try

            {

                myPort = new SerialPort();

                myPort.BaudRate = 9600;

                myPort.PortName = cmbPort.Text; // COM port from user selection

                myPort.Open();

                myPort.DataReceived += myPort\_DataReceived; // call method

            }

            catch (Exception)

            {

                MessageBox.Show("Not Connected. Please Try Again", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

            }

        }

        private void myPort\_DataReceived(object sender, SerialDataReceivedEventArgs e)

        {

            string line = myPort.ReadLine(); //read data from Arduino

            BeginInvoke(new LineReceivedEvent(LineReceived), line);

        }

        private delegate void LineReceivedEvent(string line);

        // conver raw data into correct format

        private void LineReceived(string line)

        {

            String[] dataHumTemp = line.Split(',');

            tbOutsideTemp.Text = dataHumTemp[0];

            tbInsideTemp.Text = dataHumTemp[1];

            tbInsideHumidity.Text = dataHumTemp[2];

        }

        // turn on Green RGB light when radio button is checked

        private void RbGreen\_CheckedChanged(object sender, EventArgs e)

        {

            try

            {

                if (rbGreen.Checked)

                {

                    myPort.WriteLine("101");

                }

                else

                {

                    myPort.WriteLine("200");

                }

            }

            catch (Exception)

            {

                MessageBox.Show("Please Select A Port and Press Connect First", "Try Again", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

            }

        }

        //Turn on Red RGB light when radio button is checked

        private void RbRed\_CheckedChanged(object sender, EventArgs e)

        {

            try

            {

                if (rbRed.Checked)

                {

                    myPort.WriteLine("102");

                }

                else

                {

                    myPort.WriteLine("201");

                }

            }

            catch (Exception)

            {

                MessageBox.Show("Please Select A Port and Press Connect First", "Try Again", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

            }

        }

        //Turn on Blue RGB light when radio button is checked

        private void RbBlue\_CheckedChanged(object sender, EventArgs e)

        {

            try

            {

                if (rbBlue.Checked)

                {

                    myPort.WriteLine("103");

                }

                else

                {

                    myPort.WriteLine("202");

                }

            }

            catch (Exception)

            {

                MessageBox.Show("Please Select A Port and Press Connect First", "Try Again", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

            }

        }

        private void RbOff\_CheckedChanged(object sender, EventArgs e)

        {

            try

            {

                myPort.WriteLine("200");

                myPort.WriteLine("201");

                myPort.WriteLine("202");

            }

            catch (Exception)

            {

                MessageBox.Show("Please Select A Port and Press Connect First", "Try Again", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

            }

        }

        //Red LED toggle indicated when button is clicked

        bool RedLedStatus = true;

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

        {

            if (RedLedStatus)

            {

                myPort.WriteLine("300");

                RedLedStatus = false;

            }

            else

            {

                myPort.WriteLine("301");

                RedLedStatus = true;

            }

        }

        //Method to set the max temperature

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

        {

            try

            {

                if (int.Parse(cmbTempMax.Text) == 25)

                {

                    myPort.WriteLine("25");

                }

                else if (int.Parse(cmbTempMax.Text) == 26)

                {

                    myPort.WriteLine("26");

                }

            }

            catch (Exception)

            {

                MessageBox.Show("Please Select Something", "Try Again", MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation);

            }

        }

    }

}