

|  |
| --- |
| **F I N A L D E S I G N R E P O R T** |
|  |

SMART COOLING FAN

**GROUP E**

Mahmoud Kakouri (7059875)

Suraksha Kotte (6996656)

Mohammed Zaid (7001897)

Michael Madu (6861477)

Table of Contents

[1. Executive summary 3](#_Toc106576197)

[2. Team Roles 4](#_Toc106576198)

[3. Introduction and overview 4](#_Toc106576199)

[4. Final Design Physical specifications 5](#_Toc106576200)

[4.1 DHT11 Sensor 5](#_Toc106576201)

[4.2 Motor Driver 6](#_Toc106576202)

[4.3 Potentiometer 7](#_Toc106576203)

[4.4 Arduino UNO 8](#_Toc106576204)

[5. Final code design specifications 9](#_Toc106576205)

[5.1 Serial monitor 9](#_Toc106576206)

[5.2 LCD 10](#_Toc106576207)

[5.3 Motor 10](#_Toc106576208)

[5.4 BLYNK 11](#_Toc106576209)

[6. Construction details 12](#_Toc106576210)

[7. Testing Details 13](#_Toc106576211)

[8. Budget 14](#_Toc106576212)

[8.1 Consultation Budget: 15](#_Toc106576213)

[8.2 Cash Flow chart: 16](#_Toc106576214)

[9. Marketing 16](#_Toc106576215)

[10. Conclusion 16](#_Toc106576216)

[11. Appendix 17](#_Toc106576217)

[10.1 Appendix A: 17](#_Toc106576218)

[10.2 Appendix B: 18](#_Toc106576219)

[10.3 Appendix C: 19](#_Toc106576220)

[10.4 Appendix D: 19](#_Toc106576221)

[12. References 20](#_Toc106576222)

# Executive summary

Intelligent systems are being introduced every day as technology advances. Everything is becoming more complex and understandable. The demand for cutting-edge technologies and sophisticated electrical systems is increasing. However it appears that even the advancement of technology cannot effectively and efficiently prevent themselves without the help of a manual controller or human in areas that could destroy them such as fires which is proven to be one of the most causes of destruction of information and lives, this problem can be solved using a device called a smart cooling system which uses same technology from a temperature controlled fan applied to fan with cooling abilities which takes in the ambient air in the environment and circulates it back as cool air.

This device was made used a microcontroller which is one of the most important parts of basically the *brain* of the project. Microcontrollers are crucial in the creation of smart systems because they provide the system with a brain. They have become the brains behind many of the new technologies that are being released on a daily basis.

A microcontroller is a single-chip microprocessor that is used to automate and control machines and processes. On a single chip, a microcontroller like Arduino contains a Central Processing Unit (CPU), timers and counters, interrupts, memory, input/output ports, and analog to digital converters (ADC). The size of the control board is decreased, and power consumption is minimized thanks to the microcontroller's single chip integrated circuit design.

The microcontroller used in this project is *Arduino*. Arduino is a free and open-source electronics platform with simple hardware and software. They can take inputs - such as light from a sensor, a finger on a button, or a Twitter message - and convert them to outputs - such as turning on an LED, triggering a motor, or publishing anything online. By providing a set of instructions to the board's microcontroller, you may tell it what to do.

The Arduino programming language (based on Wiring) and the Arduino Software (IDE) (based on Processing) are used to do this. Thousands of projects have used Arduino throughout the years, ranging from simple household items to complicated scientific apparatus (*Kaushik,2018*).

This paper shows how this project was carried out as well as showing how to create and simulate a fan speed control system depending on room temperature. The temperature of the room is measured using a DHT11 temperature sensor, and the fan speed changes according to the temperature.

# Team Roles

Everyone in the team had an important role in this project, and everyone worked hard to reach a successful outcome. The team members were well-coordinated and helped each other accomplish the desired result. Suraksha was in charge of the project's hardware components and design. Mahmoud, on the other hand, oversaw the project's software and managed the marketing strategies needed, whereas Mohammed Zaid was in charge of the circuit connections, and lastly, Michael Chisom handled the virtual connections to the project, which included the BLYNK layout. All group members contributed in constructing the final circuit and device.

# Introduction and overview

The title of this project is called **Smart Cooling System** and its function is to control the speed of the fan attached to the device according to the temperature read by the temperature sensor on the device and this is achieved by using a temperature-controlled fan. The device can be used in various areas which includes server rooms, pharmaceutical labs, homes etc... We added functionalities in our project that would help the user control and monitor the temperature when they are away from the device using Blynk which therefore fulfils the theme of the project which is the **Internet of Things (IOT)**.

Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and many other things. (Rif'an, 2022).

DHT11 sensors are used to monitor and control temperature, and this is the temperature sensor we used in our project. It is a basic digital temperature sensor with a low-price tag. It spits out a digital signal on the information pin using an electrical phenomenon and a semiconductor device to live the surrounding air. It's simple to use, but it takes a little longer to set up. The DHT11 sensor collects data, which is then sent to the Arduino board, which is a microcontroller circuit.

The temperature-based fan speed control system can be implemented using an Arduino board and an electronic circuit. Because the Arduino board is now quite advanced among all electronic circuits, we used it to control the fan speed. The proposed system is intended to detect the room's temperature and deliver that data to the Arduino board. The Arduino board then compares the current temperature to the set temperature using the Arduino's built-in application. The result of the operation is sent to the LCD display of associated data via the output port of an Arduino board.

A temperature-based fan is useful for "more efficiently" cooling the processor in laptops and personal computers. In most laptop fans, there are just two or three speeds available. As a result, there is an increase in power consumption. The fan used in this project has several speed settings depending on the temperature. This can also be used to cool electrical/mechanical equipment in small businesses. The entire circuit, with the exception of the motor and fan, can be built on a single PCB and used for temperature-based control activities.

# Final Design Physical specifications

This section will explain the different components used in our project design and their specifications.

## 4.1 DHT11 Sensor

The DHT11 sensor is an inexpensive digital temperature and humidity sensor. It is mainly composed of a capacitive humidity sensing element and a temperature sensing thermistor. It sends a digital signal to the data pin linked to the LCD display. It is simple to use, but data collecting needs precise timing. (ElProCus - Electronic Projects for Engineering Students, 2019)

DHT11 is a small device with an operational voltage range from 3 to 5 volts. The highest measuring current is 2.5mA. DHT11 has a temperature range of 0 to 50 degrees Celsius with a 2-degree accuracy. This sensor has a humidity range of 20 to 90 percent with a 5 percent accuracy. This sensor has a sampling rate of 1Hz, which means it takes one reading per second. (Components101, 2021)

A picture containing text, electronics

Description automatically generated

*Figure 4.1.1: DHT11 sensor*

+ (VCC) pin supplies power for the sensor. Although the supplied voltage ranges from 3.3V to 5.5V, a 5V supply is suggested.

Data pin is used to communication between the sensor and the Arduino. It sends both temperature and humidity as serial data.

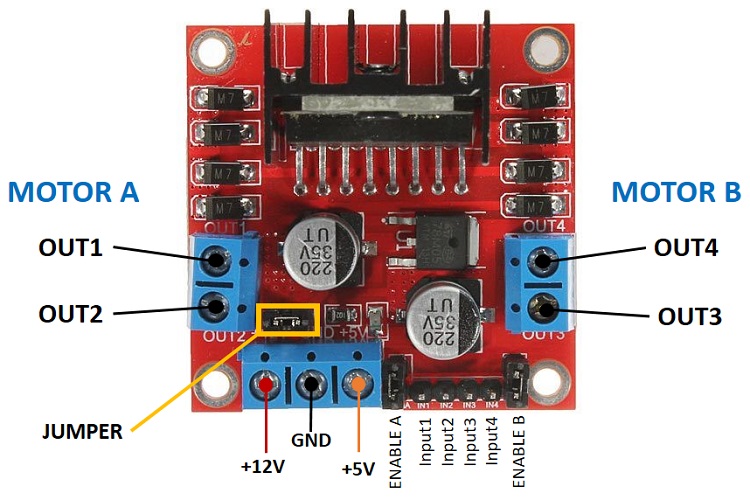
– (GND) should be connected to the ground of Arduino.

(Components101, 2021)

## 4.2 Motor Driver

The L298N Motor Driver is an H-Bridge controller that allows you to quickly control the direction and speed of up to two DC motors. Motor drivers are the interface between motors and control circuits. The motor needs a large current, yet the controller circuit works with minimal current signals. Motor drivers are responsible for converting a low-current control signal into a higher-current signal capable of operating a motor. (Works, 2017)

PWM (pulse width modulation) facilitates in voltage regulation and is thus used to control motor speed. To regulate the motor driver's speed, we must provide PWM signals to the motor enable pins. The motor's speed will change depending on the width of the pulses. The quicker the motor turns, the wider the pulses. (Pelayo, 2018)



*Figure 4.2.1: Motor Driver*

The Double H Bridge L298N Drive Chip logical voltage is 5V and the drive voltage is from 5V to 35V. The range of the logical current varies from 0 to 36mA and the drive current is 2A (MAX single bridge). Also, the maximum power is 25 Watts. (Robotpark.com, 2015)

## 4.3 Potentiometer

A potentiometer is a three-terminal variable resistor that can be adjusted manually. Two terminals are attached to opposing ends of a resistive element, while the third terminal is connected to a sliding contact, known as a wiper, that moves over the resistive element. The potentiometer is fundamentally a variable resistance divider.

When a reference voltage is provided across the end terminals, the position of the wiper controls the potentiometer's output voltage. The sliding contact on the semi-circular resistance can be moved by twisting the knob. The voltage is measured across a resistance end contact and a sliding contact. (Keim, 2022)

The Rotary, also known as the Radio POT, is available in a variety of resistance levels, including 500, 1K, 2K, 5K, 10K, 22K, 47K, 50K, 100K, 220K, 470K, 500K, and 1 M. The resistance or value determines how much opposition it causes to the flow of current. The electricity will flow more slowly as the resistance value increases. The power rating is around 0.3 Watts, with a maximum input voltage of 200 Vdc. The rotational life is estimated to be around 2000K cycles. (Components101, 2017)

Diagram

Description automatically generated

*Figure 4.3.1: Potentiometer*

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Pin Name** | **Description** |
| 1 | Fixed End | This end is connected to one end of the resistive track |
| 2 | Variable End | This end is connected to the wiper, to provide variable voltage |
| 3 | Fixed End | This end is connected to another end of the resistive track |

*Table 4.3.1: Connections*

(Components101, 2017)

## 4.4 Arduino UNO

The ATmega328P-based Arduino UNO is a microcontroller board. It is a low-cost, adaptable, and user-friendly open-source programmable microcontroller board that can be integrated into a range of electronic applications. As an output, this board can operate relays, LEDs, servos, and motors and may be interfaced with other Arduino boards, Arduino shields, and Raspberry Pi boards.

A picture containing text, electronics, circuit

Description automatically generated

*Figure 4.4.1: Arduino Uno*

This microcontroller board has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. (Arduino.cc, 2022).

The Arduino Uno board can be powered either by USB or by an external power supply. The power source is automatically selected. External (non-USB) power can be supplied by either an AC-to-DC adaptor (wall-wart) or a battery. The operating voltage is 5V.​The recommended input voltage is from 7 to 12 Volts while the input voltage limit varies from 6 to 20 volts. The DC current per input/output pin is 20 mA. (TOMSON ELECTRONICS, 2018)

# Final code design specifications

Moving on to the code specifications of our air-cooling system, we decided to break the code down to the following four parts:

## 5.1 Serial monitor

The serial monitor is a way to show or print the output of a code in the app itself (software), and we decided to use that feature to display the current temperature and humidity, and to do that we wrote and compiled the following code:

Graphical user interface, text, application

Description automatically generated

Figure 5.1.1

As you can see in figure 5.1, we started by defining the temperature sensor’s pin as A0 and then using the “DHT.read” function, the code was able to read and save the current temperature and humidity, we decided to use a delay function so that the code waits 1 second between each read and print of the results, and finally, we used the “serial print” function and we were able to print the current humidity and temperature as you can see below:

Text

Description automatically generated

Figure 5.1.2

## 5.2 LCD

The next part of the code focuses on the LCD, the LCD is just our way to show the current temperature to the user on a hardware device unlike the serial print, and following a similar approach to the serial monitor, we were able to produce the following part of the code:

Graphical user interface, text, application

Description automatically generated

Figure 5.2.1

We started by defining a variable “Vout” to read the DHT sensor’s values using analogRead, and then using the “lcdprint” function, we printed the temperature in Celsius and Fahrenheit.

## 5.3 Motor

Moving on to the third part of the code, we will be looking at the motor code as well as the if statements, as we mentioned in previous parts of this report, if the temperature is less than 16 degrees Celsius, the fan will turn off, else if the temperature is between 16 and 35 degrees, the fan will work on a low speed, and finally if it is above 35 degrees, the fan will work on full speed, using that information, we compiled the following code section:

Graphical user interface, text, application

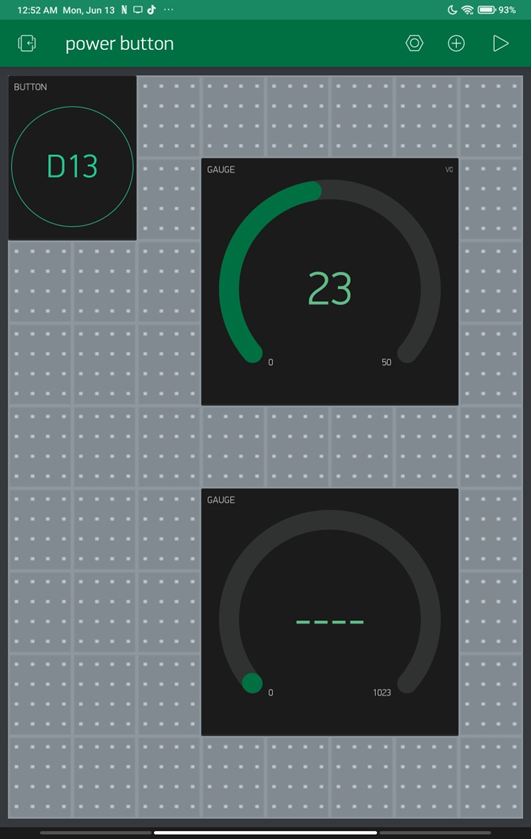
Description automatically generated

Figure 5.3.1

As you can see above, we used a digitalWrite function to turn off and turn on the fan, where HIGH is on and LOW is for off, and finally to control the speed of the fan, there was two approaches, one is by using the motor driver speed control, while the other one is by using a short delay in the code, and we decided to use a delay function in the code when we wanted the fan to work at a moderate speed as you have seen in figure 5.3.1.

## 5.4 BLYNK

A temperature and humidity gauge function were also added in the blynk layout along with the power button. The functionality of the graph is to display the temperature changes and it is done live i.e. it updates regularly to display change in temperature. This function is important because it helps the user to monitor the temperature. As stated before this device is going to be used in the server rooms ,control units and labs to the monitor and maintain the temperature of those rooms and if there happens to be any sudden change in temperature or humidity the user or management can discover it early and take measures even if they are not around the area as it works remotely to prevent loss of information, lives or data which simultaneously minimizes/eliminates cost for repair and saves time .

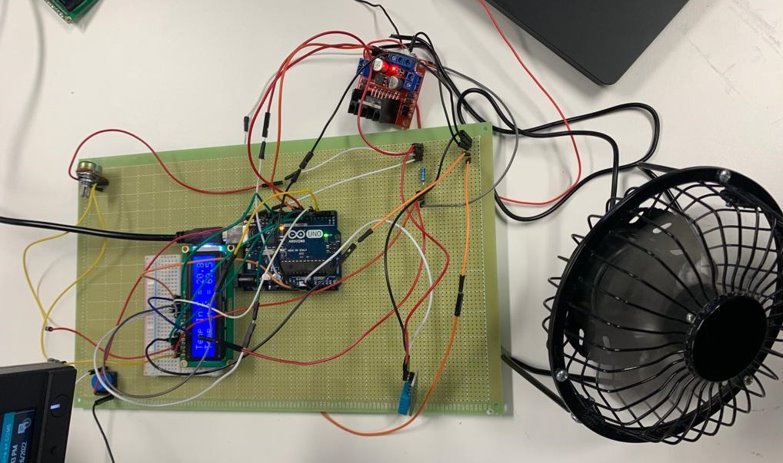


*Figure 5.4.1: Blynk App Layout*

# Construction details

This cooling system unit has been assembled in such a way that it's less messy, lowering the number of wires and giving it a minimal look. The way we went about the assembly is that we made a virtual representation of the cooling system on tinkercad and on multisim, tried to simulate it and see the working of it. After making a virtual prototype, the construction of the cooling system started, and we put all together as we did on the tinkercad.

As mentioned in the apparatus used, we have the DHT11 which is a temperature and humidity sensor, motor driver for the fan to spin, Arduino Uno which controls the whole module and has the code, perfoboard for the connections and a manual button for it to turn on and we have a virtual button on blynk app as well. Firstly, we started with connections from LCD to the Arduino, as you can see the in the below picture that the 5v of the Arduino has been connected on to the perfoboard and joined together with the other components which need the power source and have been soldered on together. And the components which need ground have been soldered on to the perfoboard together too. As you can see, we are using an L298N motor driver for the fan to spin and a potentiometer for the dimming and brightening of the lcd. The connections of the lcd have been connected the same way we had learned in the lab and the dht11 temperature and humidity sensor has been connected so that it can sense the temperature the values can be displayed onto the lcd screen. As the temperature has been sensed, the fan spins accordingly as it has been written in the code section of the report.



*Figure 6.1: Final Circuit*

As mentioned above in the hardware components section, the following parts were used in our circuit:

1. DHT11 Sensor

2. Motor Driver

3. Potentiometer

4. Arduino Uno

# Testing Details

After completing both, the final physical design and the software code design for our product, the last step was to test and check if our device is meeting the requirements and expectations set.

To do that, we started by connecting our Arduino to a computer to upload the designed code (refer to section 5), then we moved on to connect our fan to a DC voltage power supply with around 8-9V, that way we were able to give our fan enough power to start spinning and cool down when needed, after setting up, we uploaded the code and opened the serial monitor to check the current temperature which was 21 degrees Celsius at the time as well as the LCD that showed the same exact temperature, this led us to believe that the fan is working at a moderate speed since it is between 16 and 35 degrees Celsius, and to make sure that the fan would speed up if the temperature exceeded 35 degrees, we decided to take our device outside the university in the afternoon heat with a 39 degrees Celsius weather, and as we expected, the fan was spinning faster than it was when the temperature was at 21 degrees Celsius, as well as showing the temperatures on the LCD accurately at real time, this proves that our final physical and code design of the device met the requirements and that it was done successfully.

# Budget

We have a budget of 900 Dirhams for parts for this project, which we hope to use efficiently while leaving a modest amount for any future revisions. A breakdown of the parts and their prices may be found below.

|  |  |
| --- | --- |
| ***Components*** | ***Costs in AED*** |
| *Arduino and Arduino Shield Kit* | *600* |
| *DC Air Cooling Axial Fan* | *50* |
| *DHT11 Temperature sensor* | *10* |
| *DHT22 Humidity Sensor* | *15* |
| *LCD Display* | *13* |
| *Full Sized Breadboards and wires* | *Included* |
| *555 Timer* | *5* |

*Table 7.1: Components Cost*

As you can see in Table, the Arduino and Arduino shield kit will cost us around 600 Arab Emirates Dirhams, and includes the Arduino Uno, Arduino Shield, full sized Breadboard, Ethernet cable, and finally male to male wires. The second part we'll need to get this project working is a DC air cooling axial fan, which is the fan we'll use to cool the air in the room as mentioned previously in the design section of this report. After doing some research, we discovered that this sensor has been proven to work the best in terms of sensing temperature when working with Arduino Uno; thus, using this specific component would be the safest option, and it costs 10 Dirhams, as well as the DHT22 Humidity sensor, which we would also need to precisely measure the temperature of the air and use it to determine how fast the fan speed should be spinning; this sensor costs a little more than the DHT11 temperature sensor, and it costs a little more than 15 AED, as well as the speed of the fan and it costs 13 Dirhams to be exact, the next thing you will notice on the table is the breadboard and the wires, but as mentioned before, these parts are included in the Arduino Uno and Arduino shield Kit with the cost of 600 AED, The final component we will need would be the 555 Timer, this part would able us to make delays and control the fan speed and when to stop.

*Figure 8.1.1: Pie Chart*

## 8.1 Consultation Budget:

The consultation Budget divides into the 3 following parts:

**Consultation with academic staff**

* Budget of 1000 AED and the price is 500 AED/Hour
* Minimum Consultation allowed is 30 minutes
* one 30 minutes meeting every two weeks (odd weeks) to ask for advice and clear doubts.

**Consultation with Tutor/Lab instructor**

* Budget of 800 AED
* Costs 400 AED/Hour
* Minimum of 30 minutes Consultation
* 130 minutes meeting every two weeks (even weeks) to discuss issues in building our cooling system and get the advice needed.

**Consultation with mentor**

* No budget
* No costs
* Free 15 minutes weekly
* A meeting every week with the mentor to seek their professional opinion and advice as well as briefly updating them with our progress.

## 8.2 Cash Flow chart:

Our product is expected to sell for 900 AED, and parts will cost roughly 705 AED, thus our cash flow table will look like this:

|  |  |  |
| --- | --- | --- |
| **Year** | **Cash outflow** | **Cash Inflow** |
| First | 800+705x10 =7850AED | 10 Sold, 9000 AED |
| Second | 705x20 =14100AED | 20 Sold, 18000AED |
| Third | 705x40 =28200AED | 40 Sold, 360000AED |
| Fourth | 705x80 =56400AED | 80 Sold, 720000AED |
| Fifth | 56400AED | 80 Sold, 720000AED |

*Table 8.2: Cash flow chart*

# Marketing

For our commercializing and marketing strategy, we decided to follow a 5-step plan as you can see below:

1. Defining the product
2. Picking the target audiences
3. Defining the advantages of your product
4. Choosing a main marketing platform
5. Launch your product

The first step for any marketing plan is to define the product and as mentioned in the previous parts of this report, we are building a cooling fan that changes the speed according to the temperature variances.

We are mainly focused on cryptocurrency miners who use a heavy-duty personal computers to mine cryptocurrencies and that puts a huge level of strain on the computers which raises the temperature of it and causes it to crash and turn off, as well as places with server rooms that generate a lot of heat where you can use our cooling system to cool off the servers and stops them from overheating, it can also be used as a daily use cooling system for people.

We decided to use social media as our main marketing and commercializing platform, as it is the most used marketing way in the 21st century, we started by making an Instagram account called @uowdaircoolingsystem and we posted our journey of making the product which builds trust and confidence with the customers.

# Conclusion

As a team, we would like to conclude this project that has been going on for two semesters by putting all of our hard work and dedication into bringing everything together and making this work. We would like to thank Mr. Malek, our lecturer for the course, and Mr. Kiyan for always being available to assist us with the project. We began by wondering whether we could manage the temperature of the room without physically changing or moving a finger, and if the fan could speed up and slowdown in response to temperature and humidity. N ow, after two semesters, we've been able to make the notion a reality.

# Appendix

## 10.1 Appendix A:

We had 6 meetings overall in this semester.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Time of commencement | Attendance | Agenda | End of meeting | Location |
| 23/04/2022 | 15:00pm | All members of the team | Assigned roles for deliverable 4. Each member was assigned to work on 2 circuits. | 15:30pm | Webex |
| 30/04/2022 | 13:00pm | All members of the team | Discussed about the progress and wrote a list of questions to ask the professor. | 13:45pm | Webex |
| 14/05/2022 | 13:00pm | All members of the team | Discussed about the feedback given from the lab mentor and professor. We brushed through the next deliverable guidelines. | 13:25pm | Webex |
| 21/05/2022 | 14:00pm | All members of the team | Assigned roles for deliverable 5. Discussed what changes we should do on campus. | 14:35pm | Webex |
| 28/05/2022 | 12:00pm | All members of the team | Talked about the circuit modifications. We also worked on the blynk app online. | 12:30pm | Webex |
| 11/06/2022 | 13:00pm | All members of the team | Rechecking with each other if the work assigned to each other is done. Also, talked about the final design report and presentation guidelines. | 13:25pm | Webex |

## 10.2 Appendix B:

Schematics of the prototype circuits

A picture containing electronics

Description automatically generated

*Figure 6.1: Final Circuit*

A screenshot of a cell phone

Description automatically generated with medium confidence

*Figure 5.4.1: Blynk App Layout*

## 10.3 Appendix C:

**Arduino source code**

#include <dht.h>

#define BLYNK\_PRINT DebugSerial

#define dht\_apin A0 // Analog Pin sensor is connected to

#define BLYNK\_PRINT Serial

dht DHT;

#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

SoftwareSerial DebugSerial(2, 3);

#include <BlynkSimpleStream.h>

#include <SPI.h>

BlynkTimer timer;

char auth[] = "kdzv8IH5fvQeXyKWocoICzgwiKKGiBP-";

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int tempsensor = A0; // Assigning analog pin A0 to variable 'tempsensor'

float tempc; //variable to store temperature in degree Celsius

float tempf; //variable to store temperature in Fahreinheit

float vout; //temporary variable to hold sensor reading

const int buttonPin = 7; // the number of the pushbutton pin

const int ledPin = 13;

int ledState = HIGH; // the current state of the output pin

int buttonState; // the current reading from the input pin

int lastButtonState = LOW;

int motor1pin1 = 6;

int motor1pin2 = 7;

unsigned long lastDebounceTime = 0; // the last time the output pin was toggled

unsigned long debounceDelay = 50;

void setup() {

// Debug console

DebugSerial.begin(9600);

// Blynk will work through Serial

// Do not read or write this serial manually in your sketch

Serial.begin(9600);

Blynk.begin(Serial, auth);

pinMode(buttonPin, INPUT);

pinMode(ledPin, OUTPUT);

// Debug console

Serial.begin(9600);

// set initial LED state

digitalWrite(ledPin, ledState);

delay(500);//Delay to let system boot

Serial.println("DHT11 Humidity & temperature Sensor\n\n");

delay(1000);//Wait before accessing Sensor

pinMode(tempsensor, INPUT); // Configuring pin A0 as input

lcd.begin(16, 2);

delay(500);

pinMode (motor1pin1, OUTPUT);

pinMode (motor1pin2, OUTPUT);

}

void myTimerEvent()

{

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V5, 55);

}

//void setup2()

//{

// // Debug console

// Serial.begin(9600);

//

// Blynk.begin(auth);

//

// // Setup a function to be called every second

// timer.setInterval(1000L, myTimerEvent);

//}

//

//void setup1()

//{

// // Debug console

// DebugSerial.begin(9600);

//

// // Blynk will work through Serial

// // Do not read or write this serial manually in your sketch

// Serial.begin(9600);

// Blynk.begin(Serial, auth);

//}

void loop() {

//Start of Program

Blynk.run();

timer.run();

int reading = digitalRead(buttonPin);

// check to see if you just pressed the button

// (i.e. the input went from LOW to HIGH), and you've waited long enough

// since the last press to ignore any noise:

// If the switch changed, due to noise or pressing:

if (reading != lastButtonState) {

// reset the debouncing timer

lastDebounceTime = millis();

}

if ((millis() - lastDebounceTime) > debounceDelay) {

// whatever the reading is at, it's been there for longer than the debounce

// delay, so take it as the actual current state:

// if the button state has changed:

if (buttonState == LOW) {

Serial.print("0");

// only toggle the LED if the new button state is HIGH

if (buttonState == HIGH) {

Serial.print("1");

}

}

}

// set the LED:

digitalWrite(ledPin, ledState);

// save the reading. Next time through the loop, it'll be the lastButtonState:

lastButtonState = reading;

DHT.read11(dht\_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

Serial.println("C ");

delay(1000);//Wait 5 seconds before accessing sensor again.

if ((DHT.temperature > 16 && DHT.temperature < 35)) {

digitalWrite(motor1pin1, HIGH);

// turn off others

}

else if ((DHT.temperature > 35)) {

digitalWrite(motor1pin1, HIGH);

}

else if ((DHT.temperature < 16)) {

digitalWrite(motor1pin1, LOW);

}

vout = analogRead(DHT.temperature);

DHT.temperature = vout; // Storing value in Degree Celsius

lcd.setCursor(0, 0);

lcd.print("Temp In C = ");

lcd.print(vout / 45.6);

lcd.setCursor(0, 1);

lcd.print("Temp In F = ");

lcd.print(((vout / 45.6) \* 1.8) + 32);

delay(1000);

//Fastest should be once every two seconds.

}

## 10.4 Appendix D:

The members had equal individual contribution to the report.

|  |  |
| --- | --- |
| Name | Team Roles |
| Mahmoud Kakouri | Software design, Marketing, Testing Details |
| SuraksHa Kotte | Hardware design, Appendix, Overall editing of report |
| Michael Madu | Blynk app, Introduction, Executive summary |
| Mohammed Zaid | Construction details, Budget, Team roles, Conclusion |

# References

Rif'an, M., 2022. *Internet of Things (IoT): BLYNK Framework for Smart Home | KnE Social Sciences*. [online] Knepublishing.com. Available at: <https://knepublishing.com/index.php/Kne-Social/article/view/4128/8495#:~:text=Blynk%20is%20an%20IoT%20platform,address%20on%20the%20available%20widgets.> [Accessed 4 June 2022].

Kaushik, S., Chouhan, Y.S., Sharma, N., Singh, S. and Suganya, P., 2018. Automatic fan speed control using temperature and humidity sensor and Arduino. *Int. J. Adv. Res*, *4*(2), pp.453-467.

Components101. (2021). *DHT11–Temperature and Humidity Sensor*. [online] Available at: https://components101.com/sensors/dht11-temperature-sensor [Accessed 15 Jun. 2022].

ElProCus - Electronic Projects for Engineering Students. (2019). *DHT11 Sensor Definition, Working and Applications*. [online] Available at: https://www.elprocus.com/a-brief-on-dht11-sensor/ [Accessed 18 Jun. 2022].

Works, R. (2017). *Choosing the right motor-driver*. [online] Sproboticworks.com. Available at: https://sproboticworks.com/blog/choosing-the-right-motor-driver [Accessed 18 Jun. 2022].

Pelayo, R. (2018). *How to Use L298N Motor Driver*. [online] Microcontroller Tutorials. Available at: https://www.teachmemicro.com/use-l298n-motor-driver/ [Accessed 18 Jun. 2022].

Robotpark.com. (2015). *L298N Dual H Bridge Motor Driver*. [online] Available at: https://www.robotpark.com/L298N-Dual-H-Bridge-Motor-Driver [Accessed 18 Jun. 2022].

Components101. (2017). *Potentiometer*. [online] Available at: https://components101.com/resistors/potentiometer [Accessed 19 Jun. 2022].

Keim, R. (2022). *Potentiometer*. [online] Eepower.com. Available at: https://eepower.com/resistor-guide/resistor-types/potentiometer/# [Accessed 19 Jun. 2022].

Arduino.cc. (2022). *UNO R3 | Arduino Documentation | Arduino Documentation*. [online] Available at: https://docs.arduino.cc/hardware/uno-rev3 [Accessed 19 Jun. 2022].

TOMSON ELECTRONICS. (2018). *Arduino Uno Specification*. [online] Available at: https://www.tomsonelectronics.com/blogs/news/arduino-uno-specification [Accessed 19 Jun. 2022].