Project Report



**Title:** **Mini Weather Station Using Arduino Uno**

**Submitted to:**

**Faculty:** Mr. Intisar Tahmid Naheen

**Submission Date:** 27th December, 2019

**Course:** CSE299 [Junior Design]

**Section:** 03

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| 1. Momena Akhter Shukhi | 1620763642 |
| 2. Tahrim Faroque Tushar | 1621148642 |
| 3. Samiha Ferdous | 1620834042 |

# **Acknowledgement**

A big thanks and gratitude to Mr. Intisar Tahmid Naheen, for giving us the opportunity to make this project. We are thankful to those friends who helped us greatly with their valuable suggestions, comments, and constructive criticism. The GitHub, Trello, Slack, YouTube community have hugely helped us with what we needed to know about Arduino and progress with our project.

The project has helped us in gaining knowledge and experience regarding embedded systems. Thank you for giving us this wonderful opportunity.

**Table of Contents**

[**Acknowledgement** 2](#_Toc28375313)

[**Abstract** 4](#_Toc28375314)

[**Introduction** 4](#_Toc28375315)

[**Budget** 6](#_Toc28375316)

[**Work distribution:** 6](#_Toc28375317)

[**Time-plan** 7](#_Toc28375318)

[**Methodology** 8](#_Toc28375319)

[**Procedure** 9](#_Toc28375320)

[**Data Analysis** 11](#_Toc28375321)

[**Result** 16](#_Toc28375323)

[**Discussion** 17](#_Toc28375324)

[**Conclusion** 18](#_Toc28375325)

[**Reference List** 18](#_Toc28375326)

[**Appendix** 19](#_Toc28375327)

[**Screenshots:** 24](#_Toc28375328)

# **Abstract**

Our project is about creating a mini weather station using the popular microcontroller, Arduino Uno. The report focuses on the process of creation of this system and less so on its application. We have included the code in the appendix and discussed about in details in the methodology part. We have briefly discussed about the different parts that we have used in our project.

Arduino Uno is used here as the microcontroller, DHT11 to measure temperature and humidity, A water sensor to detect the water level for various applications and MQ135 the Air quality sensor for detecting a wide range of gases. When the sensors will start taking data, they will be stored in an excel sheet (the time interval is 2 minutes in our project).

# **Introduction**

Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. It can be used to program various electronic devices. We are using this microcontroller to create a mini weather station. Our mini weather station can be implemented in various real-world cases. It is considered a technical method that allows measuring weather parameters based atmospheric conditions for a proposed location with specific devices in order to realize forecasted weather conditions, and to study climate properties. The collected data allows deciding and confirming the warranty of the proposed location.

Mini weather station is a simple project with a vast array of applications. It’s simplicity makes it cheap and easy to manufacture in mass scale.

## **Background Research**

Back in last semester we did a project on measuring the temperature and humidity using an API on what’s app on the course operating system. We were influenced by that. So we thought of making a project based on that so we went for studying it and came across a project which is made in much bigger form so we thought of making a smaller version of this project as this weather plays a vital role in our everyday life.

https://howtomechatronics.com/tutorials/arduino/arduino-wirelessweather-stationproject/?fbclid=IwAR12EiqKcA0liw1RYh38DL2ngtkoGxXgKgO1D 7xkpsjSTVneINegoEytnro .

## **Aim**

Our aim of this project is to make a website that can be checked by users to know about real-time weather updates in a particular place and inform others about the current weather station of methodology department to submit their weather synopsis to the system, every 2 minutes.

## **Objective**

* Study about weather forecasting applications and system.
* Observe, measure, and record the basic elements of weather.
* Design and develop a system which can properly get updates of the current weather status.
* Evaluate proposed solutions.
* Handover successful system to the company.

# **Budget**

The total estimated budget to complete the project is provided in this table:

|  |  |
| --- | --- |
| **ITEM** | **COST** |
| 1.Arduino UNO and Genuino UNO | BDT 405 |
| 2. Water Sensor Module | BDT 105 |
| 3.Arduino ProtoShield/ Breadboard | BDT 79 |
| 4. HC-SR04 ULTRASONIC SENSOR | BDT 89 |
| 5. Coin Type Vibration Motor (10mm) | BDT 73 |
| 6. Barometric Pressure Sensor BMP180 | BDT 277 |
| 7. DHT11 Temperature and Humidity Sensor | BDT 118 |
| 8.Resistors- 1k,10k,4.7k | BDT 15 |
| **Total** | **BDT 1155** |

# **Work distribution:**

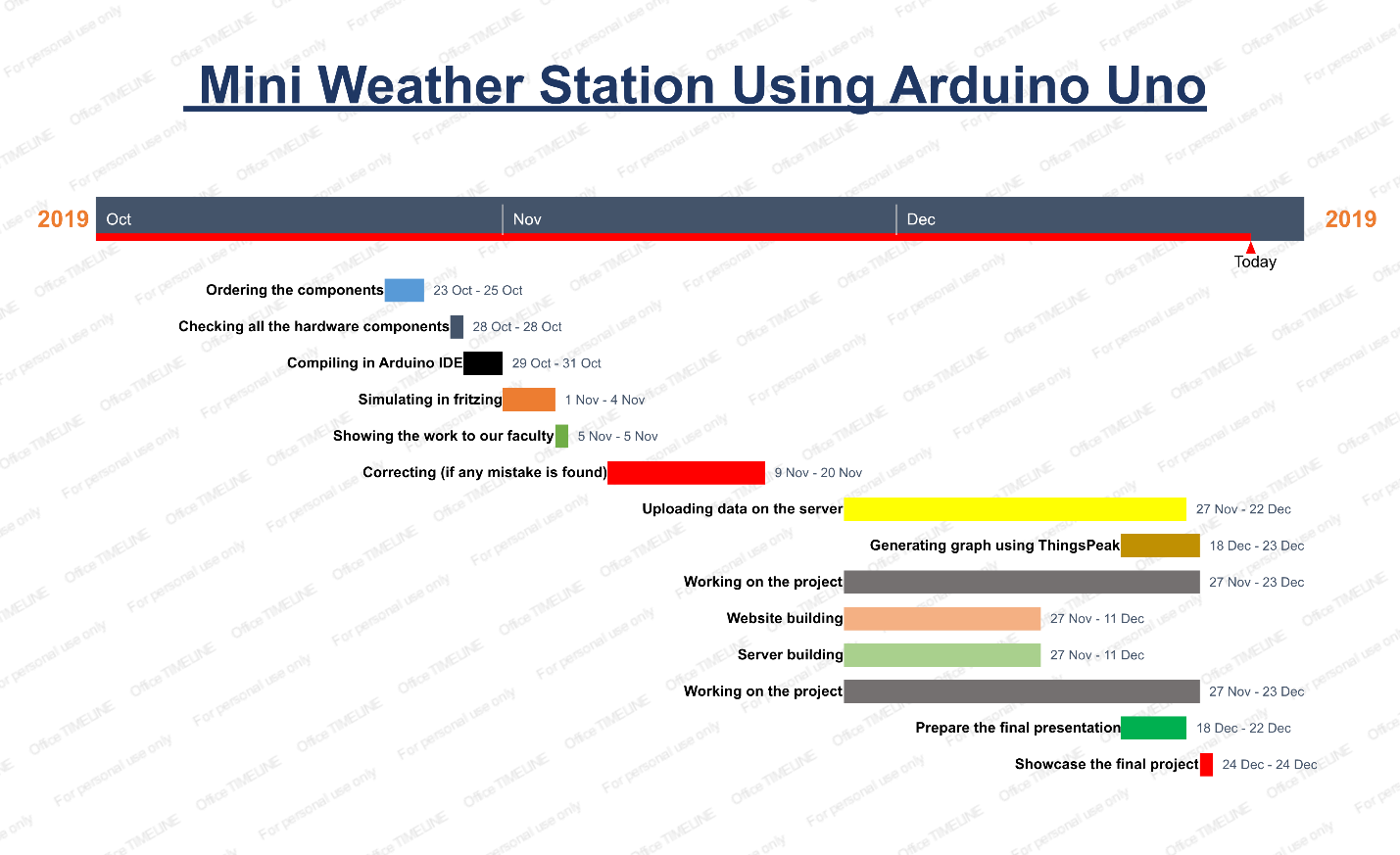
|  |  |
| --- | --- |
| Integrating circuit | Tahrim, Momena, Samiha |
| Code for dht11 and water sensor | Samiha |
| Code for coin type vibration and pressure sensor | Momena |
| Code for gas,mq135 and Wi-Fi module | Tahrim |
| Collecting data | Momena |
| Frontend | Samiha |
| Backend and database | Tahrim |

# **Time-plan**

List of the deliverables with specific dates so that you can make concerted effort to achieve them.

|  |
| --- |
| **Serial**   **Description**  Task 1 Ordering the components  Task 2 Checking all the hardware components  Task 3 Compiling in Arduino IDE  Task 4 Simulating in fritzing  Task 5 Showing the work to our faculty  Task 6 Correcting (if any mistake is found)  Task 7 Uploading data on the server  Task 8 Generating graph using ThingsPeak  Task 9 Working on the project  Task 10 Website building  Task 11 Server building  Task 12 Working on the project  Task 13 Prepare the final presentation  Task 14 Showcase the final project |

## **Gantt Chart**



# **Methodology**

## Development Stages

* Collect and analyze the requirements
* Identify and collect the necessary hardware and software
* Assemble the project
* Program the Arduino Uno
* Run tests to ensure everything is working as intended

## Software & Language Requirements

* Arduino IDE
* Fritzing
* Bootstrap 4, HTML5, CSS
* Django
* Windows PC
* Tera Term
* Excel 2016

## Hardware Requirements

* Arduino UNO
* Dht11 Sensor
* Jumper Wires
* Water Level Sensor
* MQ135 Sensor
* Breadboard

# **Procedure**

DHT11 Sensor:

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It’s fairly simple to use, but requires careful timing to grab data.

It works in the following way:

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The sensor measures temperature with a surface mounted NTC temperature sensor (thermistor) built into the unit.

Connecting the module with Arduino:

* DHT11 is a 4-pin sensor, these pins are VCC, DATA, GND and one pin is not in use.
* The DHT11 sensor attach to the Breadboard
* The VCC connect to the Arduino Board +5V
* The GND connect to the Arduino Board GND
* The Data connect to the Arduino Board Digital I/O 8

Water Level Sensor:

Water Level Sensors. Level sensors are used to detect the level of substances that can flow. Such substances include liquids, slurries, granular material and powders. ... Such measurements can be used to determine the amount of materials within a closed container or the flow of water in open channels. For our project the water level will be shown in mm.

It works in the following way:

Basic water level sensors can be used to identify the point at which a liquid falls below a minimum or rises above a maximum level. Some types use a magnetic float, which rise and fall with the liquid in the container. Once the liquid, and by extension, the magnet, reach a certain level, a reed magnetic switch is activated.

Connecting the module with Arduino:

* The Water Level sensor has 3 pins to connect. VCC, GND, A0
* The sensor attach to the Breadboard
* The VCC connect to the Arduino Board +5V
* The GND connect to the Arduino Board GND
* The A0 connect to the Arduino Analog pin 0

MQ135 Gas Sensor:

This sensor is for detecting a wide range of gases, including NH3, NOx, alcohol, benzene, smoke and CO2. Ideal for use in office or factory. MQ135 gas sensor has high sensitivity to Ammonia, Sulfide, also sensitive to smoke and other harmful gases.

It works in the following way:

It is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage.

Connecting the module with Arduino:

* The MQ135 gas sensor has 3 pins to connect. VCC, GND, A1
* The sensor attach to the Breadboard
* VCC to Arduino 5V pin
* GNG to Arduino GND pin
* Output to Arduino Analog A1 pin

Programming the LCD display and Ultrasonic Sensor:

We need to import the “DHT”, “MQ135”, “Water level sensor” library to control the DHT11, MQ135 and water level sensor with our Arduino Uno.

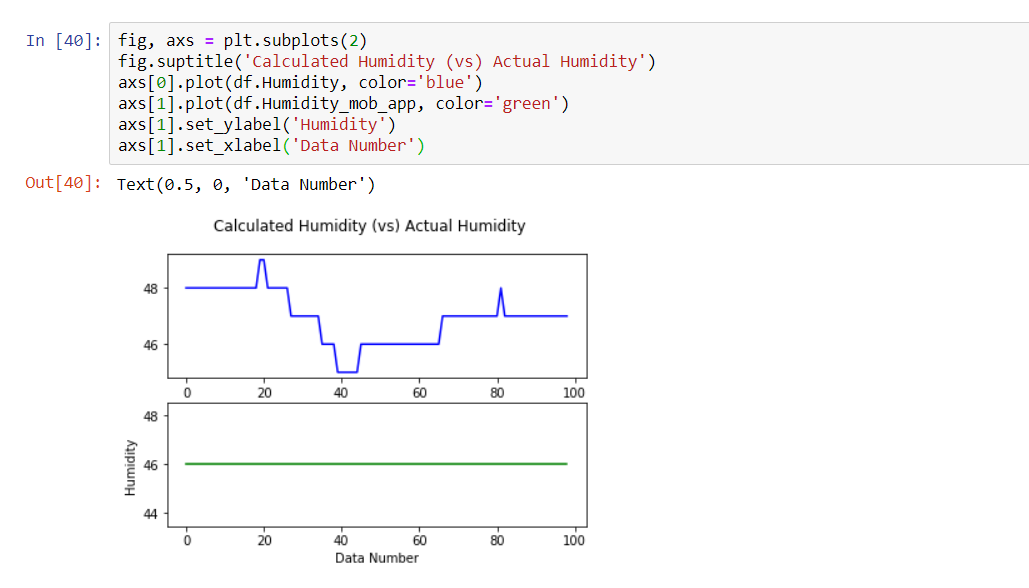
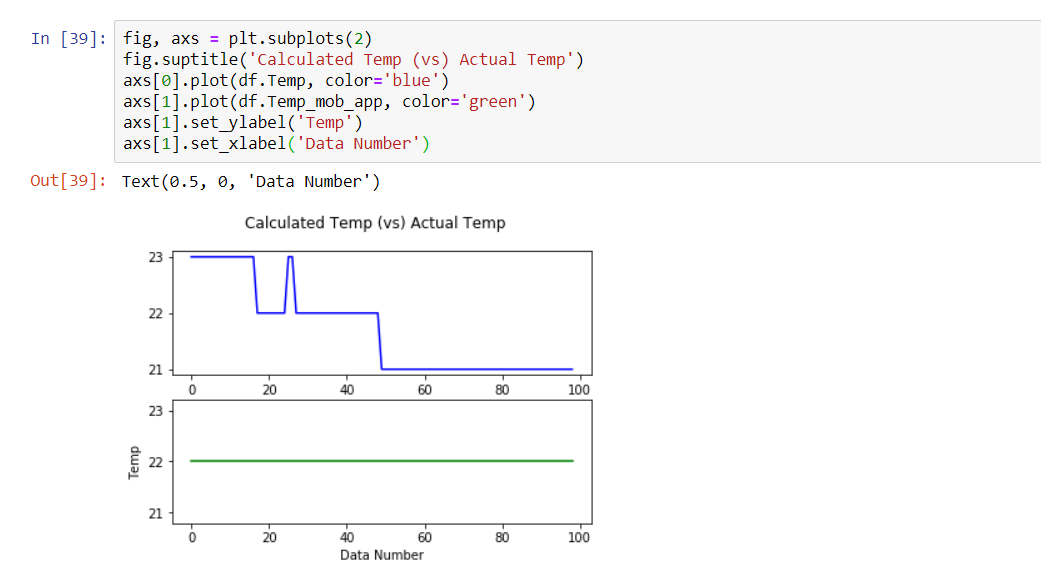
The time interval is two minutes. So after every two minutes we will get readings.

**The faults:**  
We bought few more sensors for this project. But as they were faulty, we had to discard those sensors. They were:

* WIFI module
* BMP 180 (Atmospheric pressure)
* Coin vibration motor

# **Data Analysis**

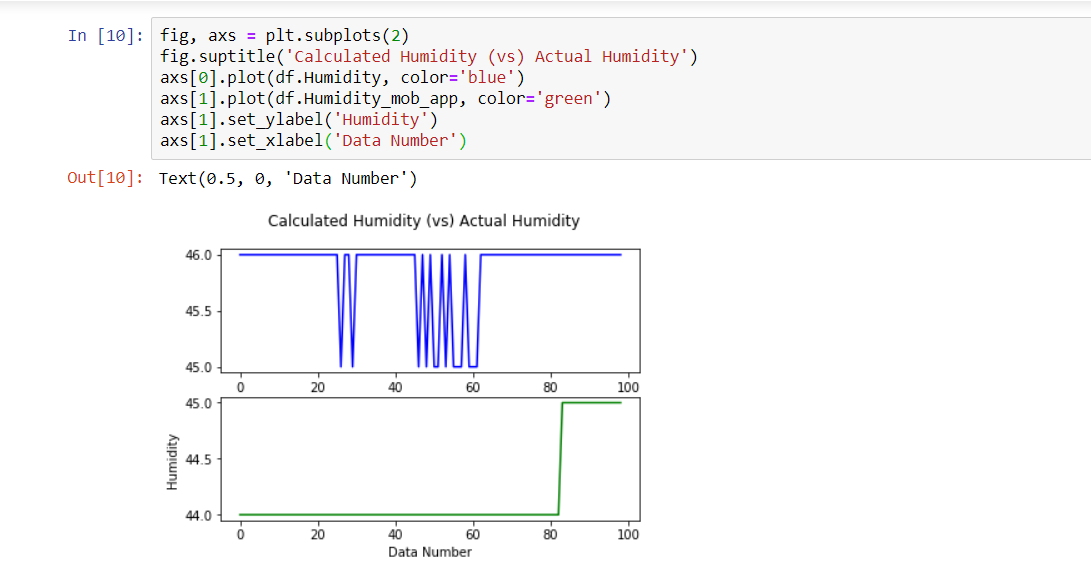
**In air conditioned room:**

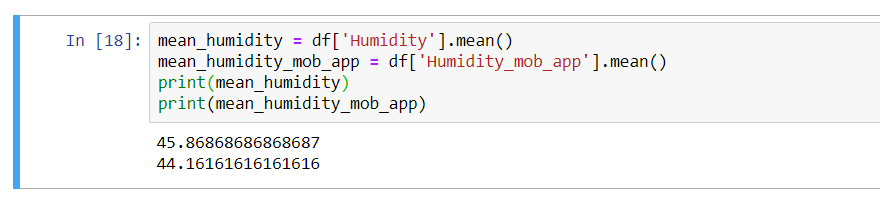
**Air conditioned room (average):**



**In normal room:**

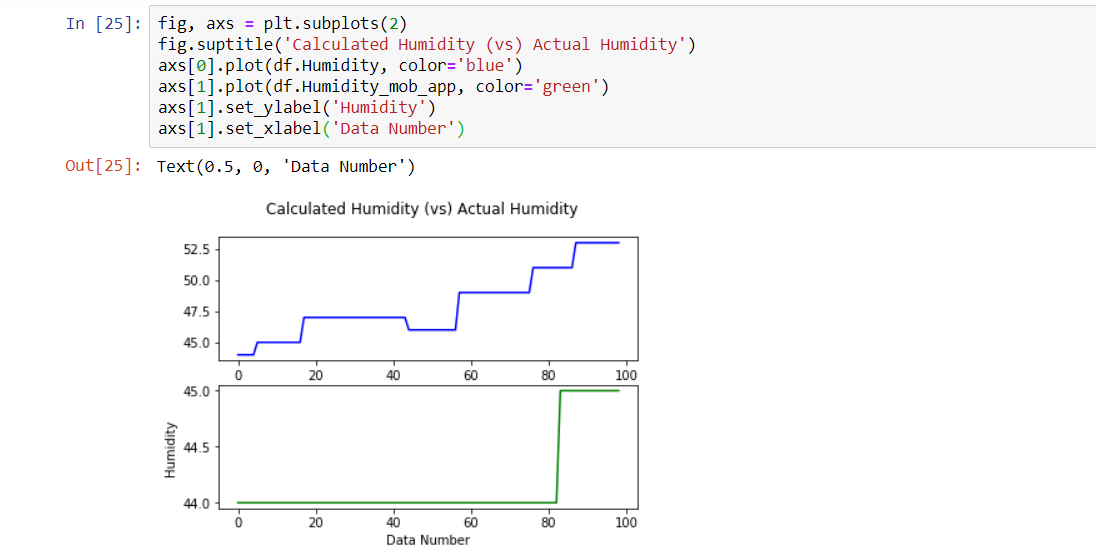
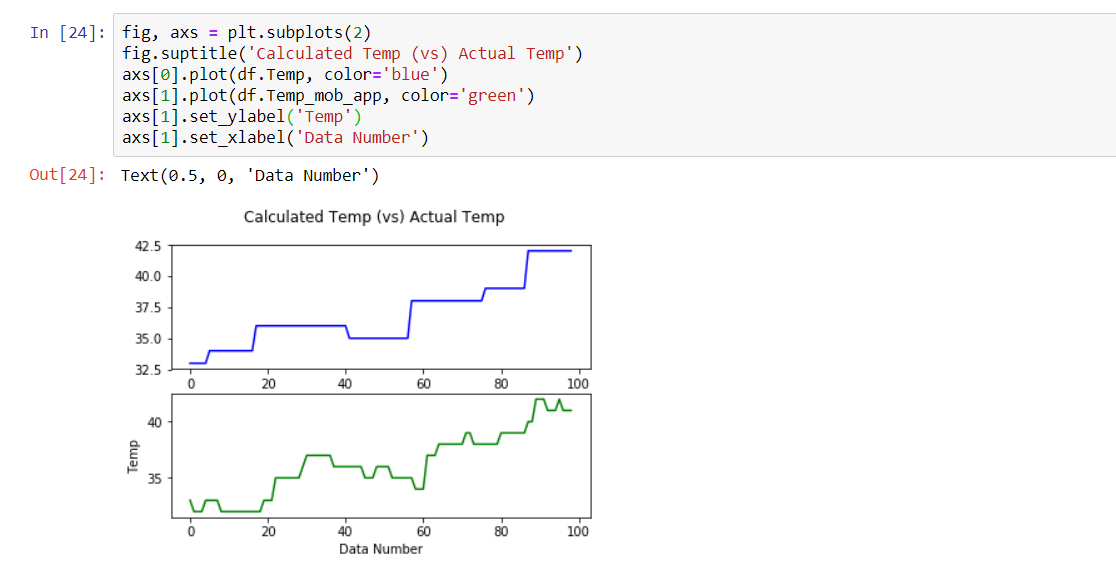
** **

**In normal room (average):**



# 

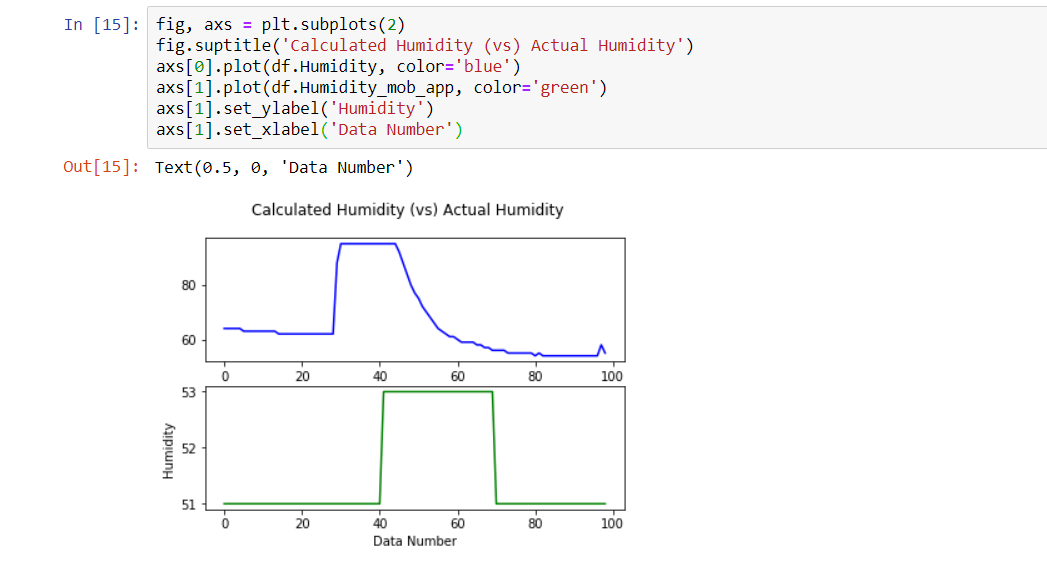
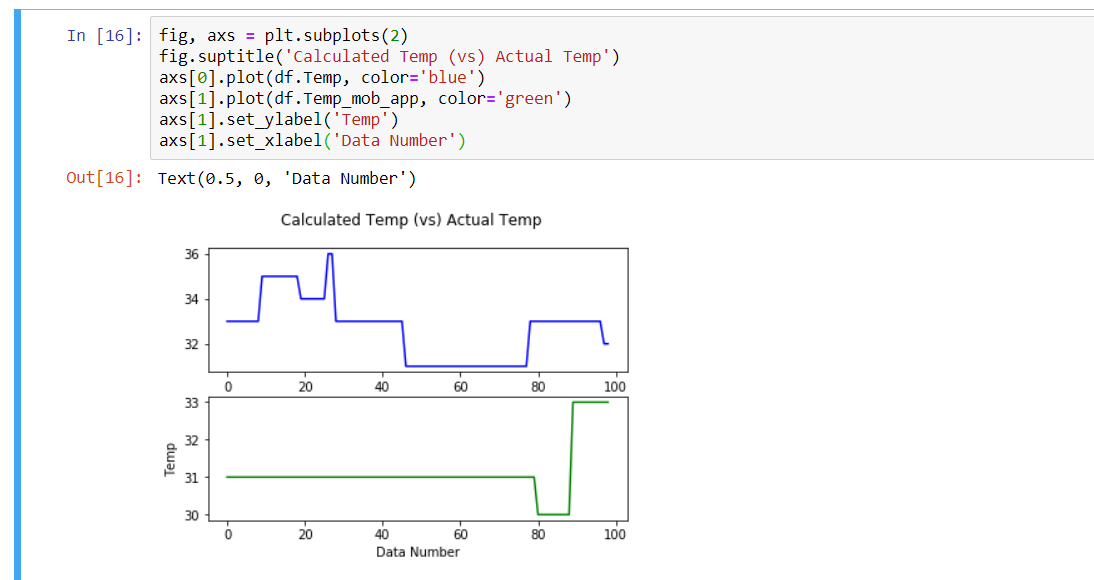
**In kitchen:**

** **

**In kitchen (average):**

****

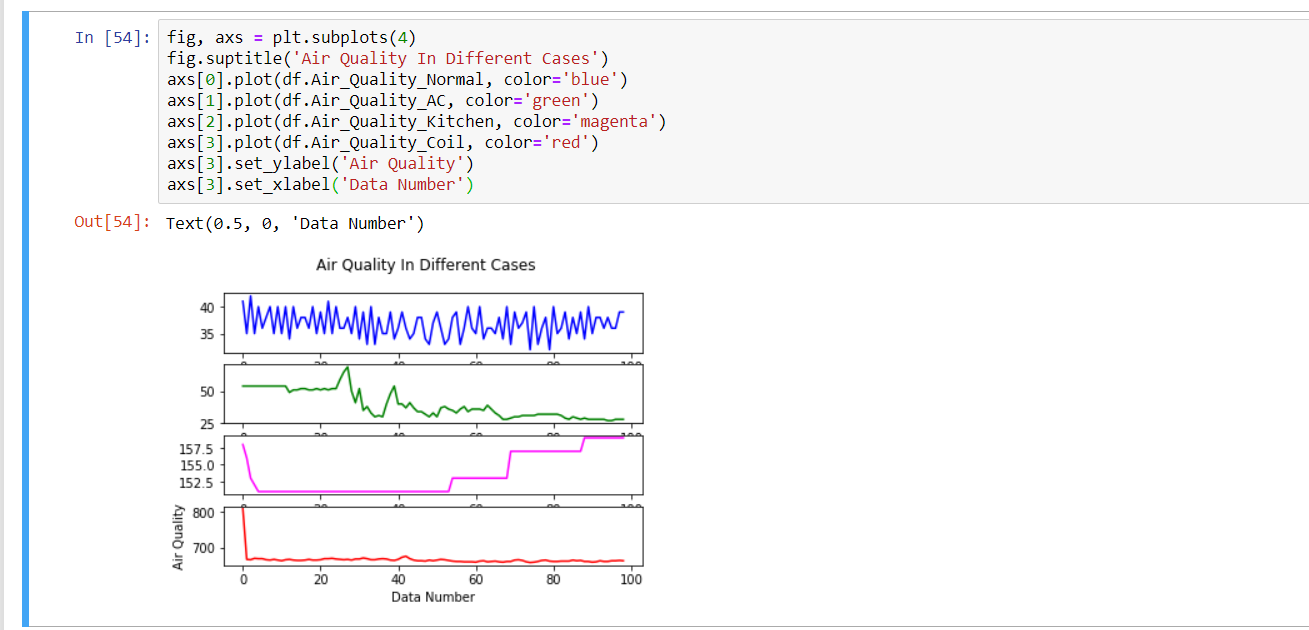
**In coil room:**

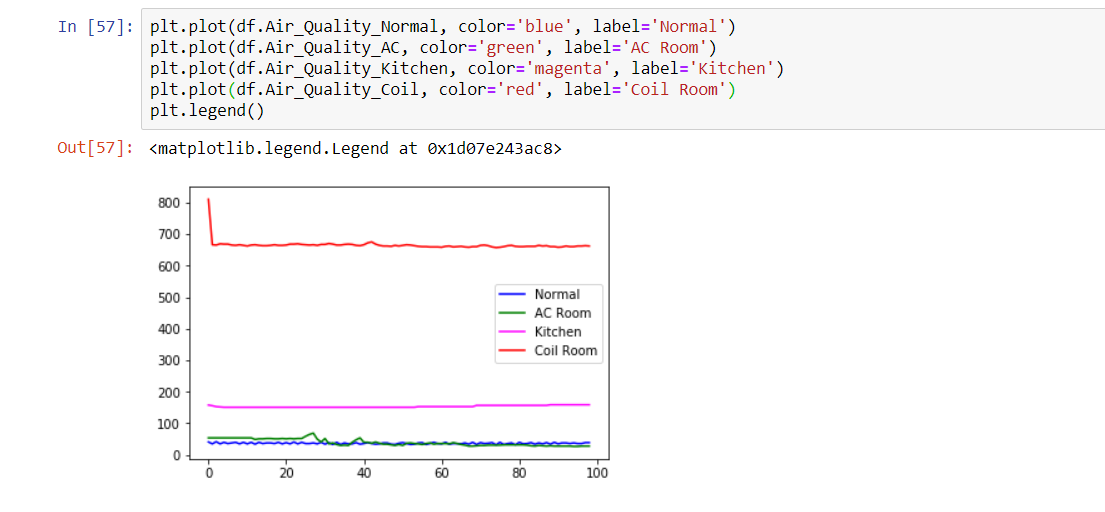
** **

**In coil room(average):**



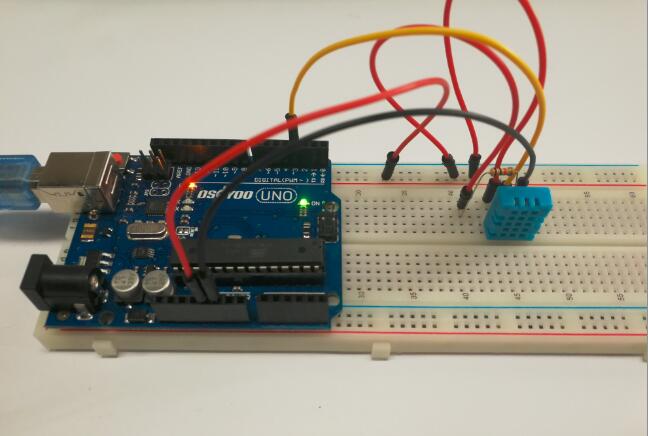
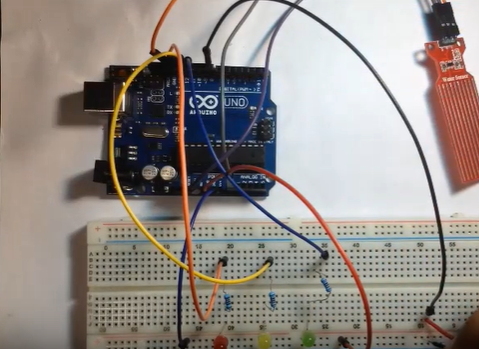
**Air quality (in different cases):**

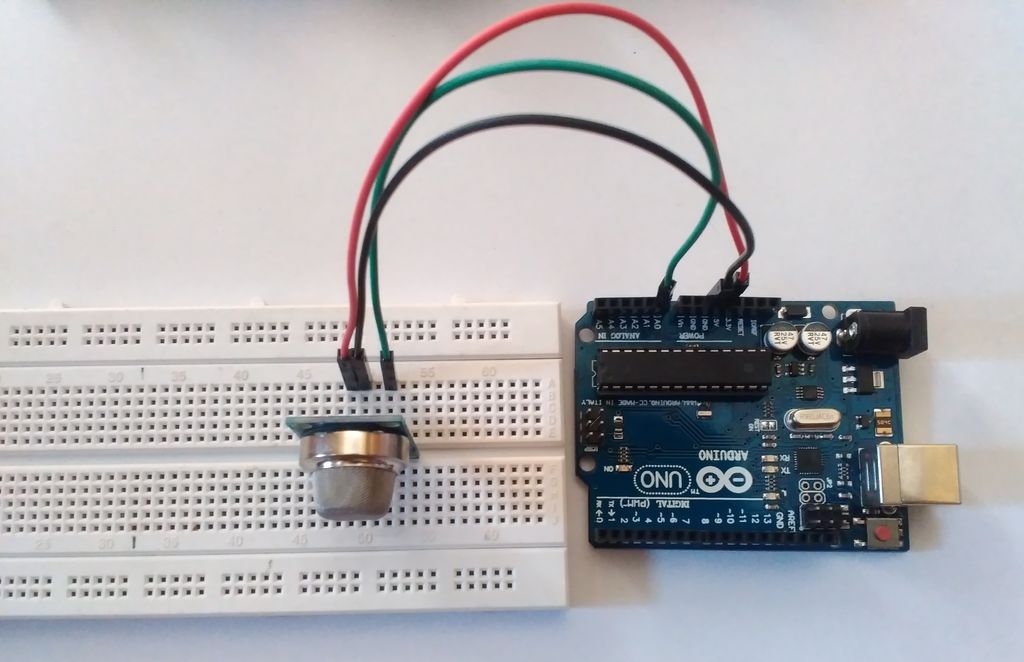


**Air quality:**

# **Result**

After assembling all the parts and programming the Arduino using the Arduino IDE, we powered up our project. It was able to accurately detect the Temperature, Water levels and Air quality of the environment. We collected the data using the Tera Term software and moved it to an excel sheet. And everything worked as we theorized.



# 

# **Discussion**

For data collection part we have used mainly 3 sensors- temperature, humidity and air quality. We have taken these data in five different cases. They are normal room environment, kitchen room, AC room, room with coil and room after the coil is being removed. When we try to measure the data in these conditions, we faced a problem. Which is every data was showing in the Tera Term software (which we use to extract the data from the Arduino UNO to CSV file). Also, the update interval was very high(5000ms). So, with the help of our instructor, we used one single sensor at a time and increased the interval to 2 min which is 120000ms. Then we took the data one by one in total 100 data. Also we have measured the actual values of the data with our mobile app. We could not compare the water sensor value because of other equipment limitations such as how much the water is evaporating during the data collection. Also, we could not compare the air quality due to the lack of sensor in our mobile phone. So, we compared the data of air quality with other conditions. After that we did the data analysis part, using Jupiter notebook, python and some it's libraries such as Numpy, Panda, Matplotlib, Plotly. We plotted the graph for showing the deviation from the actual graph. Also, we taken the mean from the calculated data and the actual data to determinate the error rate of our project data. After that, we created a database to store our findings. Also, we created a website just to demonstrate the plotted graph and other useful information. So far, we did have some problem such as some of our sensors broke, then we had to recalibrate them and also some sensor such as pressure sensor and Wi-Fi module totally stopped from working. So, we had to remove them and find alternative such as Tera Term software to send the raw values to documented format in the database. Finally, we have some great results. We had to write a simple program to make everything work and then used the Arduino IDE to push the code to our Arduino UNO. Then we ran some tests to make sure we are getting the correct readings.

The Mini Weather Station is accurate enough to be implemented in our use cases.

# **Conclusion**

The Arduino Weather Station is cheap and easy to build but can be used in various real-world scenarios. Our project focuses on making weather and climate measurement accessible to everyone. Our device can calculate Temperature, Water levels, Humidity and Air quality and its cost effective ness means it can be used by the mass in their day to day applications. For example, Nursery owners can find perfect conditions to grow the right plants, Farmers will know when to grow the crops in a more optimized way and much more. It can also be used in office or factories to maintain a healthy work environment.

|  |  |
| --- | --- |
|  |  |

# **Reference List**

[1] google, YouTube, <https://www.hackster.io/igorF2/arduino-uno-mini-weather-station31b555>

[2] a project that was showcased last semester in capstone.

# **Appendix**

Code:

#include "dht.h"

#include "MQ135.h"

#include "MQ2.h"

//change this with the pin that you use

int pin = 9; //for MQ-2

int lpg, co, smoke; //for MQ-2

MQ2 mq2(pin);

MQ135 gasSensor = MQ135(A0);

int sensorPin = 8; //for MQ-135

int sensorValue = 0; //for MQ-135

dht DHT;

#define DHT11\_PIN 7

/\* Arduino Tutorial - Watel Level Sensor 40mm

More info: \*/

const int read = A0; //Sensor AO pin to Arduino pin A0

int value; //Variable to store the incoming data

void setup(){

Serial.begin(9600);

Serial.println("Date & Time, Humidity %, Temp \*C, Air Quality, Water Level");

// mq2.begin();

}

void loop()

{

// Temp & Humidity

int chk = DHT.read11(DHT11\_PIN);

float temp = DHT.temperature;

float humidity = DHT.humidity;

// Serial.print("Temperature = ");

// Serial.print(temp);

// Serial.println(" C");

// Serial.print("Humidity = ");

// Serial.print(humidity);

// Serial.println("%");

// MQ135

sensorValue = analogRead(A0);

// sensorValue = digitalRead(8);

// Serial.print("Air Quality=");

// Serial.print(sensorValue);

// //concentration of carbon dioxide(PPM-parts per million)

// Serial.println(" PPM");

// MQ2

/\*read the values from the sensor, it returns

\*an array which contains 3 values.

\* 1 = LPG in ppm

\* 2 = CO in ppm

\* 3 = SMOKE in ppm

\*/

// float\* values= mq2.read(true); //set it false if you don't want to print the values in the Serial

// //lpg = values[0];

// lpg = mq2.readLPG();

// //co = values[1];

// co = mq2.readCO();

// //smoke = values[2];

// smoke = mq2.readSmoke();

// Water Sensor

value = analogRead(read); //Read data from analog pin and store it to value variable

if (value<=480){

// Serial.println("Water level: 0mm - Empty!");

}

else if (value>480 && value<=530){

// Serial.println("Water level: 0mm to 5mm");

}

else if (value>530 && value<=615){

// Serial.println("Water level: 5mm to 10mm");

}

else if (value>615 && value<=660){

// Serial.println("Water level: 10mm to 15mm");

}

else if (value>660 && value<=680){

// Serial.println("Water level: 15mm to 20mm");

}

else if (value>680 && value<=690){

// Serial.println("Water level: 20mm to 25mm");

}

else if (value>690 && value<=700){

// Serial.println("Water level: 25mm to 30mm");

}

else if (value>700 && value<=705){

// Serial.println("Water level: 30mm to 35mm");

}

else if (value>705){

// Serial.println("Water level: 35mm to 40mm");

}

// Pressure Sensor

// ESP8266

Serial.print(",");

Serial.print(temp);

Serial.print(",");

Serial.print(humidity);

Serial.print(",");

Serial.print(sensorValue);

Serial.print(",");

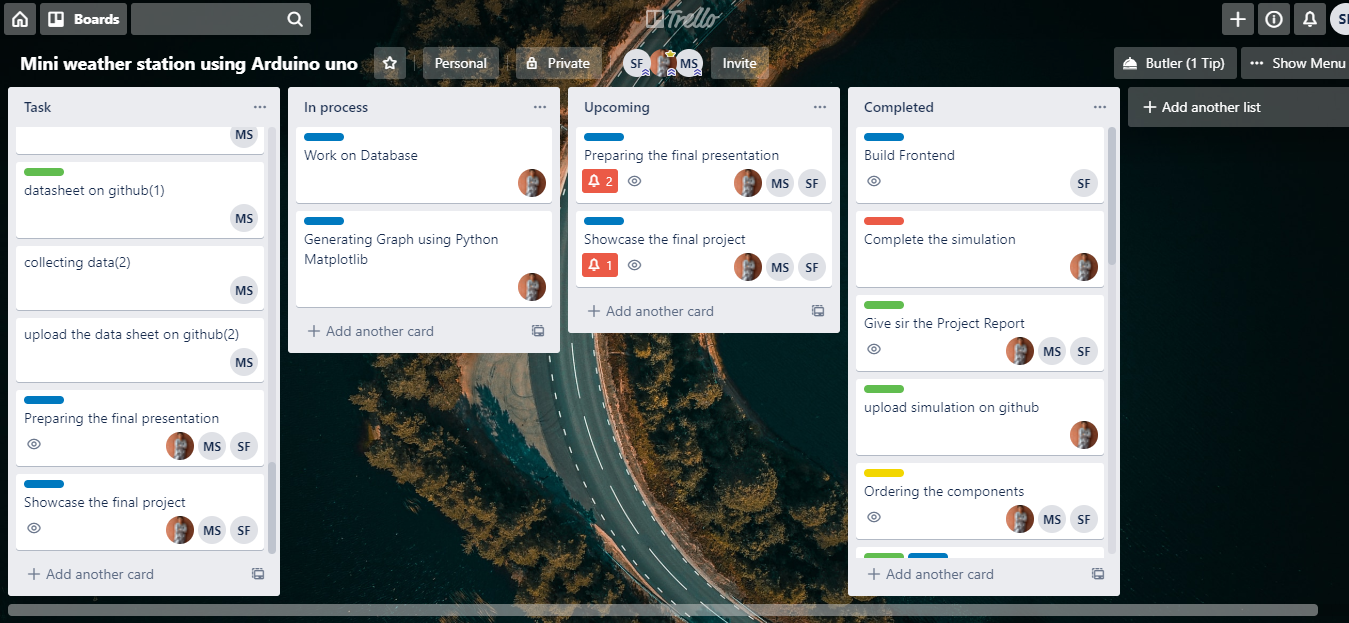
Serial.println(value);

delay(120000);

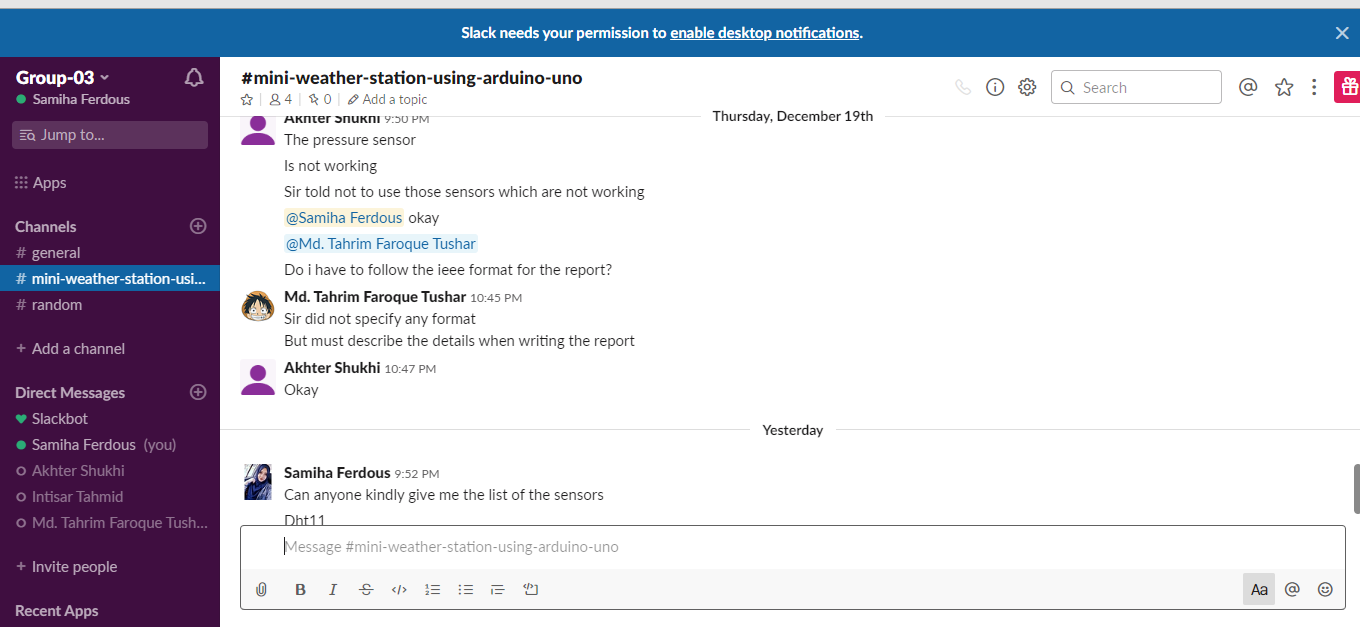
}

# **Screenshots:**

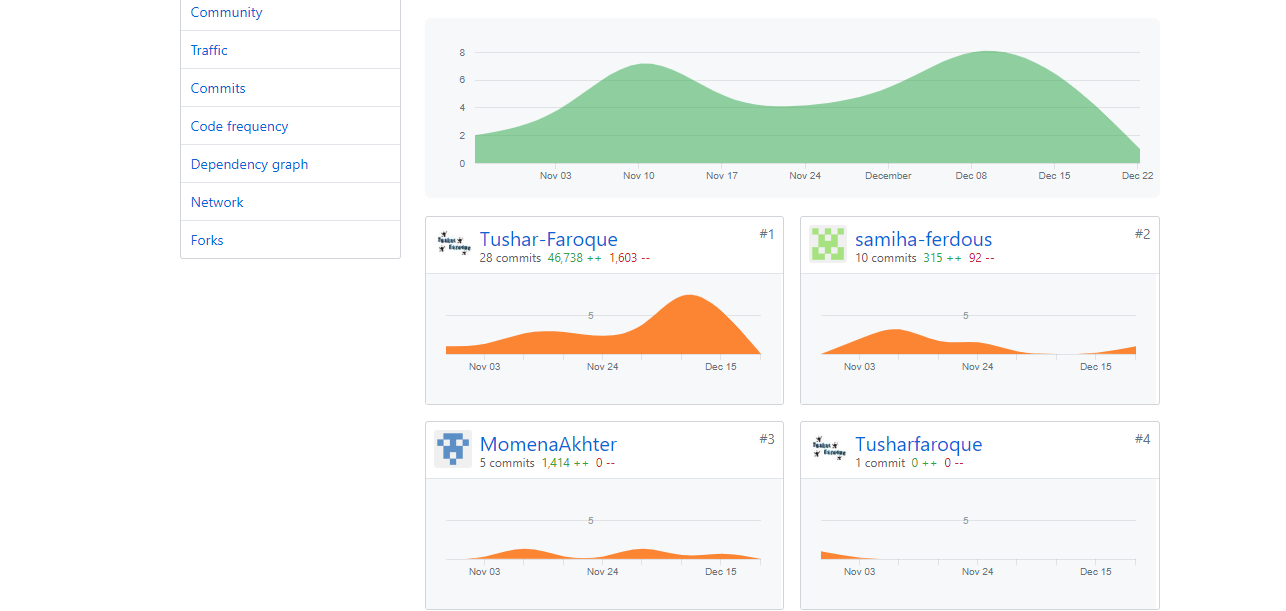
Trello board (updated on 21st December)



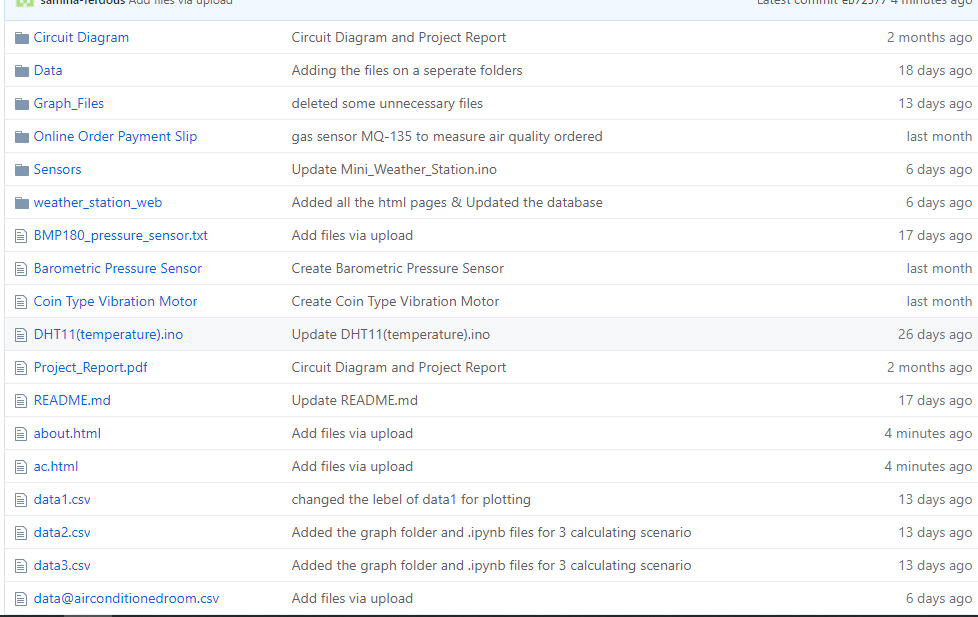
Slack (updated on 21st December)



GitHub (updated on 21st December)



GitHub files (updated on 21st December)



Github files (updated on 21st December)

