IoT based Smart Irrigation System

Department of Electrical Engineering

Mureed Qasim Shah **2018-Uet-NML-Elect-37**

2022

Final year project report submitted in partial fulfilment of requirement
for degree of Bachelors of Science in Electrical Engineering

Namal, Mianwali
Affiliated with
University of Engineering and Technology, Lahore

DECLARATION

The project report titled "IoT based Smart Irrigation System" is submitted in partial fulfilment of the degree of Bachelors of Science in Electrical Engineering, to the Department of Electrical Engineering at Namal Institute, Mianwali, affiliated with University of Engineering and Technology, Lahore.

It is declared that this is an original work done by the team members listed below, under the guidance of our supervisor "Mr. Muhammad Usama". No part of this project and its report is plagiarised from anywhere, and any help taken from previous work is cited properly.

No part of the work reported here is submitted in fulfilment of requirement for any other degree/ qualification in any institute of learning.

Team Members	**University ID**	**Signatures**
Mureed Qasim Shah	2018-Uet-NML-Elect-37	____________________

Supervisor

Mr. Muhammad Usama

Signatures with date

____

____

Table of Contents

Table of Contents

List of Figures

List of Tables:

ACKNOWLEDGMENTS

ALLAH Almighty as the most merciful and most beneficent. Without acknowledging the people who contributed to the accomplishment of this project, we would not be able to fully express our joy and satisfaction. Their constant guidance, support, and encouragement resulted in the success of our project. We were able to finish the project with Allah's grace and the prayers of our family and teachers. We would like to thank our project supervisor, Mr. Muhammad Usama, and our co-supervisors, Dr. Majid Ali. The experience of working with them was truly extraordinary. As a result of this project, we have acquired a great deal of knowledge that will inspire our future professional endeavours. Throughout this project, their guidance and expertise kept us on track toward success.

Abstract

Agriculture of farming has started past 12000 many years straight back, Neolithic age gave the beginning of civilization, Farming and later becoming continued as traditional farming techniques. Agriculture is the backbone of Pakistan economy. Pakistan is one of the fifth largest producer of sugarcane in the world, the seventh-largest producer of wheat, and the tenth-largest producer of rice. More than 70% of the population in Pakistan is related to Agriculture. Its share of GDP is mostly about 25 percent which can be more than the share of any other sector. Many problems related to Agriculture are constantly destroying the development of the country. One of the most advanced problem is wastage of water during its transportation and irrigation into the field. Agriculture sector in Pakistan uses 93% of its water that is available for irrigation but more than 60% of that water is wasted during irrigation, its application and its transportation into the field. The reason behind significant amount of water wastage is the lack of knowledge about proper irrigation methods and irrigation schedule. A possible option for these irritating dilemmas is to decide on an advanced agriculture system that consists of contemporary styles. So we can make farming advanced and easy using IoT. IoT is the system of connected devices that is a fastest-growing technology in every domain of society, especially in agriculture. The main objective of this project is to reduce the wastage of water during irrigation alongside many more advantages like maintaining proper moisture levels for the crops, reducing the farmer's effort, and increasing the productivity of crops. The sensor used in the system collects the data from the soil and sends it to the IoT web server where it can be further examined. We will design an android mobile application that will provide essay access to information to the farmer.

Chapter 1

1. Introduction

IoT is a system consisting of computing devices that can sense the data as well as convert data over the web without any human involvement. The project group shall recommend an IOT-based Irrigation and Weather Reporting System. This method will consist of two parts which can be crucial. The first part is linked to the irrigation monitoring and managing system plus the second is the weather reporting system. The project group can manage and monitor the way to obtain water coming from a place. The suggested system permits the people to directly look at the climate status online without the need for a weather forecasting company in this project. It will help to know about the current temperature, humidity, moisture, and rain by using a temperature sensor and humidity sensor respectively. The system will transfer the data to the microcontroller or process this data and the microcontroller will send this information to the web host over a WIFI Module.

In this project, sensors keep checking the values of humidity, temperature, soil moisture, and rain status and update the data on the website. A user can see the details of the sensor and according to the need of crops or plants he can supply the water just by a click on motor status. Farmers should work on the other parts of the field so that they will not care about the irrigation systems. The crops get damaged if the farmer does not supply the exact amount of water to crops or plants. So by using this system, farmers don't have to worry about the crops plants getting damaged as a result of drought waterlogging. So in this way, we will be able to save the wastage of water using proper irrigation systems.

1.1 Background Study

The contribution of the agriculture sector to Pakistan's economy is huge. Every has ensured to give food security for the propels of Pakistan. The share of the agriculture sector in the Pakistan economy is more than 50% and its share of GDP is more than 25% which is like a backbone of the Pakistan economy. [1]

A major part of Pakistan is using the old method for irrigation purposes where more than 63% of the water is wasted during irrigation and its transportation into the field. [9] So the irrigation method used by farmer either end up giving too little water or providing too much water which leads to poor production and many other disadvantages.

Water is one of the most important part of the better growing of plants. As we know that different lands require a different amount of water for its production. So the farmer will be able to monitor the field via mobile app or website remotely so that he can pay attention to the others aspect like crop cultivation and their transportation into the market.

Farmers need to take care of all these possible cases when watering their crops. The purposed system will provide automatic water to the crops or fields when required.

1.2 Problem Statement

The irrigation system used in past by the farmers either leads to too much water for the crops and fields which ultimately leads to wastage of water or provides too little water which leads to insufficient production of crops. To design a system that will reduce water wastage during irrigation as well as increase the productivity of the crops.

1.3 Objective of Project

Given below are the following objective of a smart irrigation system

1.3.1 Main Objective:

To design a smart irrigation system that will operate based on soil moisture content integrated with a rain sensor so that it will help for reducing the wastage of water during irrigation.

1.3.1 Specific Objective

- To reduce the wastage of water during Irrigation so that the maximum amount of water can be saved for the future purpose
- Maintain the right level of moisture for the soil
- Promote the proper growth of plants and crops so that the farmer can get benefits from it
- To increase the productivity of crops the one who is responsible for our survival on this earth can get benefit from it
- To reduce the human effort, the farmer has to pay attention to the other aspect of the field.
- To minimize wastage of water and over-irrigation to save water
-

So these are some of the objectives that our project will try to cover.

1.4 Motivation

The project is designed to save time for the farmers as farmers need to go to their farming fields to see the moisture level. If they need to water the land, they switch on the motor and wait until the water level is enough. It is disgusting and time-consuming. Any unconsciousness may cause over-irrigation. When soil is over irrigated, it is given too much water. This is harmful to crops and their production. Moreover, it is a waste of water. More power is used. So irrigation cost increases.

This project has been designed in such a way that it does not require any manual checking of irrigation systems. For example, if you are staying in any city of Pakistan, and have your farm in Mianwali or anywhere else then it is not possible for you to go to the farms every time to check how it's working. Instead, the leading characteristic of this project is that you can monitor your field remotely.

1.5 Business scope

This project can also prove to be useful for lawns, parks, soil monitoring, watering the football ground, and agriculture purpose as well. Because we know that agriculture is one of most important element in human life and it is the main source of food. It plays a very important role for the growth of a country’s economy. Growth in agriculture is important for the development of Pakistan's economy. By the way, our farmers are using the old traditional methods for farming. The project team wants to replace these old techniques with automated systems.

Behind that now a day the whole world is afraid due to lack of water in the future. And that is a big problem for the whole world. The project team wants to save the wastage of water and eliminate the manual labour on the field for irrigation by proposing this system. This project will help farmers to save time, cost, and wastage of water.

1.6 Useful Tools and Technologies

1.6.1 Hardware Specification

Wi-Fi Module:

We have used a Wi-Fi module which is ESP8266.

Temperature sensor:

A temperature sensor is used for temperature measurement through an electrical signal.

Humidity Sensor:

It is used to measure the amount of water present in the surrounding air.

Rain Sensor:

To measure the rain so that if our motor pump is on, we can turn it off automatically.

Moisture Sensor:

Measure the moisture content in the soil via two probes that are fed into the soil.

Microcontroller:

Arduino Uno is used.

Relay:

A 5 Volt relay module is used for switching purposes.

1.6.2 Software Specification:

Proteus:

For simulation purposes, we have used the Proteus software where we have designed and simulated the whole circuit diagram of the project.

Arduino:

For the coding of the project, we have used Arduino IDE.

Chapter 2

2. Literature Review

Archana and Priya published a paper to determine the value of moisture and temperature and humidity in the air. They did not propose the proper irrigation system for the watering of plants and crops. They only used soil moisture and temp and humidity sensors. [2] Karan Kansara builds an automatic irrigation system. It was not capable to determine the nutrient value of the plants and it will not provide any information about the temperature and humidity sensor and how we can control the motor pump when it's raining. [3] The published paper on "Automatic Irrigation System on Sensing Soil Moisture Content" only includes measuring the moisture in the soil. They also used only a single soil moisture sensor to measure the soil content in the soil. This system will not provide any remote monitoring of the field which could be via mobile app or website. Prof C.H.Chavan and P.V.Karnade proposed a system smart wireless sensor network for monitoring environmental parameters using Zigbee. The major drawback of this system was that it was not giving any information about weather forecasting. [4] So from the above projects, we see that many people in the world worked on the smart irrigation system. They recommended many ways to reduce the wastage of water during irrigation and increase the productivity of the crops. Some of the work you can find is given below in the references. One of the major drawbacks that we have found in their project is that they only use one soil moisture sensor to run the motor pump. Some of them also used rain sensors. So it's nearly impossible to irrigate the field using one soil moisture sensor. So here in our project, we have used four soil moisture sensors to control the motor pump and to start the irrigation. In case of excess amount of water, we also proposed a water drainage system. Our project will store the excess amount of rainwater and will use it for irrigation when needed. We also can change the soil moisture sensor limit according to temperature and humidity values.

Chapter 3

3. Methodology

The System is based on a micro-controller. Smart irrigation is used to save water. The soil, temperature and humidity, and rain sensor will give the live readings. These sensors will collect the data. These sensors are further connected with a Wi-Fi module to send the data on the ThingSpeak for easy monitoring.

This method includes the design of a prototype device that can be easily monitored via web or mobile app. For this purpose, the first step is drawing of timeline and reading the previous work after looking at the pros and cons of other projects that have been done before, we can start to implement and execute our plan. The complete guide for the methodology is given below in the chart:

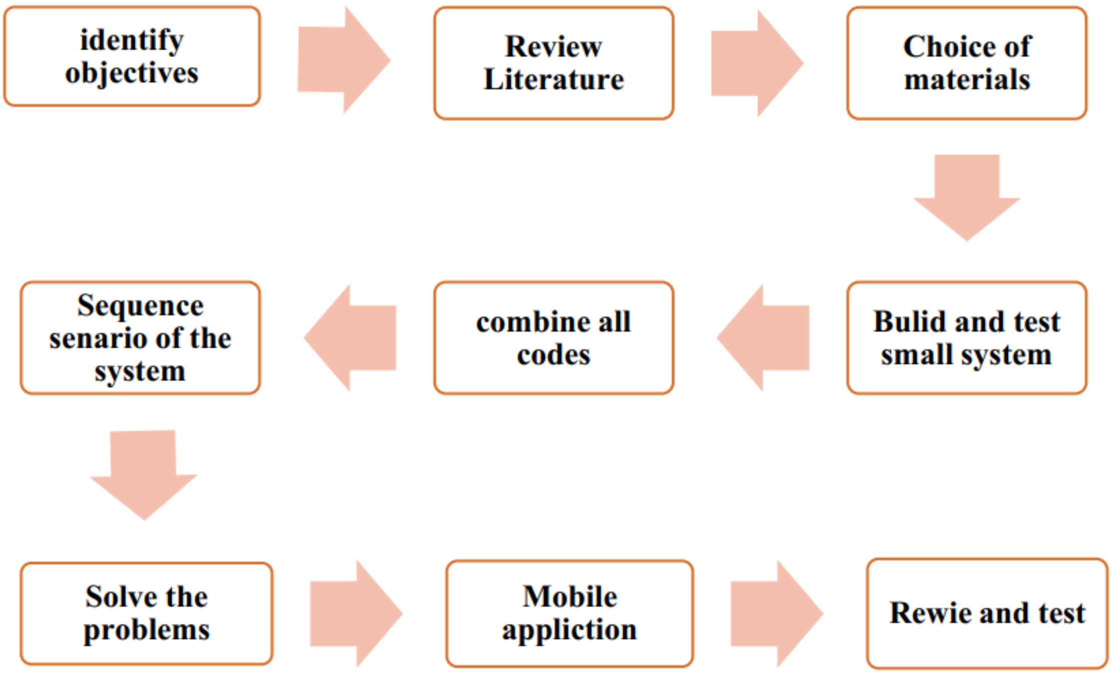

Figure 1: Process flow chart

3.1 Design Process

First of all, we have designed and simulated our whole circuit diagram for the smart irrigation system on the software Proteus.

This process includes two steps which are

1. Software Design
2. Hardware Design

3.1.1 Software Design

First of all, we have designed a circuit diagram of our project in Proteus software. The circuit diagram is given below.

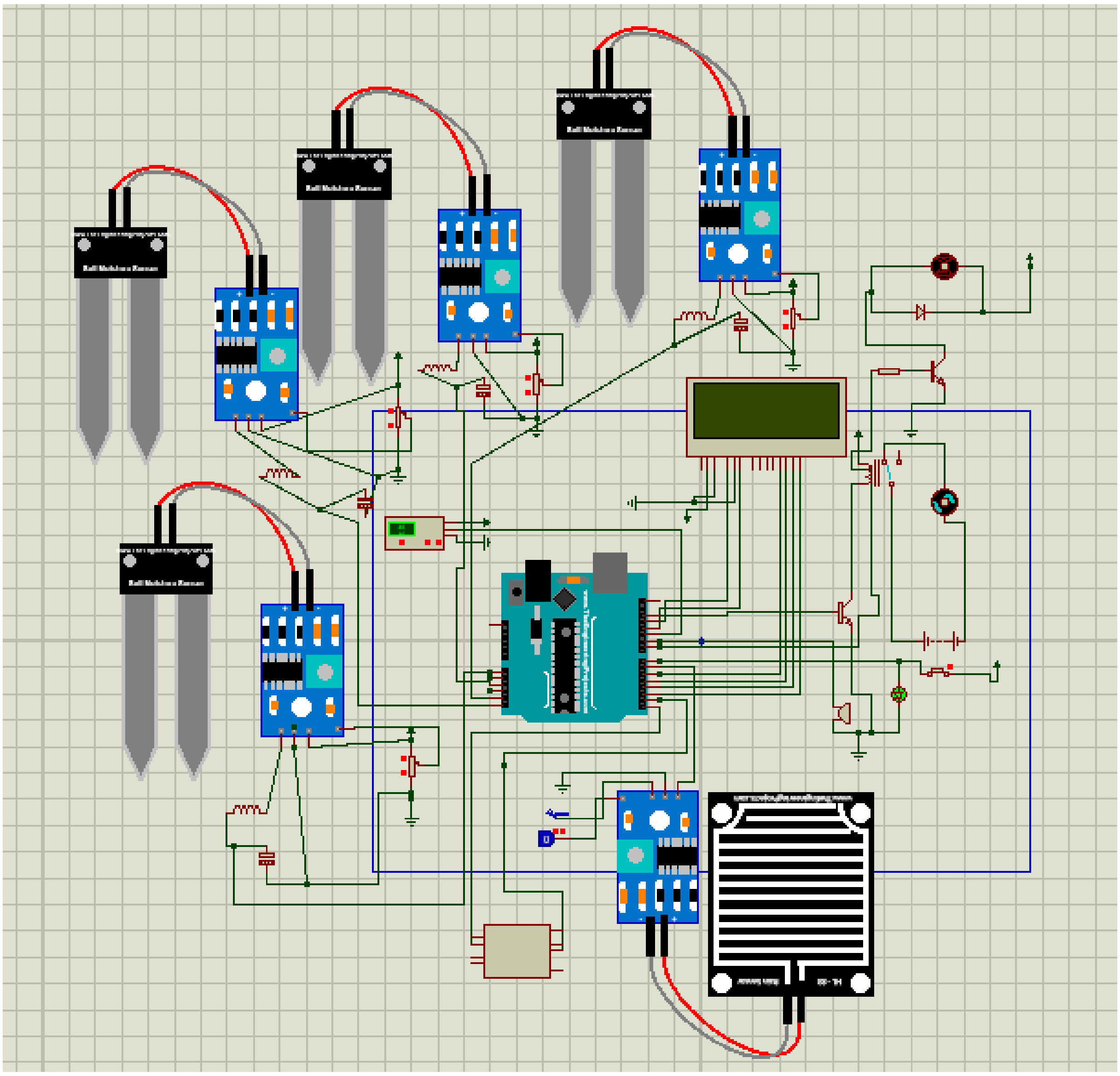

Figure 2: Simulation Diagram

This is The simulation circuit diagram that we have designed in Proteus.

3.1.1.1 Component Used in Simulation Part

Given below are the components that we used in our Project which are

1. Soil moisture sensor
2. Temperature and humidity sensor
3. Rain sensor
4. Motor pump
5. LCD
6. Arduino Uno
7. WIFI Module

3.1.1.2 Components Working Explanation

In this section, we will explain how every component is working in the simulation of this project. How do we integrate these sensors and how to get the output from this sensor?

3.1.1.3 Soil Moisture Sensor

Soil moisture sensor working is very simple. Resistance present in the soil is inversely proportional to the soil moisture. The fork-shaped probe acts just like a potentiometer whose resistance changes according to the water content present in the soil. An excess amount of water present in the soil means better conductivity. So the resistance will be low. If there is less water present in the soil means poor conductivity. So the resistance will be high.
Sensor output values will be according to the resistance. So by measuring these values we can measure the moisture level. It only has 4 pins to connect as shown below:

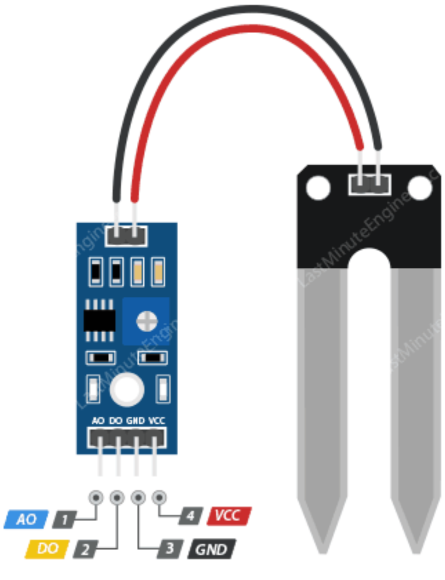

Figure 3: Soil moisture sensor

As we know the module provides an analog and digital output. We will measure the soil moisture by reading the analog output. So in this Project, we have used 4 Soil Moisture sensors. We interface these sensors with Arduino Uno as shown above in the circuit diagram figure. We are constantly measuring the percentage of moisture present in the soil. These four sensors are connected to the Arduino pins A0, A1, A2, and A5. These four pins are acting as output. By using the analogRead command we are reading the values of moisture present in the soil. As shown below

3.1.1.4 Specification of soil moisture sensor

Given below are the technical specification of the Soil Moisture Sensor.

Table 1: Technical Specification of Soil Moisture Sensor:

Type	Specifications
Range	Moisture: 0 ~ 99.9 % Temperature: 0 ~ 60 C
Accuracy	Moisture: + - 3% Temperature: + 0.5 C
Sensor type	Frequency Domain Reflectometry
Operating range	0 – 60C
Power supply	DC: 9 - 15 V
Current	25mA
Size	Probe Length 12cm 38mm

3.1.1.5 Temperature and Humidity Sensor

The DHT11 is a simple and low-cost sensor. It uses a humidity sensor and a thermistor to determine the temperature and relative humidity present in the air. It's very simple to use.

3.1.1.6 How the DHT11 measures humidity and temperature

It is used to measure water vapor by measuring the electrical resistance between two electrodes. The relative humidity is proportional to the change in resistance between two electrodes. Greater relative humidity means there is a decrease in the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

There is a thermistor built into the unit of DHT11 which is used to measure the temperature.

DHT11 is shown below:

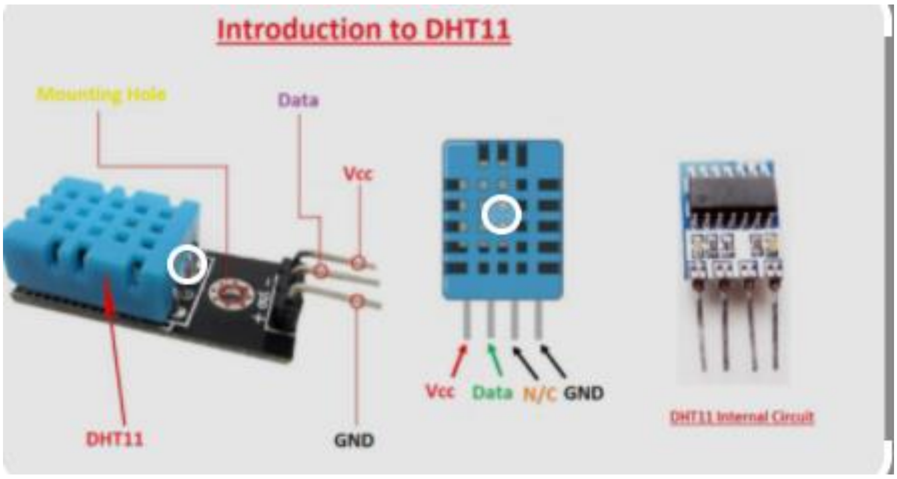

Figure 4: DHT11 Sensor

3.1.1.7 How it works in our Project:

In our project, we are measuring the values of temperature and humidity for two reasons. As we know Temperatures are high in summer and low in winter. Therefore, crops need less water in winter than in summer. So based on temperature and humidity sensor values in winter and summer, we can change the moisture limit of our project.
Another reason for using this is that we can observe the weather condition without going to a weather forecasting agency or without opening Google.

3.1.1.8 Specification of DHT11 Sensor

Table 2: Temperature Sensor Specification

Sensor Type	DHT11
Unique ID	-1
Driver Ver	1
Max Value	50°C
Min Value	0.0°C
Resolution	2°C

Table 3: Humidity Sensor Specification:

Sensor Type	DHT11
Unique ID	-1
Driver Ver	1
Max Value	80%
Min Value	20 %
Resolution	5 %

Given below is the value of the temperature and humidity sensor measured by the DHT11 Sensor

3.1.1.9 Rain Sensor

In our smart irrigation project, we also included a rain sensor for the detection of rain. It uses a rain detection plate to detect the effect of rain. The rain sensor is shown below

Figure 5: Rain Sensor

The rain sensor detects water that comes short-circuiting the tape of the printed circuits. It acts as a variable resistance that will change status. When the sensor is wet its means high resistance and when the sensor is dry it means low resistance.

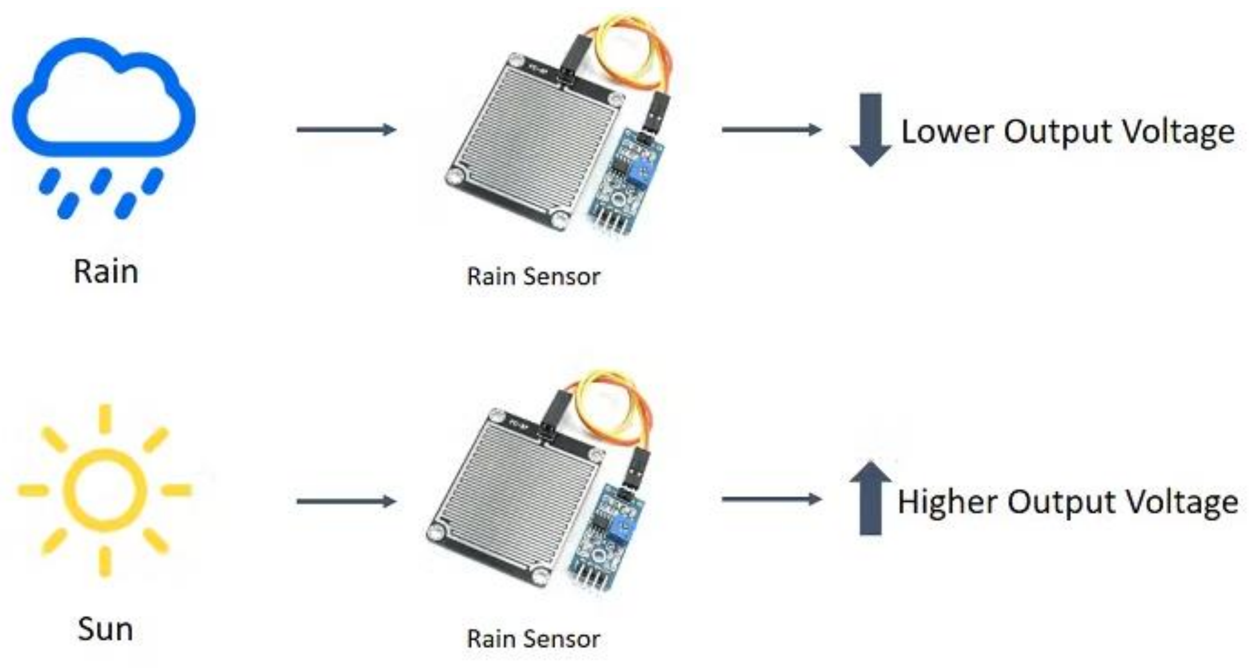

Figure 6:Rain Sensor Working

The comparator has 2 outputs connected to the rain sensor.

3.1.1.10 Technical specification

- Voltage: 3, 3v to 5v
- Sensor dimension: 3.9 * 5.4 cm

- Sensitivity potentiometer
- 2 control LEDs

3.1.1.11 Adjust the sensitivity

There is a potentiometer present on the comparator of the rain sensor. By changing the sensitivity of the potentiometer, detection can be realized on a drop of rain or in a glass of water.

3.1.1.12 How it works in our project

We have successfully connected the rain sensor with our Arduino Uno. When it starts to rain and the drops fall on the plate then its digital output value will be high and we will receive the message that rain has started.

3.1.1.13 LCD

We also used a 20x4 LCD to display the values of

- Temperature
- Humidity
- Soil Moisture
- Rain Status
- Motor Status

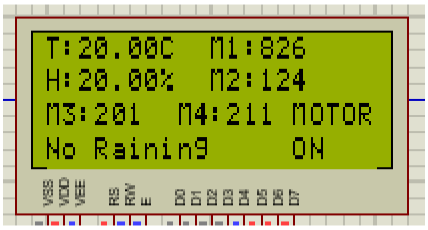

Figure 7: LCD Display

3.1.1.14 Arduino Uno

The main heart of the project is Arduino. Given below is the figure of Arduino that we used in our project.

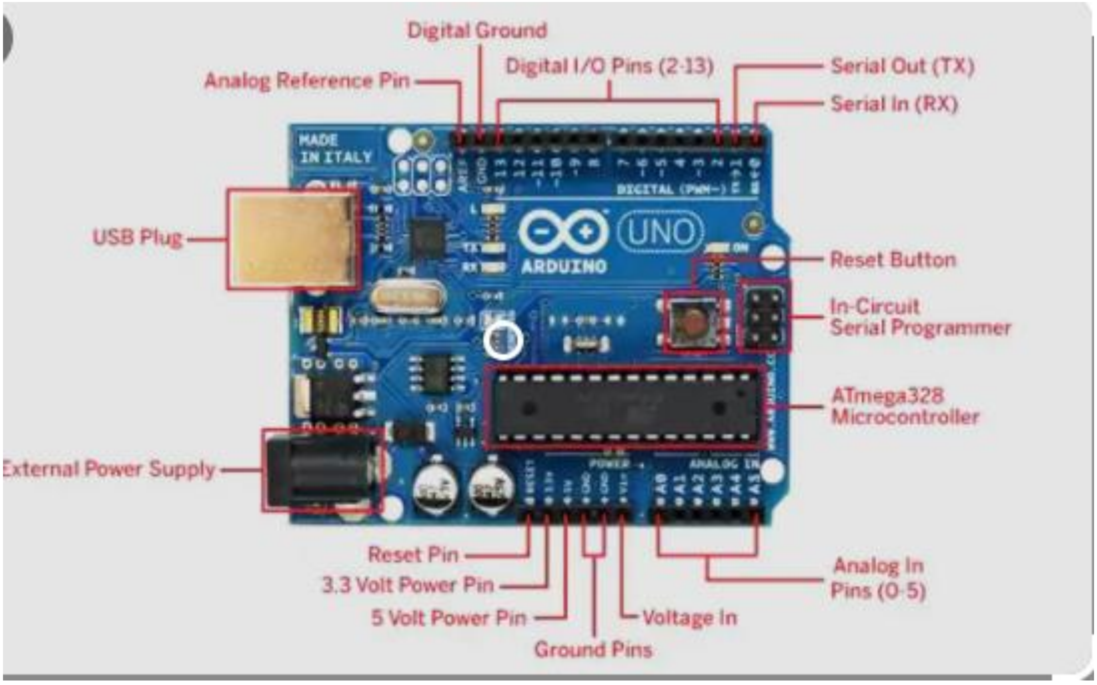

Figure 8: Arduino Uno

3.1.1.15 ESP8266 WIFI Module

So in our project, we have also interface an ESP8266 WIFI module to our Arduino for sending the data on the webserver. It is a low-cost wireless transceiver that can be used for end-point IoT developments. It uses UDP/TCP protocols to connect with the server.

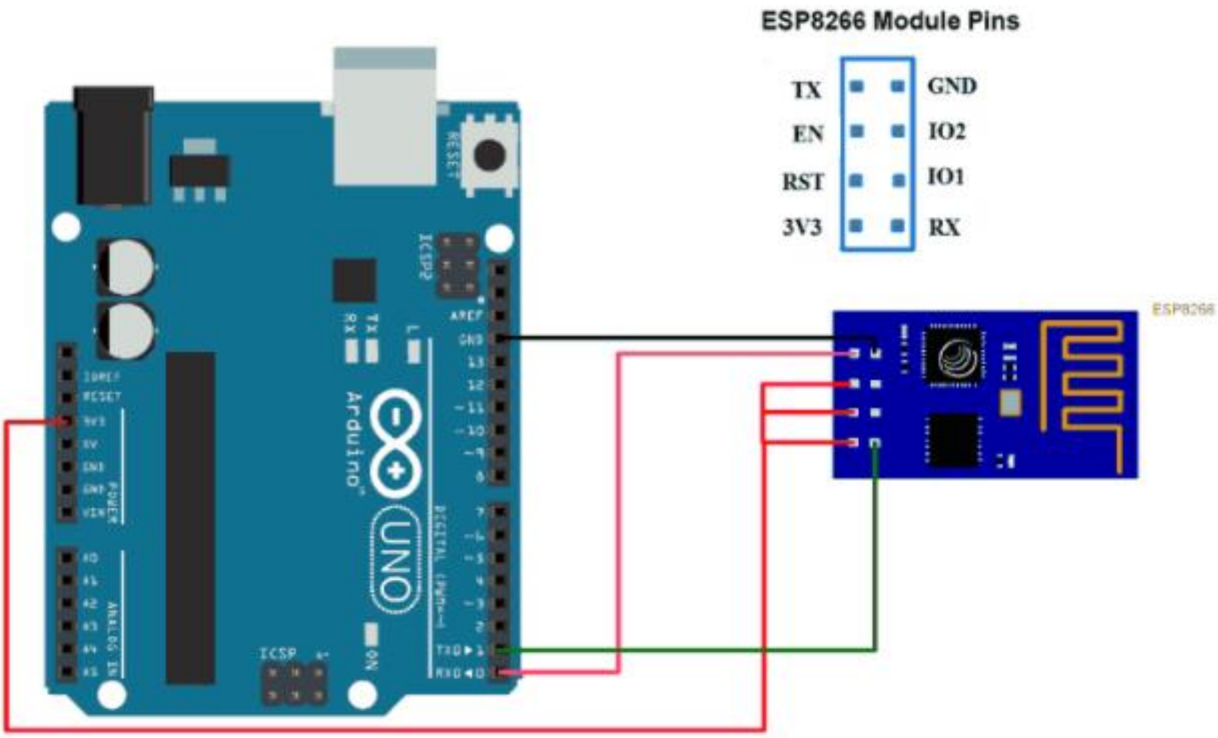

Figure 9: ESP8266 WIFI Module

So by using this WIFI module we are sending the values of Temperature, Humidity, and Soil Moisture Sensor on the webserver. We are also showing the status of Rain and Motor status on the web dashboard.

3.1.2 Hardware Design

The next part was to design and run the hardware from the simulation. We have successfully designed and run the hardware whose circuit diagram is given below:

3.1.2.1 Interfacing soil moisture sensor with Arduino

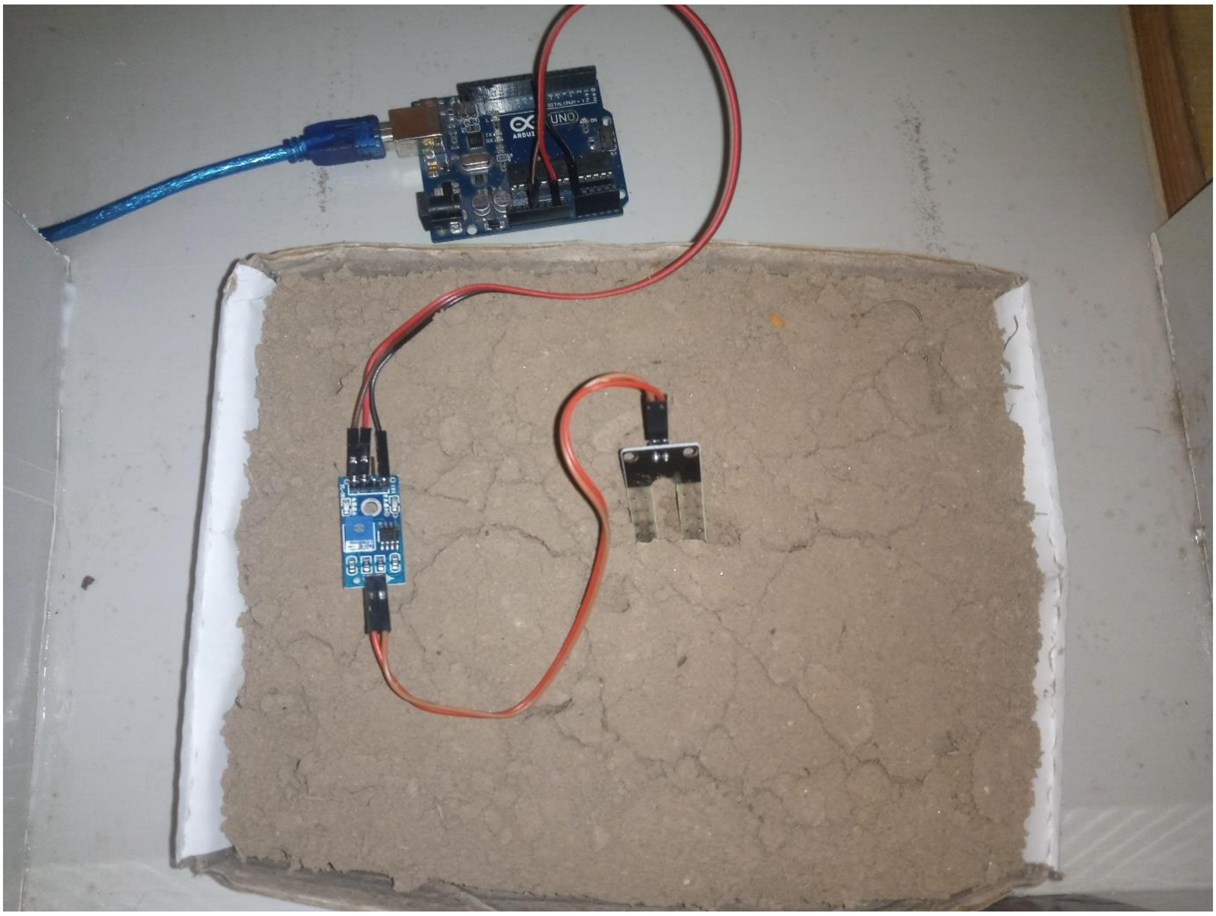

Figure 10: Soil Moisture

3.1.2.2 Interfacing temperature and humidity sensor with Arduino

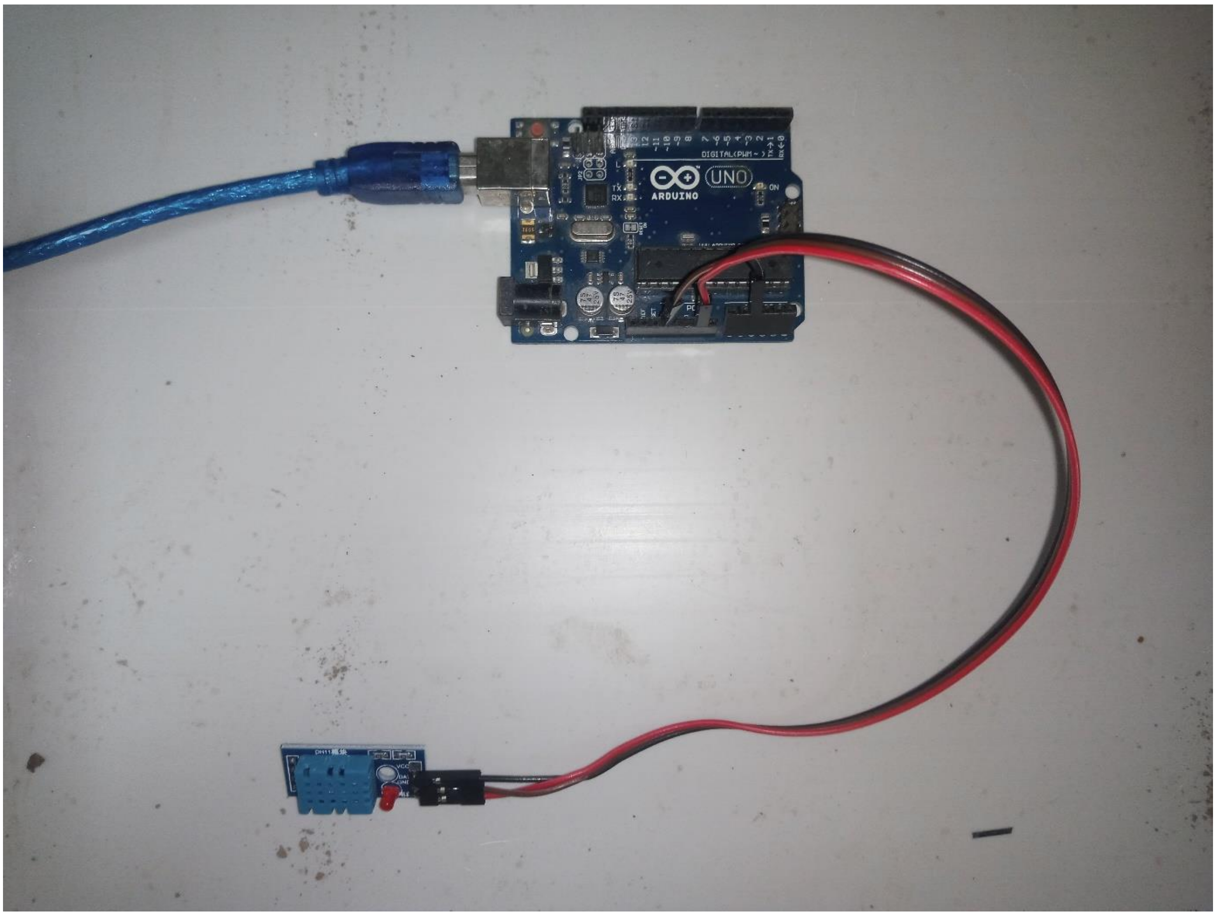

Figure 11: DHT11 Sensor

3.1.2.3 Interfacing rain sensor with Arduino

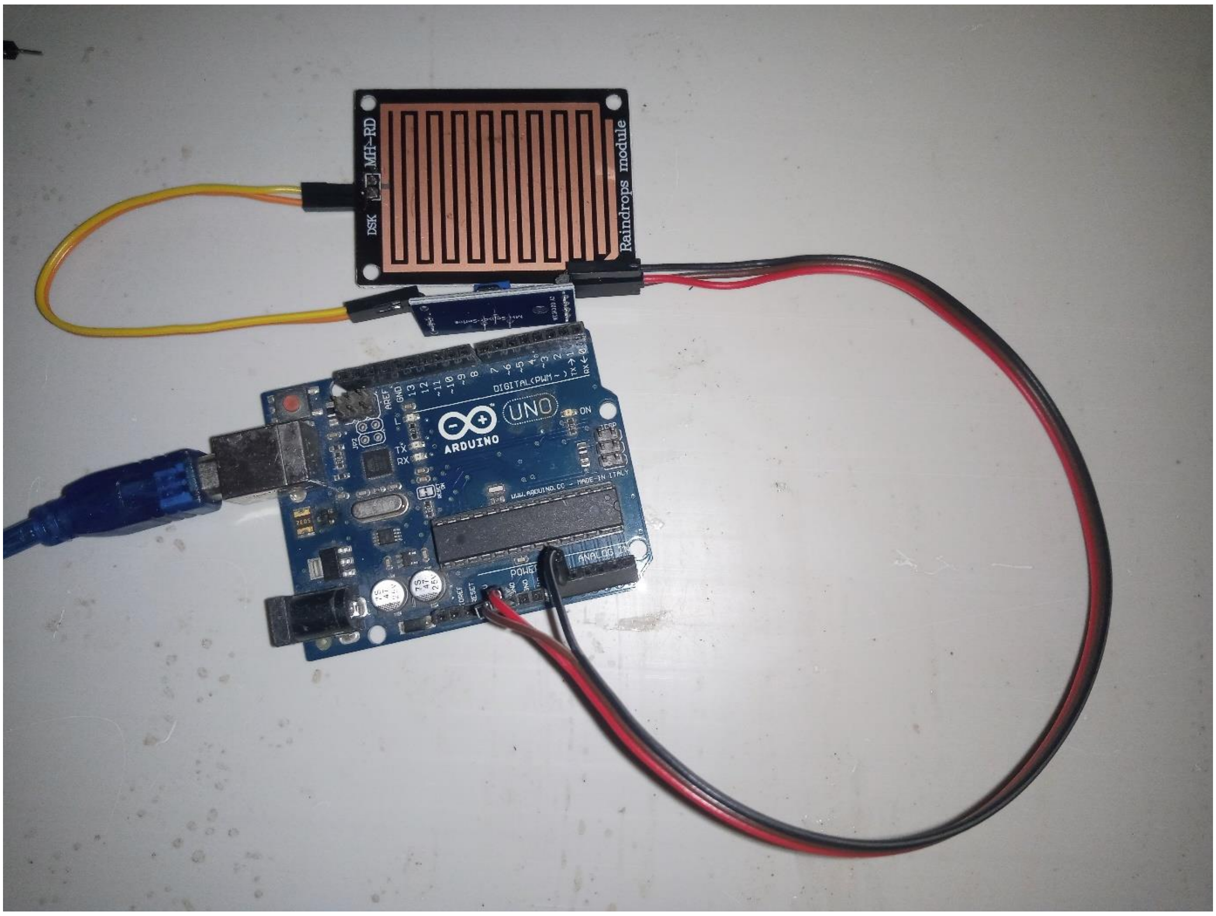

Figure 12: Rain Sensor

3.1.2.4 Interfacing relay and motor pump with Arduino

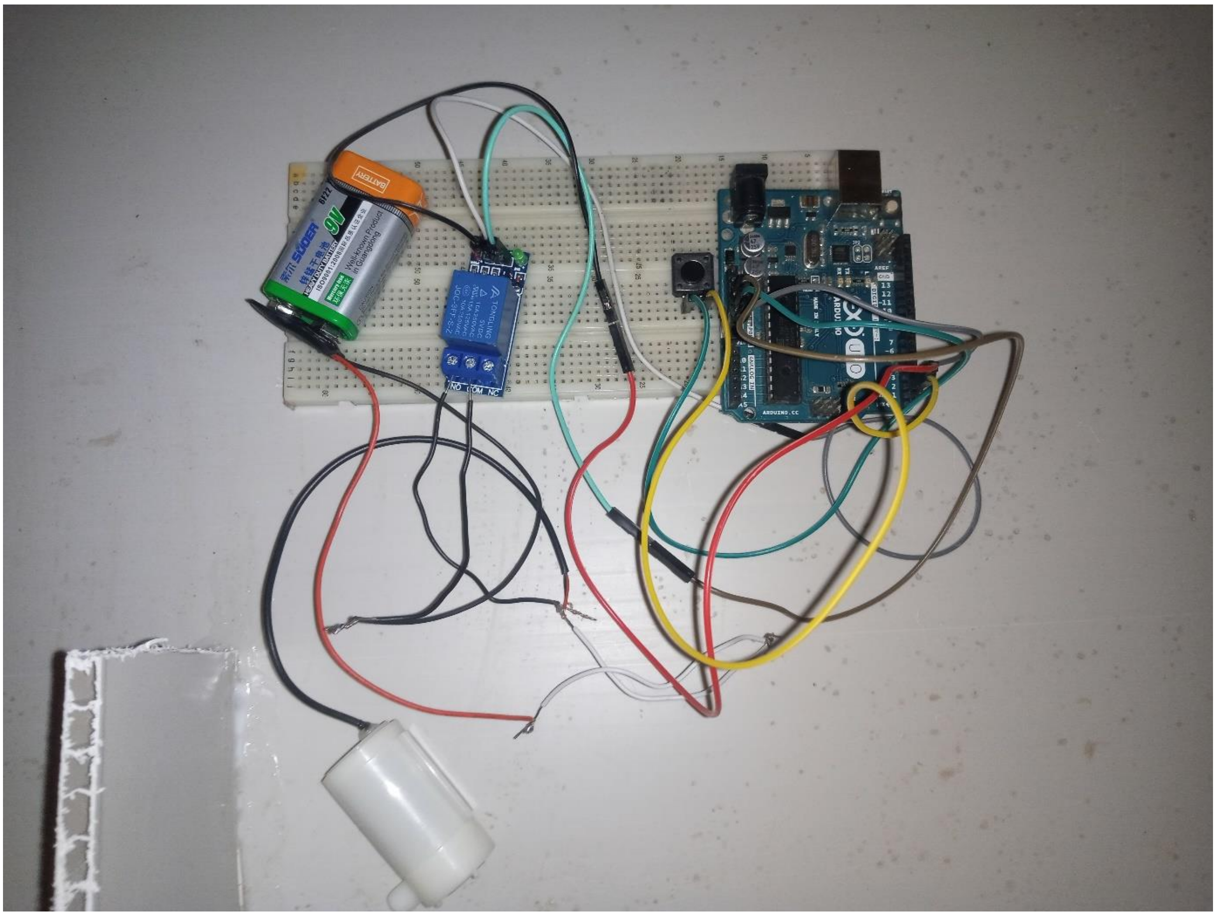

Figure 13: Water Pump Connection

3.1.2.5 Interfacing ESP8266 Wi-Fi module with Arduino

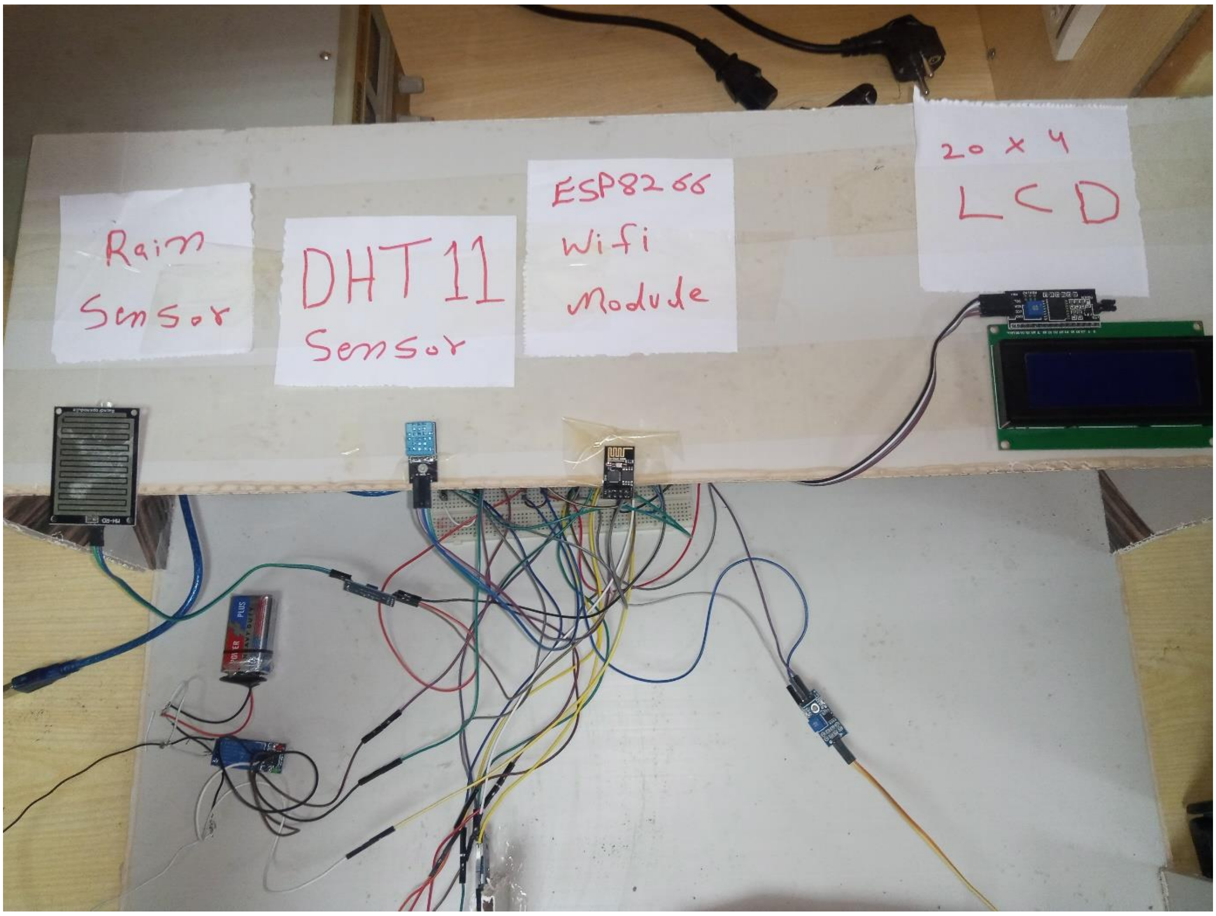

Figure 14: ESP8266 WIFI Module

3.1.2.6 Water drainage system circuit diagram

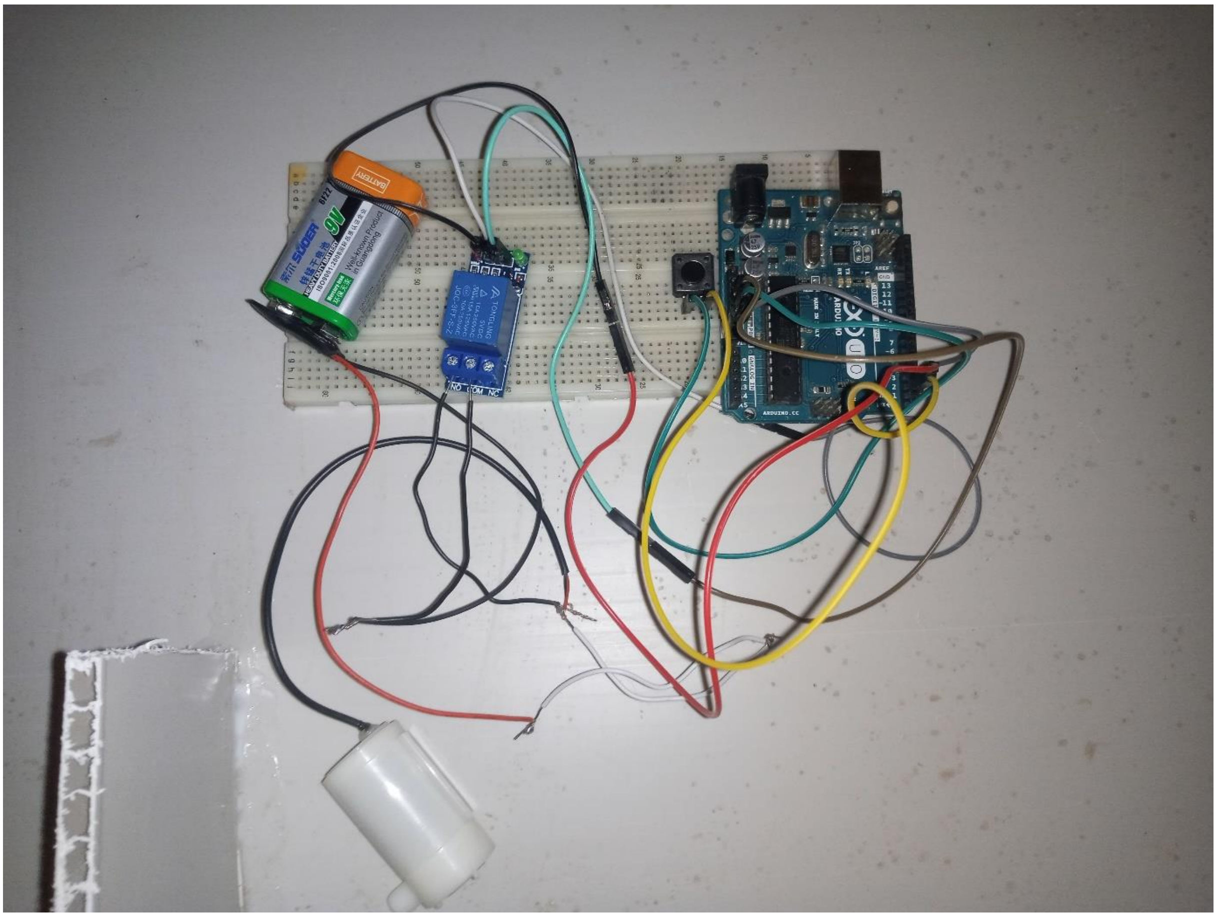

Figure 15: Water Drainage System

3.2 Complete Hardware Circuit Diagram

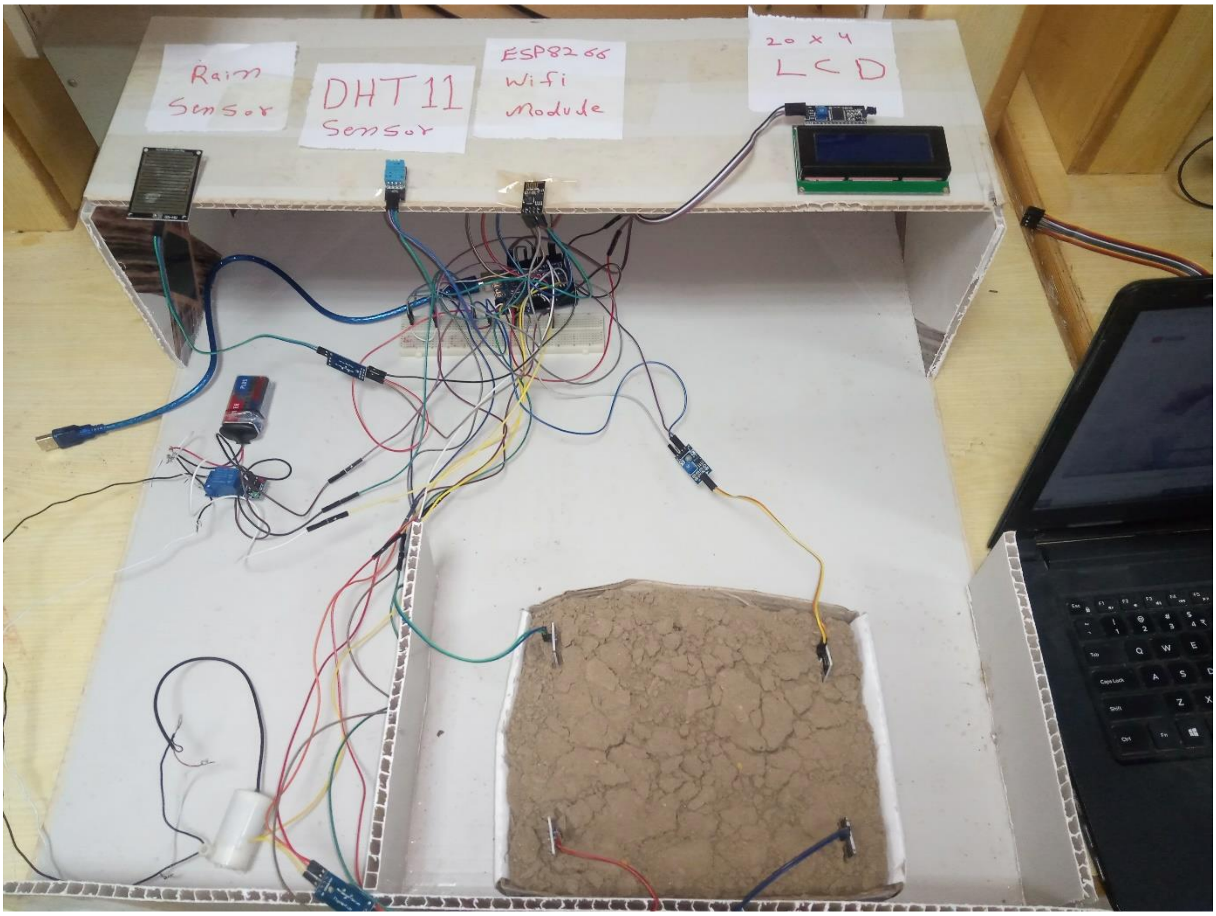

Figure 16: Complete Project Diagram

3.3 Coding of Hardware with Arduino

So now we have an interface for the different components used in the project. All the connections are properly made.

3.3.1 Code for soil moisture sensor and rain sensor:

```
 //Soil Moisture Part
// Function for the soil moisture and rain sensor
```

```
//Rain sensor function
String getRainvalue()
{
  val_analogique=analogRead(capteur_A);
  Serial.println("Rain Sensor Value = ");
  Serial.print(val_analogique);
  Serial.println("\n");
  return String(val_analogique);
  delay(100);
}

String getMoisture0()
{
   Moist0 = analogRead(sensor_pin_zero);
   Serial.println("Mositure0 = ");
   Serial.print(Moist0);
   return String(Moist0);
}
```

Figure 17: Sensor Coding

3.3.2 Coding of temperature and humidity:

```
String getTemperatureValue(){

   dhtObject.read(dht_apin);
   Serial.println(" Temperature(C)= ");
   Serial.println("\n");
   int temp = dhtObject.temperature;
   Serial.println(temp);
   //delay(50);
   return String(temp);

}


String getHumidityValue(){

   dhtObject.read(dht_apin);
   Serial.println(" Humidity = ");
   int humidity = dhtObject.humidity;
   Serial.println(humidity);
   Serial.print(" %");
   //delay(50);
   return String(humidity);

}
```

Figure 18: Sensor Coding 2

So now here we have put the basic code required for all the sensors to interface them with Arduino.

Note:

The complete code is given in the Appendices section of the report.

3.4 Flow Chart

Given below is the workflow of our project as shown.

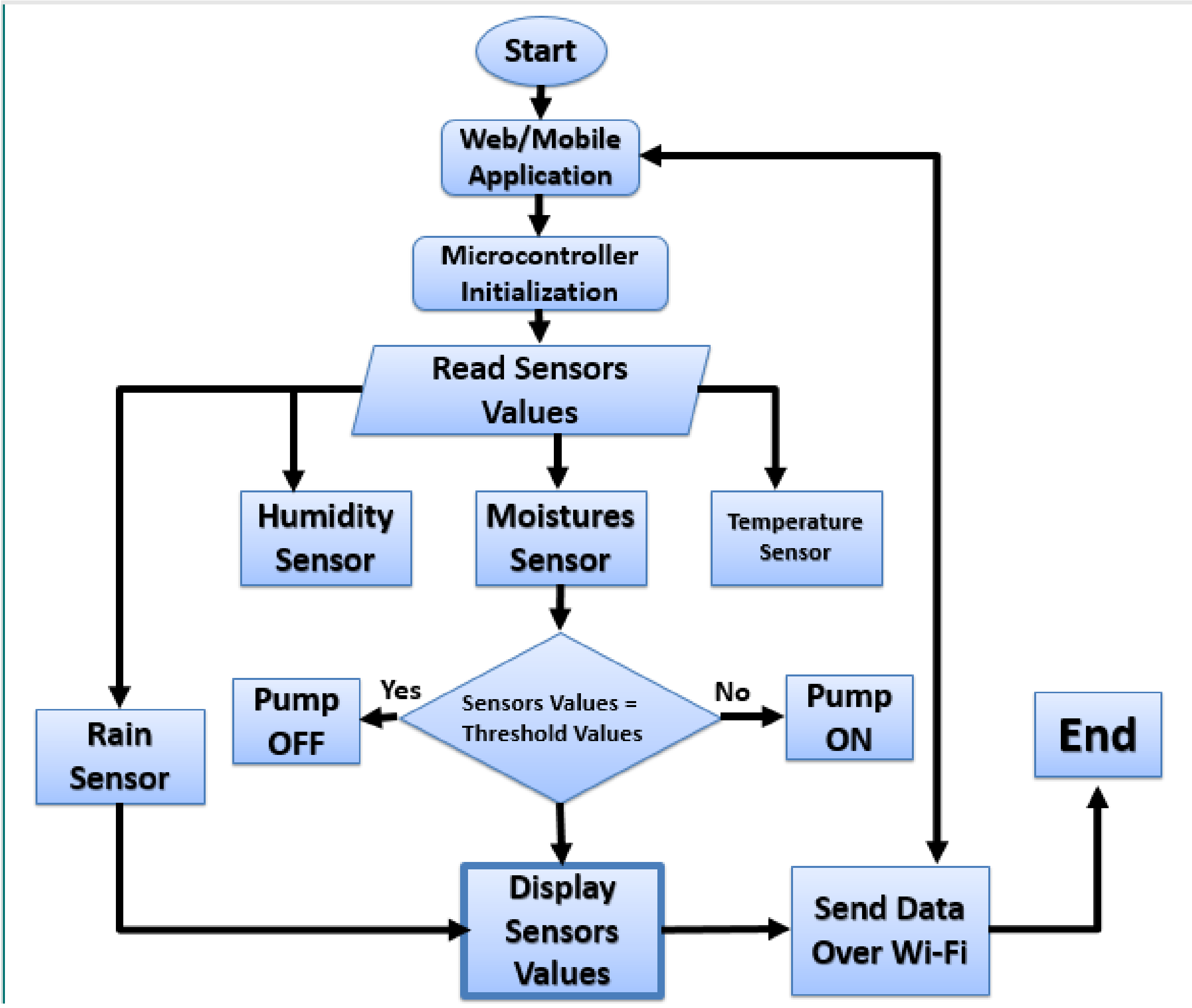

Figure 19: Flow Chart

3.5 Block Diagram

Give below is the block diagram of our project.

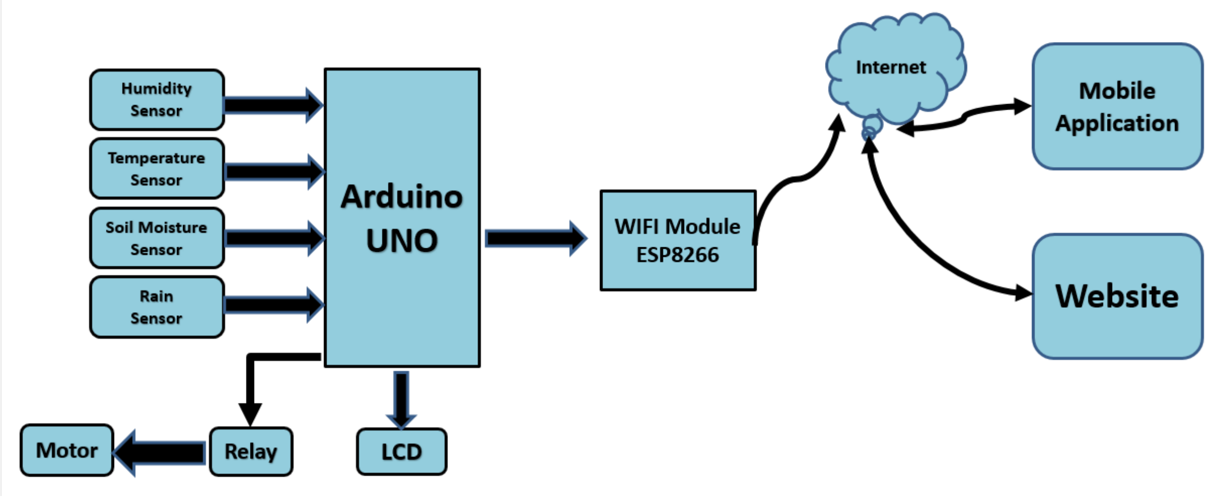

Figure 20: Block Diagram

So the above block diagram we can see that we have a unit of sensors which includes Soil, Temp, Humidity, and Rain sensors, and our microcontroller which is Arduino Uno is constantly reading the values of these sensors. We have also connected a motor pump with Arduino which is further connected with the Relay module. This motor pump is controlled by soil moisture and rain sensor values which we have discussed above.

We have also connected a Wi-fi module with Arduino which will collect the data from the sensors and send it to the IoT web server.

3.6 Implementation Constraints

Provide information about the constraints that affected the project implementation, and the measures used as a solution as a result.

Given below are the constraints that affected the project implementation

1. **Faulty Hardware Components:** As we have purchased all the components online why some of them were not working properly. It costs too much time. To solve this, we buy 2 or 3 extra components so that if one gets damaged or not working properly, we can use another one.
2. **Technical specification of hardware:** The sensors and components were not working that way as mentioned in their specification due to it takes time to complete this.
3. **Arduino IDE errors:** With excessive use of Arduino, it started to show error sometime without any changes in the code. So it solution was to restart the Arduino IDE.
4. **ThingSpeak**: One of the most important challenge faced during this system implementation was finding the most appropriate cloud platform to use for the project. Then we selected ThingSpeak because it provides a free version for the user.

3.7 Web Dashboard Design

We explained above that we are sending the values of temperature, humidity, and soil moisture sensor on a web server by using the esp8266 wifi module. So we have designed a web dashboard for displaying the values of

- Temperature
- Humidity
- Soil Moisture Sensor
- Rain Status

we have used the ThingSpeak platform to display the data of our sensors. So here we have created a private channel for the farmer. As shown in the below figure

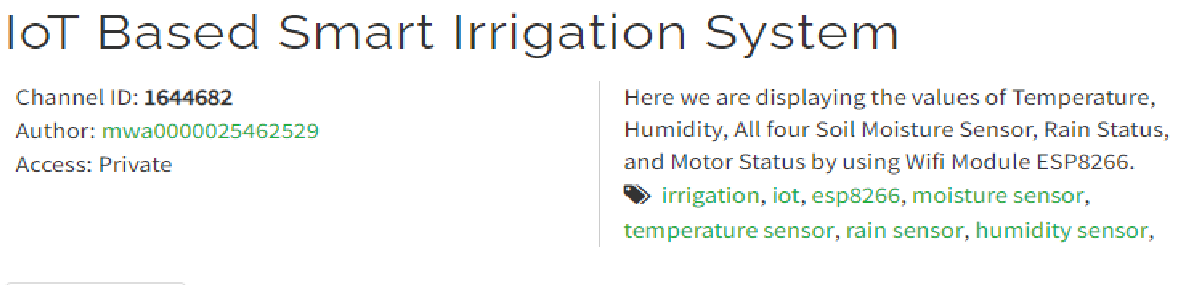

Figure 21: Web Design

So here we have also created 7 fields for

- Temperature Values
- Humidity Values
- M_Moisture Sensor Values
- Q_Moisture Sensor Values
- S_Moisture Sensor Values
- T_Moisture Sensor Values
- Rain Sensor

The output of these can be seen in the result section.

3.8 Mobile App Design

We explained above that we are sending the values of temperature, humidity, and soil moisture sensor on a web server by using the esp8266 Wi-Fi module. Then we have to design a mobile application using an MIT app inverter which will display

- Temperature
- Humidity
- Soil Moisture Sensor

The designed result is given below:

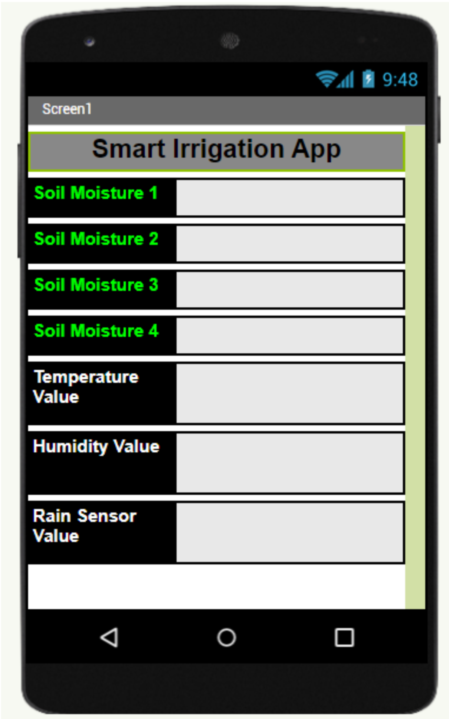

Figure 22: Mobile App Design

Then we have downloaded the application and run it on our smart phone.

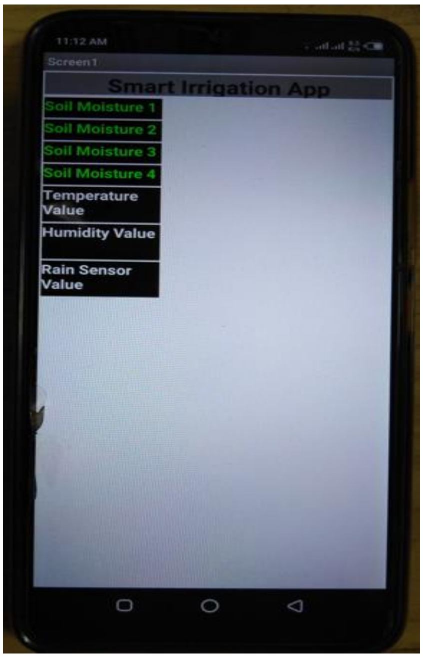

Figure 23: App

We were not able to retrieve the data from the ThingSpeak server into our designed mobile app. It is because we have used the free version of the ThingSpeak server and for retrieving the data from the server into MIT App inventor, it asks for the paid version so that's why we are unable to get data into the mobile app.

So here in the below figure, we can see that it is denying the permission.

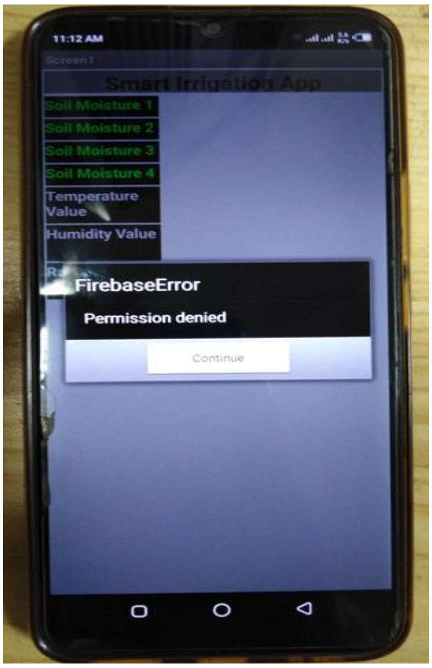

Figure 24: Permission Denied

Chapter 4

4. Results

4.1 Motor pump control using a soil moisture sensor and Rain Sensor

In this section, we will explain how we are controlling the motor pump using the values of our soil moisture sensors and rain Sensor. Soil moisture that we used in this project measured the values from 0 to 1023.

4.1.1 Condition 1: (Soil Moisture Values > 700 && No Raining)

As we are continuously reading the values of all four soil moisture sensors. When all four sensor values are greater than 700 it means that our soil is dry and the motor Pump will be ON only if there is no rain.

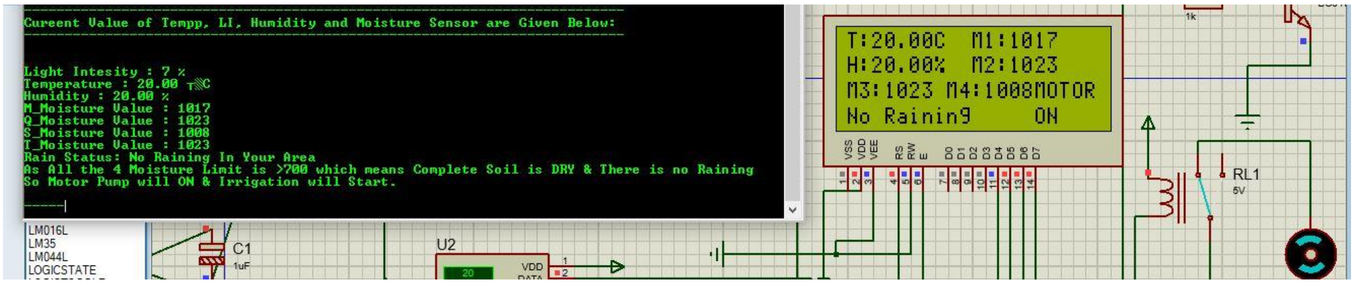

Figure 25: Moisture>700 & No Raining

As shown in the above picture we can see the rain status which shows "No Raining in Your Area". All four sensor values are > 700 and there is no rain, so there is a need for irrigation so the motor pump is ON.

4.1.2 Condition 2: (Soil Moisture Values between 300 & 700 & No Raining)

As we are continuously reading the values of all four soil moisture sensors. When all four sensor values are less than 700 and greater than 300 it means that our soil is moist so the motor pump will be on and irrigation will start only if there is no rain.

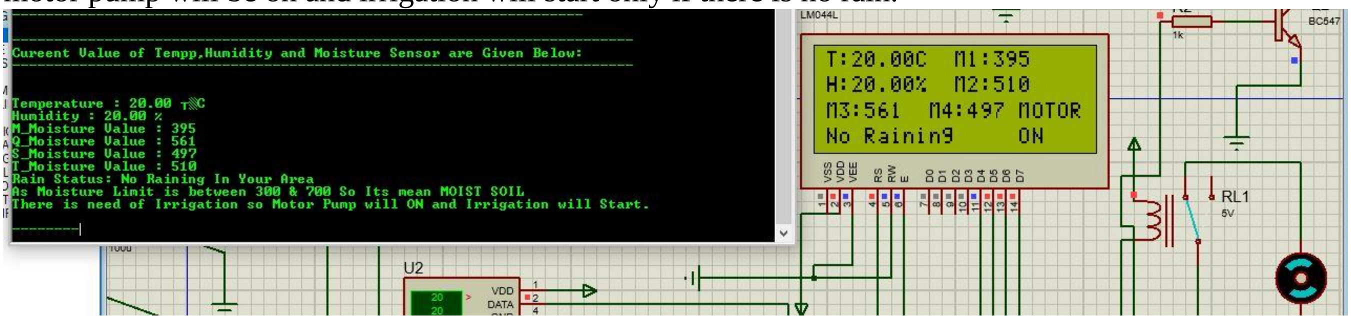

Figure 26:Moisture>300&& Moisture<700 && No Raining

As shown above, our motor is on and irrigation will start in case of moist soil.

4.1.3 Condition 3: (Soil Moisture Values < 300 & No Raining)

As we are continuously reading the values of all four soil moisture sensors. When all four sensor values are less than 300 it means that our soil is completely wet so the motor pump will remain OFF.

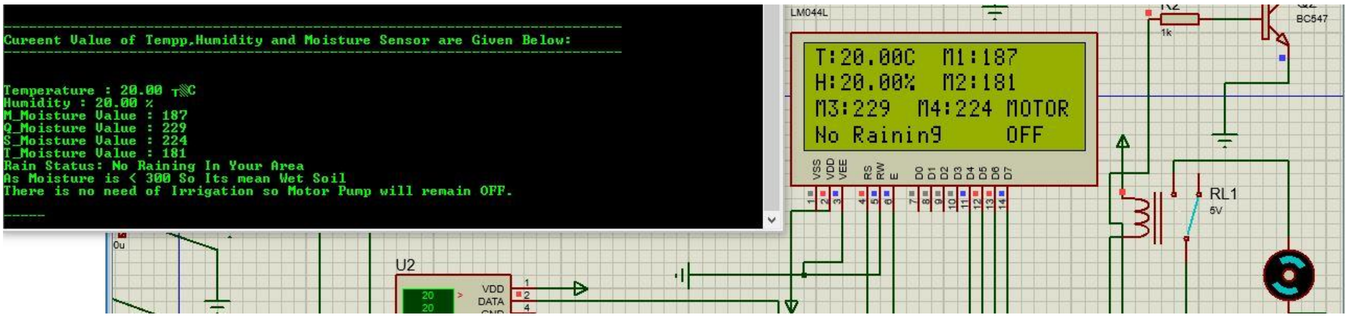

Figure 27: Moisture<300 & No Raining

4.1.4 Condition 4: (Anyone soil moisture sensor value > 700 & No Raining)

As we are continuously reading the values of all four soil moisture sensors. We have connected four soil moisture sensors and integrated them. Any one of the four sensors will give a moisture value greater than 700 then which means that one part of the soil is completely dry. so our motor pump will be on and irrigation will start. As shown in the below figure

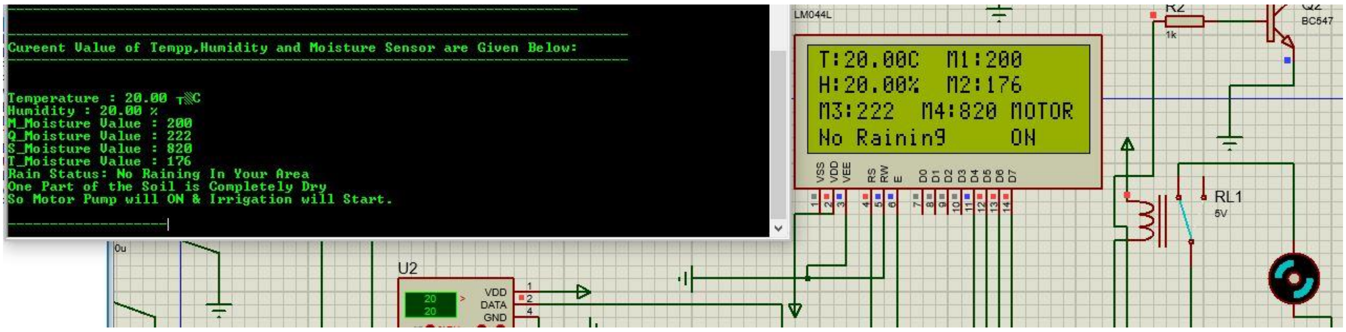

Figure 28: Anyone Moisture>700 & No Raining

4.1.5 Condition 5: (Anyone soil moisture sensor value > 700 & Raining)

As we are continuously reading the values of all four soil moisture sensors. We have connected four soil moisture sensors and integrated them. Any one of the four sensors will give a moisture value greater than 700 then which means that one part of the soil is completely dry. Its means that there is a need for irrigation but rain has started. So in this case our motor pump will remain off and irrigation will stop. As shown in the below figure

Figure 29:Anyone Moisture>700 & Raining

4.1.6 Condition 6: (Soil Moisture Values between 300 & 700 & Raining)

As we are continuously reading the values of all four soil moisture sensors. When all four sensor values are less than 700 and greater than 300 it means that our soil is moist so there is a need for irrigation but the rain has started so the motor pump will remain OFF and irrigation will stop.

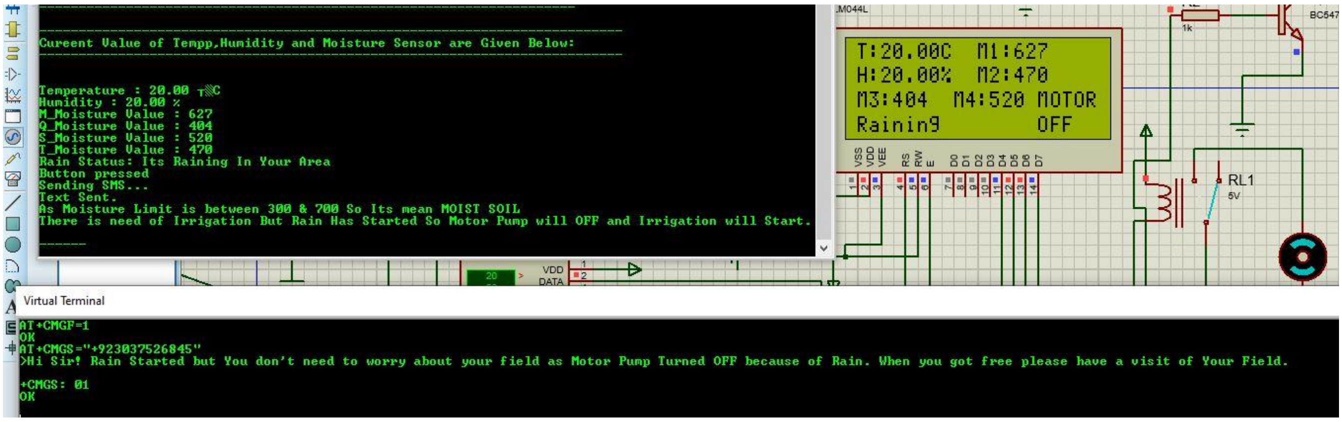

Figure 30:Moisture>300&& Moisture<700 && Raining

4.1.7 Condition 7: (Soil Moisture Values > 700 && Raining)

As we are continuously reading the values of all four soil moisture sensors. When all four sensor values are greater than 700 it means that our soil is dry and there is a need for irrigation but the rain has started so our motor Pump will remain OFF and irrigation will stop.

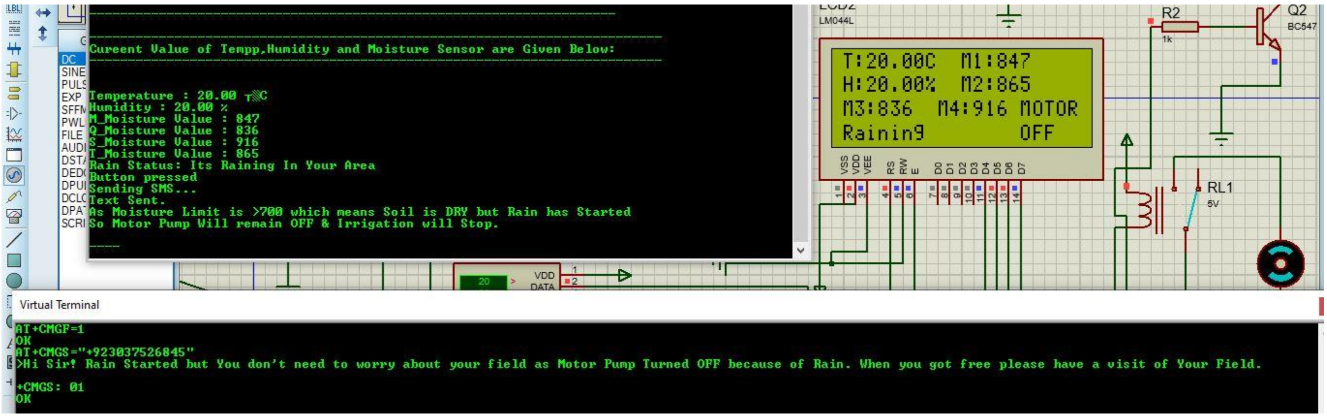

Figure 31:Moisture>700 && Raining

4.2 Temperature and Humidity Sensor:

Given below is the value of the temperature and humidity sensor measured by the DHT11 Sensor

Figure 32: DHT11 Output

4.3 Hardware Result:

4.3.1 Web Dashboard Result:

So as we discuss previously that our sensors are constantly sending the data on the webserver. Given below are the output result of sensor values that are displayed over the web dashboard. These values are sent by the ESP8266 Wi-Fi module on the ThingSpeak dashboard.

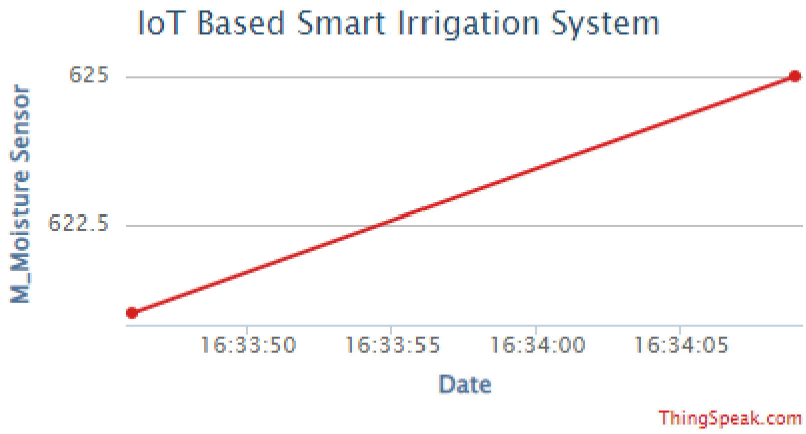

Figure 33: ThingSpeak Ch 1

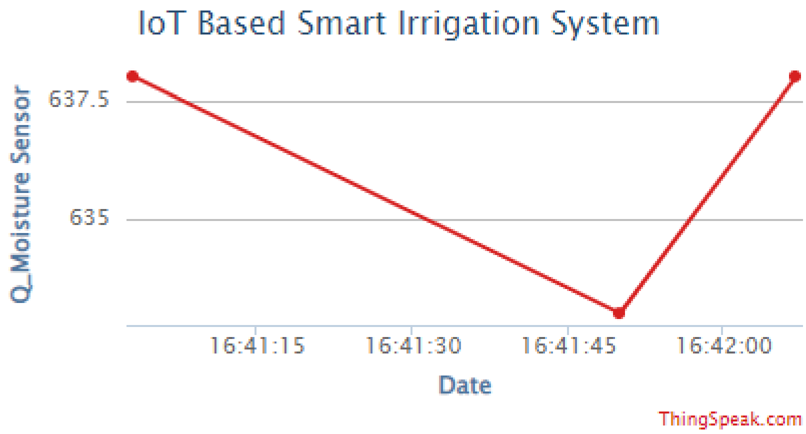

Figure 34: ThingSpeak Ch 2

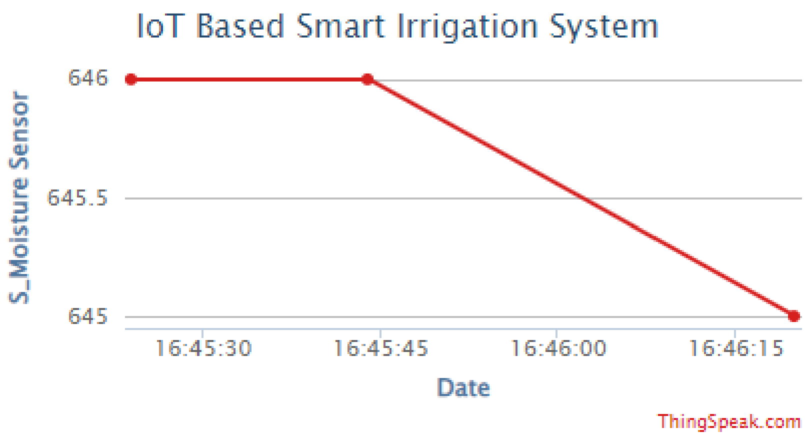

Figure 35: ThingSpeak Ch 3

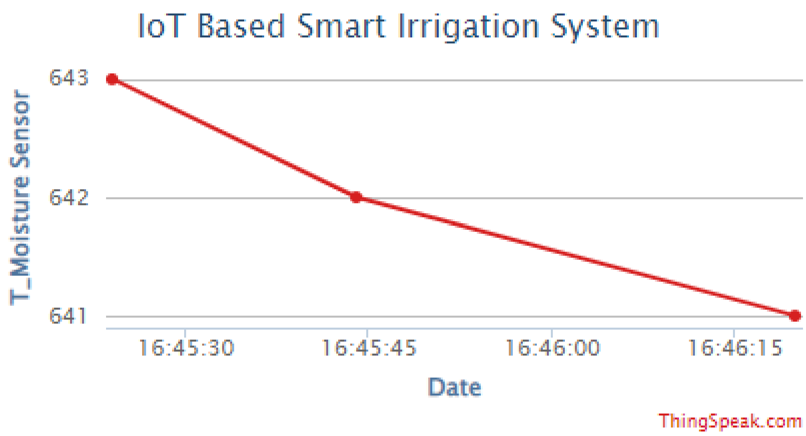

Figure 36: ThingSpeak Ch 4

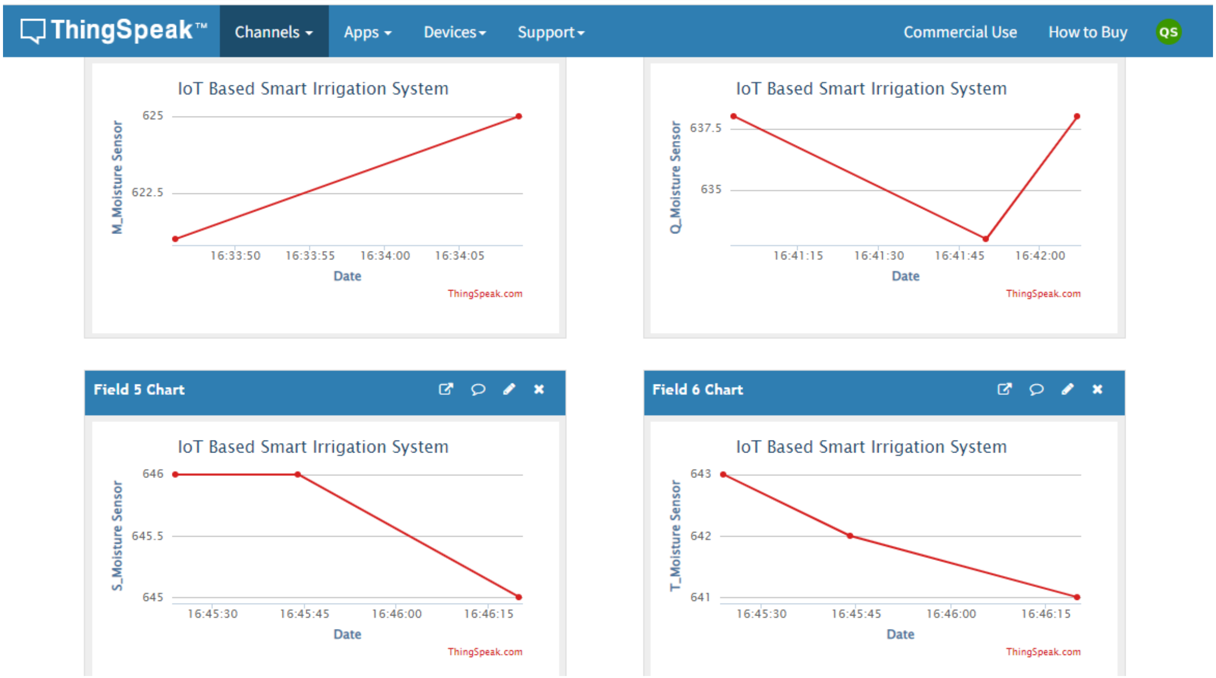

Figure 37: ThingSpeak Ch 5

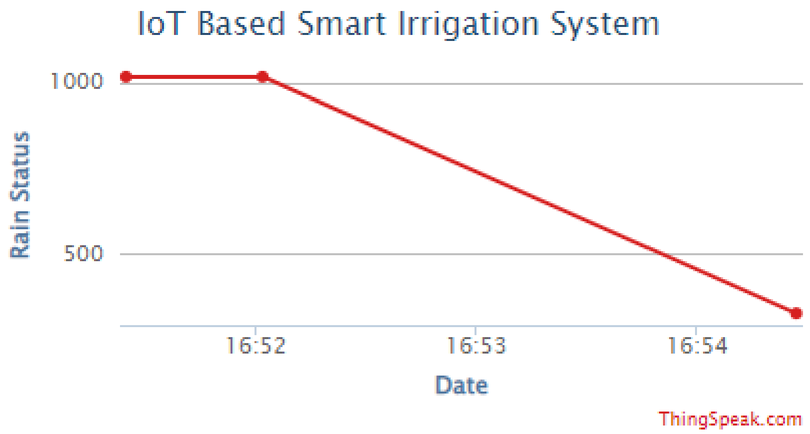

Figure 38: ThingSpeak Ch 6

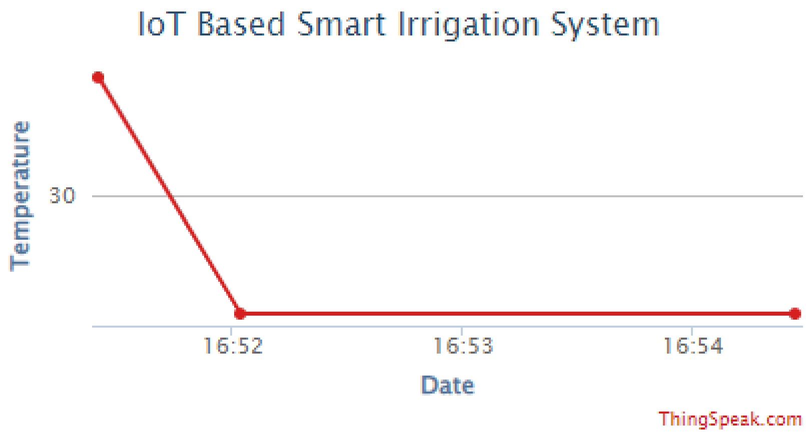

Figure 39: ThingSpeak Ch 7

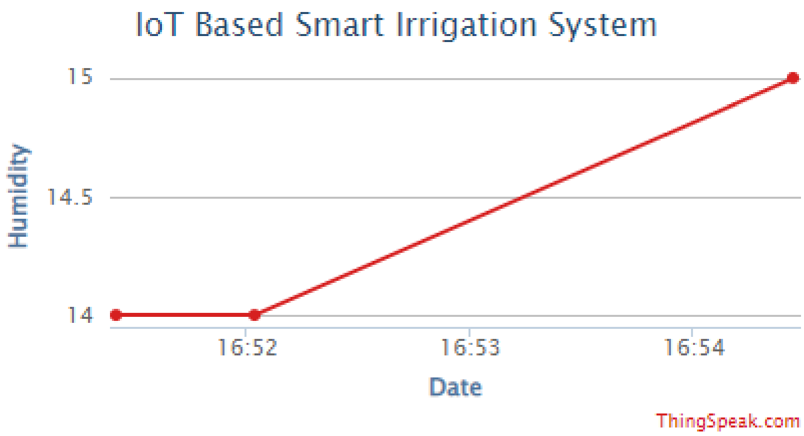

Figure 40: ThingSpeak Ch 8

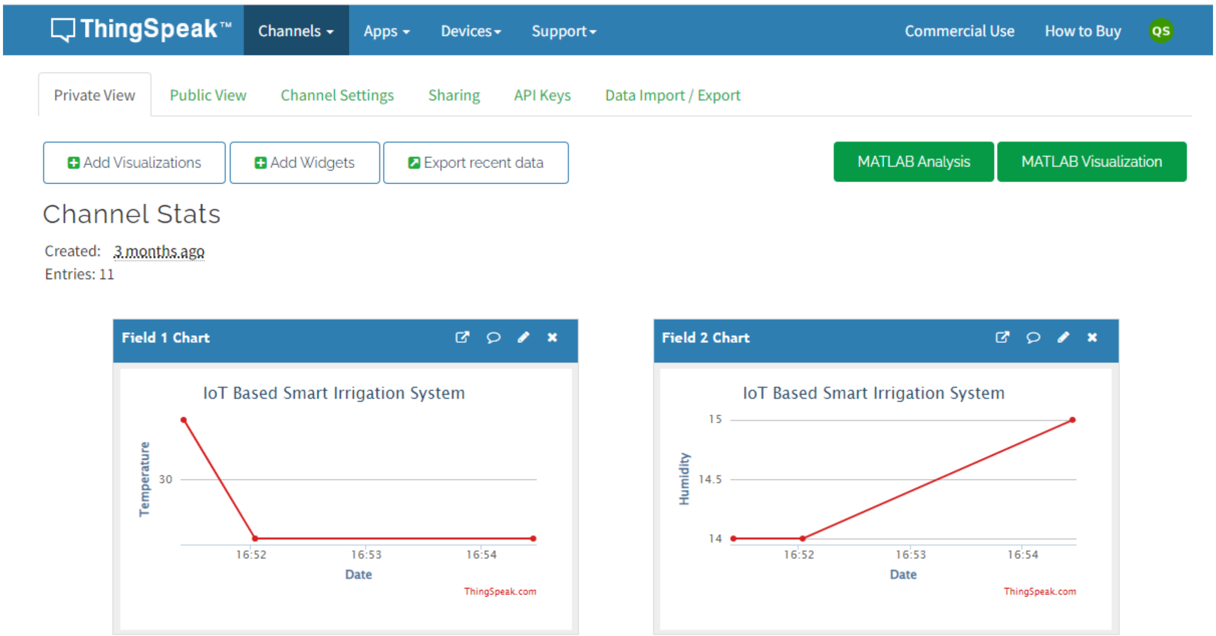

Figure 41: ThingSpeak Ch 10

Chapter 5

5. Discussion

After completing the software and hardware simulation and execution for the smart irrigation system, we achieved our goal. All the necessary and proposed requirements have been done smartly in this smart Irrigation system, that's why it is finalized now.

After this step, we tested the system on Proteus where its works properly. All the relevant results and their work can be seen in the previous section. We run our system on more than five conditions in the simulation part for its better working and also verified the results.

After that, we designed the hardware part of the project. First of all, we have tested and verified all the components including sensors and motor pump. Then based on our simulation circuit diagram we have started to design the hardware. We have connected four soil moisture sensors in the soil and connected them with Arduino. Now Arduino can get soil moisture sensor values to form the soil. This system is configured in this way that if there is rain then no irrigation should be done to reduce the wastage of water. For this purpose, we have used the rain sensor which will send the signal to Arduino if there is rainy or not. If it's raining, then our Arduino will send a signal and the motor pump will off automatically.

In our project, we are also measuring the values of temperature and humidity for two reasons. As we know Temperatures are high in summer and low in winter. Therefore, crops need less water in winter than in summer. So based on temperature and humidity sensor values in winter and summer, we can change the moisture limit of our project.

Another reason for using this is that we can observe the weather condition without going to a weather forecasting agency or without opening Google.

So in our project, we have also interface an ESP8266 WIFI module to our Arduino for sending the data on the webserver.

5.1 Limitation:

Given below are the limitation of this project.

- The farmer must have an excess to the mobile phone or laptop.
- The farmer must have a strong internet connection.

- We were not able to retrieve the data from the ThingSpeak server into our designed mobile app. It is because we have used the free version of the ThingSpeak server and for retrieving the data from the server into MIT App inventor, it asks for the paid version so that’s why we are unable to get data into the mobile app.

Chapter 6

6. Conclusion

As IoT-based irrigation systems are making it workable for a farmer to gather important information which leads to better crop production. So if we shift from the old irrigation system to IoT based system then the need for the increasing population of Pakistan can be fulfilled. In this system, we have provided a new approach to the old used irrigation system where a huge amount of water is wasted. The soil moisture, temperature, humidity, and rain sensor values are measured and sent to the Arduino for further analysis. The farmers can monitor their farms from anywhere in the country via a mobile app or website dashboard.

Chapter 7

7. Future Work

To provide a predictive model so that the farmers can predict the future forecasting related to their field as he is constantly collecting the data from the sensors. It can be done by using machine learning models. To control the system via Zig Bee instead of wires which can provide very good results. We can also make this system workable using renewable energy which is solar power instead of using batteries. So by using solar energy we can reduce the future cost easily. Shifting towards the agri-business which focus on providing fresh product quality. Using the agriculture drones for the better monitoring and protection of the crops and fields. Adding the AI system to predict the production of crops.

Chapter 8

8. Reflections on Learning

8.1 Technical learning

We have learned many things by doing this project. Many technologies we have learned by doing this project and it gives us a good command of languages like C++.

Given below are the technical skills that we have learned by doing this project:

Arduino:

Proteus:

We have also learned about the internet of things. How to connect different devices using the internet of things and how it works.

One of the most important learning that this project gives us is to design a web dashboard and how to send data on this dashboard using Arduino.

We also learned the C++ language for the coding of this project.

8.2 Non-Technical skills:

Given below are the non-technical skills that we have learned by doing this project.

- Time Management
- Communication skills
- Team working
- Hard-working
- Data collection
- Things management
- Hardware designing
- Writing skills
- Presentation skills

9. References

[1] Moller, J. (2010). A Versatile Technology in automation of agriculture machinery. Computer virsion, 17.

[2] Design and Implementation of Automatic Plant Watering System “Archana and Priya” presented at International Journal of Advanced Engineering and Global technology, vol- 04, Issue-01, Jan-2016.

[3] Sensor Based Automated Irrigation System with IOT “Karan Kansara and Vishal Zaweri,” presented at International Journal of Computer Science and Information Technologies, vol-06, 2015.

[4] C.H.Chavan and V.Karnade, “ Wireless Monitoring of Soil moisture, Temperature and Humidity using Zigbee in Agriculture” presented at International Journal of Engineering Trends and Technology (IJETT), vol-11, May- 2014.

[5] Asian Journal of Science and Technology Vol. 10,Issue 0 6, pp.9756-9768, June, 20.

[6] Measurement of Temperature and Humidity by using Arduino Tool and DHT11 “Deeksha Srivastava, Awanish Kessarwani, Shivani Dubey”, International Research Journal of Engineering and Technology (IRJET), volume5, Issue 12, Dec 2018.

[7] The working principle of an Arduino “Yusuf Abdullahi Badmasai” published in 2014 11th International Conference on Electronics, Computer and Computation (ICECCO).

[8] Shen Jin, Song Jingling, Han Qiuyan, Wang Shengde, and YangYan, School of Electric and Electronic Engineering, A Remote Measurement and Control System for Greenhouse Based on GSM-SMS, IEEE 8th International Conference on Electronic Measurement and Instrument, 2007,pp. 45-82

[9] Webin Huang, Guanglong Wang, Research of Wireless Sensor Networks for an Intelligent Measurement System Based on ARM, pp.1074- 1079, 2011.

[10] Sezen SM, Yazar A, Irrigation Management on Yield And Quality Of Tomatoes Grown in different Soilless Media in Glasshouse, 41-48, 2010.

[11] Daniel K.Fisher and HirutKebede, a Low Cost Microcontroller-Based System to Monitor Crop Temperature and Water Status, pp. 168-173, 2010.

[12] Joaquín Gutiérrez, Juan Francisco Villa-Medina Automated Irrigation System Using A Wireless sensor Network And Gprs Module Ieee Transactions On Instrumentation And Measurement, Vol. 63, No. 1, January 2014

[13] F.S. Zazueta., A. G Smajstrla And G. A Clark,"Irrigation System Controllers. IFAS Extension, Agricultural And Biological Engineering Department, Florida Cooperative Extension Service, University Of Florida. Gainsville, FL. 2002.

[14] Purnima, S.R.N. Reddy, PhD. Design of Remote Monitoring and Control System with Automatic Irrigation System using GSM-Bluetooth International Journal of Computer Applications (0975 – 888)Volume 47– No.12, June 2012

[15] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel Porta-Gándara. Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 63, NO. 1, JANUARY 2014

10. APPENDICES

10.1 Appendix A:

Coding of the Project

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h> //LCD Library
#include <dht11.h>
#define RX 2
#define TX 3
#define dht_apin 11 // Analog Pin sensor is connected to
dht11 dhtObject;
String AP = "Namal-Net";       // AP NAME
String PASS = "namal123"; // AP PASSWORD
String API = "WQ3MB9BO9E7VJFDB";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial esp8266(RX,TX);
//Rain Sensor
const int capteur_A = A4;
int val_analogique;
//Ended
```

```
  //Soil Moisture Part
  int sensor_pin_zero = A0; // Soil Sensor input at Analog PIN A0
  int sensor_pin_one = A1; // Soil Sensor input at Analog PIN A1
  int sensor_pin_two = A2; // Soil Sensor input at Analog PIN A2
  int sensor_pin_three = A3; // Soil Sensor input at Analog PIN A3

  int Moist0 ;
  int Moist1 ;
  int Moist2 ;
  int Moist3 ;
  //Soil Moisture Part
void setup() {
  Serial.begin(9600);
  Serial.println(F("*******************************************************************************"));
  Serial.println("\n");
  Serial.println("***********             WelCome To The Project                   ***********\n");
  Serial.println("\n");
  Serial.println(F("*** IoT Based Smart Irrigation System By EE Students of Namal University ***\n"));
  Serial.println("\n");
  Serial.println(F("***---------------------------------------------------------------------***\n"));
  Serial.println(F("**************************************************************************\n"));
  Serial.println("\n");
  Serial.println(F("Reading the Value of Moisture, Temp, Humidity and Rain Sensor\n"));
  pinMode(capteur_A, INPUT);//Rain Sensor Part
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}
```

```
void loop() {

 String getData = "GET /update?api_key="+ API +"&field1="+getTemperatureValue()+"&field2="+getHumidityValue()+"&field7="+getRainvalue();

 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");

  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(getTemperatureValue());
  lcd.print("C");
  lcd.setCursor(9, 0);
  lcd.print(" M1:");
  lcd.print(Moist1);
  lcd.setCursor(0, 1);
  lcd.print("H:");
  lcd.print(getHumidityValue());
  lcd.print("%");
  lcd.setCursor(9, 1);
  lcd.print(" M2:");
  lcd.print(Moist2);
  lcd.setCursor(0, 2);
  lcd.print("M3:");
  lcd.print(Moist3);
  lcd.setCursor(8, 2);
  lcd.print("M0:");
```

```
   //Case1: If all the soil moisture sensor values is greater than 600; Dry Soil
   if(Moist0>600)
   { if(Moist1>600)
   { if(Moist2>600)
   { if(Moist3>600)
   {
    digitalWrite(relayPin, HIGH);
        lcd.setCursor(15, 2);
        lcd.print("MOTOR");
        lcd.setCursor(15, 3);
        lcd.print("ON");
        Serial.print("\n");
        Serial.println(F("As All the 4 Moisture Limit is >700 which means Complete Soil is DRY & There is no Raining"));
        Serial.print("\n");
        Serial.print("\n");
        Serial.println(F("So Motor Pump will ON & Irrigation will Start."));
        Serial.println();
        Serial.println("---------------------------------------------------------------------------");
        Serial.println();
   }}}
 //  delay(1000);
  }
```

```
  //Case2: If all the soil moisture sensor values is between 370 to 600; Moist Soil

if(Moist0>=370 && Moist0<=600)
 {
  if(Moist1>=370 && Moist1<=600)
 {
  if(Moist2>=370 && Moist2<=600)
 {
  if(Moist3>=370 && Moist3<=600)
 {
  digitalWrite(relayPin, HIGH);
  Serial.println(F("As Moisture Limit is between 370 & 600 So Its mean MOIST SOIL"));
  Serial.print("\n");
  Serial.print("\n");
  Serial.println(F("There is a  need for Irrigation so Motor Pump will ON and Irrigation will Start.""));
  Serial.print("\n");
  Serial.println();
  Serial.println("--------------------------------------------------------------------");
  Serial.println();
 }}}
 delay(1000);
 }


 //Case3: If all the soil moisture sensor values is less than 370; Wet soil
 if(Moist0<=370)
 { if(Moist1<=370)
 { if(Moist2<=370)
 { if(Moist3<=370)
 {
  digitalWrite(relayPin, LOW);
  lcd.setCursor(15, 2);
  lcd.print("MOTOR");
  lcd.setCursor(15, 3);
  lcd.print("OFF");


  Serial.print(F("As Moisture is < 370 So Its mean Wet Soil"));
  Serial.println();
  Serial.println(F("There is no need of Irrigation so Motor Pump will remain OFF."));
  Serial.println();
  Serial.println("--------------------------------------------------------------------");
  Serial.println();
 }}}
// delay(1000);
}
```

```
//  //Any one soil moisture sensor value is greater than 600
  if(Moist0>600 || Moist1>600 || Moist2>600 || Moist3>600)
  {
   digitalWrite(relayPin, HIGH);
   lcd.setCursor(15, 2);
   lcd.print("MOTOR");
   lcd.setCursor(15, 3);
   lcd.print("ON");


   Serial.print(F("One Part of the Soil is Completely Dry"));
   Serial.println();
   Serial.println(F("So Motor Pump will ON & Irrigation will Start."));
   Serial.println();
   Serial.println("--------------------------------------------------------------------");
   Serial.println();
  }
  //delay(30000);
}

}
```

```
String getTemperatureValue(){

   dhtObject.read(dht_apin);
   Serial.println(" Temperature(C)= ");
   Serial.println("\n");
   int temp = dhtObject.temperature;
   Serial.println(temp);
   //delay(50);
   return String(temp);

}


String getHumidityValue(){

   dhtObject.read(dht_apin);
   Serial.println(" Humidity = ");
   int humidity = dhtObject.humidity;
   Serial.println(humidity);
   Serial.print(" %");
   //delay(50);
   return String(humidity);

}
```

```
//Rain sensor function
String getRainvalue()
{
  val_analogique=analogRead(capteur_A);
  Serial.println("Rain Sensor Value = ");
  Serial.print(val_analogique);
  Serial.println("\n");
  return String(val_analogique);
  delay(100);
}

String getMoisture0()
{
   Moist0 = analogRead(sensor_pin_zero);
   Serial.println("Mositure0 = ");
   Serial.print(Moist0);
   return String(Moist0);
}

String getMoisture1()
{
    Moist1 = analogRead(sensor_pin_one);
   Serial.println("Mositure1 = ");
   Serial.print(Moist1);
   return String(Moist1);
}

String getMoisture2()
{
   Moist2 = analogRead(sensor_pin_two);
   Serial.println("Mositure2 = ");
   Serial.print(Moist2);
   return String(Moist2);
}

String getMoisture3()
{
   Moist3 = analogRead(sensor_pin_three);
   Serial.println("Mositure3 = ");
   Serial.print(Moist3);
   return String(Moist3);
}
```

```
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }

    countTimeCommand++;
  }

  if(found == true)
  {
    Serial.println("OYI");
    countTrueCommand++;
    countTimeCommand = 0;
  }

  if(found == false)
  {
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
  }

  found = false;
}
```