Project Report

On

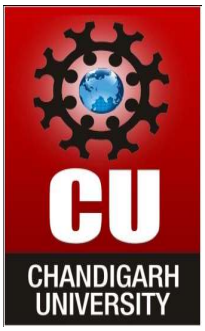
Weather Station using IOT

Submitted for the requirement of

Project

BACHELOR OF ENGINEERING

COMPUTER SCIENCE & ENGINEERING



Submitted to:

Mohit Sir

Submitted By:

NAME: Nitin Mishra

UID: 18BCS1988

NAME: DHRUV BHATTACHARJEE

UID: 18BCS2072

Mentor Signature

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY, GHARUAN

June 2021

CERTIFICATE

This is to certify that the work embodied in this Project Report entitled " " being submitted by **" 18BCS1988 " , " 18BCS2072"** , 5th Semester for partial fulfillment of the requirement for the degree of **" Bachelor of Engineering in Computer Science & Engineering "** discipline in **" Chandigarh University "** during the academic session Jan-Jun 2021 is a record of bonafide piece of work, carried out by student under my supervision and guidance in the **" Department of Computer Science & Engineering ", Chandigarh University.**

APPROVED & GUIDED BY:

DECLARATION

I, student of **Bachelor of Engineering in Computer Science & Engineering, 5th Semester, session: Jan – June 2021, Chandigarh University**, hereby declare that the work presented in this Project Report entitled " " is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Student details and Signature

Name: Nitin Mishra

UID: 18BCS1988

Name: Dhruv Bhattacharjee

UID:18BCS2072

APPROVED & GUIDED BY:

To our parents, teachers and all the well-wishers out there . .

ABSTRACT

The environmental conditions play the major effects on human beings and the weather parameters are very important roles in our daily life. Many research efforts have paid to solve the environmental problems. So the collecting of data about different parameters of the weather is necessary for planning in home and environment, and the database of weather parameters become more important for living things. In this work, two weather parameters: temperature and relative humidity have been measured by Arduino with DHT11 sensor for solving the environmental problems. The collected data from the system have been stored and transmitted to the cloud by MONGODB with JavaScript and Python programming language and then the comparison of the data collected from the sensor and the Mongo database on cloud has been made for the accuracy of the project. This sensor plays a major role wherever not every human can be all day. Solving the problem of weather condition and solving the problem before is great help for everyone. Weather Station can be made and installed anywhere around the world like in home school or anywhere and every data can be mapped daily and can also alert and predict conditions with AI and ML algorithms. This Weather Station is made in very low cost and effective help to analyse all data stored in MongoDB.

TABLE OF CONTENT

ACKNOWLEDMENTS

I would like to earnestly acknowledge the sincere efforts and valuable time given by Mentor and Project teacher. Their valuable guidance and feedback has helped me in completing this project

Also, I would like to mention the support system and consideration of my parents who have always been there in my life.

I am Thankful to my Project partner who helped me through the project without him. I could never had completed this task.

I am thankful to all my friends who helped me to understand and complete the project.

GLOSSARY

IDE - An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program.

List of Figures

1. Introduction

Weather condition plays an important role in our daily life as weather and climate are the most ubiquitous factors for home and environment planning. Moreover, the tremendous development of Internet nowadays made possible to monitor weather conditions and collect the respective data in-situ. All the objects, sensors and devices can be linked through Internet to share and analyze the data collected at various locations. Most Internet applications focus on providing information for human beings. IoT is Internet of Things, known as M2M between smart devices that collect data, relay information to the others, process the information collaboratively, and take action automatically. IoT can be much more extensive in predicting and knowing the weather conditions in particular place by connecting the weather station to the Internet. IoT is a system consists of things attached with sensors, connected to the internet via wire or wireless network. In this paper, the attempt to support the solving the environmental problems and the real time weather conditions have been made to get the smart system for human beings and, living things or non-living things. DHT22 is a digital sensor with built-in analog-to-digital converter. It consists of both temperature sensor with negative temperature coefficient and humidity sensor. So, it has been used to detect the temperature and relative humidity of the desired environment. Raspberry Pi 4 reads the output from DHT22 first. And then the data collected are continuously stored in SD card of the Pi-board. By using MONGODB, these data are transmitted to the cloud to make the sharing data with the others, and the predicting the weather condition of specific area

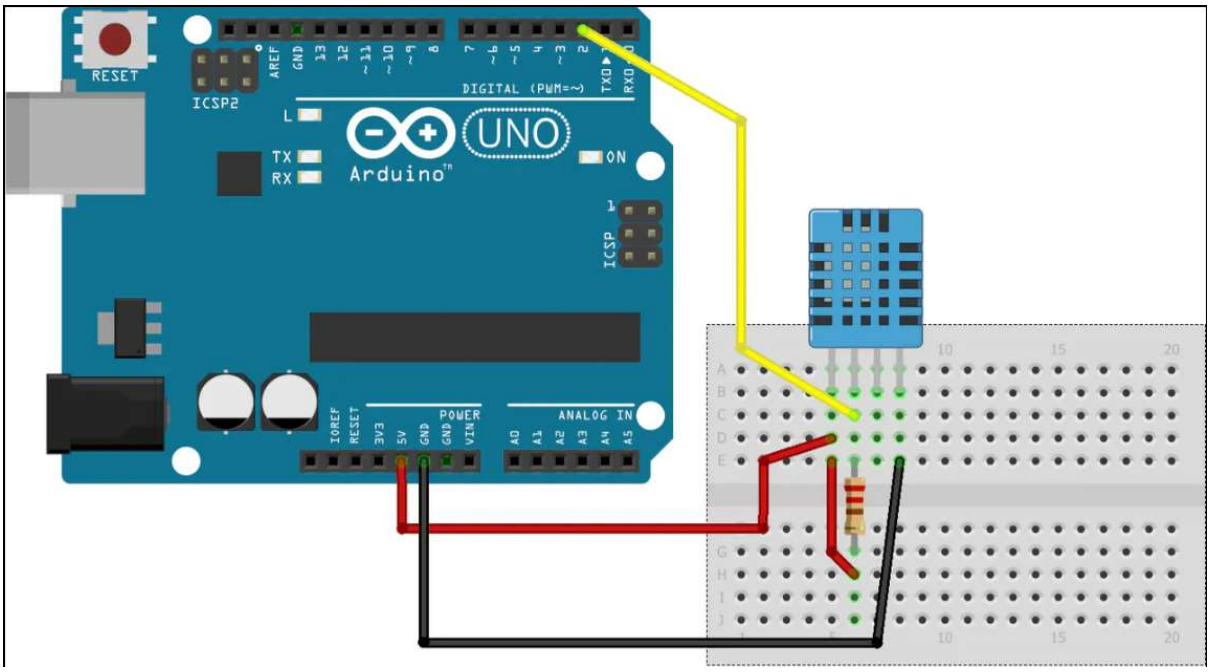


Figure 1.1 DHT11 and Arduino

Working Principle

The device works by taking readings from various sensors at different pins in Arduino microcontroller. For this purpose we've used an Arduino compatible Wi-Fi shield stacked upon our Arduino microcontroller which adds up extra functionality to our Arduino board. It increases the scope of this project. The various sensors are attached to the microcontroller each of them taking 5V input from Arduino except one pressure sensor requiring 3.3V using a 3.3V pin out from the board. All the sensors are connected using a breadboard. For temperature sensor to prevent any damage or unstable behaviour a 10kΩ resistor is attached in parallel to the temperature sensor on the breadboard. We've used DHT11 temperature sensor to get the temperature and humidity readings connected to digital pin 7 on board for input signals. It gives us continuous reading of surrounding environment in the range of two to three seconds. A raindrop sensor module is also attached from analog pin on Arduino to take input signals from the sensor. The sensor detects either there is any rain or not in terms of values. The raindrop sensor module comes with a potentiometer attached to it. For simulation purpose we can check it by putting some water droplets on the board and we can see the readings fluctuating.

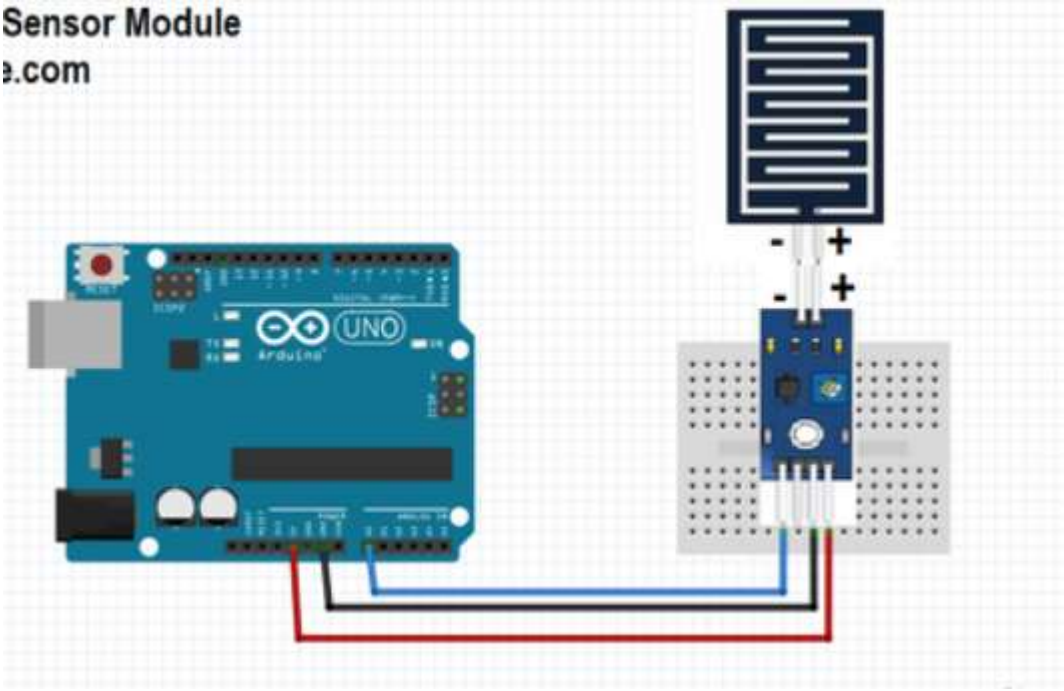


Figure 1.2 Raindrop sensor

2. Background

This section will cover all the information related to sensors and it's working. This section will also provide information related to working of the project

2.1 Weather Station

In this project we are putting together a Weather Station using a variety of sensors. The platform being used is an Arduino board and DHT11, Raindrop Sensor, ESP8266, Servo Motor. A complete set of idea for developing project in a way which can be used in house or commercially. The work of the project is to detect and analyse the weather condition and close the door/windows of the house during rain.

2.1.1 Sensors

This section will provide brief description about the sensors used in the project

i. DHT 11

This is Temperature and Humidity Sensor used to detect the temperature and humidity in specific area where installed. The DHT11 sensor can either be purchased as a sensor or as a module. Either way, the performance of the sensor is same. The sensor will come as a 4-pin package out of which only three pins will be used whereas the module will come with three pins. The only difference between the sensor and module is that the module will have a filtering capacitor and pull-up resistor inbuilt, and for the sensor, you have to use them externally if required. The DHT11 is a commonly used Temperature and humidity sensor**.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers. The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

Applications:

- Measure temperature and humidity
- Local Weather station
- Automatic climate control
- Environment monitoring

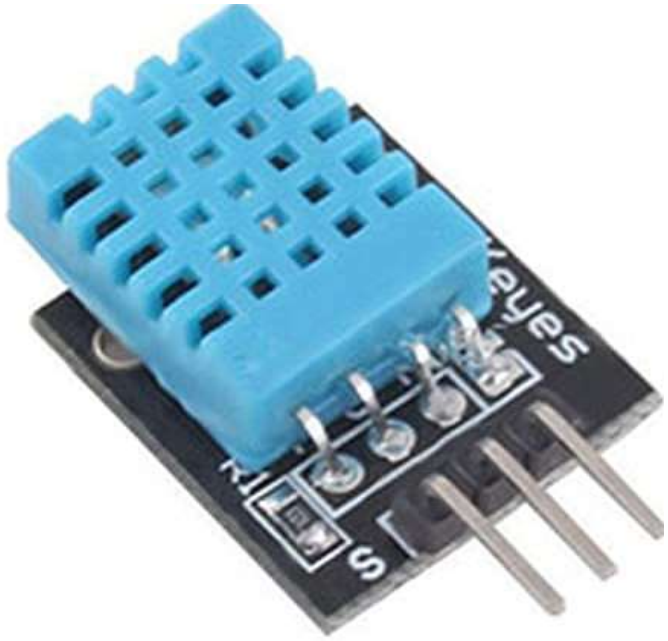


Figure 1.3 Temperature and Humidity Sensor

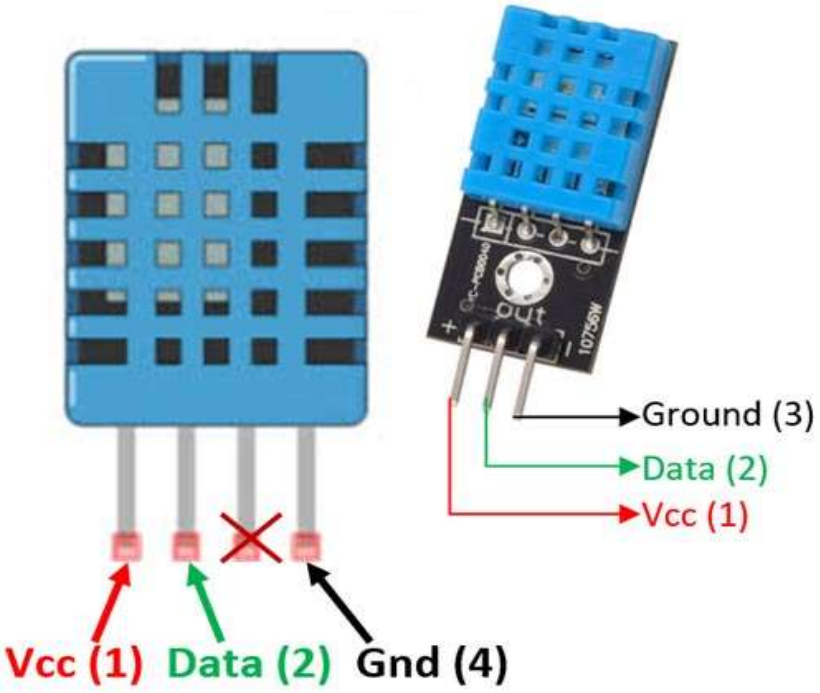


Figure 1.4 Temperature and Humidity Sensor Architecture

i. Raindrop Sensor

Raindrop Sensor is a tool used for sensing rain. It consists of two modules, a rain board that detects the rain and a control module, which compares the analog value, and converts it to a digital value. The raindrop sensors can be used in the automobile sector to control the windshield wipers automatically, in the agriculture sector to sense rain and it is also used in home automation systems.

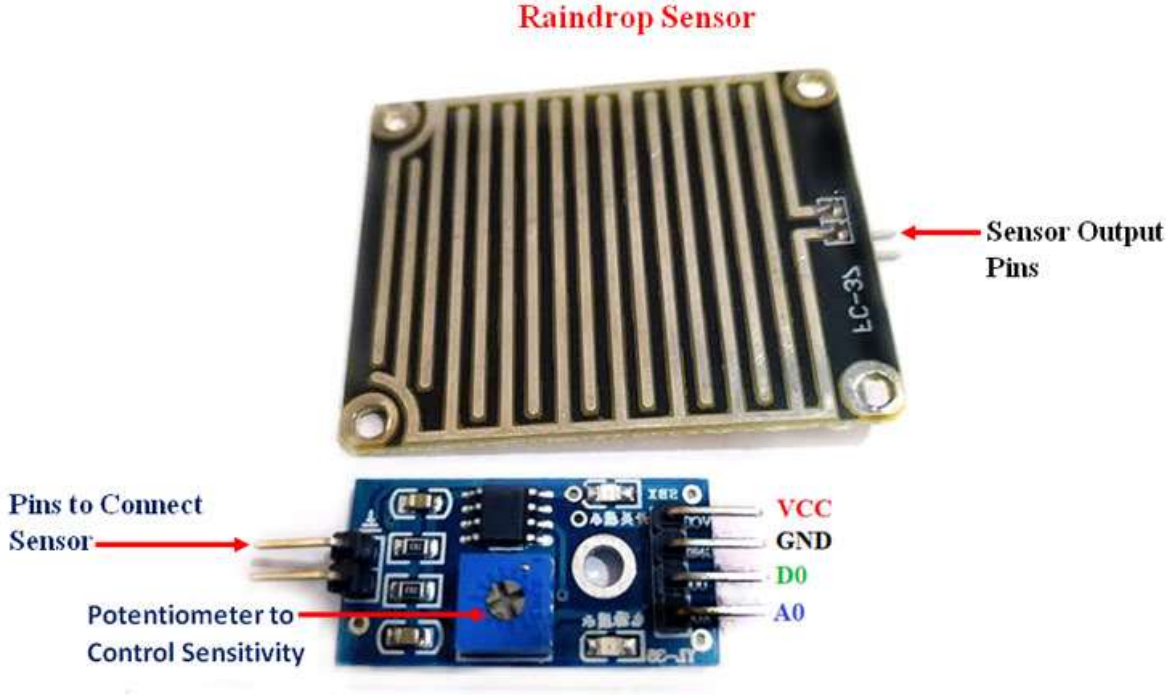


Figure 1.5 Raindrop Sensor

Interfacing the raindrop sensor with a microcontroller like 8051, Arduino, or PIC is simple. The rain board module is connected with the control module of the raindrop sensor. The control module of the raindrop sensor has 4 outputs. VCC is connected to a 5V supply. The GND pin of the module is connected to the ground. The D0 pin is connected to the digital pin of the microcontroller for digital output or the analog pin can be used. To use the analog output, the A0 pin can be connected to the ADC pin of a microcontroller. In the case of Arduino, it has 6 ADC pins, so we can use any of the 6 pins directly without using an ADC converter. The sensor module consists of a potentiometer, LN393 comparator, LEDs, capacitors and resistors. The pinout image above shows the components of the control module. The rainboard module consists of copper tracks, which act as a variable resistor. Its resistance varies with respect to the wetness on the rainboard.

ii. ESP8266

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espress if Systems in Shanghai, China.
The chip first came to the attention of Western makers in August 2014 with the ESP-01 module, made by a third-party manufacturer Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first, there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, the chip, and the software on it, as well as to translate the Chinese documentation.

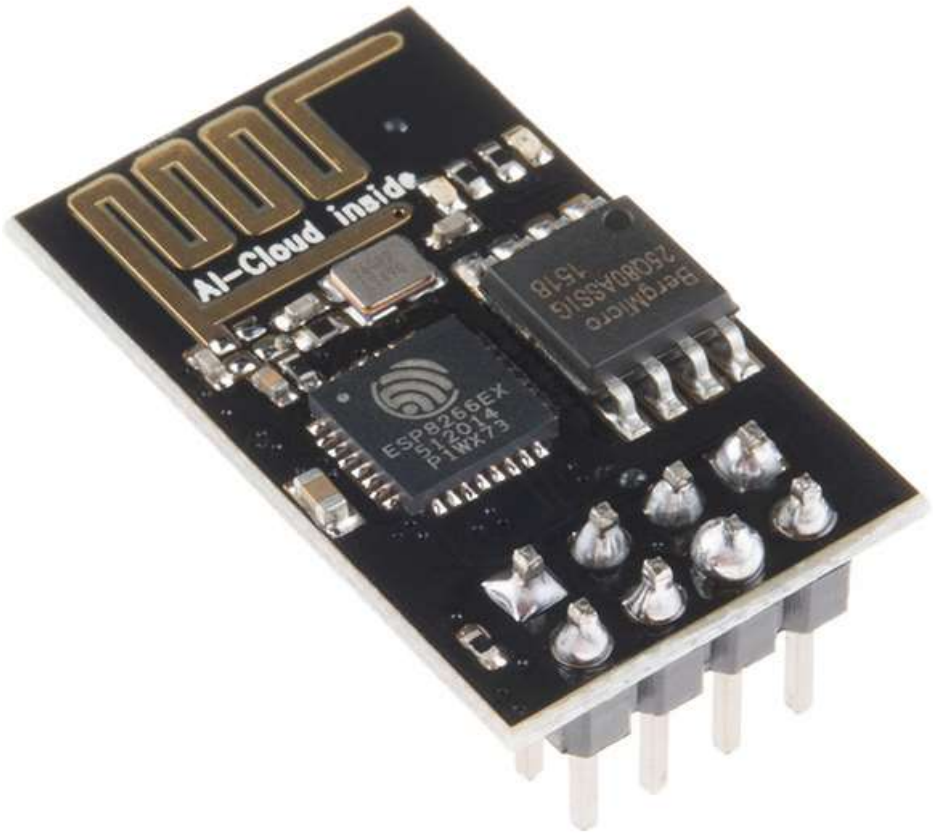


Figure 1.5 Raindrop Sensor

The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing the building of single-chip devices capable of connecting to Wi-Fi.

2.1.2 Arduino

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the "Arduino language". In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool (arduino-cli) developed in Go.

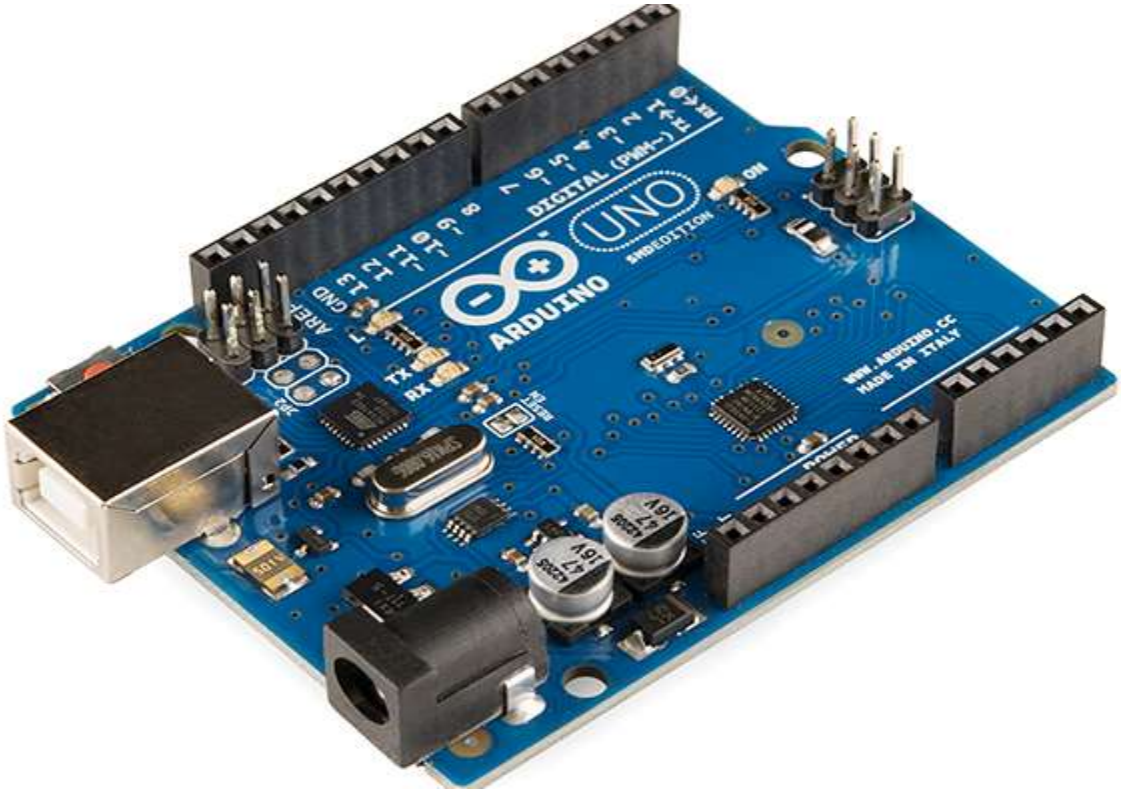


Figure 1.6 Arduino

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

Hardware

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.
Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in Arduino
Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012.The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad,run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

2.2 Motivation

The Motivation for the project is that we face many issues during Rainy season in our farm as all our vegetables and everything produced needs to take care, with this project farmer can see the condition anywhere around home or when they are not present in their farm. This project is designed to find weather condition and open or close window as required in rainfall or hot and cool weather.

2.3 Specifications and Goals

The goal of our project is to design, implement, and test the project in a way that it can predict the specific area condition and make aware and also take action automatically.

- Door and windows closed in Rainfall and in cold weather
- Door closed in night if left open
- Summarize the data and predict outcome
- Notification on Mobile devices for the action taken and the current condition

2.4 Summary

The aim of this project was to build Advanced Weather Station that can be used to take action itself. This project was a great learning opportunity for us to apply our engineering knowledge. The thesis addresses the difficulties our group faced throughout the project. It also open up the possibilities to broaden the understanding and applications of IOT, Database, Server-Client connection.

3. System Design

This section explains in details how the project was approached. All hardware and software components that will be used will be explained in this section of the report. Problems encountered and solutions to these problems will not be mentioned in this section of the report.

3.1 Project Overview

The ultimate goal of this project is to develop the weather station and build it in perfect way for longer use. This project will be used in many purpose in every aspect. The Internet connection source will be required for working of the project 24/7. Due to the time constraints of our project, it was not possible to design and build a weather station from scratch. Therefore, it was decided that our group would purchase an existing Arduino and sensors kit and interface the necessary components to detect and log data.

This project was split into four main stages; the research stage, the building and interfacing stage, the tuning and calibration stage, and the programming stage. Each stage will be explained more thoroughly in the following sections of this document.

3.1.1 The Research Stage

The research stage was a critical stage that provided our team with the knowledge necessary to complete the other stages of our project. This stage was an ongoing process that our team had to return to many times during the development process to gain the knowledge needed to continue on with the project. Our research encompassed a wide range of sources, which included studies done at different universities and hobby enthusiast sources. Our research included the working of Arduino, NoSQL Database ,Server-Client working process ,use of network, Different sensors and devices and their uses.

3.1.2 Building Stage

This stage started when the ordered parts started arriving. During this stage we focused on verifying and testing each component thoroughly. The testing process will be explained in greater detail in a later section in this document. After each part was verified to be working correctly, we combined the components together. The frame and control board were assembled, the motors and sensors were mounted, and the Wifi module and the working of the weather Station was implemented.

3.1.4 The Programming Stage

In this stage of our project we were to design Arduino design for sensors for our Weather Station. The preliminary designs for these commands have been drawn out and will be explained in greater detail in a later section of this document. Unfortunately we did not get very far in this stage and programming for these commands has yet to be started.

3.3 Physical Components

This subsection describes all the physical components of our project and how they work.

Arduino, ESP8266(Wi-fi Module),DHT11(Temperature and Humidity Sensor),Servo Motor, Raindrop Sensor, Jumper Wires,Breadboard.

3.3.1 DHT11

The DHT11 is a basic, ultra 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). Its fairly simple to use, but requires careful timing to grab data. You can get new data from it once every 2 seconds, so when using the library from Adafruit, sensor readings can be up to 2 seconds old.

Comes with a 4.7K or 10K resistor, which you will want to use as a pullup from the data pin to VCC.

Specifications:

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50 °C temperature readings +-2 °C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing
- Adafruit Learning Documentation for DHTxx Sensors
- RoHS compliant

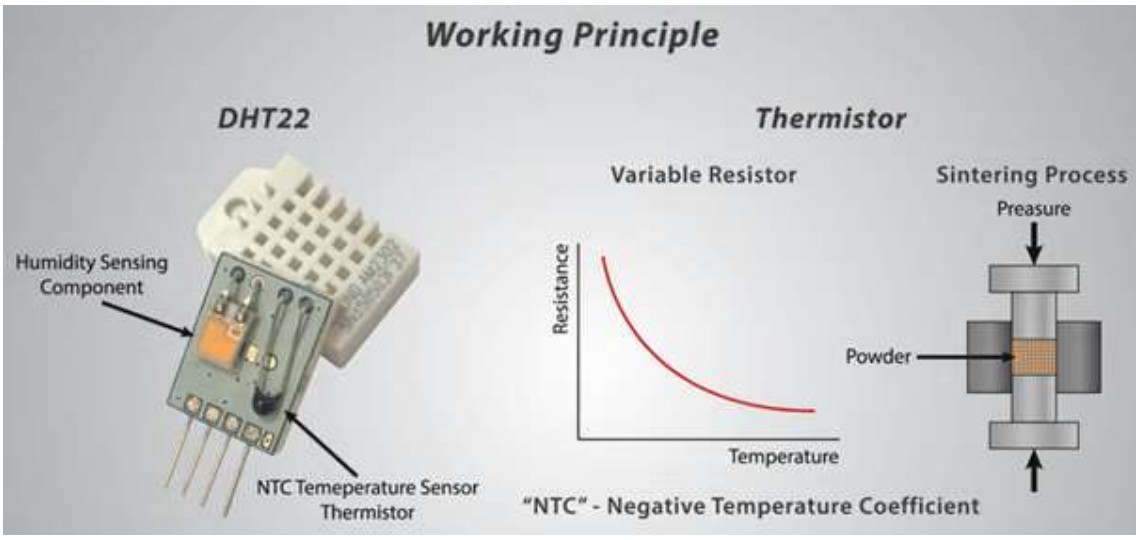


Figure 1.7 Working principle of DHT11

The percentage of water present in the air is termed as humidity. Water as gaseous state called vapor. As the temperature of the air increases more water vapor can be generate.

Humidity measurement in industries is critical because it may affect the business cost of the product and the health and safety of the personnel. So, its huge importance of humidity sensor is very important, especially in the control systems for industrial processes like chemical gas purification, dryers, ovens, film desiccation, paper and textile production, and food processing. In agriculture, measurement of humidity is important for plantation protection (green house), soil moisture monitoring, etc.

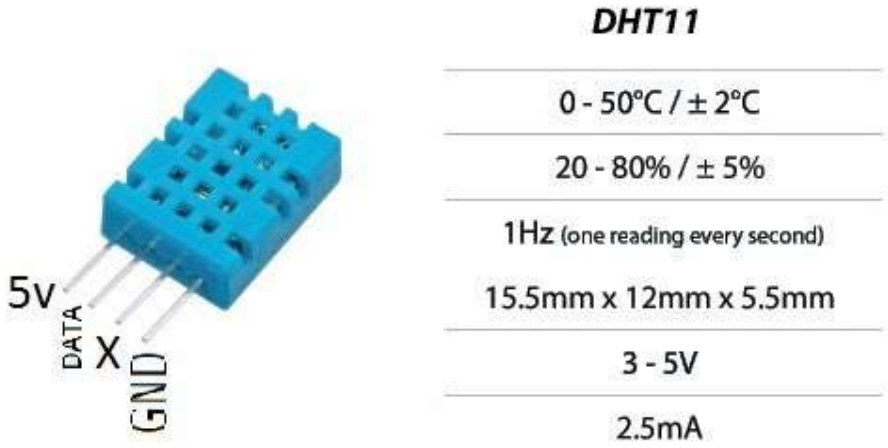


Figure 1.8 Architecture

Types of humidity:

1) Relative Humidity = (density of water vapor / density of water vapor at saturation) x 100%

2) Absolute=Mass(vapour) / volume. Unit-grams/m3

3) Specific:Mass(vapour) / total mass.

4) Dew Point:Temperature(above 0°C) at which the water vapor in a gas condenses to liquid water)

5) Frost POINT: Temperature(below 0°C) at which the water vapor in a gas condenses to ice

3.3.2 Raindrop Sensor

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance.

Rain Sensor module allows to measure moisture via analog output pins and it provides a digital output when a threshold of moisture exceeds.

The module is based on the LM393 op amp. It includes the electronics module and a printed circuit board that “collects” the rain drops. As rain drops are collected on the circuit board, they create paths of parallel resistance that are measured via the op amp.

Figure 1.9 Raindrop sensor

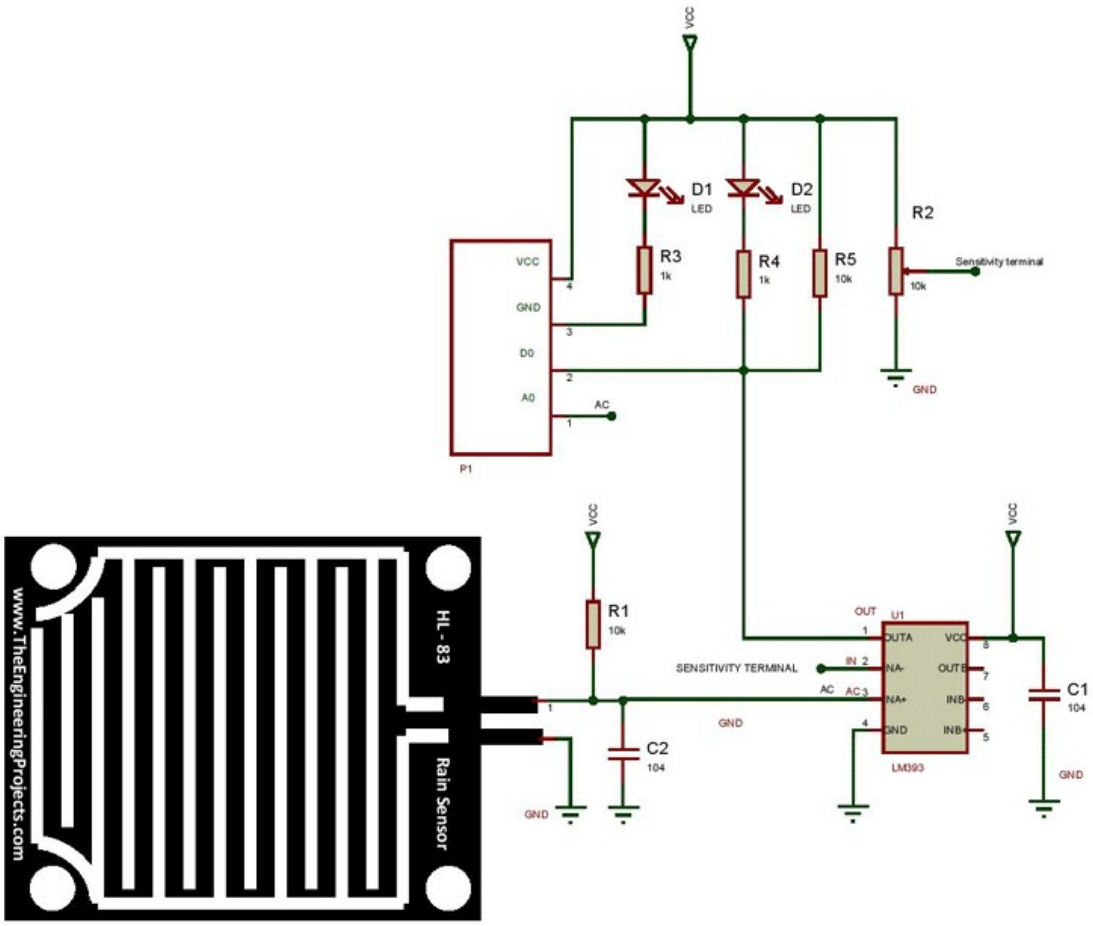


Figure 2.0 Raindrop Architecture

The sensor is a resistive dipole that shows less resistance when wet and more resistance when dry. When there is no rain drop on board it increases the Resistance so we gets high voltage according to V=IR.

When rain drop present it reduces the resistance because water is a conductor of electricity and presence of water connects nickel lines in parallel so reduces resistance and reduces voltage drop across it.

3.3.3 ESP8266

The ESP8266 can be controlled from your local Wi-Fi network or from the internet (after port forwarding). The ESP-01 module has GPIO pins that can be programmed to turn an LED or a relay ON/OFF through the internet. The module can be programmed using an Arduino/USB-to-TTL converter through the serial pins (RX,TX).

.

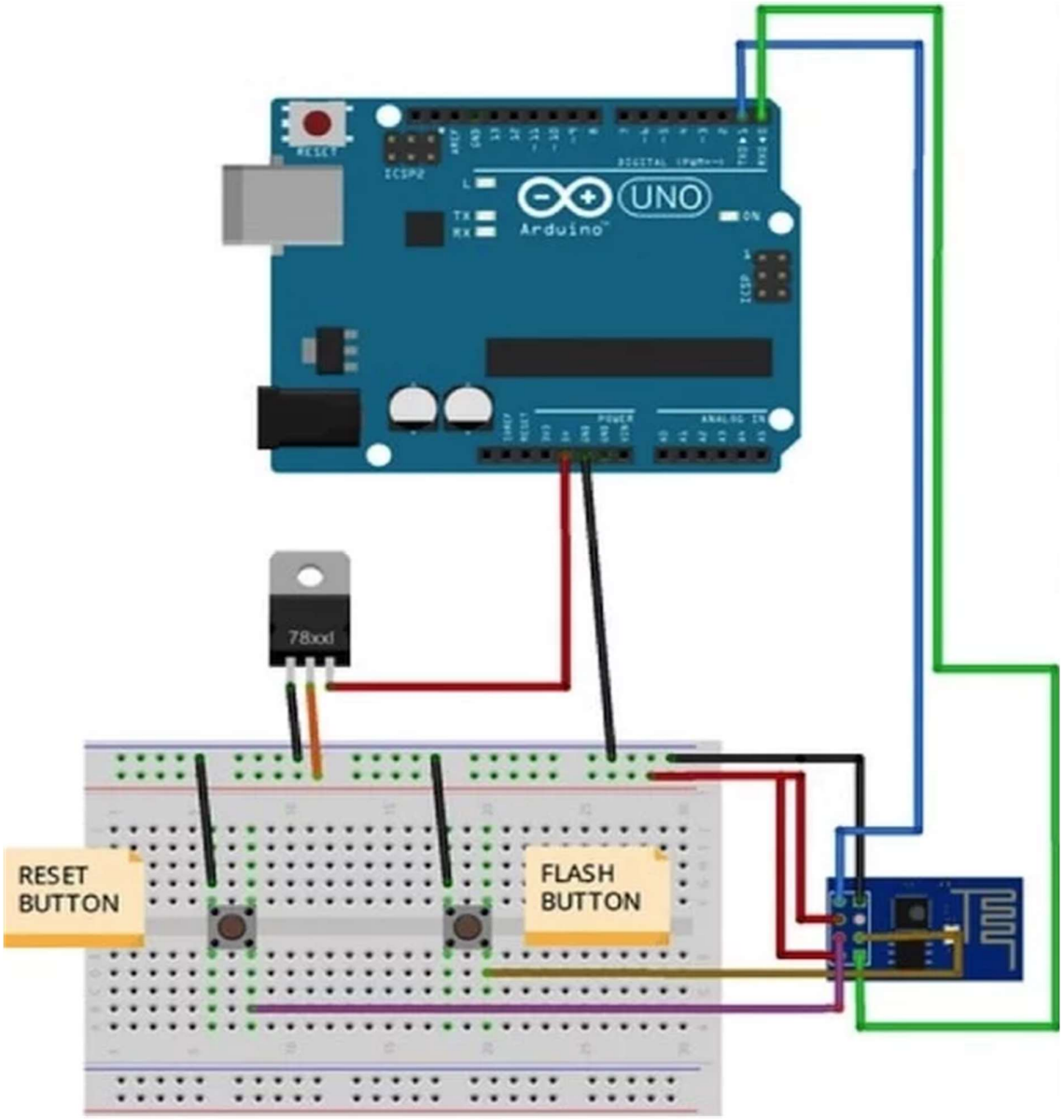


Figure 2.1 Wifi module

Using Arduino Uno to Flash the Code to the ESP8266

You can use the Arduino UNO to flash the code to ESP8266 ESP-01. While uploading the code, follow the same procedure to keep the flash button pressed while you click once on reset and release the flash button

3.3.4 Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Figure 2.2 Servo Motor

Working of Motor

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.
Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.
Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

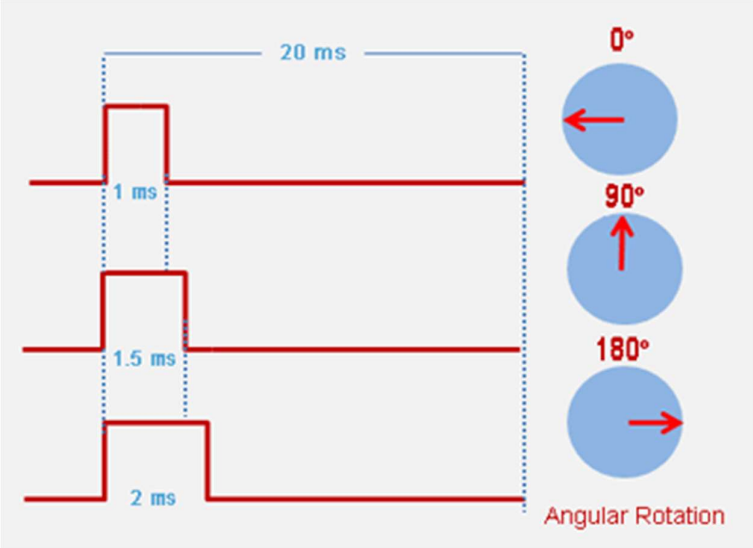


Figure 2.3 Servo motor working

3.4 Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

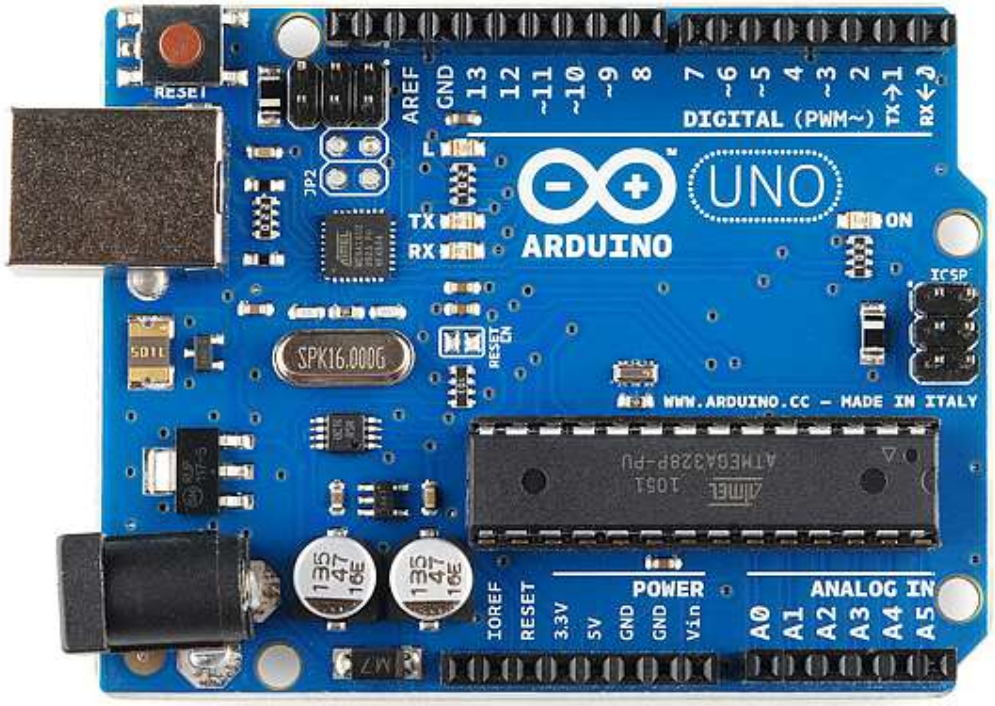


Figure 2.4 Arduino

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

3.5 Software

This subsection describes all the software components of our project and how they work

3.5.1 Windows OS

This Project is designed in Windows 10 OS version 20H2 with all the dependent libraries and working structure.

Windows 10 is a series of operating systems developed by Microsoft and released as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, released nearly two years earlier, and was released to manufacturing on July 15, 2015, and broadly released for the general public on July 29, 2015.Windows 10 was made available for download via MSDN and Technet, and as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10, which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

Windows 10 received mostly positive reviews upon its original release. Critics praised Microsoft's decision to provide a desktop-oriented interface in line with previous versions of Windows, contrasting the tablet-oriented approach of Windows 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch-oriented interface of its predecessor. Critics also praised the improvements to Windows 10's bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of the changes to operating system behaviours, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners, and adware-esque tactics used to promote the operating system on its release.

3.5.2 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++.It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

Arduino IDE is a derivative of the Processing IDE, however as of version 2.0, the Processing IDE will be replaced with the Visual Studio Code-based Eclipse Theia IDE framework.

3.5.3 NodeJs and React framework

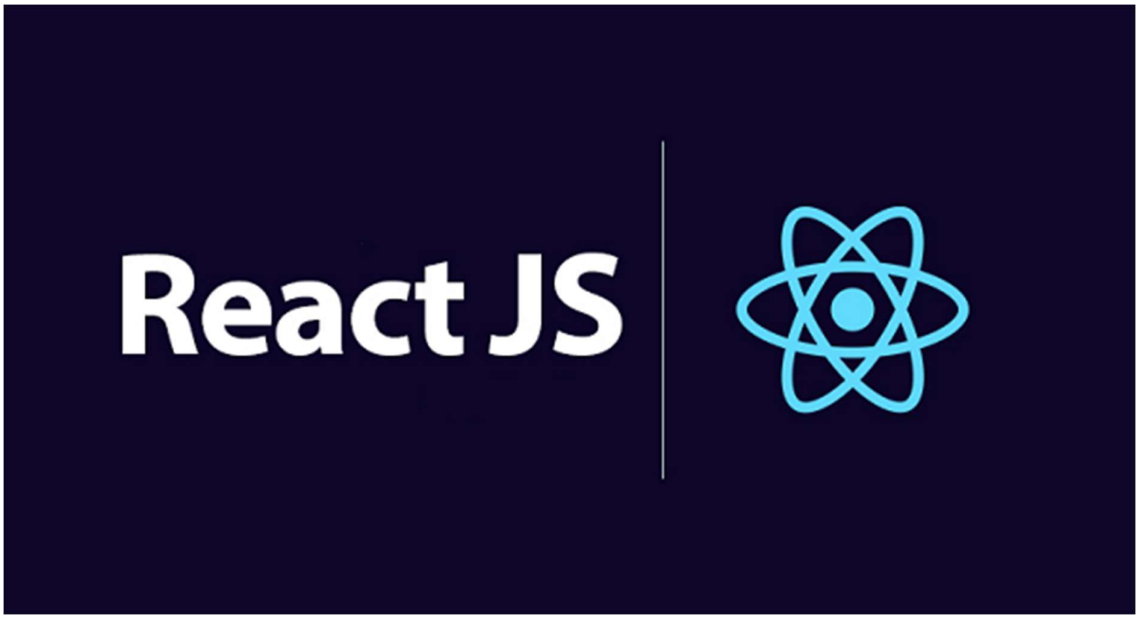
As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like EventMachine::run(). In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behaviour is like browser JavaScript — the event loop is hidden from the user.
HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

React Framework

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript libraryfor building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies.React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.



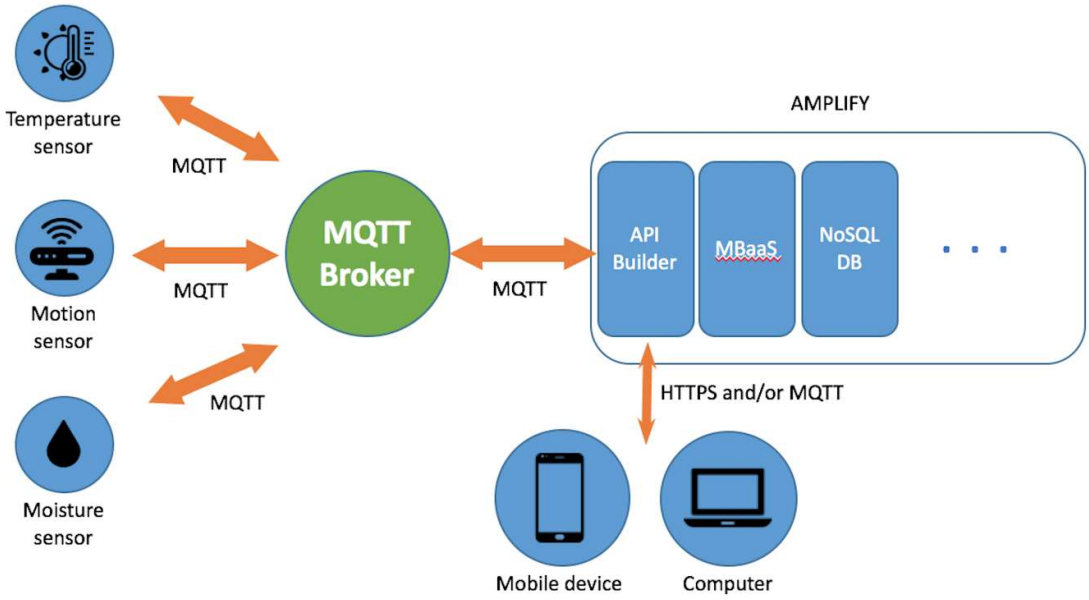
React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application.

3.5.4 Python Paho MQTT & Mosquitto Server and Client

This code provides a client class which enable applications to connect to an MQTT broker to publish messages, and to subscribe to topics and receive published messages. It also provides

some helper functions to make publishing one off messages to an MQTT server very straightforward. It supports Python 2.7.9+ or 3.5+.

The MQTT protocol is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. Designed as an extremely lightweight publish/subscribe messaging transport, it is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.



Mosquitto

Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.

The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers.

The Mosquitto project also provides a C library for implementing MQTT clients, and the very popular mosquitto_pub and mosquitto_sub command line MQTT clients.

3.6 Summary

In this section the hardware and software components were explained in detail. The reasons for choosing each component was made clear.

4.Experimental Results and Discussion

In this section of the document we will be discussing the verification and testing of each hard-ware and software component. All problems will be described in detail and the solutions we made to solve these problems. In this section we will also discuss our overall results of the project and what we could have done to improve upon our project. Future work for this project will also be mentioned in this section of the document.

4.1 System Setup

This stage will let's us know how we did all the setup of our project

Sensors and Arduino

Every sensors with Arduino were connected and was ready to programmed with taking care of connections, Firstly we checked with individual sensors and their working
DHT11,Raindrop Sensors were made to work together and later the Servo motor was designed to work on the output of the raindrop sensor.

```
//For Door(Servo Motor)
if(rainReading<100){
 myservo.attach(SERVO_PIN);
 myservo.write(0);
 Serial.println("Door Closed");
 delay(2000);
 myservo.detach();
}
if(myservo.read()==0 & rainReading>700){
 myservo.attach(SERVO_PIN);
 myservo.write(120);
 Serial.println("Door Open");
 delay(2000);
 myservo.detach();
}
```

Figure 2.5 Code work for Raindrop Sensor

```
if(!mqttClient.connected()){
  reconnect_mqtt();
}
delay(3000);
float h = dht.readHumidity();
float t = dht.readTemperature();
int rainReading = analogRead(A0);
int range = map(rainReading, rainSensorMin, rainSensorMax, 0, 2);
```

Figure 2.6 Code for DHT11 and Raindrop Sensor

As the working of Sensors was working fine we designed overall project with connection with Arduino and this worked well with some changes and every connection was made.

Now the data was to send through mosquitto server for that Wifi module was connected to the Arduino and data of the Sensors were send through the Wifi Module to the server.

```
void loop() {
  status = WiFi.status();
  if ( status != WL_CONNECTED) {
    while ( status != WL_CONNECTED) {
      Serial.print("Attempting to connect to WPA SSID: ");
      Serial.println(WIFI_AP);
      // Connect to WPA/WPA2 network
      status = WiFi.begin(WIFI_AP, WIFI_PASSWORD);
      delay(500);
    }
    Serial.println("Connected to AP");
  }
  if(!mqttClient.connected()){
    reconnect_mqtt();
  }
```

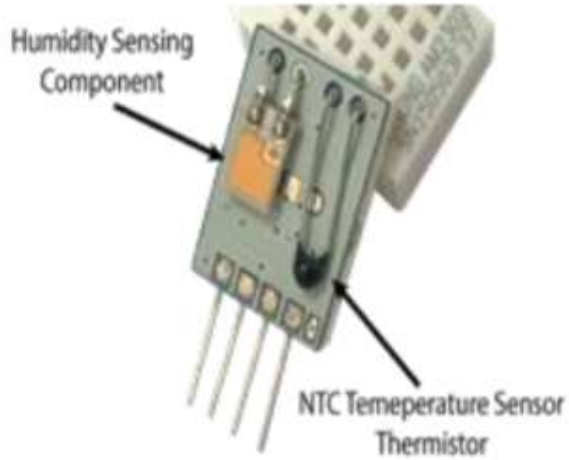
Figure 2.7 Wifi connection

We need to save the data to use it for our purpose for that we decided to send the data to Mongodb

4.2 EXPERIMENTAL DETAIL

The proposed system consists of (i) temperature sensor, (ii) humidity sensor, (iii) data input from hardware sensors and (iv) data output to the cloud for sharing with the others.

In this proposed weather data collected system, DHT11 has been used for temperature and humidity of the environment. It is a digital sensor with an inbuilt analog-todigital converter (ADC) and the data can be transmitted through wire up to 20 m away from Raspberry pi. It consists of a humidity sensing component, a NTC (negative temperature coefficient) temperature sensor and an IC on the back side of the sensor. It is necessary to put on 10 kΩ resistor between pin-1 and pin-2 of DHT11.



The weather data such as temperature and humidity, have been collected by using DHT11 with the aid of Arduino by Java Script. The first part creates the database (“test”) and the collection (“dht”) which apply the connection to the cloud. For the data sorting on CLOUD, the first creates the variable “insert” for temperature and humidity. When the term “data uploaded” appears, the data value can be inserted, and it is necessary to create ID on cloud. The duration

time between one-data and another is 10s.

```
21:05:16.826 -> {"Temperature":35.5,"Humidity":54}
21:05:17.065 -> Success sending message
21:05:20.341 -> {"Temperature":35.6,"Humidity":54}
21:05:20.853 -> Success sending message
21:05:24.129 -> {"Temperature":35.2,"Humidity":54}
21:05:24.368 -> Success sending message
21:05:27.747 -> {"Temperature":35.2,"Humidity":54}
21:05:27.951 -> Success sending message
21:05:31.241 -> {"Temperature":35.4,"Humidity":54}
21:05:31.479 -> Success sending message
21:05:34.755 -> {"Temperature":35.4,"Humidity":54}
21:05:34.993 -> Success sending message
21:05:38.265 -> {"Temperature":35.5,"Humidity":54}
21:05:38.505 -> Success sending message
21:05:41.769 -> {"Temperature":35.5,"Humidity":54}
21:05:42.002 -> Success sending message
21:05:45.371 -> {"Temperature":35.5,"Humidity":54}
21:05:45.609 -> Success sending message
```

Figure 2.8 Output of Arduino IDE

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. It works on concept of collection and document. MongoDB is also a NoSQL type database. NoSQL is not a relational database. It provides more flexibility, since all records are not restricted by the same column names and types defined across the entire table

```
Log : Sending CONNECT (u0, p0, wr0, wq0, wf0, c1, k60) client_id=b'backend-client' properties
Log : Received CONNACK (0, Success) properties=[ReceiveMaximum : 20, TopicAliasMaximum : 10]
Log : Sending SUBSCRIBE (d0, m1) [(b'uno/weatherStats', {QoS=0, noLocal=False, retainAsPubli
tainHandling=0})]
Log : Received SUBACK
Log : Received PUBLISH (d0, q0, r0, m0), 'uno/weatherStats', properties=[], ...  (34 bytes)
```

Figure 2.9 Subscribing to MQTT client

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple database. To store the weather data, “Forecast” database must be created on cloud. Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. In this work, “FORECAST_COLLECTION t” collection has been made under “test” database.

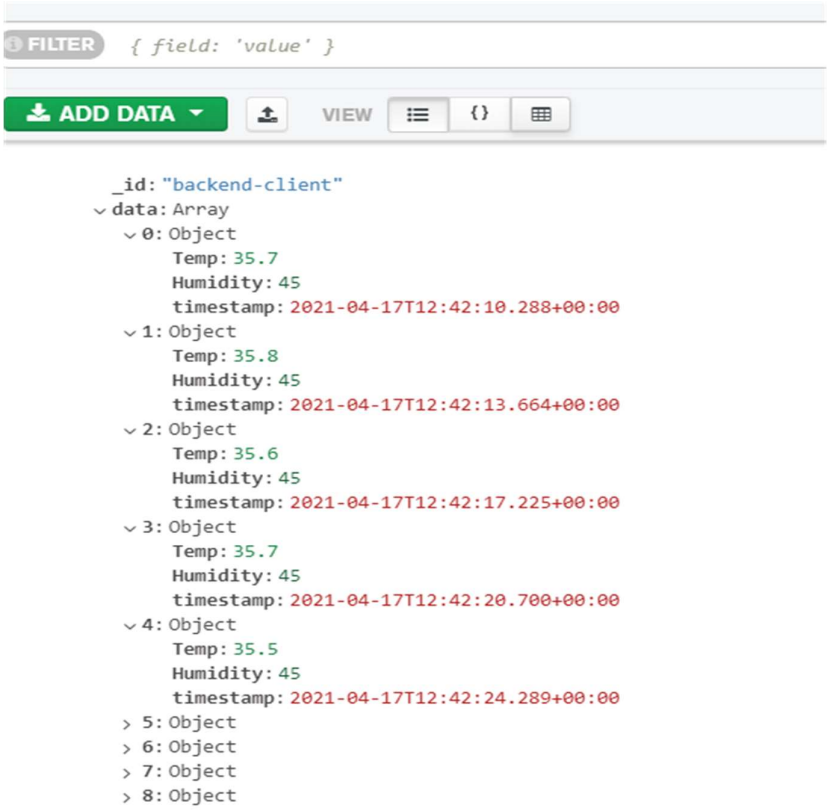


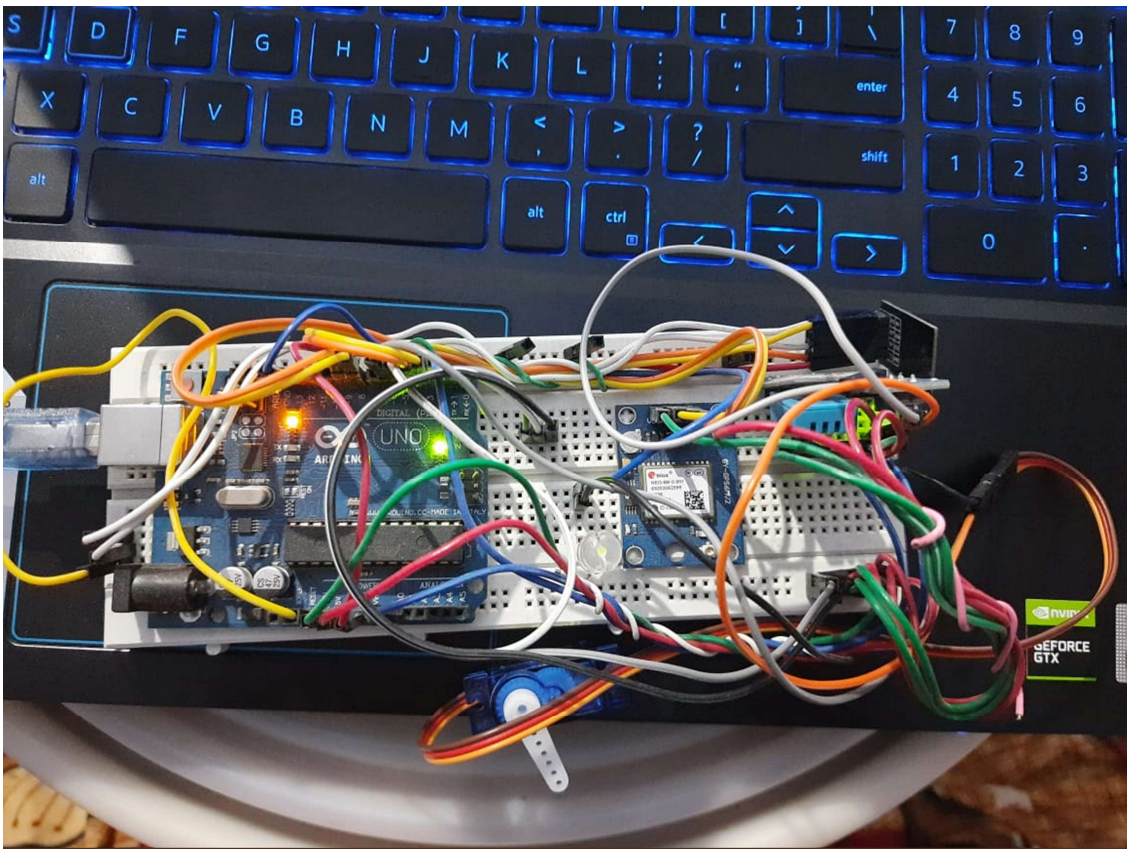
Figure 3.0 MONGODB data

In the proposed work, "FORECAST_COLLECTION" collection of "test database" has been created for temperature and humidity data . MongoDB can be used for "BIG DATA" in its collection of respective database. It is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another, and then data is stored in the form of JSON style documents.

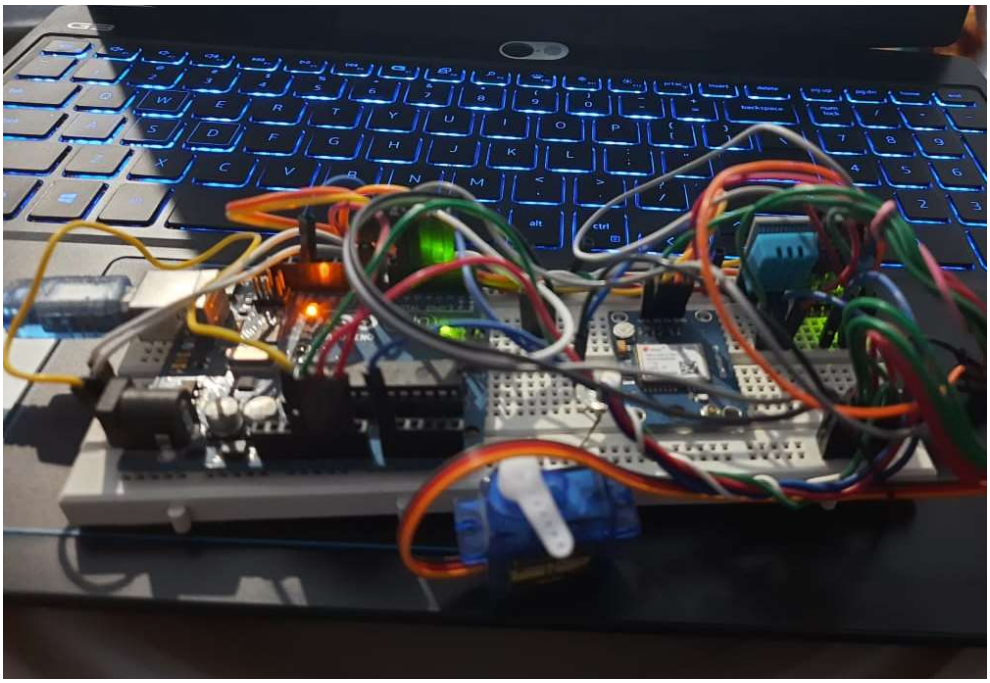
```
21:03:56.339 -> [WiFiEsp] Initializing ESP module
21:04:00.039 -> [WiFiEsp] Initilization successful - 1.5.4
21:04:00.105 -> Attempting to connect to WPA SSID:Nitin's PC
21:04:13.397 -> [WiFiEsp] Connected to Nitin's PC
21:04:13.431 -> GPS, DHT11, and, Raindrop detector started.
21:04:13.465 -> Attempting MQTT Connection
21:04:13.499 -> [WiFiEsp] Connecting to 192.168.137.1
21:04:13.670 -> MQTT connection established.
21:04:13.670 -> Publishing to: uno/weatherStats
21:04:16.777 -> {"Temperature":35.5,"Humidity":54}
21:04:16.982 -> Success sending message
21:04:20.258 -> {"Temperature":35.2,"Humidity":54}
21:04:20.497 -> Success sending message
```

Figure 3.1 Connection of Wifi and MQTT with Mosquitto

For the running program of node.js, it is necessary to make the command “node nodemongodht.js”. In this work, there are three programs such as “nodemongodht.js, “package.json” and “package-lock.json” in “projectTest” directory.



The first data output of temperature and its humidity .The second data will be produced within 10 s after the first output. The temperature and the humidity data of the desired environment for specific duration time are continuously collected by the proposed system. It is now seen that each document is printed in JSON style. The temperature and humidity data is transmitted into the cloud



4.2.1 CONTRIBUTION

The proposed system has been designed to create the low cost weather station to get the information of real time weather condition and easy to install to achieve the weather data of a specific area. And then the data can be used to share to the others by using MONGODB with node.js

4.3 Discussion

The Project was working Successfully upto prediction and as we have seen every setup was perfect and upto mark.

4.3.1 Future Work

The future design for the project is to implement ML operation of that it can calculate and analyse data itself and create and AI operation for notifying every steps so that it can work on it’s own.

4.4 Summary

Project Name : Weather Station

Brief Project description :

This Project is based on IOT weather station to find temperature and humidity and take action accordingly if it rains. This project include connection to server and also Database connection.

5. CONCLUSION

Weather prediction is a very important factor, which forecasts the climate in a region based upon the values of weather parameters. So the calculated results from this system can be used in forecasting the weather of that locality for a period of time. This research makes the understanding concepts of humidity sensor and temperature sensor of DHT11 according to the construction of them, and then how to create the MONGO database on cloud of the weather parameters of the specific area. The data from sensor are transmitted to sever where they can be viewed globally which will be easily accessible to everyone. All the weather parameters were successfully displayed via MONGO database which are accessible by both administrator and users. Because there is no concept of relationship in MongoDB, a document database, in which one collection holds different documents and it can deal with big data. The comparison between sensor data and cloud data has been made for the determination of the accuracy of mongo-node.js on private-cloud. The system can make to solve the environmental problems due to the weather condition for living-things and non-living things.

REFERENCES

[1] Adnan S, Yulong L, Mohan Z, Selim A, 2014, "Low Cost Embedded Weather Station with Intelligent System" (USA: Michigan)

[2] Aguado E & Burt J, Understanding Weather and Climate, 6th ed. Boston [Mass.]: Pearson Education, Inc., (2013).

[3] Avid Kopec, 2019, "Classic Computer Science Problems in Python" (New York: Manning Publications Company)

[4] Hamid F, Rahman W, Mohd Nizar H, "Humidity Sensors Principle, Mechanism, and Fabrication Technologies", IJIRCCE, March 2015

[5] Kodali R & Mandal S, "IoT based weather station", 2016 Interna-tional Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, India, pp. 680-683, (2016).

[6] Narasimha Karumanchi, 2016, "Data Structure and Algorithmic Thinking with Python", (Bombay: CareeMonk Publications)

[7] Peter Farrell, 2019, "Math Adventures with Python", (San Francisco: no statch Press)

[8] Ruano AE, Mestre G, Duarte H & Silva S, "A neural network based intelligent weather station", 2015 IEEE 9th International Symposi-um on Intelligent Signal Processing (WISP), Siena, Italy, pp. 1-6, (2015).

[9] Sarmad Nozad M, "Design od Weather Monitoring System Using Arduino Based Database Implementation", Journal of MultidisciplinaryEngineering Science and Technology, Vol.4, Issue 4, April-2017

[10] Satyanarayana KNV, Reddy SRN, Sai Teja PVYN & Habibuddin B, “IOT Based Smart Weather Station Using Raspberry-PI3”, Journal of Chemical and Pharmaceutical Sciences, Vol. 2016, No. 10, pp. 1-6, (2016).

[11] Stefano Z, 2018, “Wind Direction Extraction from SAR in Coastal Areas” (Italy: Padova)

[12] Yash M, Anshika M & Bhateja D, “Correlation among environmental parameters using an online smart weather station system”, India Conference (INDICON), 2015 Annual IEEE, New Delhi, India, pp. 1-6, (2015).