Project Report

On

Weather Station using IOT

Submitted for the requirement of

Project

BACHELOR OF ENGINEERING

COMPUTER SCIENCE & ENGINEERING

Submitted to:	**Submitted By:**
Charanpreet Ma'am	**NAME: Mishra Nitin Sherbahadur**
	UID: 18BCS1988
	Name: Dhruv Bhattacharjee
	UID:18BCS2072

Mentor Signature charnpreet kaur

04/29/2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY, GHARUAN

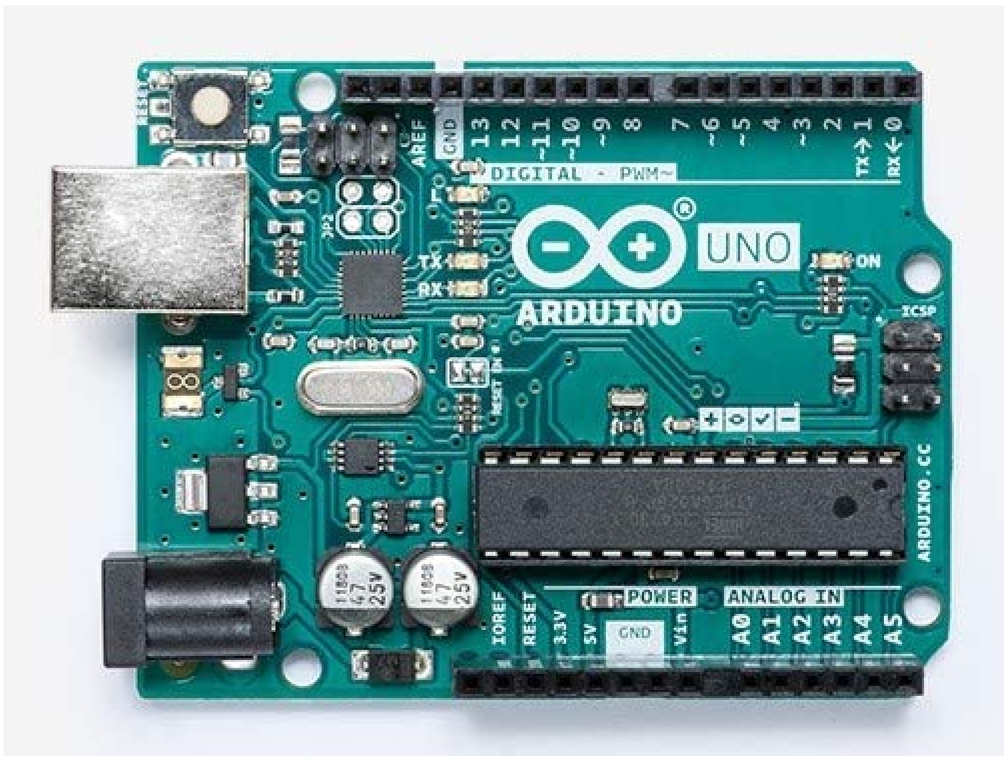
June 2021

IMPLEMENTATION:

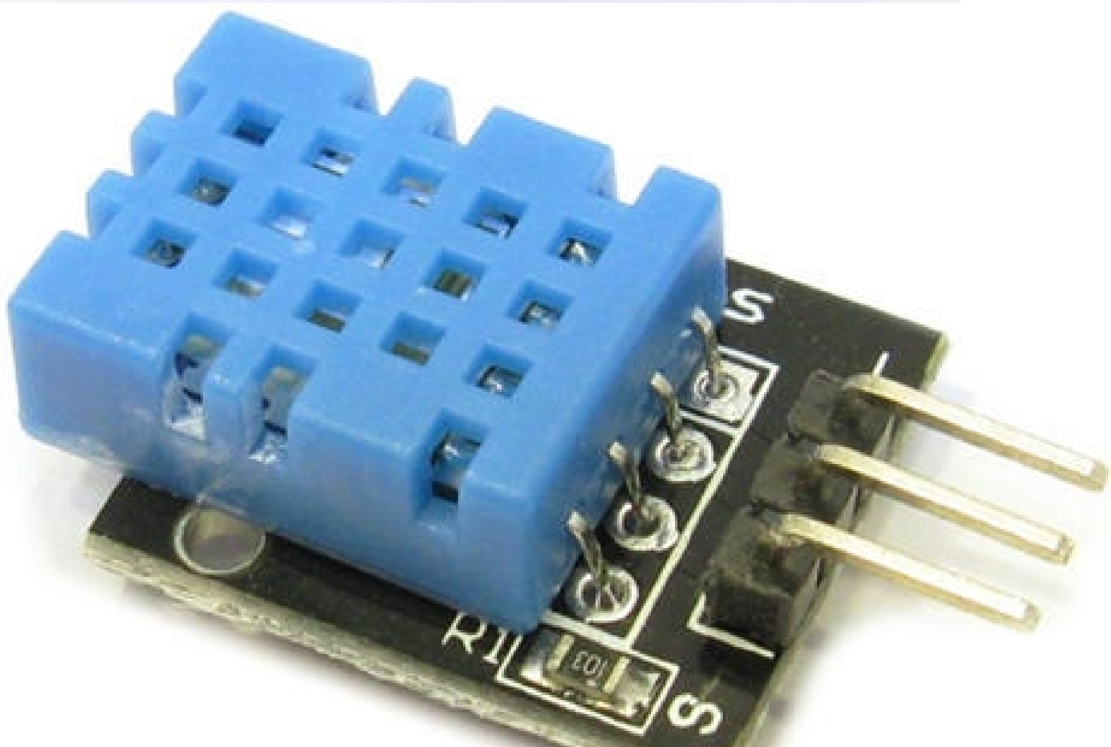
The implementation is now been achieved till 70-80% of the main project. As our project contains both software and hardware element following is list of hardware components and technologies used to implement the logic.

Hardware:

- Arduino Uno Board
- ESP8266 WIFI module
- Raindrop sensor
- DHT11 Temperature and Humidity sensor
- Servo Motor



RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL - PWM~
UNO
ARDUINO
TX
RX
ON
ICSP
ARDUINO.CC
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5



S
R1
103
S

Software technologies:

- NodeJS
- JavaScript
- MongoDB
- React
- MQTT protocol
- Mosquitto MQTT broker
- Arduino IDE
- HTML
- CSS
- Python

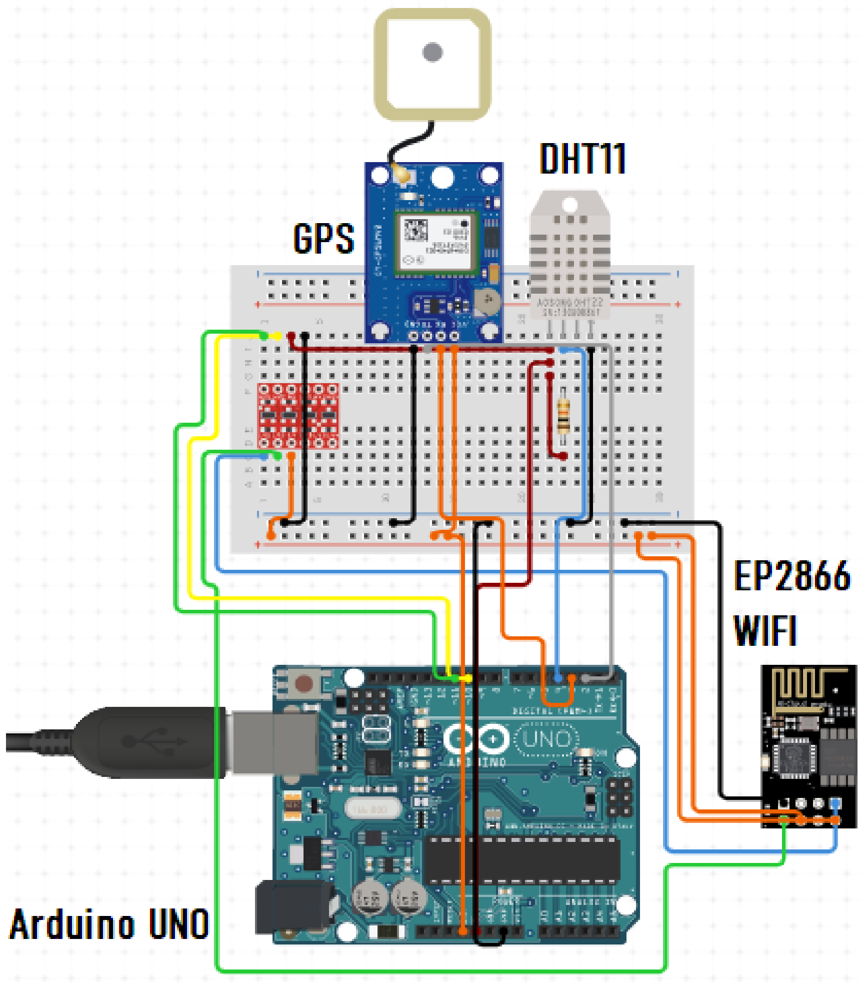
On the hardware part, the Arduino board gathers temperature, humidity, GPS coordinates, rain data from all sensors attached to it and then send it over to MQTT broker running on server. As the broker receives the NodeJS module also running on same server collects that data stores it in database which also running on the same server. The reason MongoDB is chosen is that it provides JavaScript library which makes it very easy to communicate with Database.

For the frontend we have used React, a JavaScript library created and maintained by Facebook.

Step by step process:

1. Arduino publishes the data to network through MQTT protocol.
2. The JavaScript code running on the same network subscribes to topic collects the data published by Arduino.
3. The same program which collects the data stores the data in the database by using *'mongodb'* library.
4. The collected data can now be viewed through webpage served on same network.

Hardware configuration:

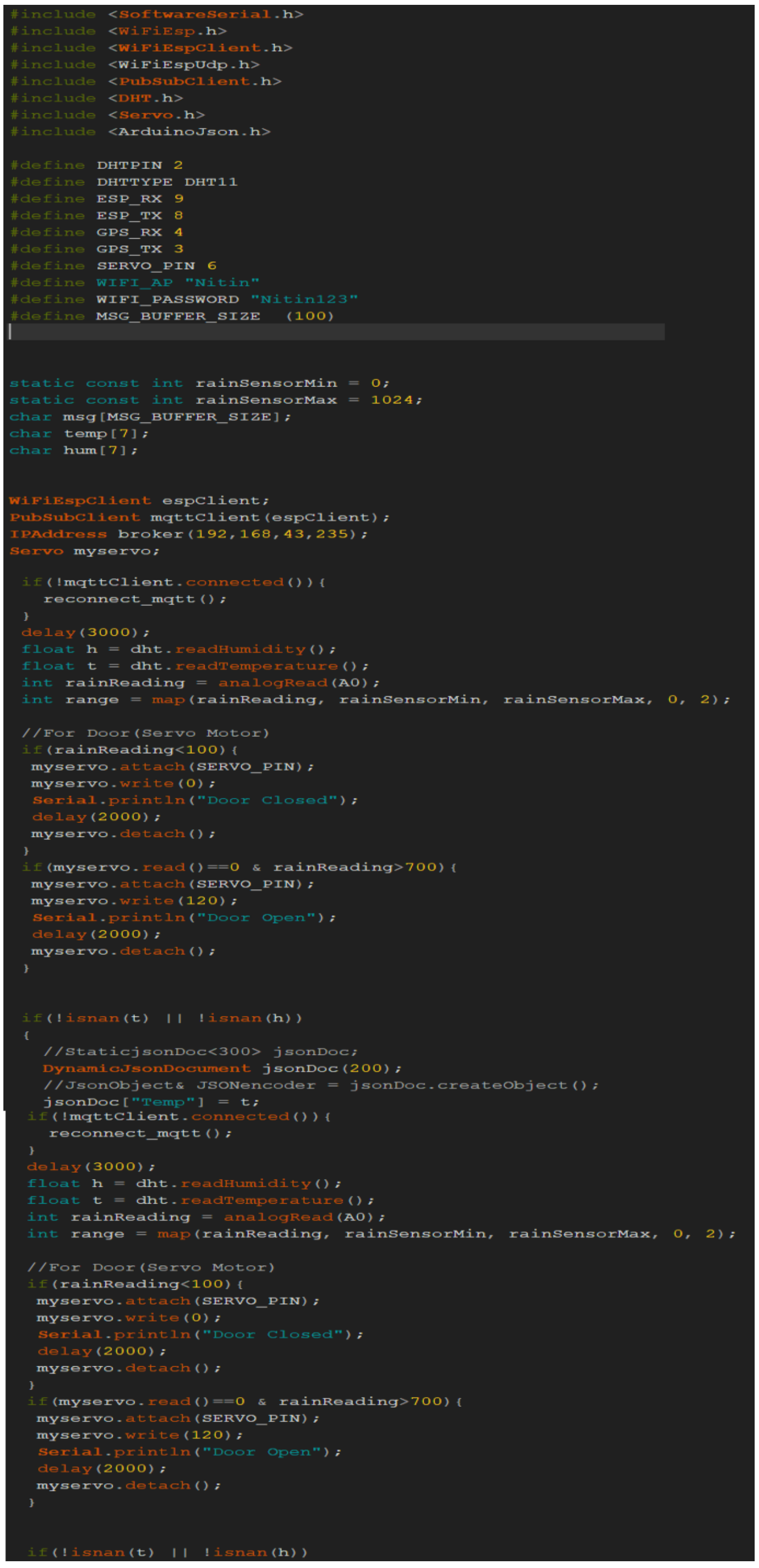


Hardware Configuration

Above is the diagram of our system. Here a Arduino Uno board is connected to different sensors and modules like DHT11, Raindrop Sensor, GPS sensor and ESP8266 WIFI module. Arduino Uno has inbuilt WIFI capabilities so we have made use of ESP8266 to connect Arduino board to network through which it will transmit the data. ESP8266 is a WIFI micro-Controller which can work in combination with Arduino uno or can also be used standalone device. The temperature and humidity are both collected by DHT11, a very reliable low-cost and accurate sensor for its purpose. The raindrop sensor is a very simple sensor which detects raindrop droplets in order to check for rain, Ublox-6 is GPS sensor which collects GPS coordinates of the place at which the device is placed. The overall power consumption of the device is very low and can run on a very simple battery for long time or can also be powered through a DC adapter.

Software Configuration:

Arduino code:

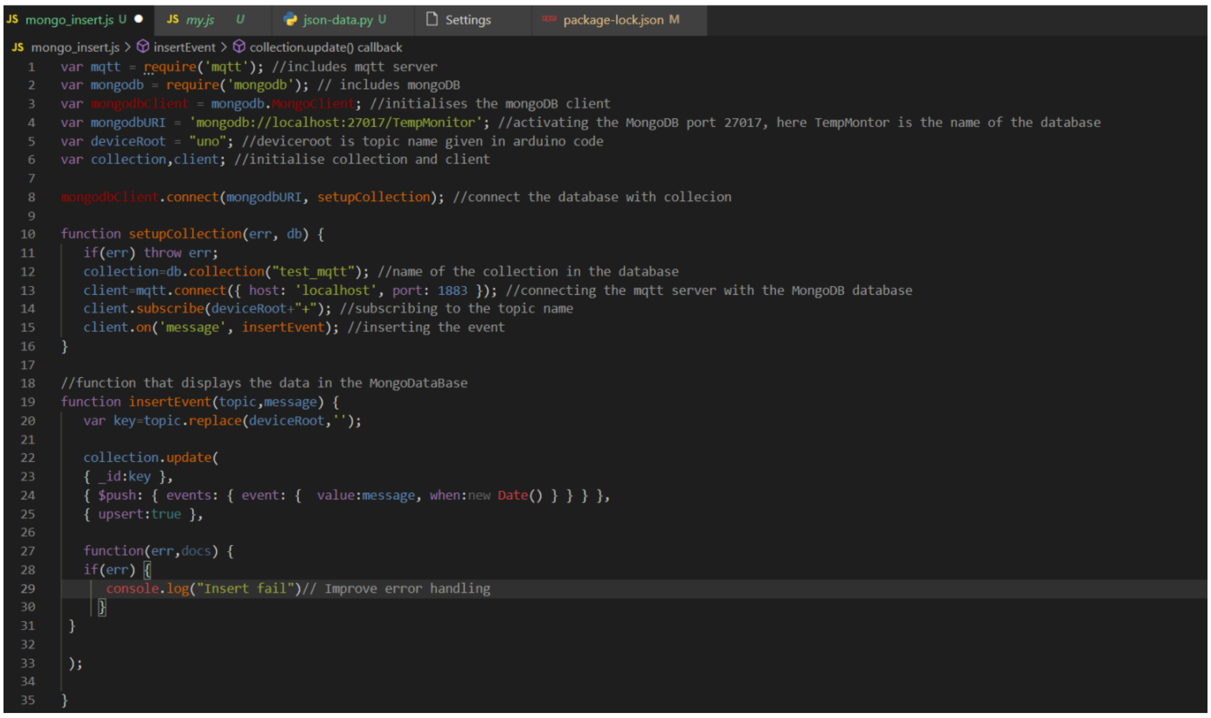


```
#include <SoftwareSerial.h>
#include <WiFiEsp.h>
#include <WiFiEspClient.h>
#include <WiFiEspUdp.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <Servo.h>
#include <ArduinoJson.h>

#define DHTPIN 2
#define DHTTYPE DHT11
#define ESP_RX 9
#define ESP_TX 8
#define GPS_RX 4
#define GPS_TX 3
#define SERVO_PIN 6
#define WIFI_AP "Nitin"
#define WIFI_PASSWORD "Nitin123"
#define MSG_BUFFER_SIZE  (100)



static const int rainSensorMin = 0;
static const int rainSensorMax = 1024;
char msg[MSG_BUFFER_SIZE];
char temp[7];
char hum[7];


WiFiEspClient espClient;
PubSubClient mqttClient(espClient);
IPAddress broker(192,168,43,235);
Servo myservo;

 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);

 //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();
 }


 if(!isnan(t) || !isnan(h))
 {
   //StaticjsonDoc<300> jsonDoc;
   DynamicJsonDocument jsonDoc(200);
   //JsonObject& JSONencoder = jsonDoc.createObject();
   jsonDoc["Temp"] = t;
 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);

 //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();
 }


 if(!isnan(t) || !isnan(h))
```

Arduino Code.

Above is a part of code that runs on Arduino board and send all data.This code is implemented to program the working of the sensors and also connection to MQTT to mosquitto server.

NodeJS code to collect and store data:



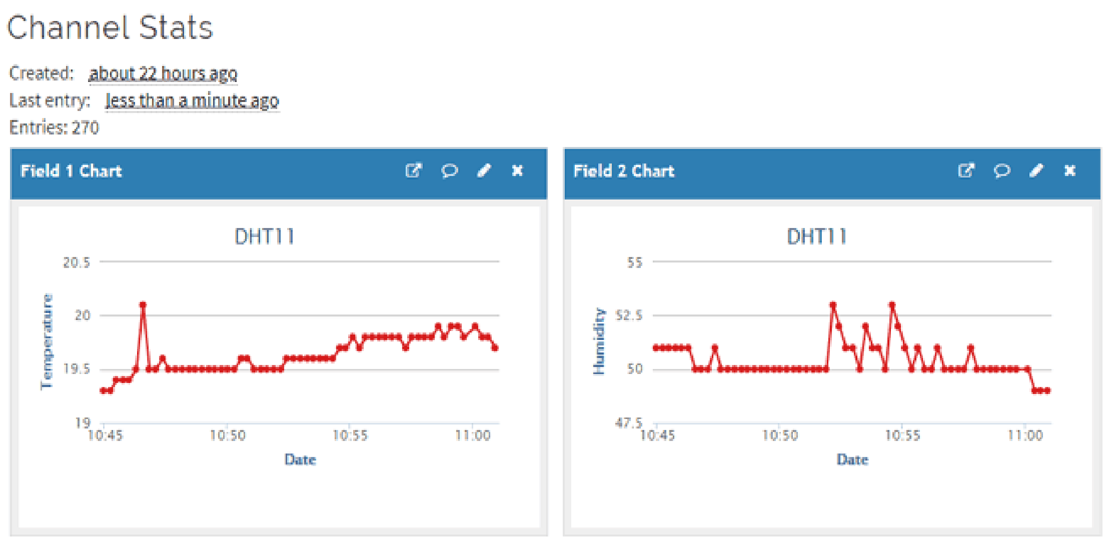
```javascript
var mqtt = require('mqtt'); //includes mqtt server
var mongodb = require('mongodb'); // includes mongoDB
var mongodbClient = mongodb.MongoClient; //initialises the mongoDB client
var mongodbURI = 'mongodb://localhost:27017/TempMonitor'; //activating the MongoDB port 27017, here TempMontor is the name of the database
var deviceRoot = "uno"; //deviceroot is topic name given in arduino code
var collection,client; //initialise collection and client

mongodbClient.connect(mongodbURI, setupCollection); //connect the database with collecion

function setupCollection(err, db) {
   if(err) throw err;
   collection=db.collection("test_mqtt"); //name of the collection in the database
   client=mqtt.connect({ host: 'localhost', port: 1883 }); //connecting the mqtt server with the MongoDB database
   client.subscribe(deviceRoot+"+"); //subscribing to the topic name
   client.on('message', insertEvent); //inserting the event
}

//function that displays the data in the MongoDataBase
function insertEvent(topic,message) {
   var key=topic.replace(deviceRoot,'');

   collection.update(
   { _id:key },
   { $push: { events: { event: {  value:message, when:new Date() } } } },
   { upsert:true },

   function(err,docs) {
   if(err) {
      console.log("Insert fail")// Improve error handling
     }
 }

 );

}
```

Here the data is been extracted by the MQTT through mosquitto and stored in MongoDB

The Data is stored in JSON format.

Frontend:

Homepage:



Channel Stats

Created: about 22 hours ago

Last entry: less than a minute ago

Entries: 270

Homepage.

This is the current layout of homepage which display the chart of temperature and humidity over time. It fetches data from the database created on local server as it get updated. Right now this the only page on website but more pages and graphs are going to be added as we keep working on it.

The frontend is made using react.

TEAM WORK:

The Team distribution and workload distribution during this development process has been given below.

Mishra Nitin Sherbahadur 18BCS1988

As the only member in the team all the things are created by me which includes:

- Configuring hardware.
- Arduino logic coding
- NodeJS server
- Creating database

Dhruv Bhattacharjee 18BCS2072

- Frontend development.

CONCLUSION:

Making this project has improved and hugely expanded my skill set. I got to learn about many new technologies. I learned JavaScript and NodeJS for the purpose of development of this project. I learned how to create webpages in react. And as I keep working on it I intend to create weather prediction of place from the collected data. Also a android app is to created so that the data can be easily viewed from mobile devices.