IOT weather reporting system

A. Introduction

. The system monitors humidity, temperature and raining state using different sensors and displays it to the user using lcd and UI.
It allows the people to directly check the weather states without the need of a weather forecasting agency.

B. Minimum Requirement

- Arduino uno board
- DHT 22 OR DHT 11 sensor
- ESP-8266 WIFI module
- Any raindrops module such as (MH-RD)
- LCD
- 330-ohm resistors, 4700-ohm resistors, and variable resistor.
- wires
- Libraries used (LiquidCrystal.h, DHT.h, ThingSpeak.h, WiFiEsp.h)
- Thingspeak platform

C. Project Setup

Since I'm dealing with 4 different components, so it is better to do the setup for each privately then combine them in one program. The most important programs are the program of the DHT 22 sensor and the setup of the WIFI which are shown down.

- WIFI module setup

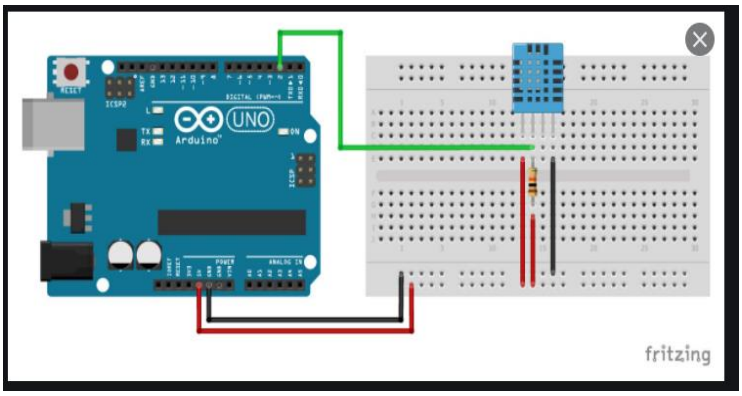
```
void setup() {
  //Initialize serial and wait for port to open
  Serial.begin(115200);

  // initialize serial for ESP module
  setEspBaudRate(ESP_BAUDRATE);

  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }

  Serial.print("Searching for ESP8266...");
  // initialize ESP module
  WiFi.init(&Serial1);

  // check for the presence of the shield
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue
    while (true);
  }
  Serial.println("found it!");

  ThingSpeak.begin(client);  // Initialize ThingSpeak
}
```

```
void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this lin
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you
  // pieces of information in a channel.  Here, we write to field 1.
  int x = ThingSpeak.writeField(myChannelNumber, 1, number, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  // change the value
  number++;
  if(number > 99){
    number = 0;
  }

  delay(20000); // Wait 20 seconds to update the channel again
}
```

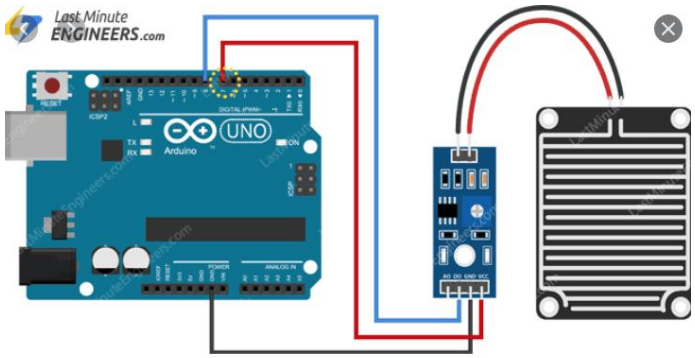
- DHT 22 SETUP

```
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
  dht.begin();
}
void loop() {
  delay(2000);
  // Reading temperature or humidity takes about 250 millisecon
  // Sensor readings may also be up to 2 seconds 'old' (its a v
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
}
```

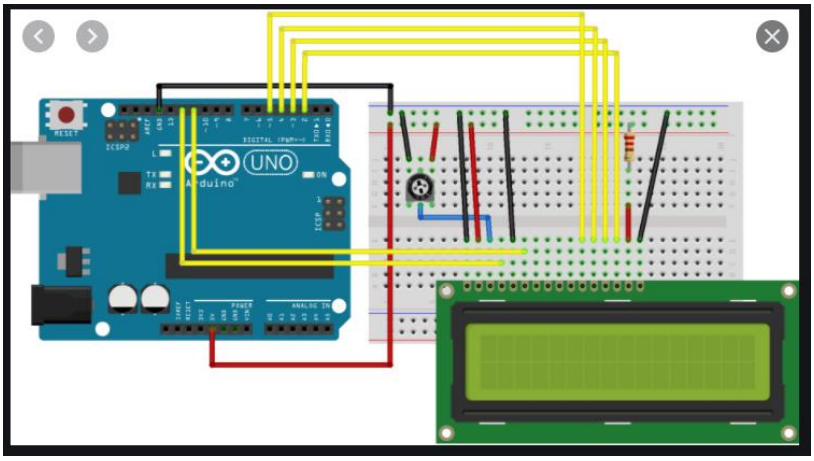
DHT connection



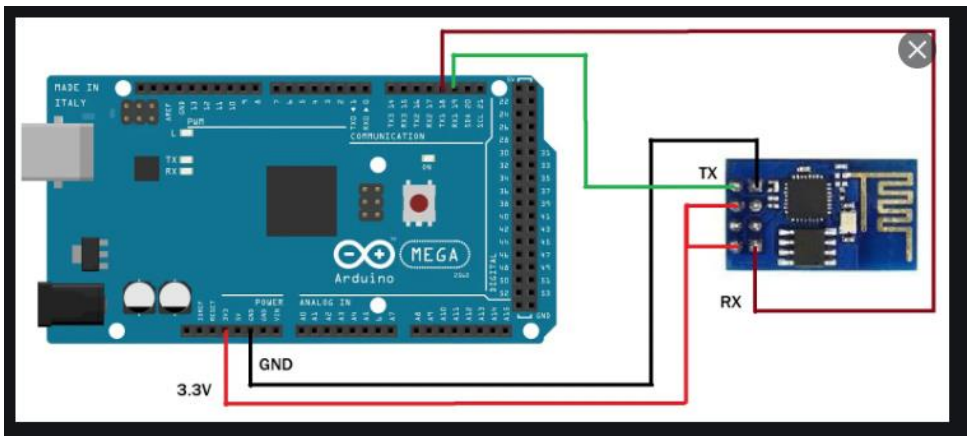
- *raindrop sensor setup*



LCD connection



WIFI module connection



D. Procedure

(all connection mentioned are showed in the setup part)

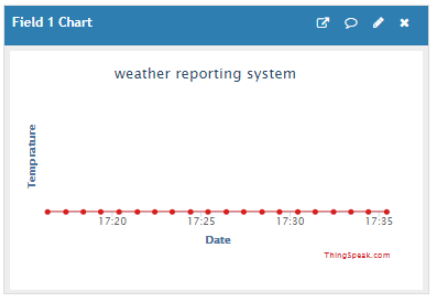
1. Make the connection for the DHT22 sensor
2. *Try the test program of the DHT22 sensor to make sure that the sensor is working well*
3. *connect and program the rain drop sensor and make sure it is giving correct output*
4. *connect and program the LCD to print 'hello world' test its functionality*
5. *create a program called MAIN to update the code used in the project inside it*
6. *make a program that collect the data from the DHT22 and the raindrop sensor and display it on the LCD (inside the main program)*
7. *setup the user interface to display data sent from the sensor and to be user friendly*
8. *connect the WIFI module and try a test example program*
9. *combine the WIFI module with the main program and make sure the data displayed on the LCD is also uploaded to the website*

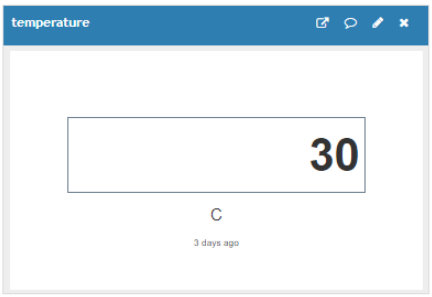
E. Result

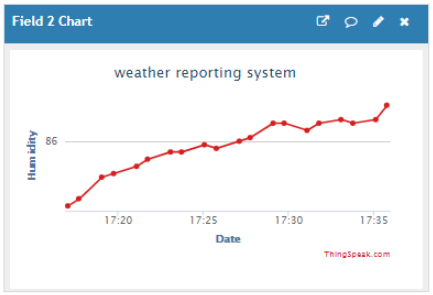
1- As shown in the picture, the first filed is displaying the current temperature value and a graph to show the previous values

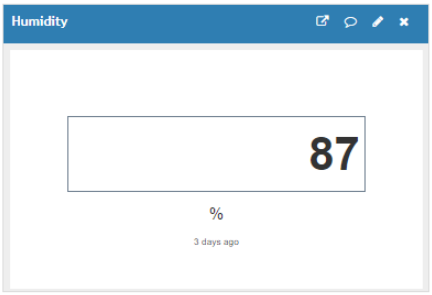
2- the second filed is used to display humidity percentage and a graph to show the previous values

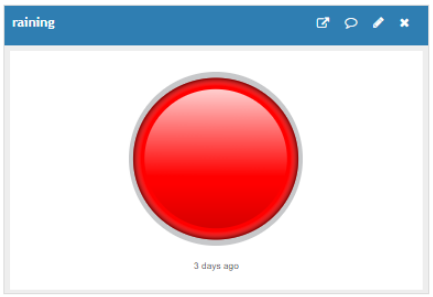
3- the third filed is to store the rain drop level and displayed as a lamb that light up if the rain level exceeded a specific level











created_at	entry_id	temprature	humidity	raindrop
2020-12-06 09:03:01 UTC	1196		82.5	
2020-12-06 09:03:21 UTC	1197	30.3		
2020-12-06 09:03:41 UTC	1198		82.7	
2020-12-06 09:04:01 UTC	1199			862
2020-12-06 09:04:21 UTC	1200	30.3		
2020-12-06 09:04:41 UTC	1201			844
2020-12-06 09:05:02 UTC	1202		83.1	
2020-12-06 09:05:22 UTC	1203	30.3		
2020-12-06 09:05:42 UTC	1204		83.1	
2020-12-06 09:06:03 UTC	1205			902
2020-12-06 09:06:22 UTC	1206	30.3		
2020-12-06 09:06:43 UTC	1207			939
2020-12-06 09:07:03 UTC	1208		83.4	
2020-12-06 09:07:23 UTC	1209	30.3		
2020-12-06 09:07:42 UTC	1210		83.2	
2020-12-06 09:08:03 UTC	1211			1008
2020-12-06 09:08:23 UTC	1212	30.3		
2020-12-06 09:08:42 UTC	1213			1011
2020-12-06 09:09:02 UTC	1214		83.4	
2020-12-06 09:09:22 UTC	1215	30.3		
2020-12-06 09:09:43 UTC	1216		83.4	
2020-12-06 09:10:03 UTC	1217			1015
2020-12-06 09:10:23 UTC	1218	30.3		
2020-12-06 09:10:43 UTC	1219			1016
2020-12-06 09:11:03 UTC	1220		83.6	
2020-12-06 09:11:23 UTC	1221	30.3		
2020-12-06 09:11:44 UTC	1222		83.5	
2020-12-06 09:12:03 UTC	1223			1016
2020-12-06 09:12:24 UTC	1224	30.3		
2020-12-06 09:12:43 UTC	1225			1016
2020-12-06 09:13:04 UTC	1226		83.7	
2020-12-06 09:13:24 UTC	1227	30.3		
2020-12-06 09:13:43 UTC	1228		83.9	
2020-12-06 09:14:04 UTC	1229			1017
2020-12-06 09:14:24 UTC	1230	30.3		
2020-12-06 09:14:44 UTC	1231			1018
2020-12-06 09:15:04 UTC	1232		83.9	
2020-12-06 09:15:24 UTC	1233	30.3		
2020-12-06 09:15:43 UTC	1234		84.3	
2020-12-06 09:16:04 UTC	1235			1017
2020-12-06 09:16:24 UTC	1236	30.3		

F. References

1. https://www.google.com/search?q=dht22+connection&rlz=1C1GCEA_enMY839MY839&sxsrf=ALeKk00YJTAHKOaa9ecwmxEWtwWrpAvpng:1607519220906&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjjrobh-8DtAhXhwzgGHcbTClEQ_AUoAXoECBAQAw&biw=1536&bih=792#imgrc=ouRo1hq0Ms3jSM
2. https://www.google.com/url?sa=i&url=https%3A%2F%2Flastminuteengineers.com%2Frain-sensor-arduino-tutorial%2F&psig=AOvVaw0dRmTApATdU4pc_OXQfmhp&ust=1607605886213000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCICp0eD8wO0CFQAAAAAdAAAAABAO
3. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.arduino.cc%2Fen%2FTutorial%2FLibraryExamples%2FLiquidCrystalDisplay&psig=AOvVaw3qoEklxD-eMcYw3qU0acP2&ust=1607606189754000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPjQzvH9wO0CFQAAAAAdAAAAABAD

Prepared by:

1. Eissa Mohamed sharaf