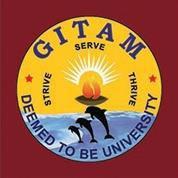
**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

# GITAM INSTITUTE OF TECHNOLOGY,

# GITAM

**(DEEMED TO BE UNIVERISTY)**



**DECLARATION**

We, hereby declare that the IOT project entitled **"HUMIDITY AND TEMRATURE MONITORING”** is an original work done in the department of Computer Science and Engineering, GITAM Institute of Technology, **GITAM** (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech

In COMPUTER SCIENCE AND ENGINEERING.

**REGD NO: IOT PROJECT MEMBERS SIGNATURE**

**121810318033**  **G.GAYATHRI**

**121810318046** **S.V.ISHWARYA**

**121810318058 J.MEGHANA**

# TEMPERATURE AND HUMIDITY MONITORING

**AIM:** To monitor humidity and temperature in surrounding area.

**MATERIALS REQUIRED:**

Hardware Requirements:

* Arduino Uno
* ESP8266-01
* DHT11 sensor
* Bread board
* Male-to-male connecting wires.

Software Requirements:

* Arduino IDE

**THEORY:**

Using Internet of Things (IOT), we can control any electronic equipment in homes and industries . Moreover , you can read a data from any sensor and analyze it either digitally or graphically from anywhere in the world .

**Arduino Uno:**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

The main advantage of Arduino is its simple programming language. It's designed to make new user to get a glimpse and boost their confidence of programming.

**ESP8266-01:**

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

**Uses:**

ESP8266. The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by manufacturer Espressif Systems in Shanghai, China. ... This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands.

**DHT11 sensor:**

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.

**Working:**

The DHT11 calculates relative humidity by measuring the electrical resistance between two electrodes. The humidity sensing component of the DHT11 is a moisture holding substrate with the electrodes applied to the surface.

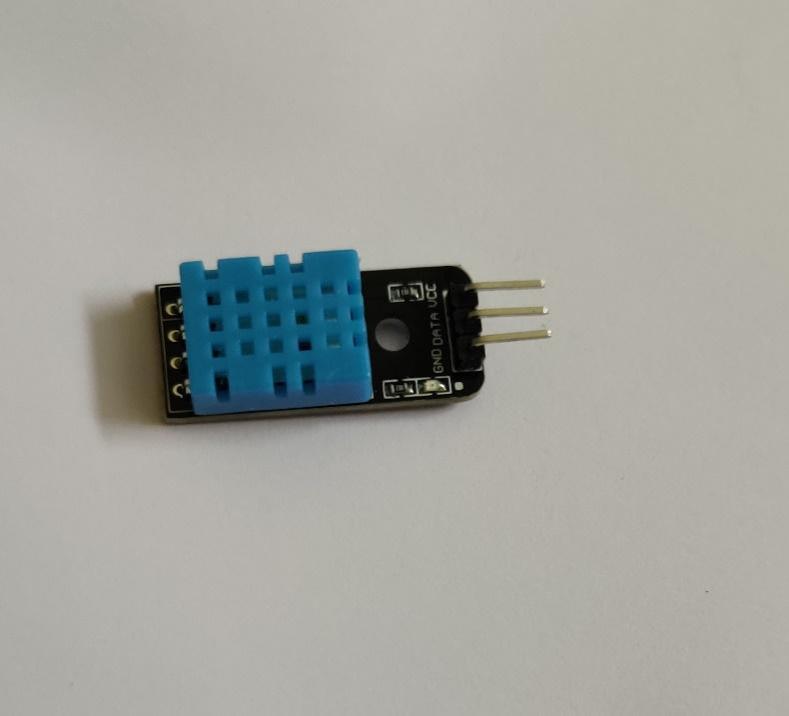
**Bread board:**

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

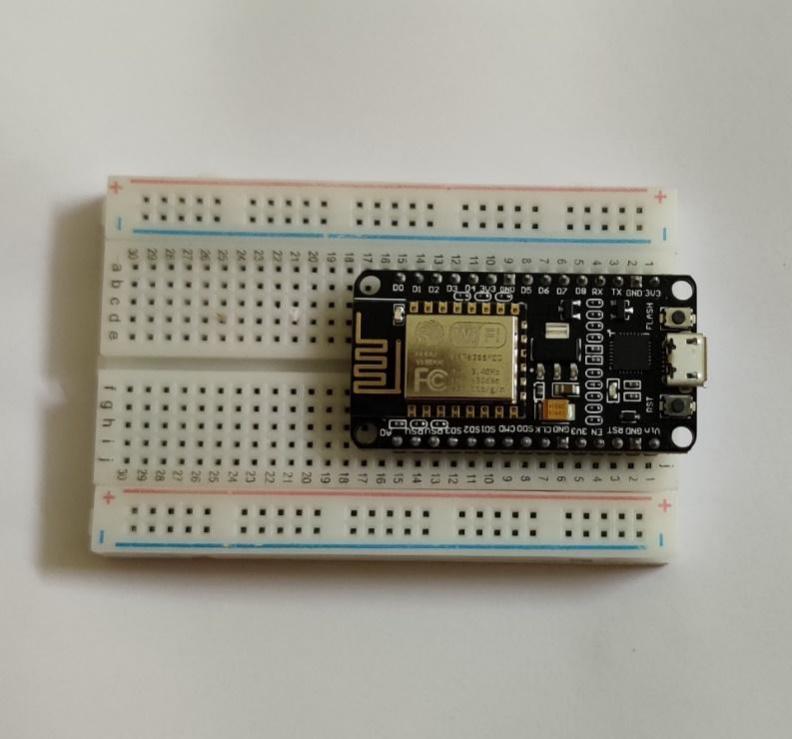
The purpose of the breadboard is to make quick electrical connections between components- like resistors, LEDs, capacitors, etc- so that you can test your circuit before permanently soldering it together.

We can also include a sensor like gas , infrared , ultrasonic sensors based on their requirements . Moreover , it is possible to control the relay , actuators through Internet once a sensor values are going above / below predetermined values.

**PROCEDURE:**

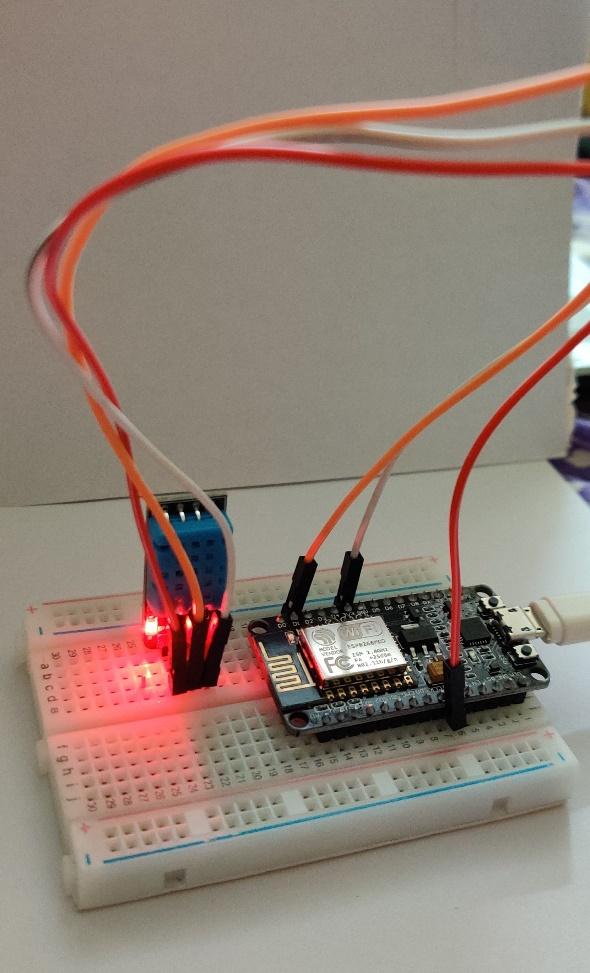
Step-1: Take the bread board and insert the Wi-Fi module on it.

Step-2: Insert the DHT11 sensor on the bread board next to the Wi-Fi module.



Step-3: As the DHT11 sensor contains 3 pins named ground, data, VCC and the wifi module have the pins from D1 to D7 connect the male-to-male connecting wires from ground to ground.

Step-4: Another wire from data pin to D1.

Step-5: Last wire from VCC to 3.3 voltage pin as shown in the figure.

Step-6: Connect the data cable from Wi-Fi module to the laptop.

Step-7: The led of the DHT11 sensor will glow up when you connect it to the laptop.

Step-8: When you compile the source code enter the token which is sent to your mail by cloud app.

Step-9: Enter the Wi-Fi name and password in source code of the mobile which is connected to the blynk.

Step-10: Open the blynk app in the mobile the tap the temperature, give the input as analog and select the V4 option because we connected the wire to D1 in the Wi-Fi module and set the temperature range from 0 to 50 Celsius.

Step-11: Tap humidity and select the V5 option and set the humidity range from 0 to100.

Step-12: Press the RST button on the Wi-Fi module.

Step-13: In the Arduino software on the right top there will be an option. Click on it, then it connects to mobile.

Step-14: Then it displays the temperature and humidity values on the mobile screen based on the Wi-Fi range.

**SOURCE CODE:**

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include "DHT.h"           // including the library of DHT11 temperature and humidity sensor

#define DHTTYPE DHT11      // DHT 11

#define DHT\_PIN D1

#define UPDATE\_TIME 2000

DHT dht(DHT\_PIN, DHTTYPE);

char auth[] = "\*\*\*\*\*\*\*\*\*\*";            // You should get Auth Token in the Blynk App.

char ssid[] = "\*\*\*\*";    // Your WiFi credentials.

char pass[] = "\*\*\*\*\*";  // Set password to "" for open networks.

float t;               // Declare the variables

float h;

void setup()

{

    Serial.begin(9600);// Debug console

    Blynk.begin(auth, ssid, pass);

    dht.begin();

}

void sendUptime()

{

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  // Read temperature as Fahrenheit

  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("Failed to read from DHT sensor!");

    return;

  }

  float hi = dht.computeHeatIndex(f, h);

  Serial.print("Humidity: ");

  Serial.print(h);

  Serial.print(" %\t");

  Serial.print("Temperature: ");

  Serial.print(t);

  Serial.print(" \*C ");

  Serial.print(f);

  Serial.print(" \*F\t");

  Serial.print("Heat index: ");

  Serial.print(hi);

  Serial.println(" \*F");

  Blynk.virtualWrite(V4, t);

  Blynk.virtualWrite(V5, h);

}

void loop()

{

  unsigned long current\_millis;

  // get current time

  current\_millis = millis();

  while(1)

  {

    Blynk.run();

    if (millis() - current\_millis >= UPDATE\_TIME)

    {

       sendUptime();

       break;

    }

  }

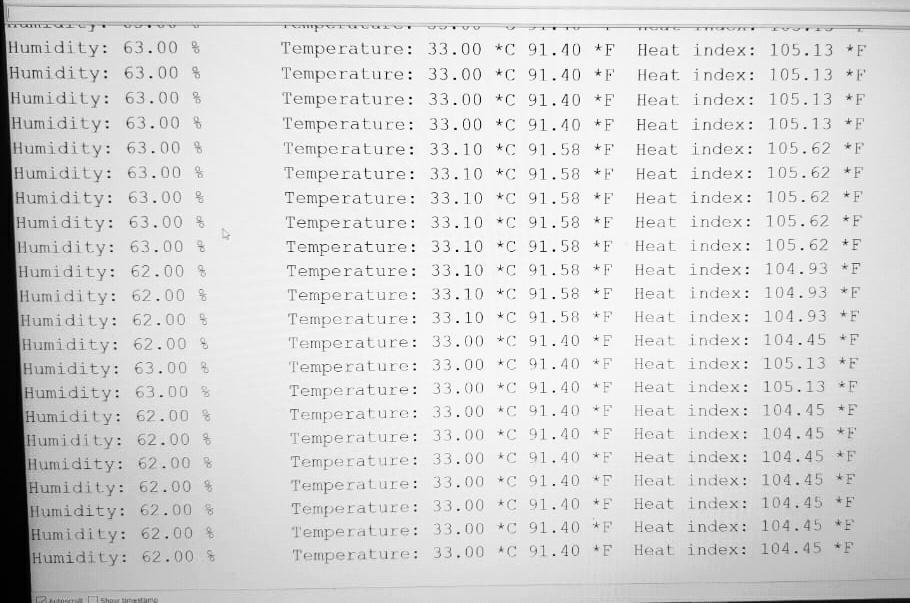
}

**OUTPUT:**

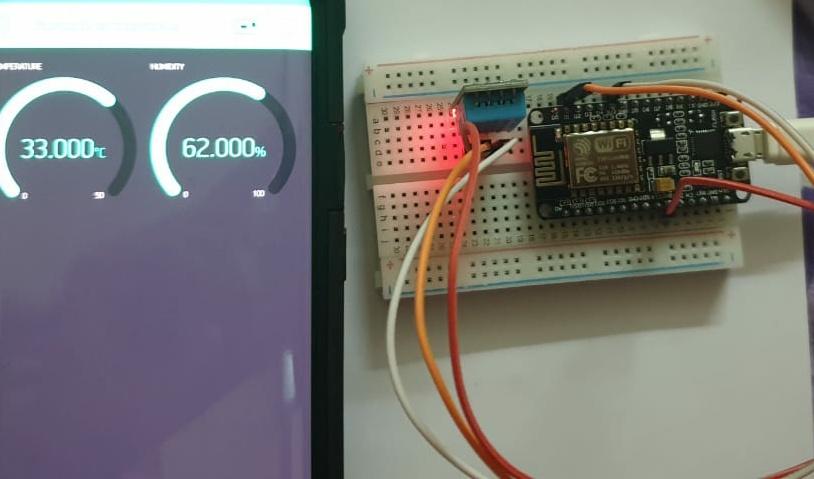
The output can be shown in three different cases:

Case-1: When the sensor is at room temperature.

On the laptop screen:

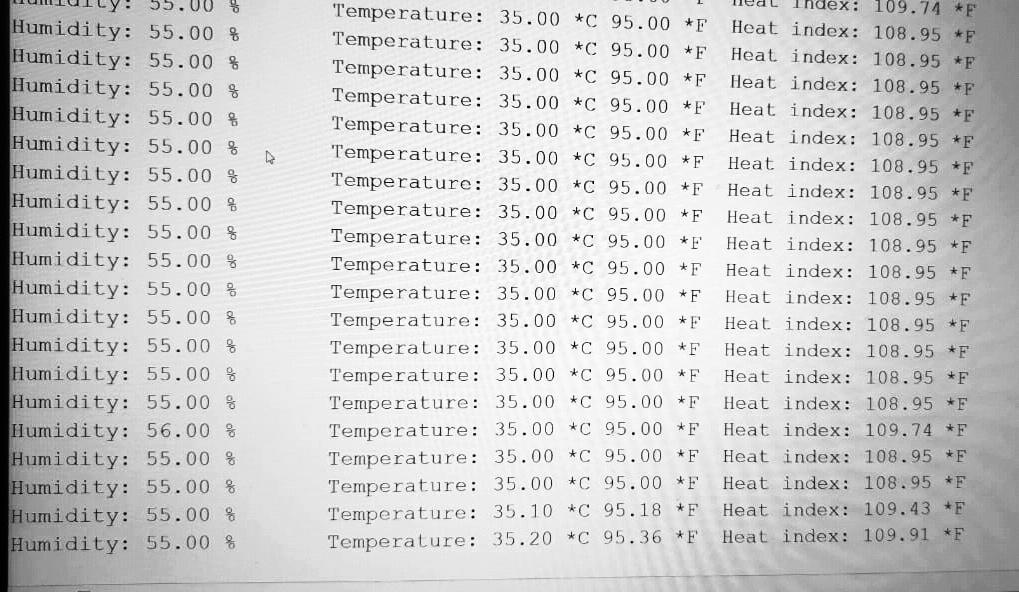


On the mobile screen:

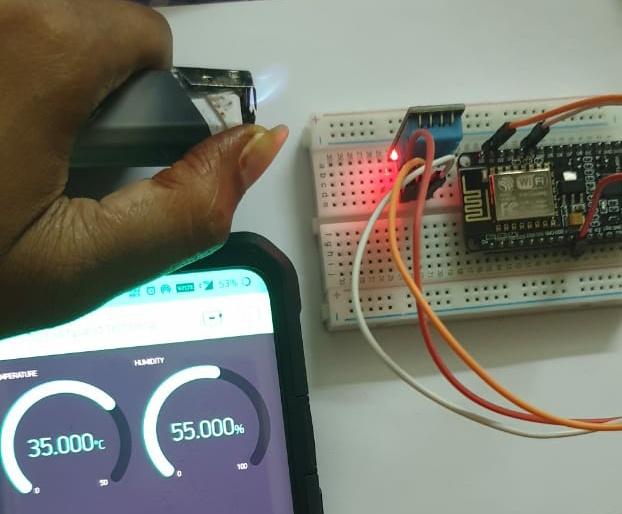


Case-2: When the sensor is placed at hot temperature.

On the laptop screen:

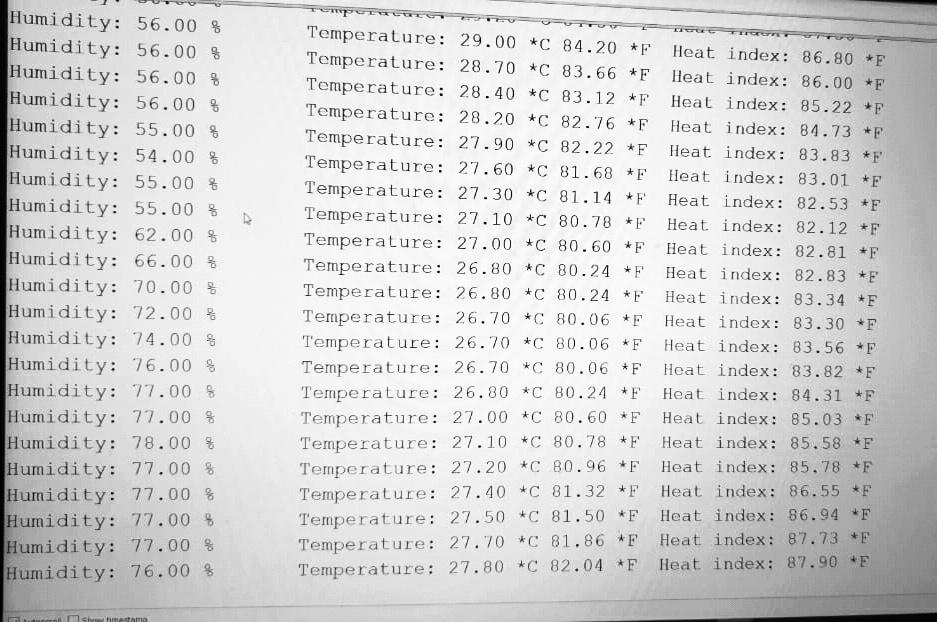


On the mobile screen:

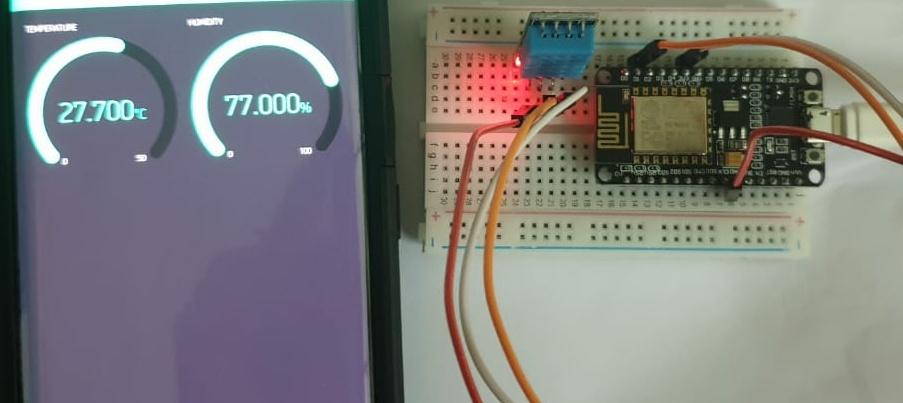


Case-3: When the sensor is placed at cool temperature.

On the laptop screen:



On the mobile screen:



**Result:**

Therefore, by following the above mentioned procedure we can monitor the temperature and humidity values using the internet of things(IOT).