
Activity Report on ELS5308

# DHT11 Humidity Temperature Monitor on ThingSpeak with NodeMCU

**Under the Guidance of**

Ms. Yogita Khandge

Ms. Panchsheela Kamble

**Submitted By**:

Ajay Singh (226902)

Mayank Bankoti (226914)

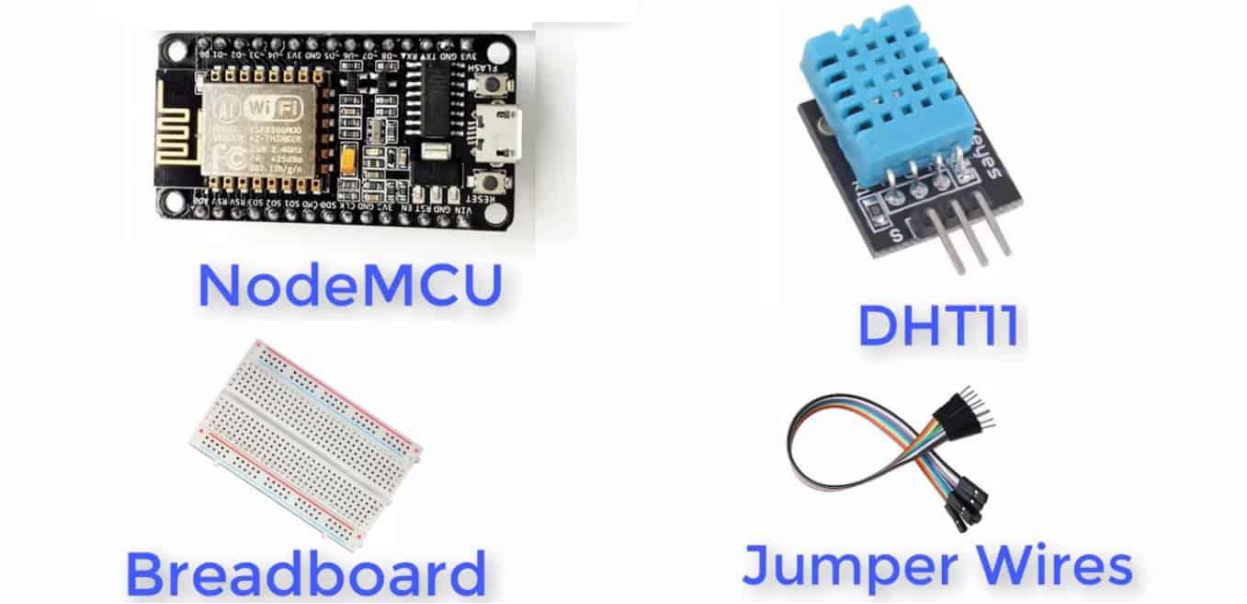
Date: 09/26/2023 Signature

# DHT11 Humidity Temperature Monitor on ThingSpeak with NodeMCU

**Introduction:**

In this project, we will create a humidity and temperature monitoring system using a DHT11 sensor and NodeMCU. The data collected by the sensor will be uploaded to ThingSpeak, a cloud-based IoT platform, for visualization and analysis. This report will guide you through the necessary steps to set up and run this project successfully.

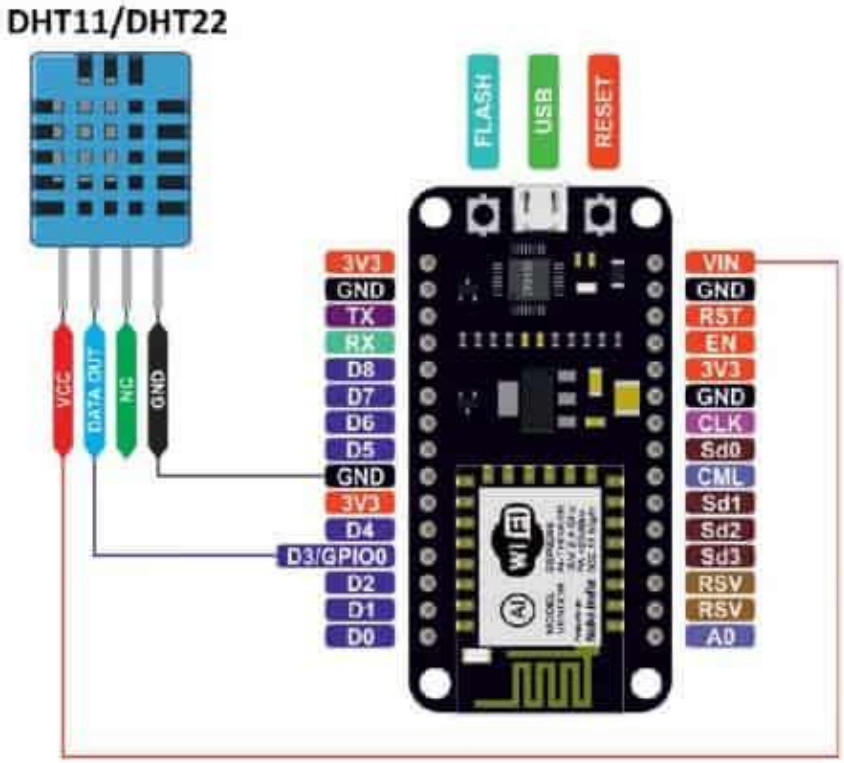
## Component Required:




**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).It’s fairly simple to use but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using the library, sensor readings can be up to 2 seconds old.

**NodeMCU:** NodeMCU is an open-source firmware and development kit designed for the ESP8266 microcontroller. This compact and affordable platform provides built-in Wi-Fi connectivity, making it exceptionally well-suited for Internet of Things (IoT) projects. NodeMCU supports both the Lua scripting language and the Arduino IDE, allowing developers to quickly prototype and deploy a wide range of IoT applications. With its ease of use, extensive community support, and low power consumption, NodeMCU has become a popular choice for makers, hobbyists, and developers looking to create IoT devices and smart solutions.

## Circuit Diagram & Connection:




- Connect the DHT11 sensor to the NodeMCU as follows:
- DHT11 VCC to NodeMCU 3.3V
- DHT11 GND to NodeMCU GND
- DHT11 DATA to NodeMCU D3
- Power the NodeMCU using the USB cable connected to your computer.

## Setting Up ThingSpeak & Getting API Key:

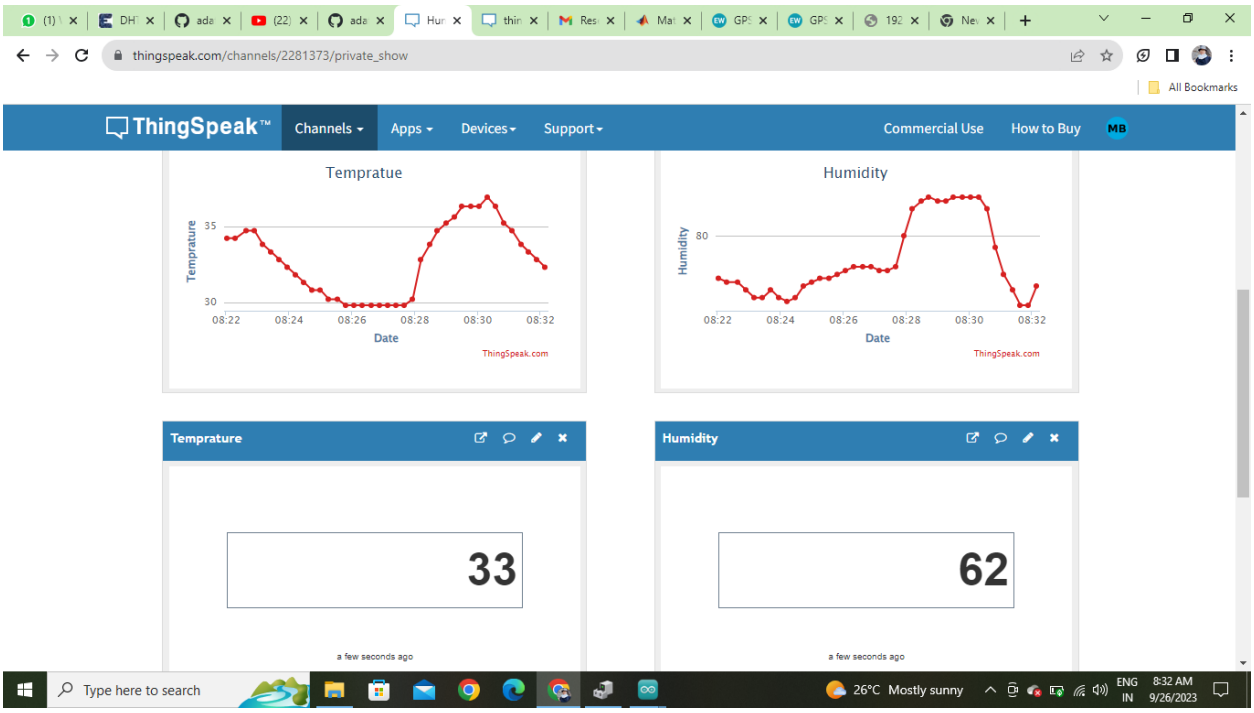
- Create a ThingSpeak account if you don't have one already.
- Log in to ThingSpeak and create a new channel.
- Note down the API Key and Channel ID for later use.

# Algorithm:

The program for Humidity & Temperature Monitoring using DHT11 & NodeMCU on ThingSpeak is given below.

- Write the program on Arduino IDE.
- Download the DHT11 library from GitHub and add it to your library manager.
- Select the NodeMCU ESP-12E board from the board manager.
- Paste your API Key from thingspeak which you created earlier on a programming section line.
- Edit the program to change the wifi SSID and password with your own.
- Compile the code and Upload it to NodeMCU board

## Monitor Humidity Temperature Data on ThingSpeak



## Source Code:

```
#include <DHT.h>  // Including library for dht

#include <ESP8266WiFi.h>

String apiKey = "H38TEGNC0XKW43BB";     //  Enter your Write API key from ThingSpeak

const char *ssid =  "Fergusson College";     // replace with your wifi ssid and wpa2 key
const char *pass =  "ELS@123";
const char* server = "api.thingspeak.com";

#define DHTPIN 0          //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
       Serial.begin(115200);
       delay(10);
       dht.begin();

       Serial.println("Connecting to ");
       Serial.println(ssid);


       WiFi.begin(ssid, pass);

      while (WiFi.status() != WL_CONNECTED)
     {
           delay(500);
           Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");

}

void loop()
{

      float h = dht.readHumidity();
      float t = dht.readTemperature();

            if (isnan(h) || isnan(t))
               {
                  Serial.println("Failed to read from DHT sensor!");
                   return;
               }

                     if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
```

```
            {

                String postStr = apiKey;
                postStr +="&field1=";
                postStr += String(t);
                postStr +="&field2=";
                postStr += String(h);
                postStr += "\r\n\r\n";

                client.print("POST /update HTTP/1.1\n");
                client.print("Host: api.thingspeak.com\n");
                client.print("Connection: close\n");
                client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                client.print("Content-Type: application/x-www-form-urlencoded\n");
                client.print("Content-Length: ");
                client.print(postStr.length());
                client.print("\n\n");
                client.print(postStr);

                Serial.print("Temperature: ");
                Serial.print(t);
                Serial.print(" degrees Celcius, Humidity: ");
                Serial.print(h);
                Serial.println("%. Send to Thingspeak.");
              }
        client.stop();

        Serial.println("Waiting...");

  // thingspeak needs minimum 15 sec delay between updates
  delay(1000);
}
```