

# Assignment No 4

**1. Problem Statement :-** Upload data from environmental sensor to cloud server.

**2. Hardware Required :-**

- NodeMCU 8266
- Temperature and Humidity Sensor (DHT11)
- hook-up wires
- breadboard

**3. Software Required :-**

- Arduino Software (IDE)
- ThingSpeak Cloud

**4. Theory**

In this Assignment, we are going to monitor temperature and humidity data over internet using ThingSpeak IoT cloud. We can view that data and graph over time on their website. It is made using ESP8266 WiFi module and DHT11 temperature and humidity sensor. ESP8266 WiFi chip reads the current temperature and humidity from DHT11 using ADC and sends it to ThingSpeak server for live monitoring from anywhere in the world.

## **What is ThingSpeak API?**

ThingSpeak is a very good platform for IoT based projects. By using channels and web pages provided by ThingSpeak we can monitor any data over the Internet from anywhere and we can also control our system over the internet. ThingSpeak 'Collects' data from sensors, 'Analyze and Visualize' data and 'Acts' by triggering a reaction.

### **• Steps to setup ThingSpeak account for this assignment.**

1. First of all, user needs to create an account on ThingSpeak.com, then Sign In and click on 'Get Started'.
2. Now go to the 'Channels' menu and click on the 'New Channel' option in the same page.
3. Now you will see a form for creating the channel, fill the Name and Description as per your choice. Then fill 'Temperature' in 'Field 1' field and 'Humidity' in 'Field 2' field. Tick the check box 'Make Public' & 'Show status' option below the form and finally Save the channel. Now your new channel is ready.
4. Now click on 'API keys' tab and note the Write and Read API keys, here we are only using Write key. You need to copy and paste this key in the code.

5. Now user need to upload the program to ESP8266 using Arduino IDE.
6. After uploading, open “PRIVATE VIEW” icon in ThingSpeak website and observe the monitored temperature and humidity values on graph.

- **Keywords used in ThingSpeak API**

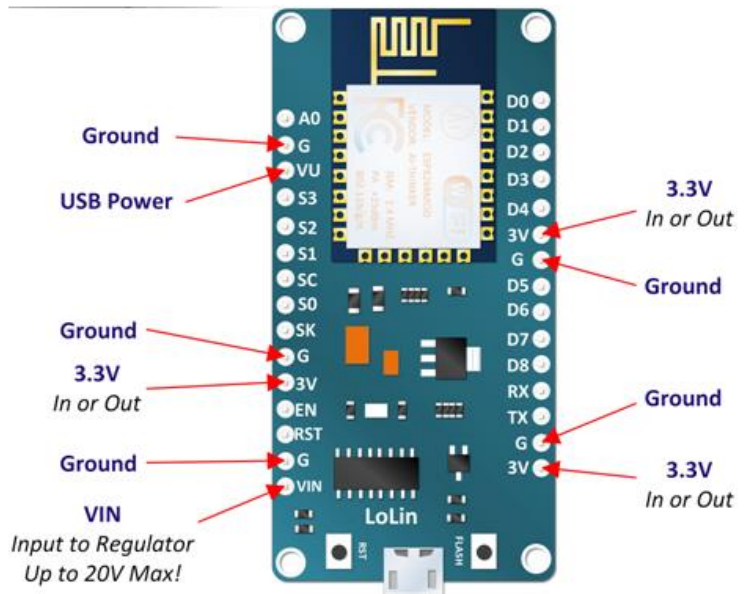
1. **Channel** – Channel can be said as a stream of data. It is identified by a numerical channel ID using which data can be inserted or retrieved using ThingSpeak APIs.
2. **Field** – Each channel is having 8 fields which can hold any type of data. For eg. you may store temperature, humidity, RFID data (alphanumeric) in different fields of the same channel.
3. **Status** – It is short status message to augment the data stored in a channel.
4. **Location** – In addition to above 8 fields we can store gps location or coordinates. For eg. we can store the location of the place from where the data is coming. It is having latitude, longitude and elevation.
5. **Write API Key** – A 16 digit API key code that allows an application to write data to a channel. You should not share this API key publicly, because anyone having this key can write data to your channel.
6. **Read API Key** – A 16 digit API key code that allows an application to read the data stored in a channel. You should not share this API key publicly, because anyone having this key can read data from your channel.

### **ESP8266:**

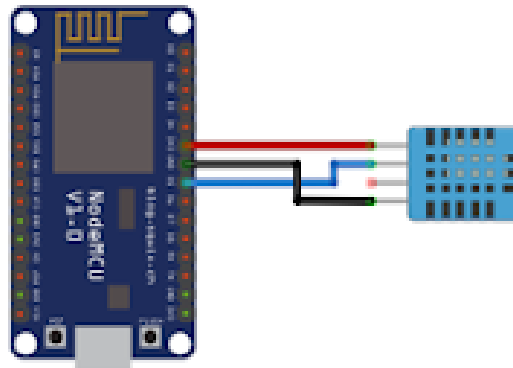
NodeMCU (Micro Controller Unit) is an open source IoT platform. ESP8266 is a low-cost, WiFi Module chip that can be configured to connect to the Internet. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

#### **Features :-**

- Processor: 32-bit RISC microprocessor core running at 80 MHz
- Memory:
  - 32 KB instruction RAM
  - 80 KB user-data RAM
- IEEE 802.11 b/g/n Wi-Fi
- 16 GPIO pins



## Interfacing Temperature sensor with ESP8266



### Steps to be followed:

1. Build interfacing circuit.
2. Create ThingSpeak account. Create new channel and note Write API key.
3. Interface DHT11 with ESP8266 with the help of jumper wires.
4. Write Arduino sketch for required task.
5. Update the write API key, SSID name & password and the Channel ID of ThingSpeak.
6. Go to Manage Libraries. Search ESP8266. Click on install.
7. Add following link to file/preferences  
[http://arduino.esp8266.com/versions/2.4.2/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/versions/2.4.2/package_esp8266com_index.json)  
 Go to Board manager. Search ESP8266. Click on install. Select board as Node MCU 1.0 (ESP-12E Module).
8. Go to Manage Libraries. Search DHT11 Sensor Library and Install the library for DHT11.

9. Go to Manage Libraries. Search Thingspeak by Mathworks Library and Install the library for Thingspeak.
10. Select port as given activated in Control panel/Device manager/Ports/Silicon Labs.
11. Compile & upload sketch.
12. Go to ThingSpeak and observe private view of your channel.

**Code:**

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
#define DHTPIN D5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "Komal";
const char* password = "nuttertool";
WiFiClient client;
unsigned long myChannelNumber = 1016656;
const char * myWriteAPIKey = "XFG3D1F5D4YGXXXZ";
uint8_t temperature, humidity;
void setup()
{
  Serial.begin(115200);
  dht.begin();
  delay(10);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Print the IP address
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);
}
void loop()
{
  static boolean data_state = false;
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
```

```
Serial.print("Temperature Value is :");
Serial.print(temperature);
Serial.println("C");
Serial.print("Humidity Value is :");
Serial.print(humidity);
Serial.println("%");
// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
// pieces of information in a channel. Here, we write to field 1 and field2.
if( data_state )
{
  ThingSpeak.writeField(myChannelNumber, 1, temperature, myWriteAPIKey);
  data_state = false;
}
else
{
  ThingSpeak.writeField(myChannelNumber, 2, humidity, myWriteAPIKey);
  data_state = true;
}
delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}
```

## 5. Conclusion :-

## 6. Sample Questions :-

1. Describe different control structures used in Arduino.
2. What is the use of Arduino WiFi library? Which classes are included in this library?
3. Describe Wifi client class and its respective functions.
4. Describe function from WiFi class
  - a. Begin
  - b. Disconnect
  - c. Status
  - d. Config
  - e. SSID

<https://www.arduino.cc/en/Reference/WiFi>