EEL 7170 : Introduction to IoT

Lab Report

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Name: **Anushkaa Ambuj**

Roll Number: **B21ES006**

Program: **B.Tech in ES**

1 Lab-1

1.1 Objective

To interface the NodeMCU with APIs for fetching and sending data

1. Fetch data from OpenWeather API and upload it to ThingSpeak API
2. Generate Random Numbers and send it to ThingSpeak API

1.2 Components Used

1. NodeMCU (ESP8266 or ESP32)
2. Laptop or desktop with Arduino IDE installed
3. USB cable (for connecting NodeMCU)
4. Internet access
5. OpenWeather API account
6. ThingSpeak API account

1.3 Procedure

- Set WIFI Configurations
- Initialize the Openweather and ThingSpeak API Keys
- Modify the API endpoint by adding the code block mentioned below:

  String serverPath = (serverName + "&field1=" + JSON.stringify(myObject["main"]["temp"]) + "&field2=" + JSON.stringify(myObject["main"]["pressure"]) + "&field3=" + JSON.stringify(myObject["main"]["humidity"]) + "&field4=" + JSON.stringify(myObject["wind"]["speed"]) + "&field5=" + String(random(126)));

- Note: The JSON should be converted into 'string' before sending to ThingSpeak API
- Alter the code by the number (21*6) = 126 for generating random numbers, in code snippet *random(126)*

1.4 Code

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <Arduino_JSON.h>

// Wifi Configurations
const char* ssid = "IDEAPAD";
const char* password = "Hello144";

// OpenWeather API Key
String openWeatherMapApiKey = "56d550e863d263ebf0f515693bc3186b";
```

```
// Add THINGSPEAK.COM API KEY
String serverName = "http://api.thingspeak.com/update?api_key=4E785DAEOKBXTSZC";

// Set your country code and city
String city = "Jodhpur";
String countryCode = "IN";

// THE DEFAULT TIMER IS SET TO 10 SECONDS FOR TESTING PURPOSES
// For a final application, check the API call limits per hour/minute to avoid getting blocked/banned
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 10 seconds (10000)
unsigned long timerDelay = 10000;

String jsonBuffer;

void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

  Serial.println("Timer set to 10 seconds (timerDelay variable), it will take 10 seconds before publishing the first reading.");
}

void loop() {
  // Send an HTTP GET request
  if ((millis() - lastTime) > timerDelay) {
    // Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
      String serverPath1 = "http://api.openweathermap.org/data/2.5/weather?q=" + city + "," + countryCode + "&APPID=" + openWeatherMapApiKey;

      jsonBuffer = httpGETRequest(serverPath1.c_str());
      Serial.println(jsonBuffer);
      JSONVar myObject = JSON.parse(jsonBuffer);

      // JSON.typeof(jsonVar) can be used to get the type of the var
      if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;
      }
```

```

      Serial.print("JSON object = ");
      Serial.println(myObject);
      Serial.print("Temperature: ");
      Serial.println(myObject["main"]["temp"]);
      Serial.print("Pressure: ");
      Serial.println(myObject["main"]["pressure"]);
      Serial.print("Humidity: ");
      Serial.println(myObject["main"]["humidity"]);
      Serial.print("Wind Speed: ");
      Serial.println(myObject["wind"]["speed"]);

      WiFiClient client;
      HTTPClient http;

      String serverPath = (serverName + "&field1=" + JSON.stringify(myObject["main"]["temp"]) + "&field2=" + JSON.stringify(myObject["main"]["pressure"]) + "&field3=" + JSON.stringify(myObject["main"]["humidity"]) + "&field4=" + JSON.stringify(myObject["wind"]["speed"]) + "&field5=" + String(random(126)));

      // Your Domain name with URL path or IP address with path
      http.begin(client, serverPath.c_str());

      // Send HTTP GET request
      int httpResponseCode = http.GET();

      if (httpResponseCode>0) {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        Serial.println(payload);
      }
      else {
        Serial.print("Error code: ");
        Serial.println(httpResponseCode);
      }
      // Free resources
      http.end();
    }
    else {
      Serial.println("WiFi Disconnected");
    }
    lastTime = millis();
  }
}

String httpGETRequest(const char* serverName) {
  WiFiClient client;
  HTTPClient http;

  // Your IP address with path or Domain name with URL path
  http.begin(client, serverName);
```

```
  // Send HTTP POST request
  int httpResponseCode = http.GET();

  String payload = "{}";

  if (httpResponseCode>0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    payload = http.getString();
  }
  else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  // Free resources
  http.end();

  return payload;
}
```

1.5 Results

We successfully obtain the Temperature, Pressure, Humidity, Wind Speed and Random number Plots on the ThinkSpeak website using NodeMCU microcontroller.

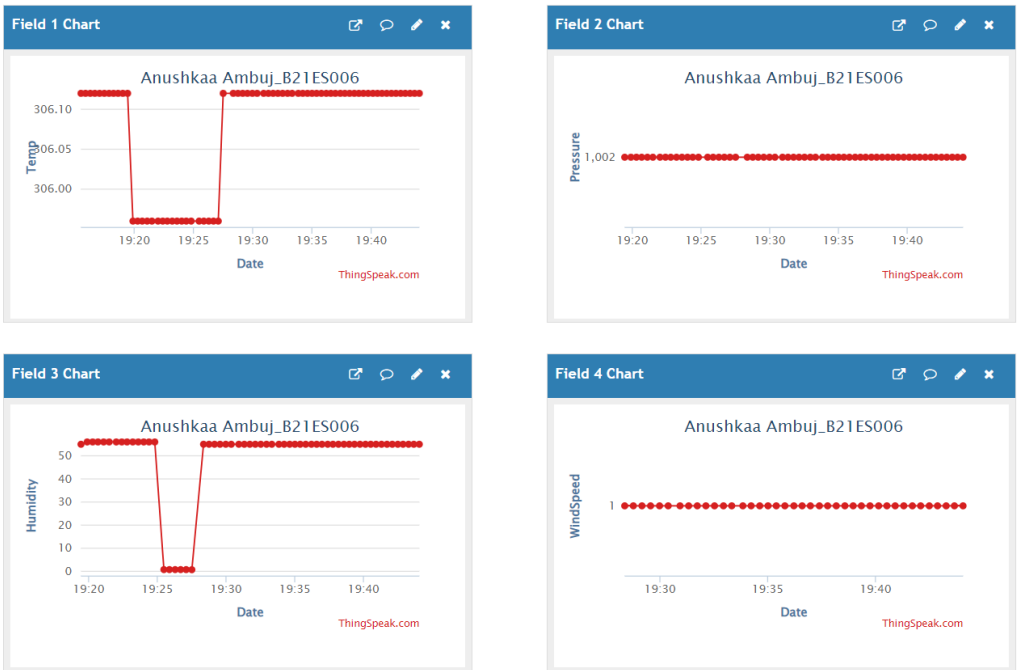

Figure 1: Plots of data received from Openweather API

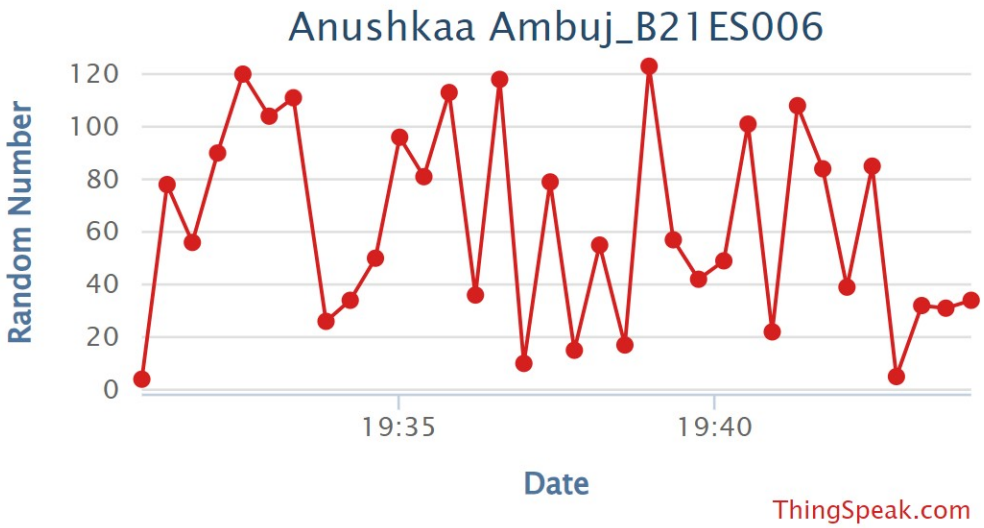

Figure 2: Plot of Random Number sent to ThingSpeak

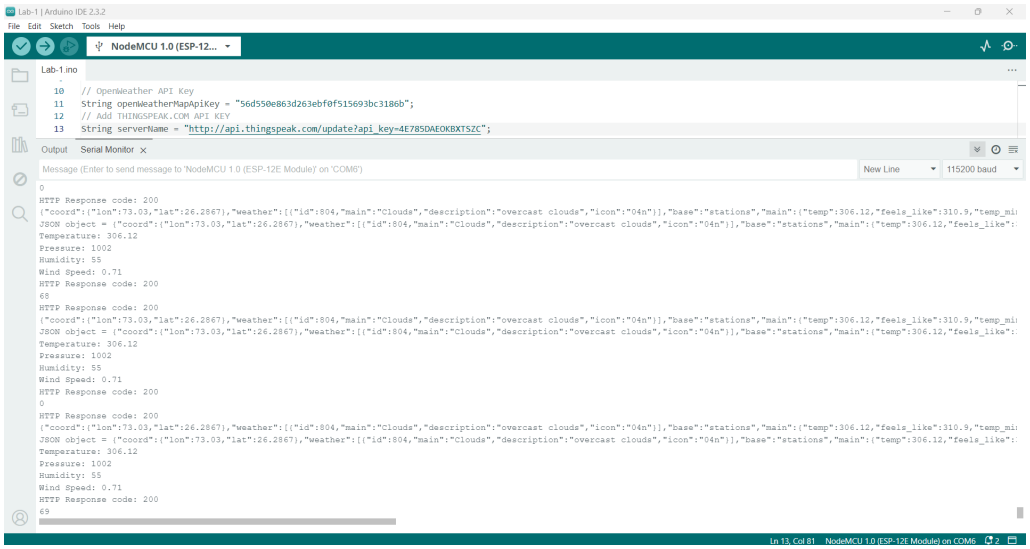

Figure 3: Serial Monitor Output

2 InLab-1

2.1 Problem 1

Objective

- Fetching and Sending Data from OpenWeather API: To fetch weather data and display it on the serial console.

Procedure

- To display data from OpenWeather on the serial monitor, we utilized the API.ino file provided through Google Classroom. First, we updated the Wi-Fi credentials by setting the ssid and password variables to match our network details. After creating an OpenWeather account to obtain the API key, we inserted the key into the openWeatherMapApiKey variable in the Arduino IDE. Finally, we uploaded the code to the NodeMCU and observed the output on the Serial Monitor.

- To begin, open the thingspeak.ino file provided on Google Classroom and update the Wi-Fi configuration by entering your network credentials. Next, create a ThingSpeak account and set up a new channel. Copy the write API key from your ThingSpeak channel and paste it into the appropriate section in the Arduino IDE. After updating the code, upload it to the NodeMCU and verify that the serial console displays an "HTTP Response code: 200". Once successful, you can visualize the uploaded data on the ThingSpeak platform.

Results

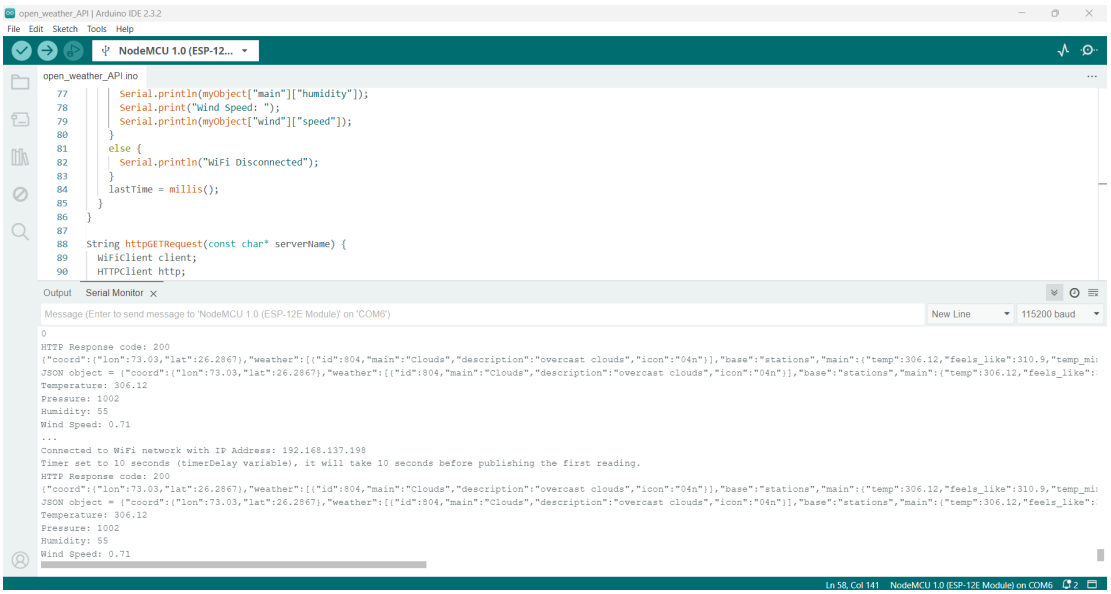

Figure 4: OpenWeather Data on the Serial Monitor

2.2 Problem-2

Objective

- Sending Data to ThingSpeak API: To upload random data to ThingSpeak and visualize it.

Results

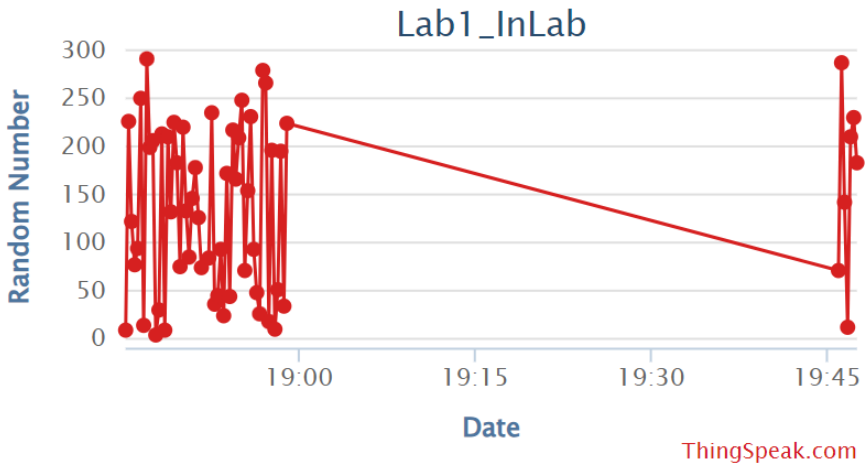

Figure 5: Plot of Random Number on Thingspeak

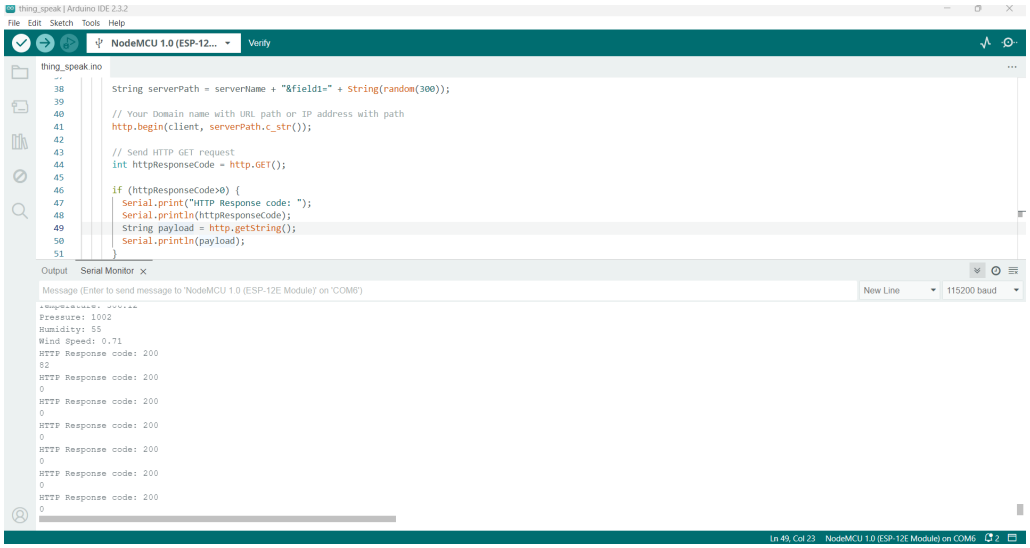

Figure 6: Random Number on the Serial Monitor