**BSc in Information Technology**
**Year 4 - Semester 1/2**

**Lab Exercise 4**

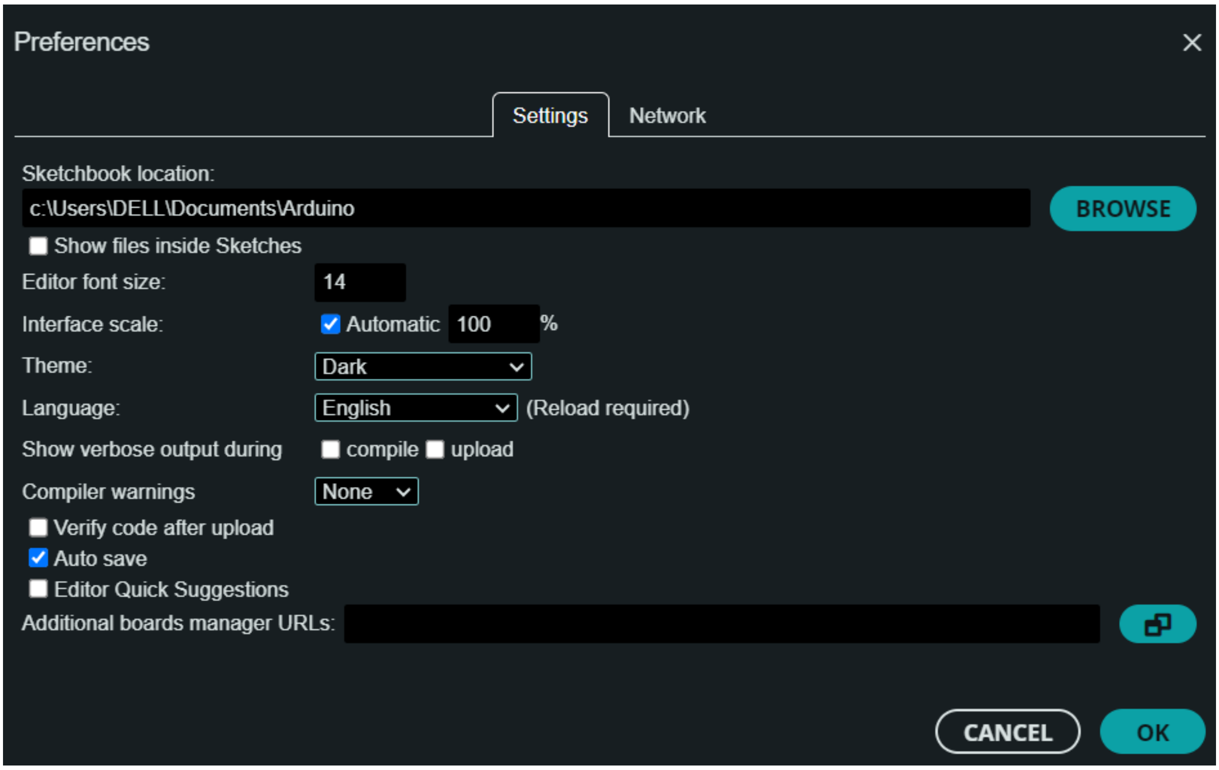
**IT4030 - Internet of Things (IoT) 2024**

# Lab 4: measure humidity and temperature using a NodeMCU (ESP8266) and a DHT11 sensor while publishing the data to Adafruit IO

1) Goto **Adafruit** and create a user account. (IO -> Feeds{Create two feeds})
   Adafruit Industries, Unique & fun DIY electronics and kits

2) i)Search **"Nodemcu ESP8266 Preferences link"** and copy the link or,
   ii)visit Quick Start to Nodemcu (ESP8266) on Arduino IDE : 3 Steps (with Pictures) - Instructables or any suitable website and copy Preferences link or,
   iii)use http://arduino.esp8266.com/stable/package_esp8266com_index.json
   link as Preferences link.

3) Goto Arduino IDE -> File -> Preferences -> Additional boards manager URLs -> Paste the link -> OK



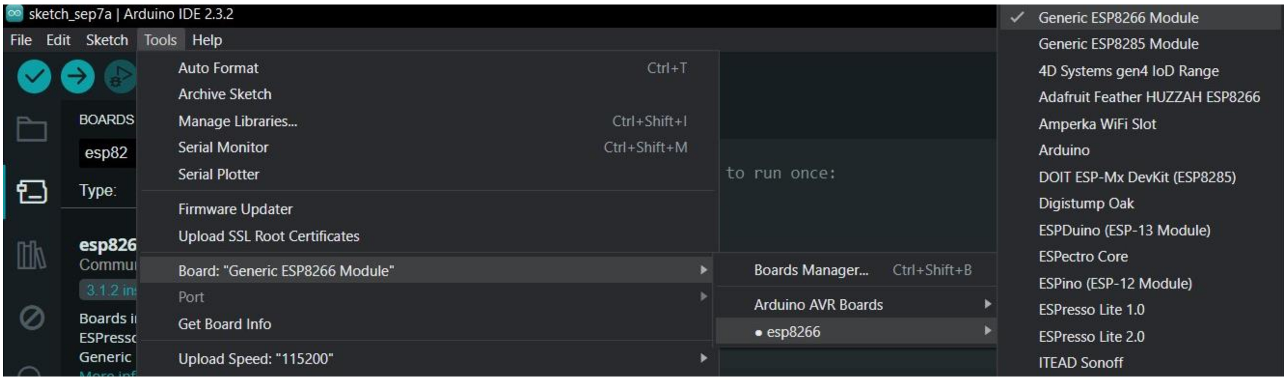
Preferences

Settings | Network

Sketchbook location:
c:\Users\DELL\Documents\Arduino
□ Show files inside Sketches
Editor font size: 14
Interface scale: ☑ Automatic 100 %
Theme: Dark
Language: English (Reload required)
Show verbose output during □ compile □ upload
Compiler warnings None
□ Verify code after upload
☑ Auto save
□ Editor Quick Suggestions
Additional boards manager URLs:

4) Arduino IDE -> Tools -> Board -> Board Manager -> Search “esp8266” -> Install

5) Arduino IDE -> Tools -> Board -> esp8266 -> Generic ESP8266 Module



6) Code

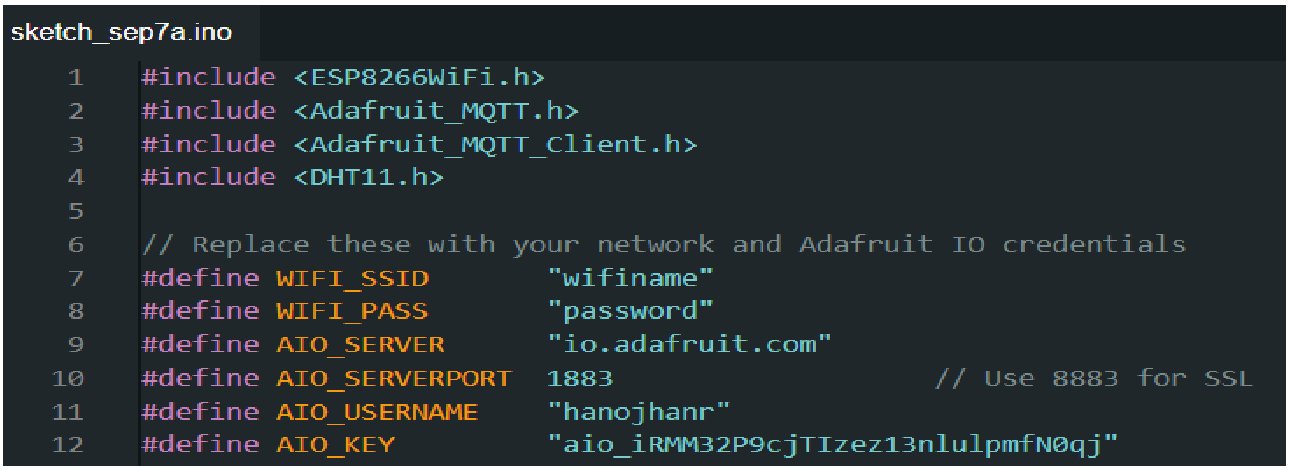
```
#include <ESP8266WiFi.h>

#include <Adafruit_MQTT.h>

#include <Adafruit_MQTT_Client.h>

#include <DHT11.h>



// Replace these with your network and Adafruit IO credentials

#define WIFI_SSID       "wifiname"

#define WIFI_PASS       "password"

#define AIO_SERVER      "io.adafruit.com"

#define AIO_SERVERPORT  1883                  // Use 8883 for SSL

#define AIO_USERNAME    "hanojhanr"

#define AIO_KEY         "aio_iRMM32P9cjTIzez13nlulpmfN0qj"



#define DHTPIN 4    // DHT11 sensor data pin connected to ESP8266 to pin 4

DHT11 dht11(DHTPIN);
```

```
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);


// Setup feeds for temperature and humidity

Adafruit_MQTT_Publish temperatureFeed = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/temperature");

Adafruit_MQTT_Publish humidityFeed = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/humidity");


void setup() {

Serial.begin(115200);

delay(10);


  // Connect to Wi-Fi

  Serial.println(); Serial.println();

  Serial.print("Connecting to ");

  Serial.println(WIFI_SSID);


  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());

  // Connect to Adafruit IO
  connectToAdafruitIO();

  // Initialize the DHT sensor
  // dht11.begin();  // Not needed for DHT11 library
}

void loop() {
  // Ensure the connection to Adafruit IO is maintained
  if (mqtt.connected()) {
    mqtt.processPackets(10000);
    mqtt.ping();
  } else {
    connectToAdafruitIO();
  }

  int temperature = 0;
  int humidity = 0;

  // Read temperature and humidity from DHT11
  int result = dht11.readTemperatureHumidity(temperature, humidity);
  // Check if any reads failed
  if (result == 0) {
    // Publish the data
    Serial.print("Publishing temperature: "); Serial.println(temperature);
    if (!temperatureFeed.publish(temperature)) {
      Serial.println("Failed to publish temperature");
    }

    Serial.print("Publishing humidity: "); Serial.println(humidity);
    if (!humidityFeed.publish(humidity)) {
      Serial.println("Failed to publish humidity");
    }
  } else {
    Serial.print("Failed to read from DHT sensor! Error: ");
    Serial.println(DHT11::getErrorString(result));
  }

  // Wait for a few seconds before sending the next reading
  delay(2000);
}

void connectToAdafruitIO() {
  int8_t ret;
  // Attempt to connect to Adafruit IO
  Serial.print("Connecting to Adafruit IO... ");
  while ((ret = mqtt.connect()) != 0) {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying in 5 seconds...");
    mqtt.disconnect();
    delay(1000);
  }
  Serial.println("Adafruit IO connected!");
}
```

7) Arduino IDE -> Tools -> Select correct Port (If port is not detected, download and install CH340 driver. Use this link [CH340 Drivers for Windows, Mac and Linux (gogo.co.nz)] or Google and download)

8) Arduino IDE -> Tools -> Manage Libraries… -> Search and Install ESP8266WiFi, Adafruit_MQTT and DHT11

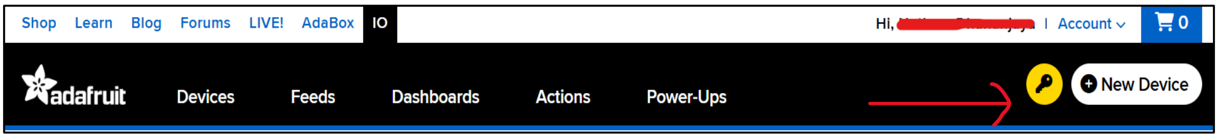
9) Change WIFI_SSID and WIFI_PASS in your code to your Wi-Fi or mobile hotspot (Only Android works) ID and Password.



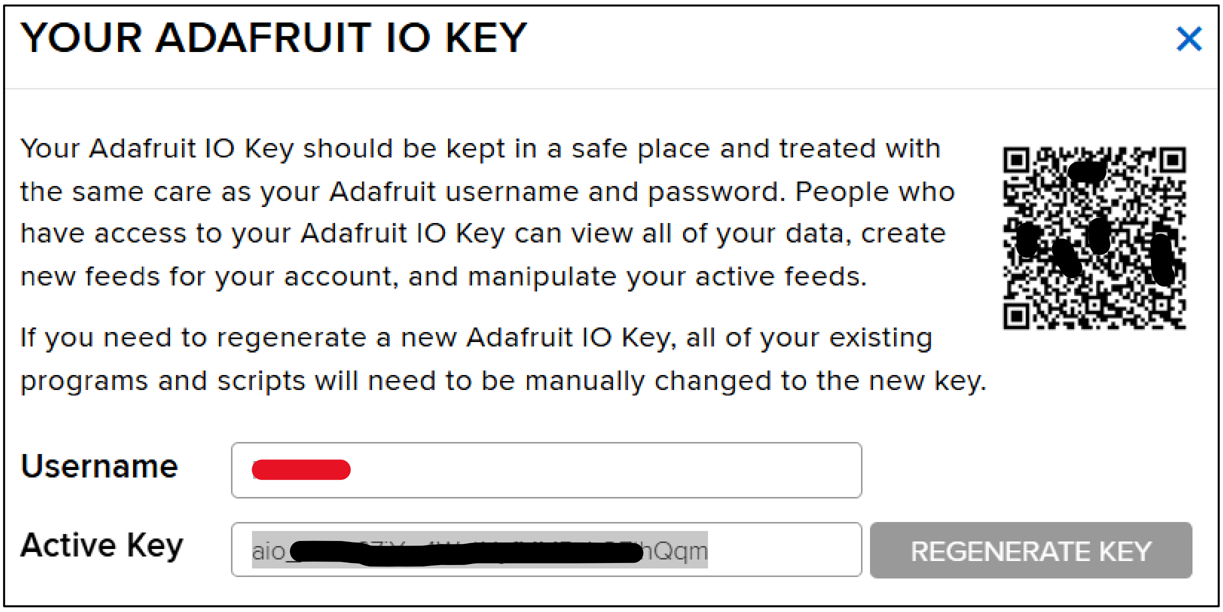
sketch_sep7a.ino

```
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <DHT11.h>

// Replace these with your network and Adafruit IO credentials
#define WIFI_SSID       "wifiname"
#define WIFI_PASS       "password"
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                  // Use 8883 for SSL
#define AIO_USERNAME    "hanojhanr"
#define AIO_KEY         "aio_iRMM32P9cjTIzez13nlulpmfN0qj"
```

10) Goto Adafruit -> IO -> Click API Key



11) Change AIO_USERNAME and AIO_KEY in your code to your Adafruit Username and Active Key.



YOUR ADAFRUIT IO KEY ✕

Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

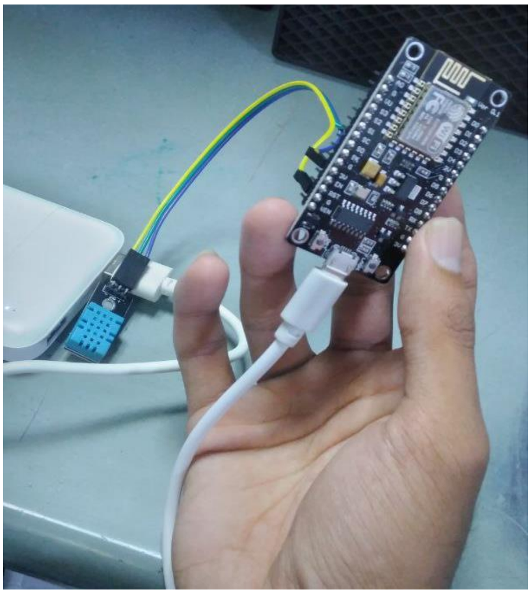
Username [redacted]

Active Key aio_[redacted]hQqm

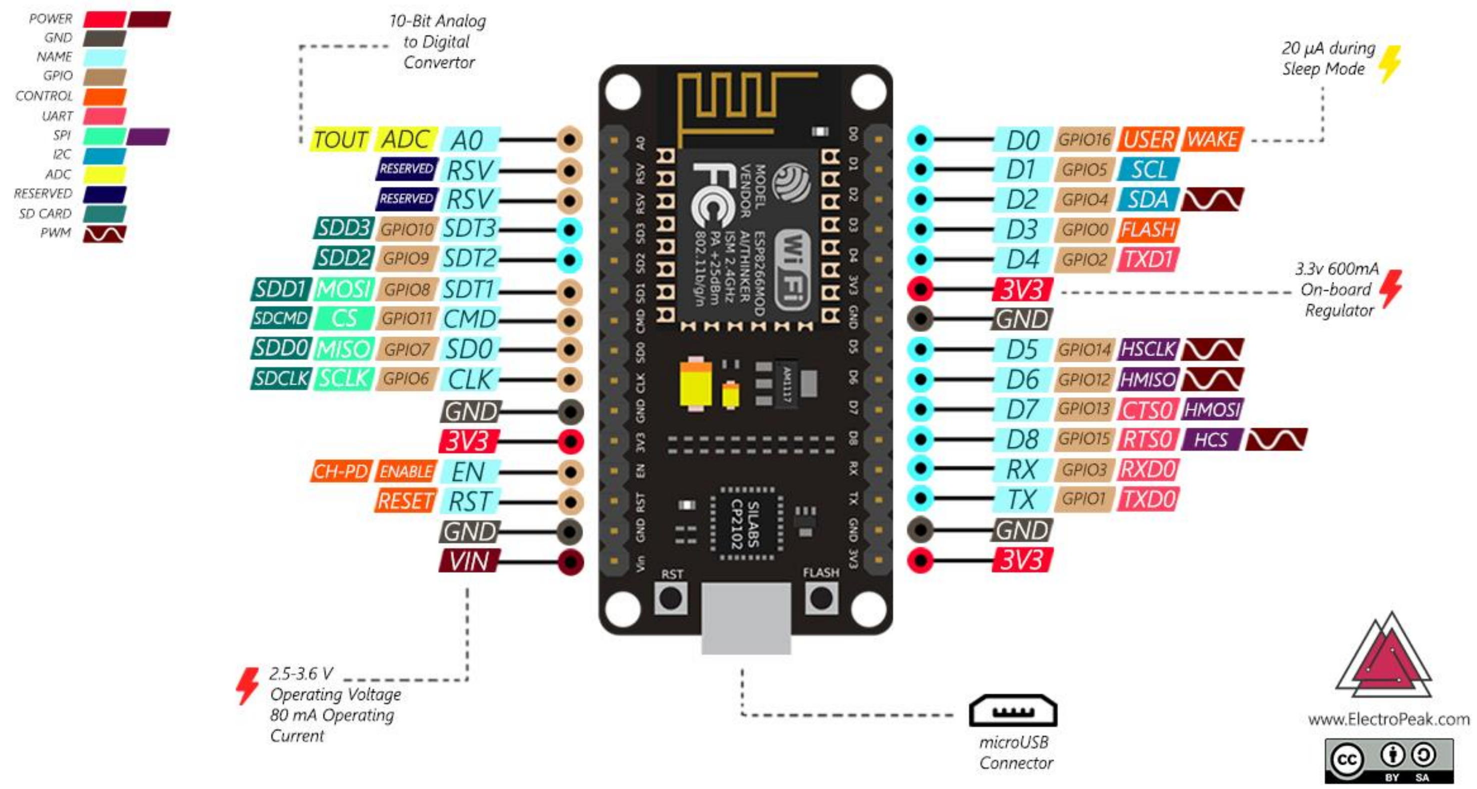
REGENERATE KEY

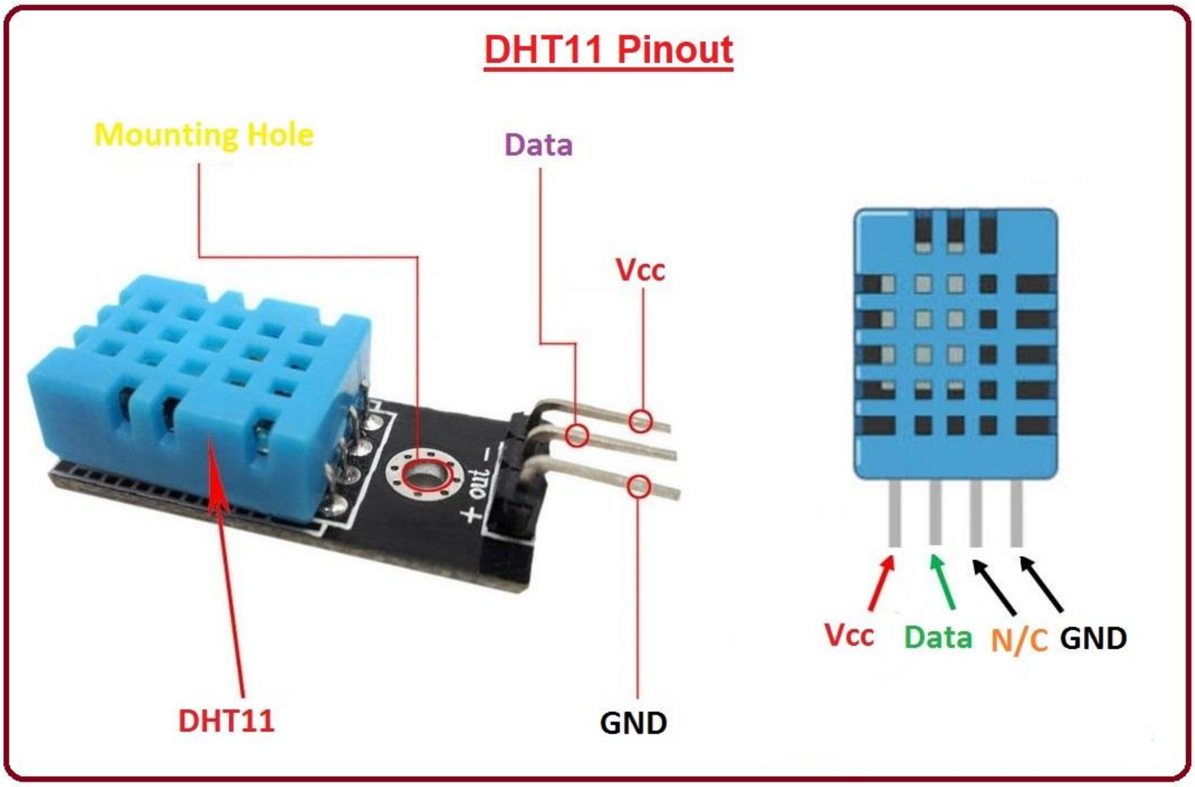
12) Tools -> Serial Monitor -> Change to 115200baud

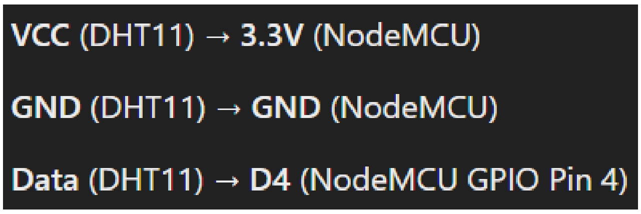
13) Upload the code to NodeMCU and monitor the output from Serial Monitor and readings for temperature and humidity will be printed, and the data will be published to Adafruit IO.



- We can use NodeMCU without connecting to the computer.
- NodeMCU is connecting to a powerbank.






**VCC** (DHT11) → **3.3V** (NodeMCU)

**GND** (DHT11) → **GND** (NodeMCU)

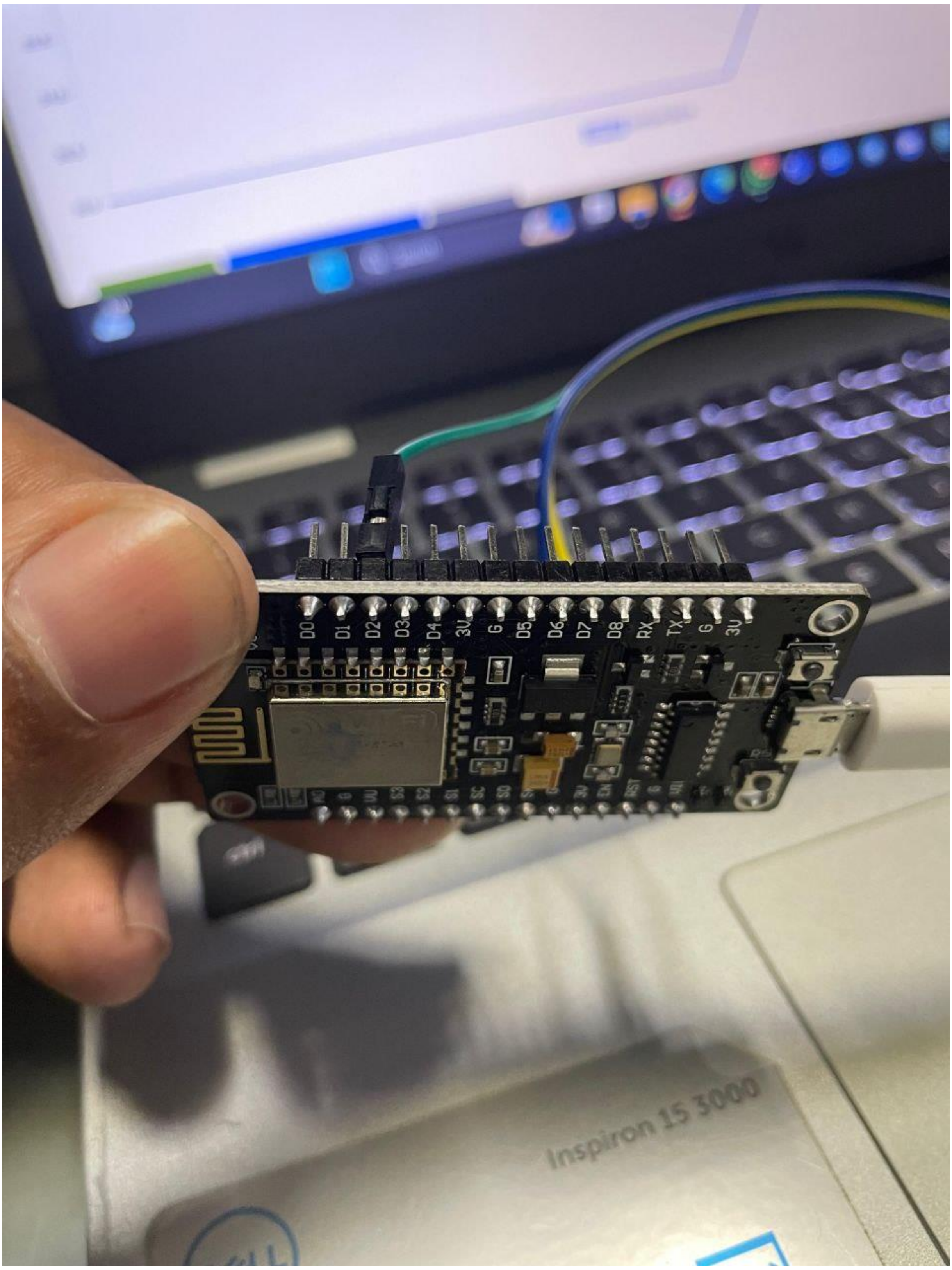
**Data** (DHT11) → **D4** (NodeMCU GPIO Pin 4)

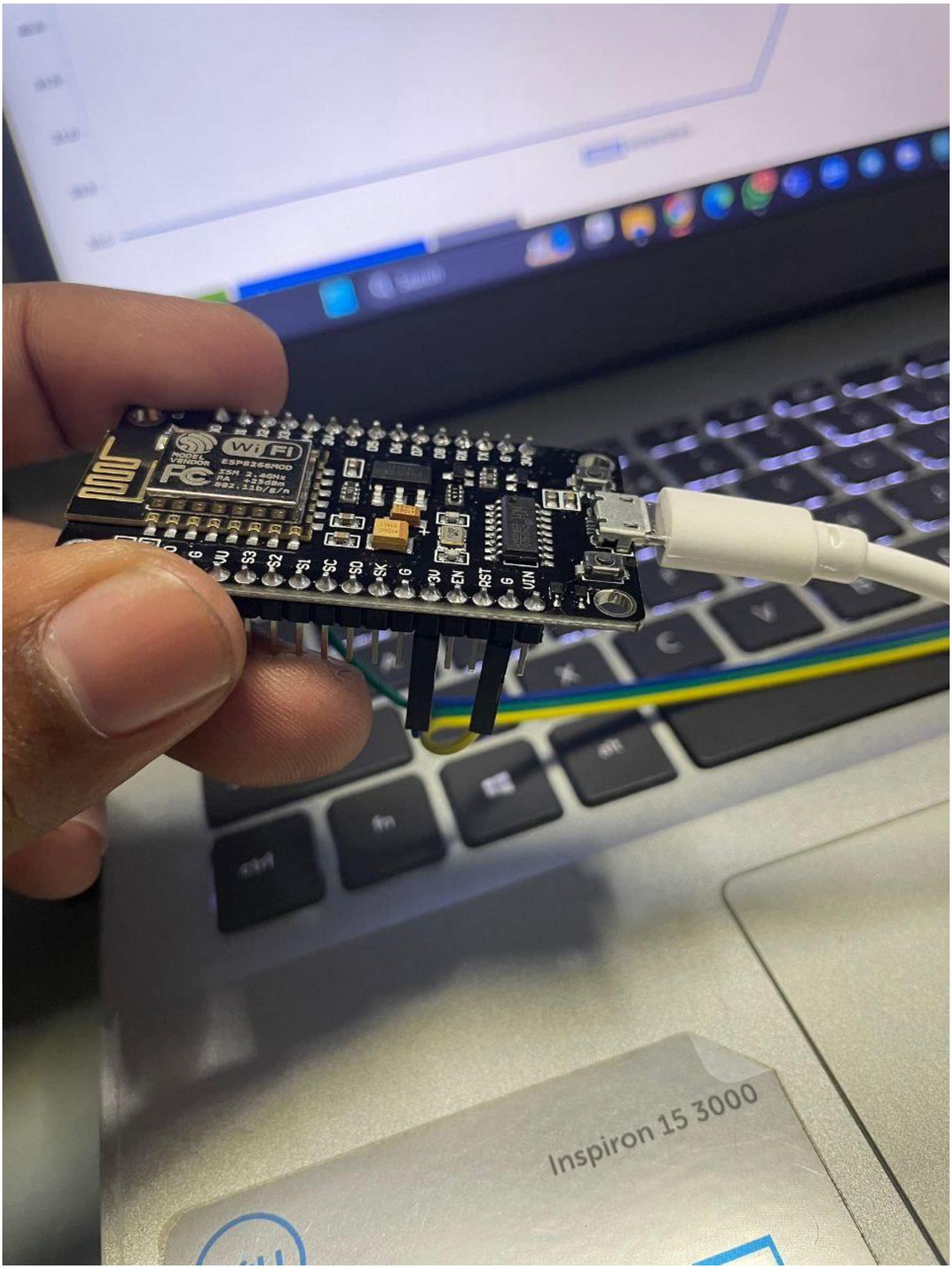
**DHT11** -> **NodeMCU**

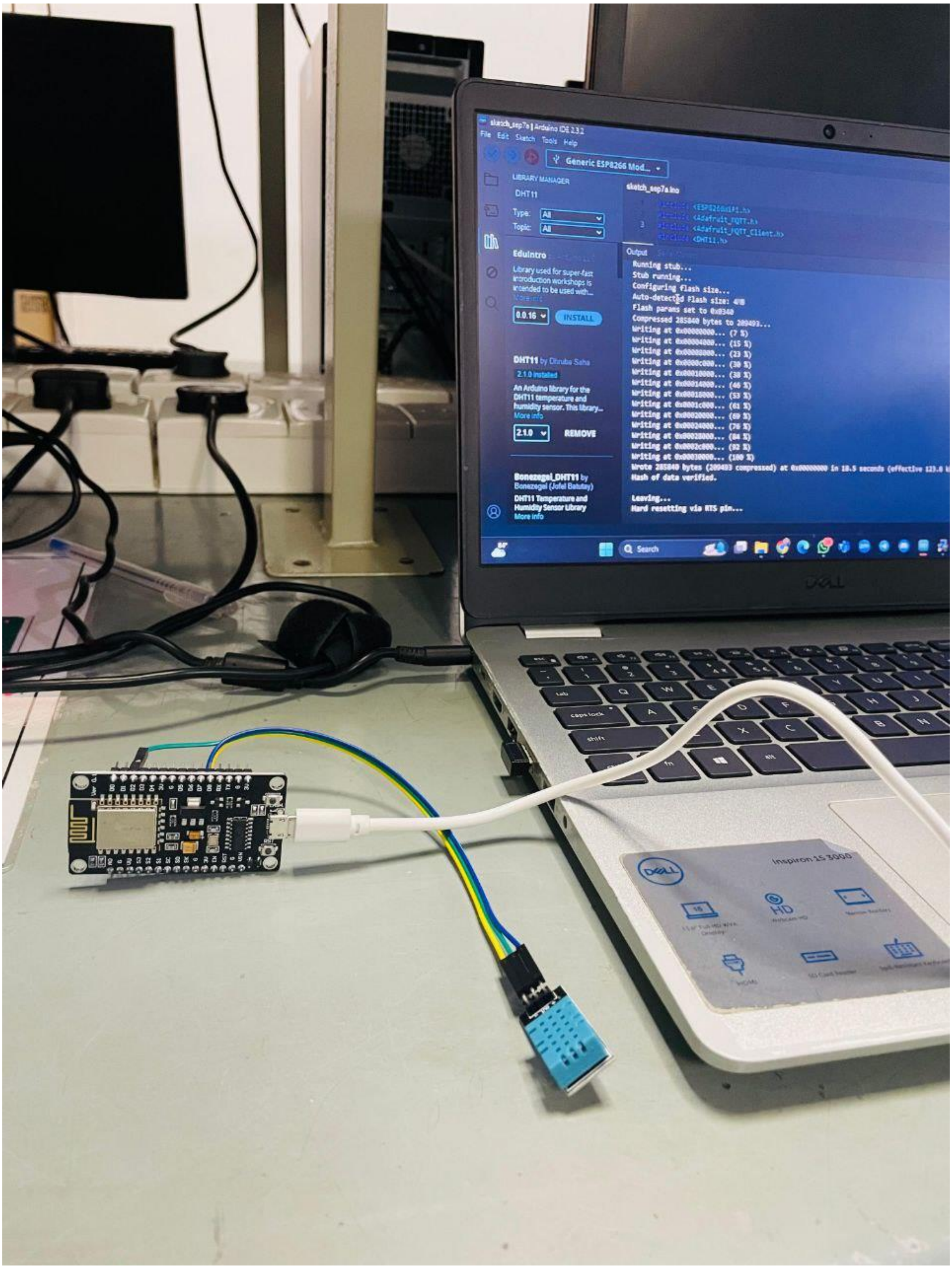
+ -> 3V

- -> G

Out -> D2


D0
D1
D2
D3
D4
3V
G
D5
D6
D7
D8
RX
TX
G
3V
Inspiron 15 3000


MODEL
VENDOR
WiFi
ESP8266MOD
ISM 2.4GHz
PA +25dBm
802.11b/g/n
D5
D6
D7
D8
RX
TX
G
3V
S3
S2
SI
SC
SO
SK
G
3V
EN
RST
G
VIN
Inspiron 15 3000



EduIntro

Library used for super-fast introduction workshops is intended to be used with...

DHT11 by Dhruba Saha

An Arduino library for the DHT11 temperature and humidity sensor. This library...
More info

Bonezegei_DHT11 by Bonezegei (Jofel Batutay)

DHT11 Temperature and Humidity Sensor Library
More info

```
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <DHT11.h>
```

```
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0340
Compressed 285840 bytes to 209493...
Writing at 0x00000000... (7 %)
Writing at 0x00004000... (15 %)
Writing at 0x00008000... (23 %)
Writing at 0x0000c000... (30 %)
Writing at 0x00010000... (38 %)
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 285840 bytes (209493 compressed) at 0x00000000 in 18.5 seconds (effective 123.8 k
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```