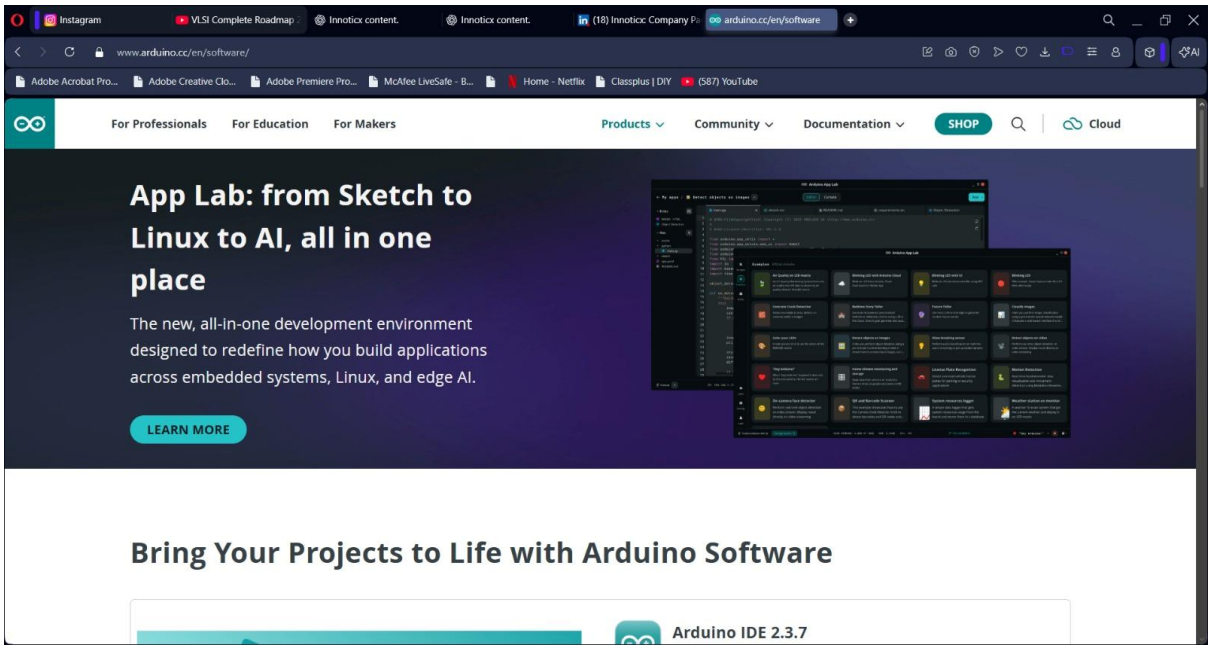
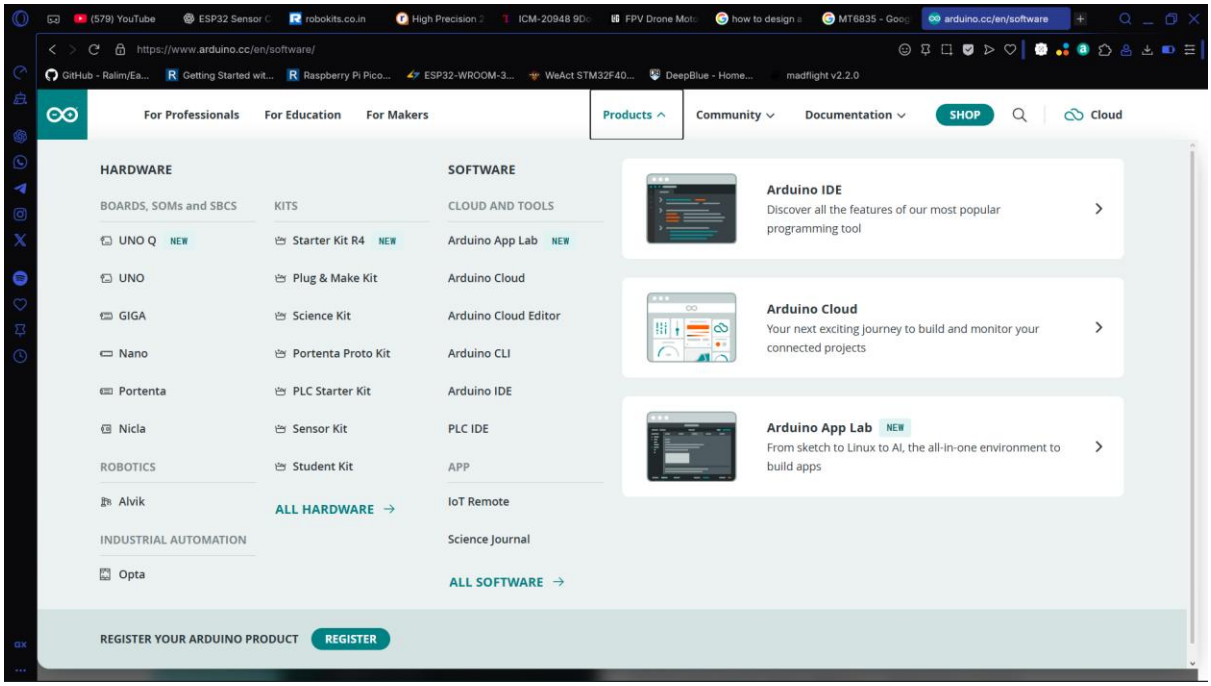
Installation of IDE

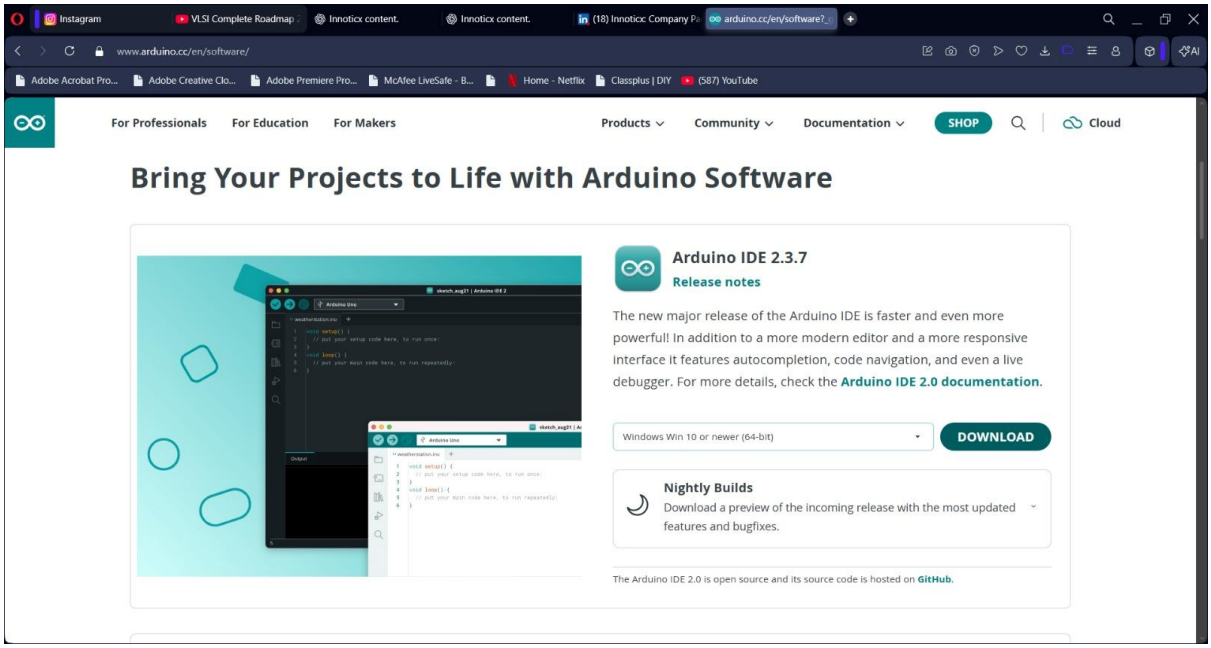
Step 1: Visit the Arduino Website



Step 2: Click on the products Tab on the navigation bar at top of the web page

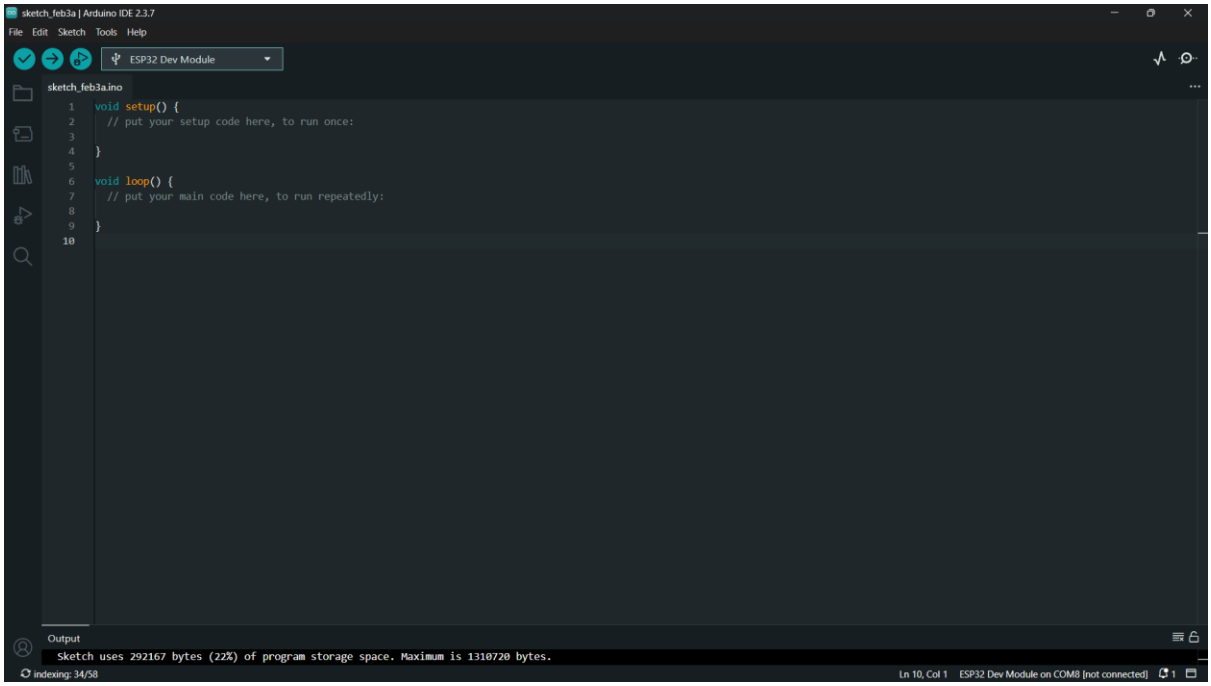


Step 3: Click on Arduino ide

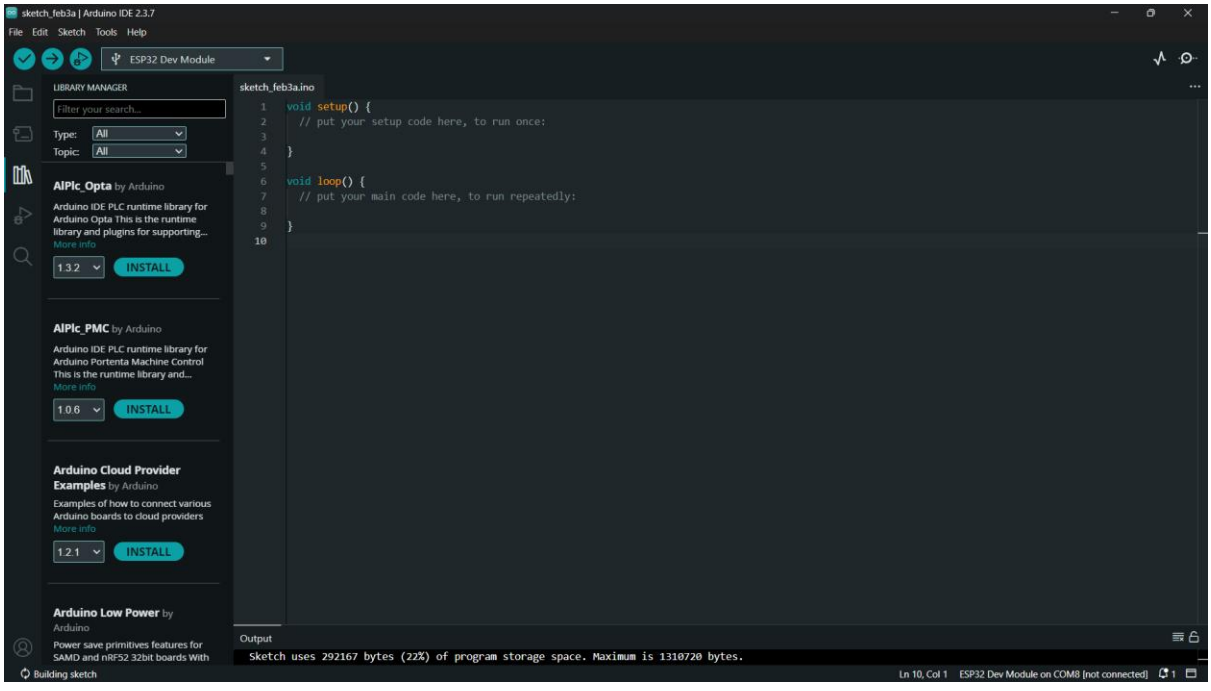


Step 4: Scroll down a little and download the installation file for your OS

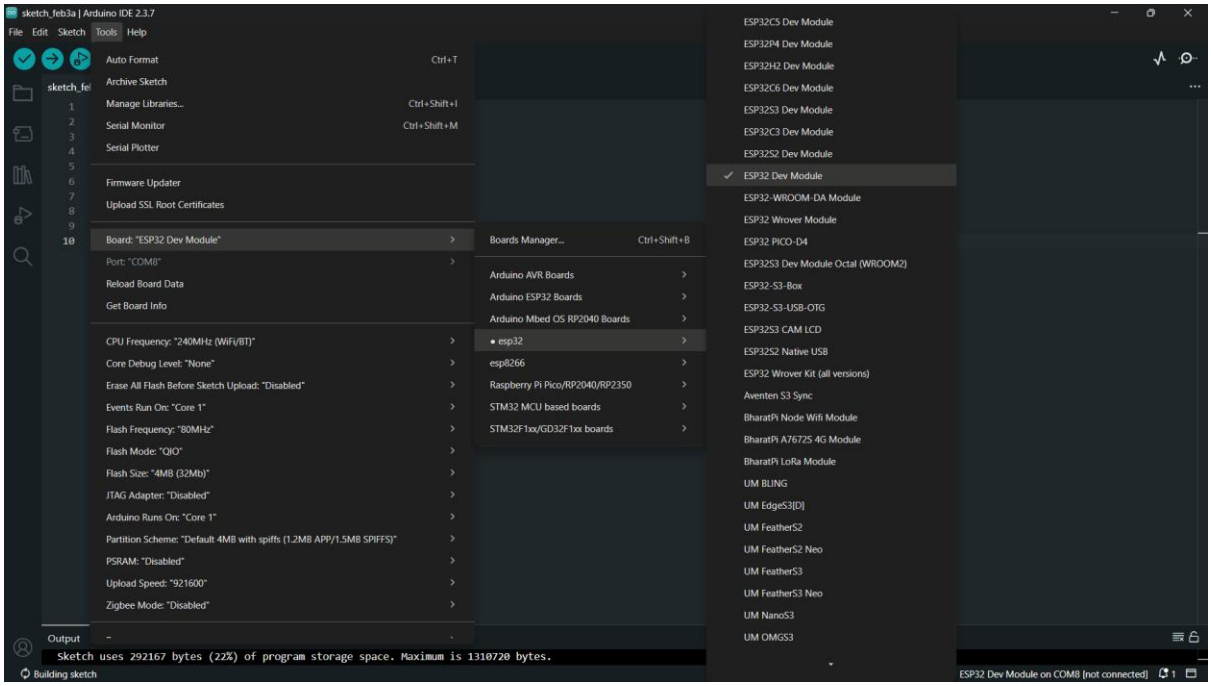
Introduction to IDE



This is How the Arduino IDE looks when you first open it on the top there are three blue buttons, The first Button which has a Checkmark on it the compile button the second button which has an arrow on it is used to upload the code to the dev Board

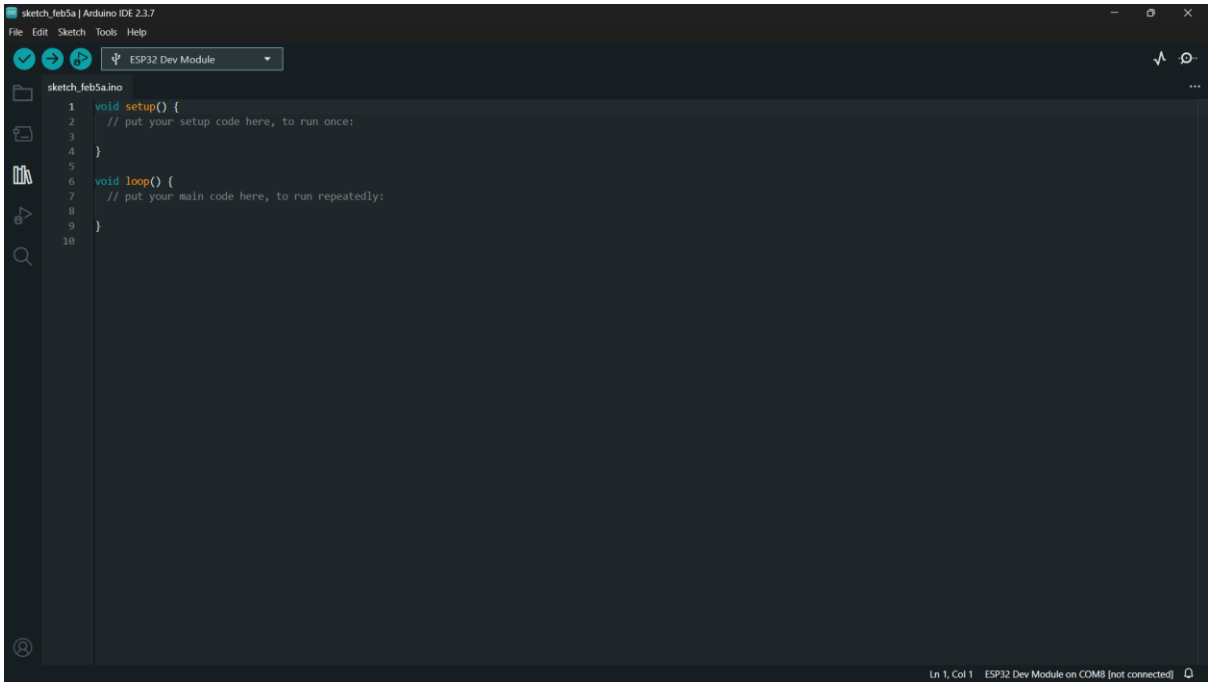


On the left hand side there are five buttons, The first button is the file manager in the IDE, the second button is used to install different Dev Boards in the IDE, the third button is used to install Different libraries in the IDE.

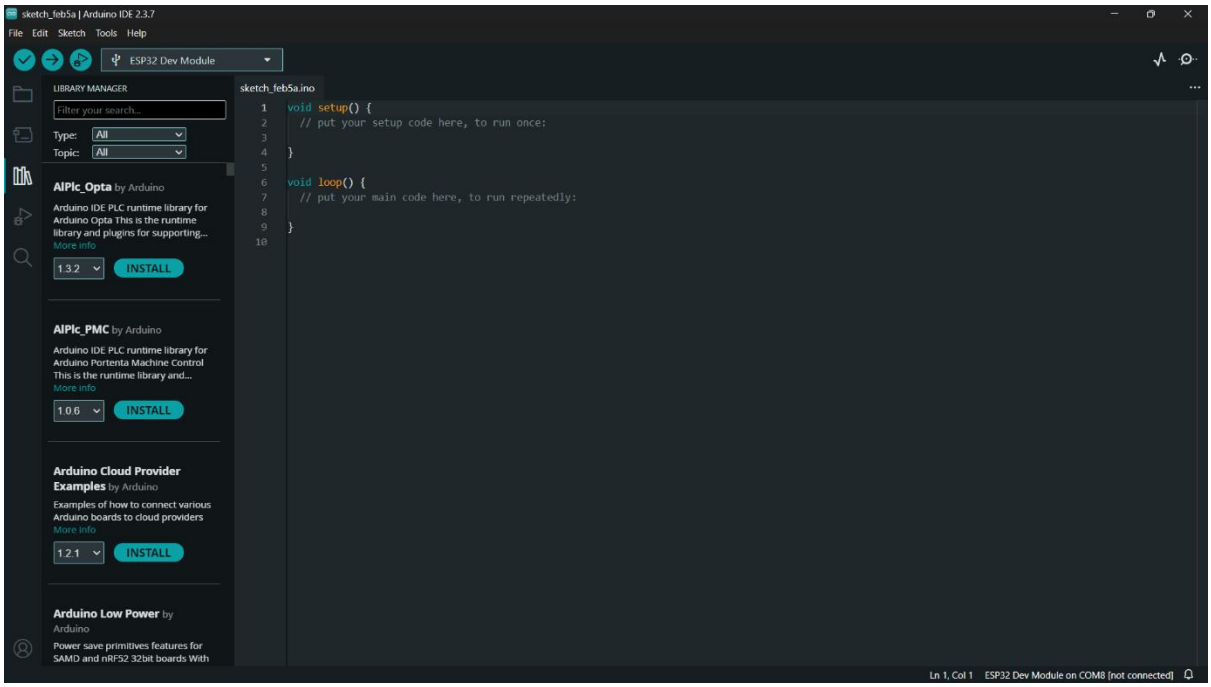


You would need to select the Exact board in the tools Tab you are trying to program and the COM Port to which the board is connected to.

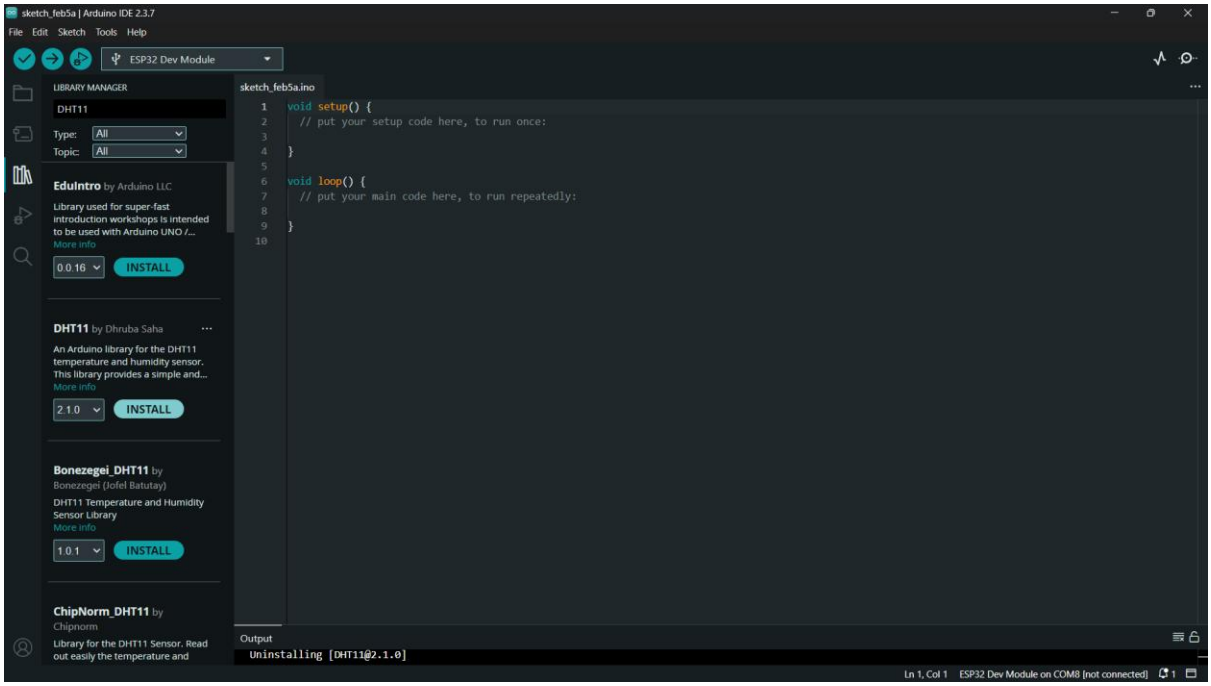
How to install a Library



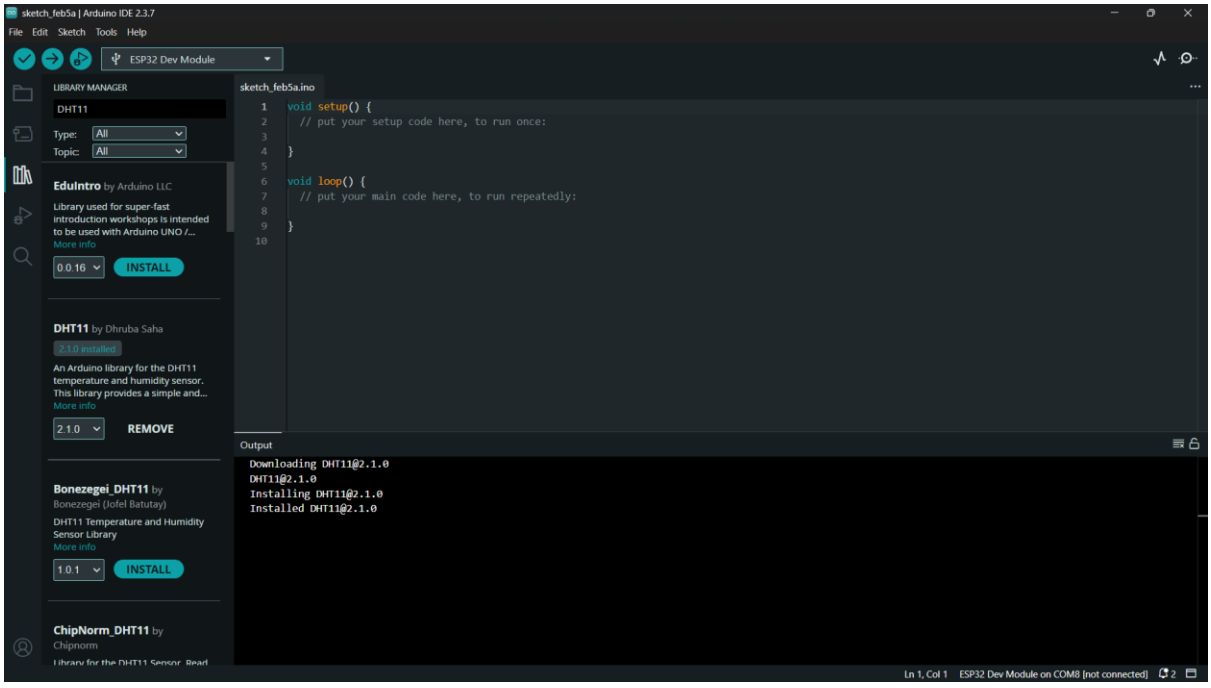
Step 1: In the Arduino Software click the library Manager Button on the Left hand side of the window



Step 2: The library manager Column appear. To install any library type its name in the search bar located on top of the library manager.

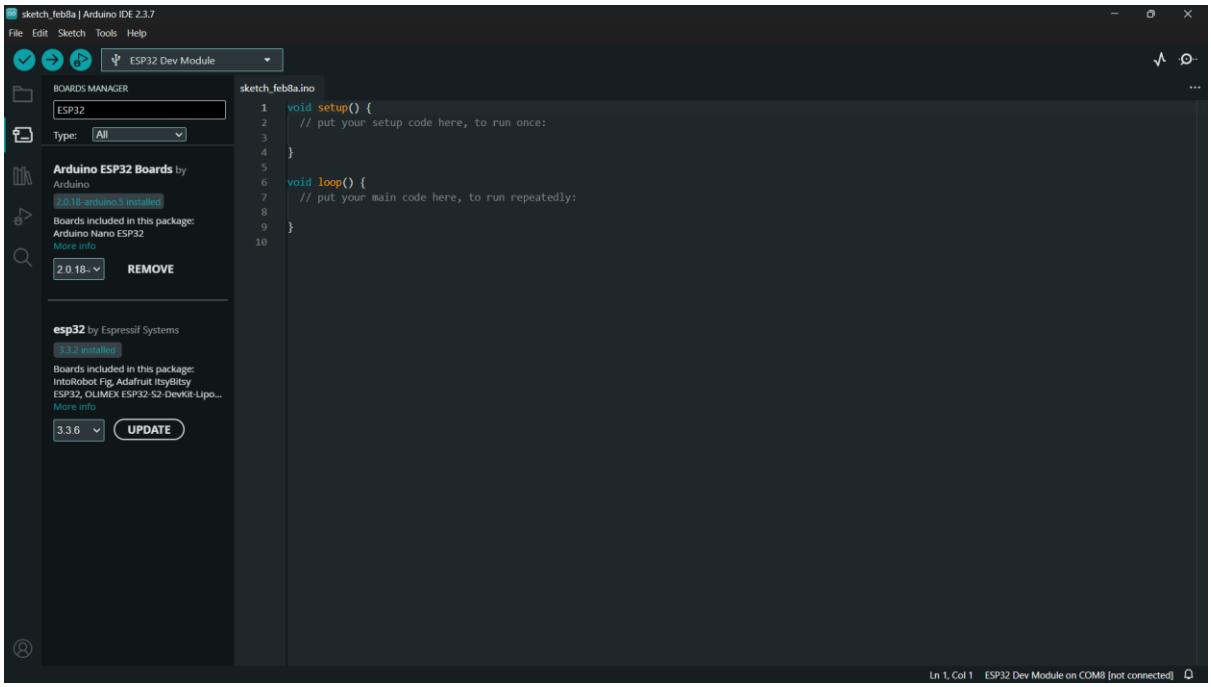


Step 3: For example lets search for DHT11 library, all the libraries related to DHT11 will appear, Scroll till you find the intended library and click the install button on it.



During the installation process you will see the that the terminal opens up and shows that the library is being installed. After installation the install button changes to remove button and a pop up appears saying that the library has been successfully installed.

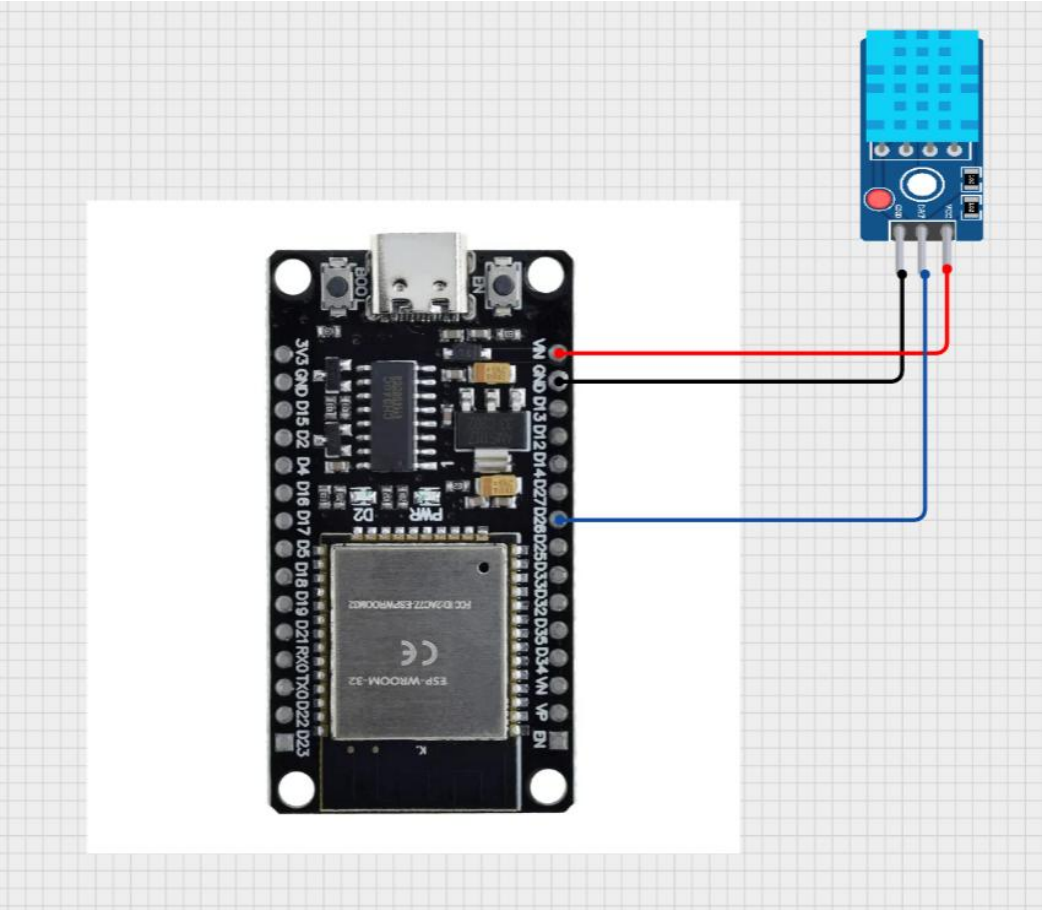
Installing ESP32 Boards in IDE



Step 1: Click on the Boards manager tab located on the left hand side column just above Library manager

Step 2: In the boards manager search menu type ESP32 and install the ESP32 by Espressif Systems.

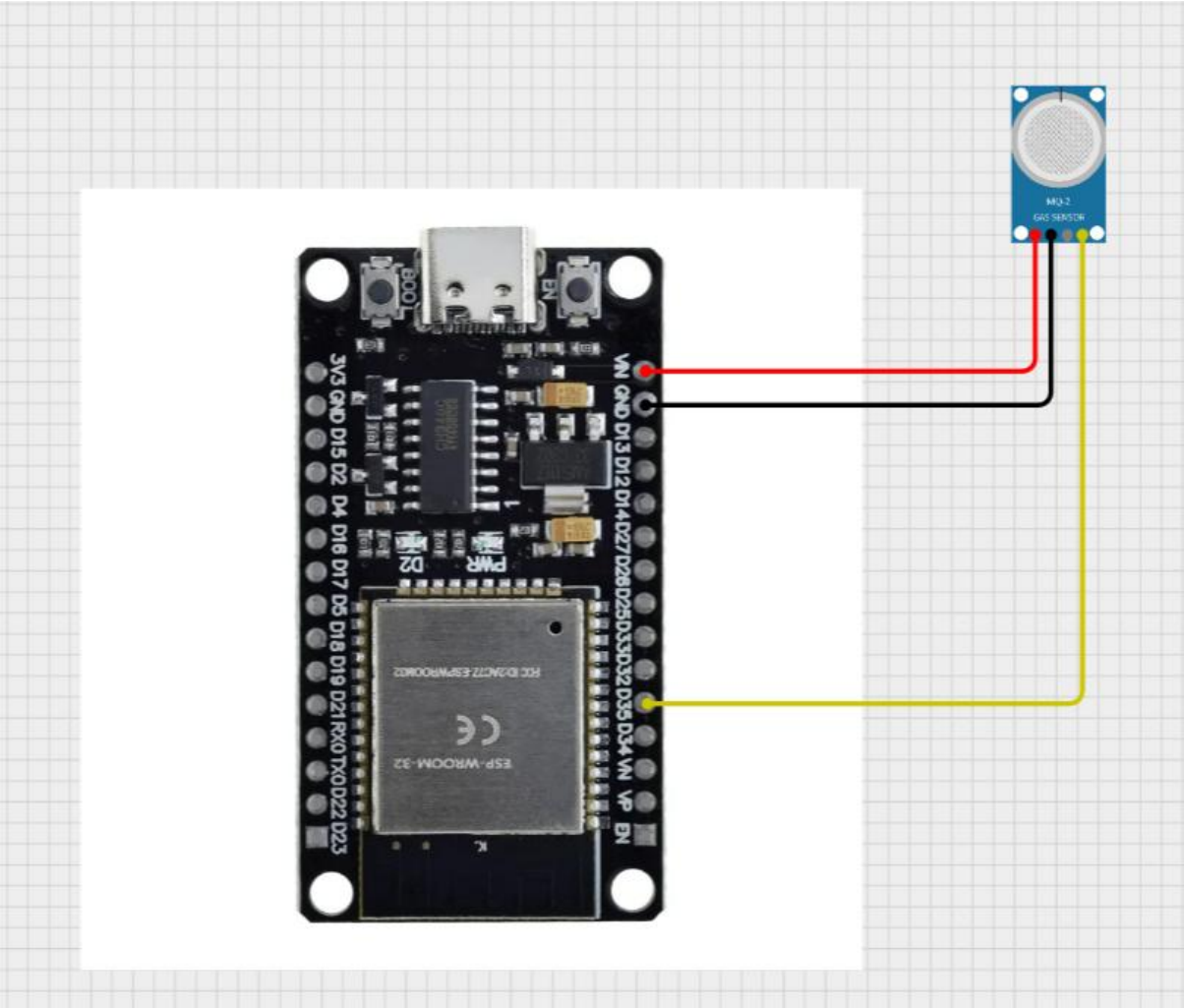
DHT11


BOOT
EN
3V3 GND D15 D2 D4 D16 D17 D5 D18 D19 D21 RX0 TX0 D22 D23
VN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
PWR
ESP-WROOM-32
GND
DAT
VCC

dht11_temperature_sensor.ino

```ino
#include <DHT.h>  // Include the DHT sensor library

// ------- Pin & Type Configuration -------
#define DHT_DATA_PIN   26   // The ESP32 pin where the DHT11 DATA pin is connected
#define DHTTYPE        DHT11 // Define the type of DHT sensor being used (DHT11)

// Create a DHT sensor object using the pin and sensor type
DHT dht(DHT_DATA_PIN, DHTTYPE);

void setup() {
  Serial.begin(9600);   // Start serial communication at 9600 baud rate

  delay(1000);          // Small delay to allow sensor and serial to stabilize
  dht.begin();          // Initialize the DHT11 sensor

  Serial.println("DHT11 Test Started"); // Print a message to indicate setup is done
}

void loop() {
  // Read humidity and temperature values from DHT11 sensor.
  // These functions return NAN if the reading fails.
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // Default is Celsius

  // Check if either reading failed
  if (isnan(h) || isnan(t)) {
    Serial.println("DHT11 Read Error"); // Message if the sensor failed to read
  } else {
    // Print the temperature and humidity values
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.print(" °C | Humidity: ");
    Serial.print(h);
    Serial.println(" %");
  }

  delay(2000); // Wait 2 seconds before the next reading
}
```

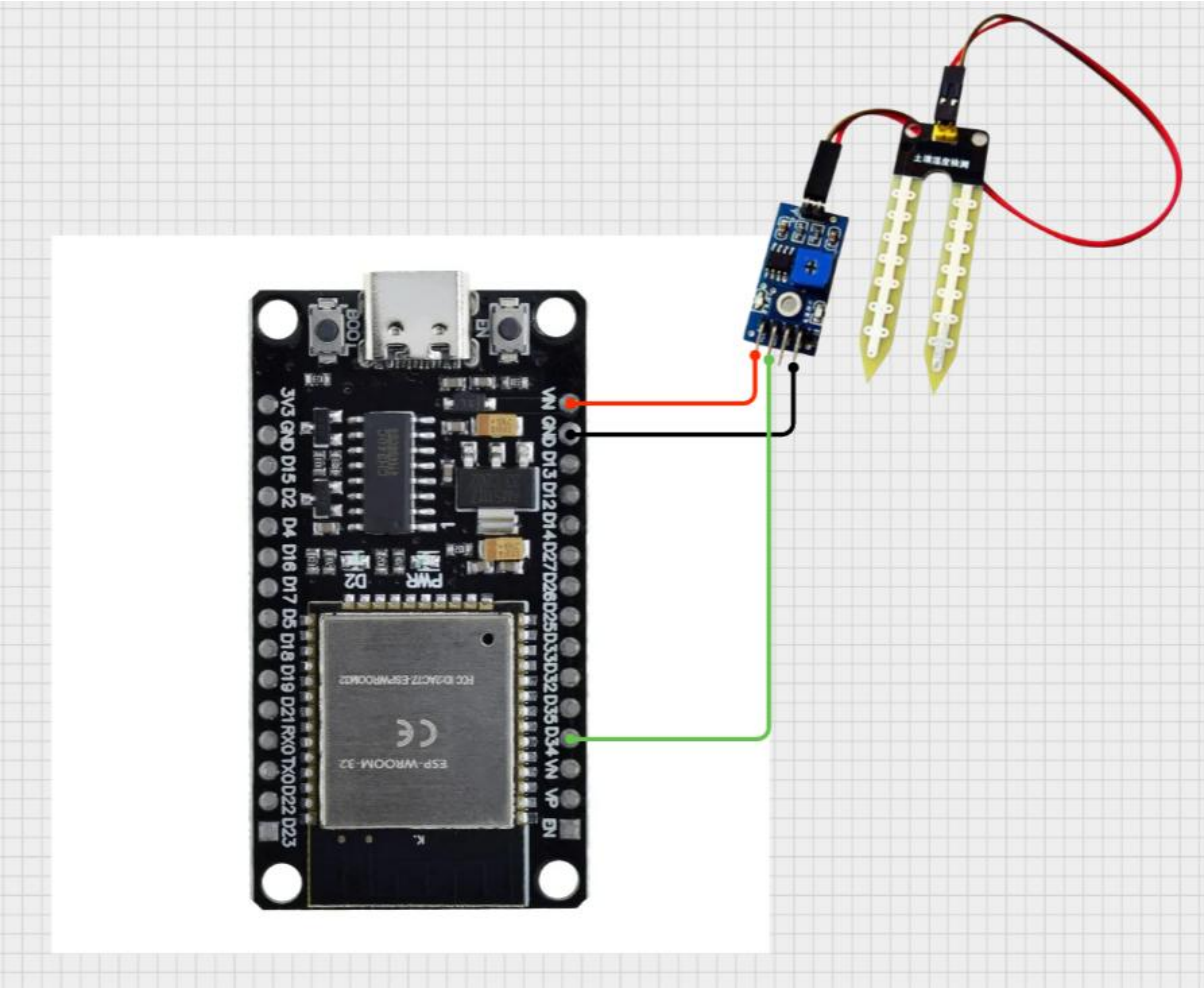
MQ-2 Gas Sensor



mq-2_gas_sensor.ino

```ino
#define MQ2_AO_PIN 35   // Define the ESP32 pin where MQ2 analog output (AO) is connected

void setup() {
  Serial.begin(9600);   // Start serial communication at 9600 baud rate
  delay(1000);          // Small delay to allow Serial to initialize properly

  // Initial message to the user
  Serial.println("MQ2 Gas Sensor Test Started");
  Serial.println("Warming up MQ2... wait 2-3 minutes");
  // MQ2 requires a warm-up time for the heating element to stabilize
}

void loop() {
  // Read raw analog value from MQ2 AO pin (0-4095 on ESP32)
  int mq2Raw = analogRead(MQ2_AO_PIN);

  // Print the raw sensor value
  // Higher value = less gas, lower value = more gas (depends on load resistor)
  Serial.print("MQ2 Gas Raw Value: ");
  Serial.println(mq2Raw);

  delay(2000);  // Wait 2 seconds before taking the next reading
}

```

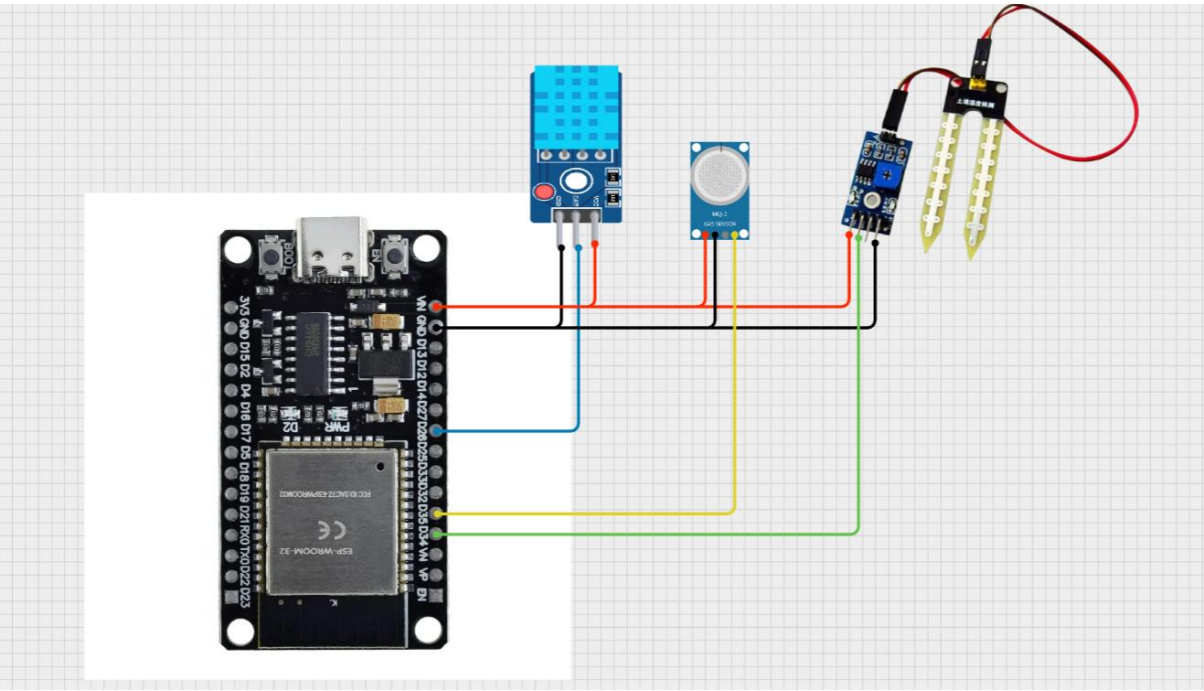
Soil Moisture sensor



soil_moisture_sensor.ino

```ino
#define SOIL_AO_PIN 34    // ESP32 pin connected to the soil moisture sensor's analog output (AO)

void setup() {
  Serial.begin(9600);     // Start serial communication at 9600 baud rate

  delay(500);             // Small delay to allow serial communication to stabilize
  Serial.println("Soil Moisture Sensor Test Started");
}

void loop() {
  // Read the raw analog value from the soil moisture sensor
  // ESP32 ADC range = 0 to 4095
  int soilRaw = analogRead(SOIL_AO_PIN);

  // Print the raw analog value to the Serial Monitor
  Serial.print("Soil Moisture Raw Value: ");
  Serial.println(soilRaw);

  // Wait for 2 seconds before the next reading
  delay(2000);
}

```

Combined code



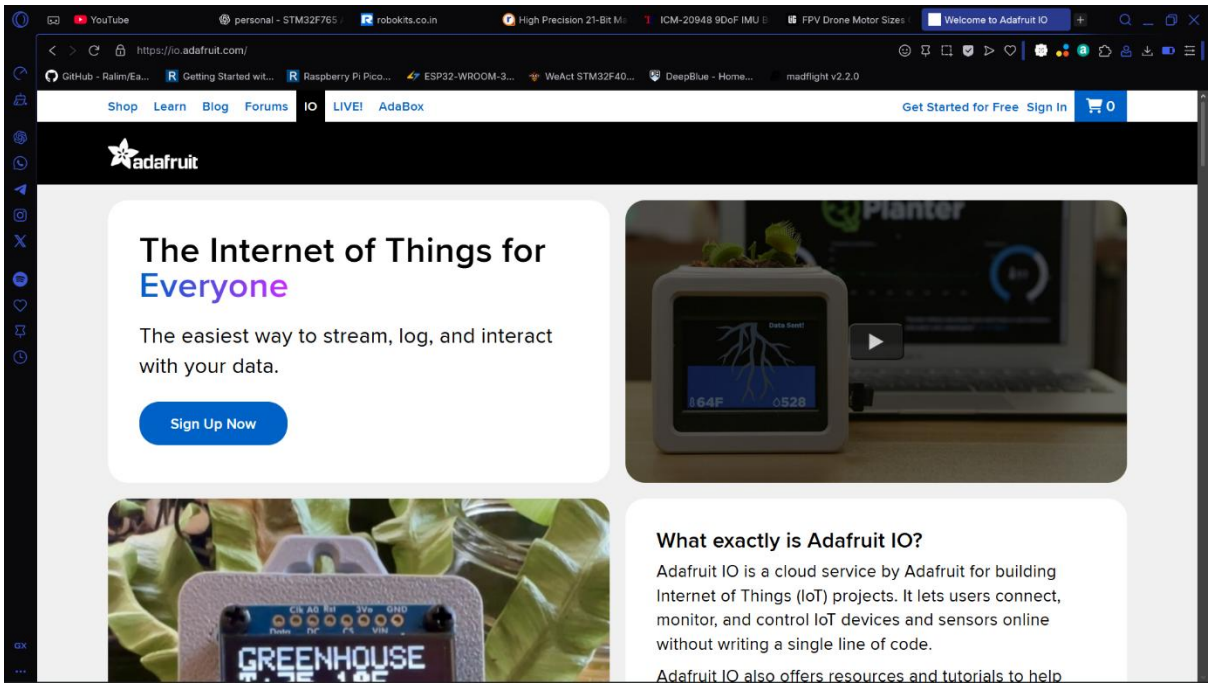
greenhouse_interfacing.ino

```ino
#include <DHT.h>  // Include the DHT library needed for temperature & humidity sensor

// ---------- PIN DEFINITIONS ----------

// DHT11 Sensor Pins
#define DHT_DATA_PIN   26     // Data pin connected to DHT11
#define DHTTYPE        DHT11  // Define the sensor type (DHT11)

// Soil Moisture Sensor Pin
#define SOIL_AO_PIN    34     // Analog output pin of soil moisture sensor

// MQ2 Gas Sensor Pin
#define MQ2_AO_PIN     35     // Analog output pin of MQ2 gas sensor

// Create the DHT object using pin 26 and sensor type DHT11
DHT dht(DHT_DATA_PIN, DHTTYPE);

void setup() {
  Serial.begin(9600);     // Start serial communication at 9600 baud rate
  delay(500);            // Small delay for Serial Monitor to initialize

  dht.begin();            // Initialize the DHT11 sensor

  // Print startup message
  Serial.println("ESP32 Sensor Interface Started");
  Serial.println("DHT11 | Soil Moisture | MQ2 Active");
  Serial.println("MQ2 warm-up: wait 2-3 minutes for stable readings");
}

void loop() {

  // -------- READ DHT11 SENSOR --------
  float humidity = dht.readHumidity();         // Read humidity (%)
  float temperature = dht.readTemperature();   // Read temperature (Celsius)

  // -------- READ SOIL MOISTURE SENSOR --------
  int soilRaw = analogRead(SOIL_AO_PIN);  // Read raw analog value (0-4095)

```

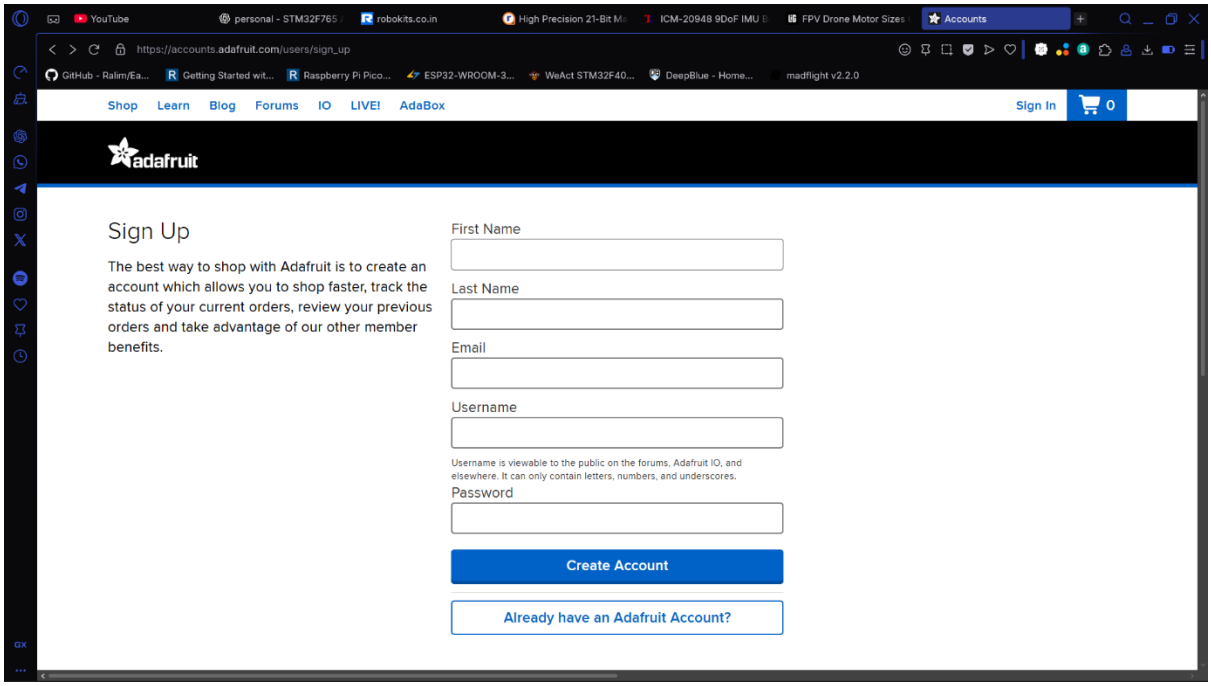
```
  // Convert the raw soil moisture reading to a percentage
  // These numbers (3500 = dry, 1200 = wet) must be calibrated for YOUR sensor
  int soilPercent = map(soilRaw, 3500, 1200, 0, 100);
  soilPercent = constrain(soilPercent, 0, 100); // Force value within 0–100%

  // -------- READ MQ2 GAS SENSOR --------
  int mq2Raw = analogRead(MQ2_AO_PIN);   // Read raw gas sensor value (0–4095)

  // -------- PRINT SENSOR READINGS TO SERIAL --------
  Serial.println("--------------------------------");

  // Print DHT11 values
  if (isnan(humidity) || isnan(temperature)) {  // Check if reading failed
    Serial.println("DHT11 Read Error — Check wiring");
  } else {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C | Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");
  }

  // Print soil moisture values
  Serial.print("Soil Moisture Raw: ");
  Serial.print(soilRaw);
  Serial.print(" | Moisture: ");
  Serial.print(soilPercent);
  Serial.println(" %");

  // Print MQ2 gas sensor value
  Serial.print("MQ2 Gas Raw Value: ");
  Serial.println(mq2Raw);

  delay(2000); // DHT11 requires at least 2 seconds between readings
}
```

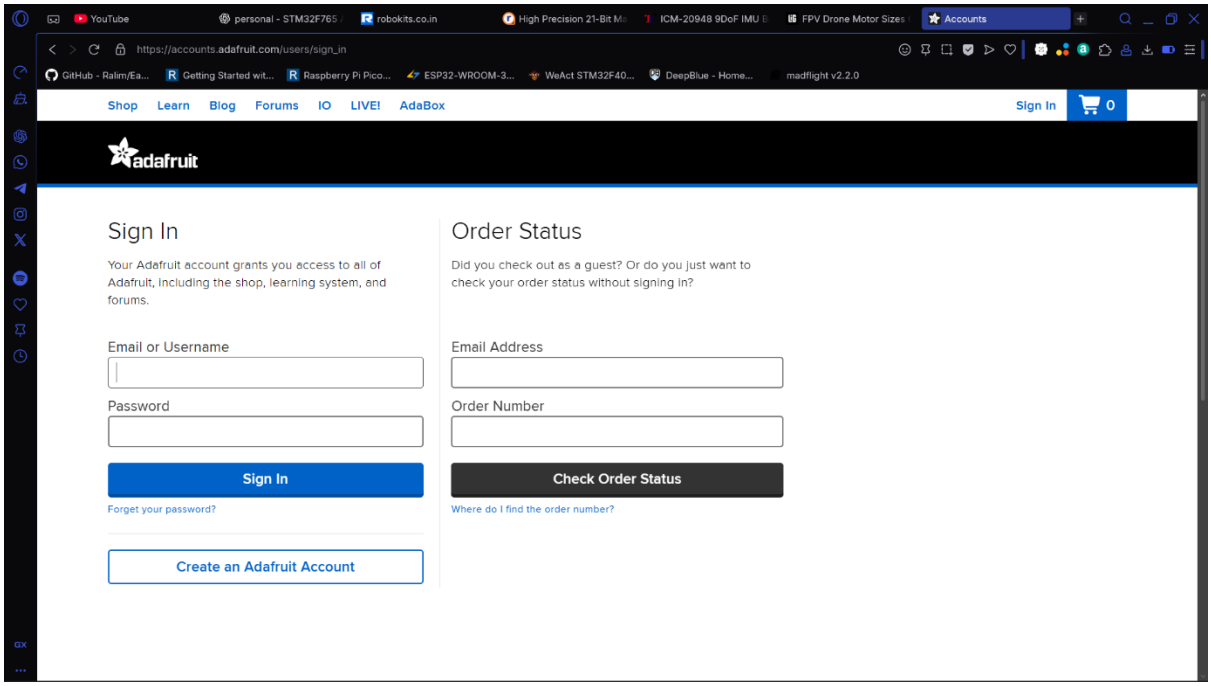
Adafruit IO Setup:

Step 1: Visit the Adafruit IO website

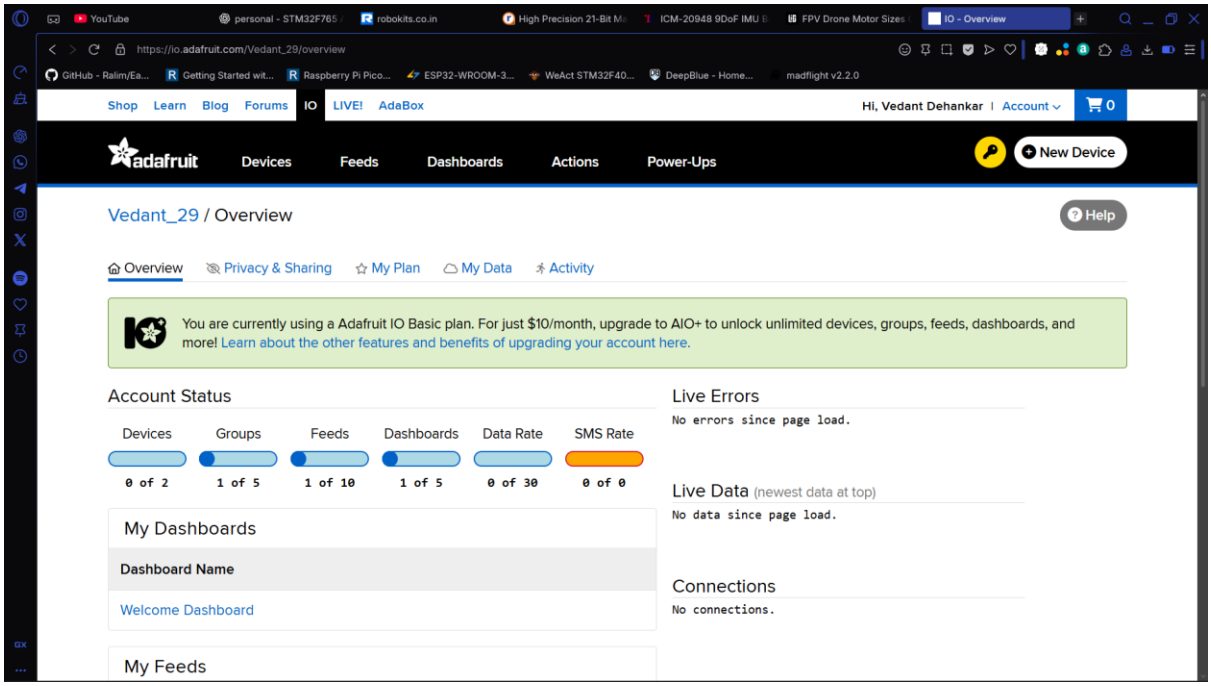


Step 2: In the Adafruit IO page Click on the Sign Up button.

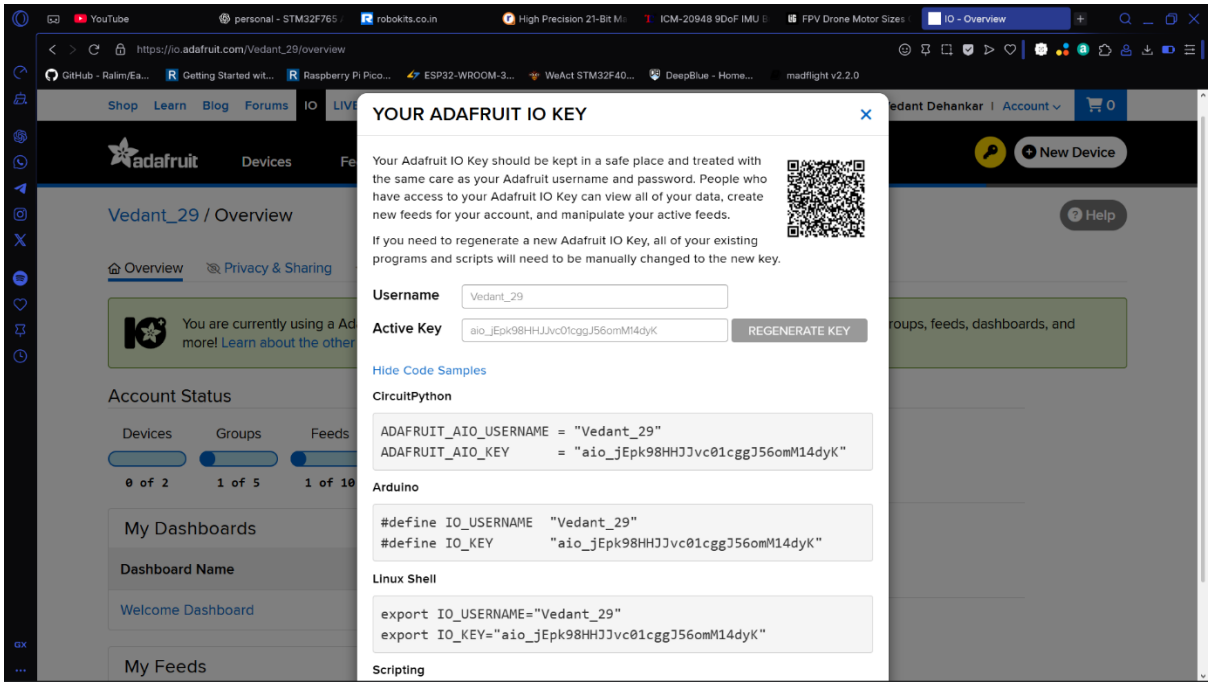




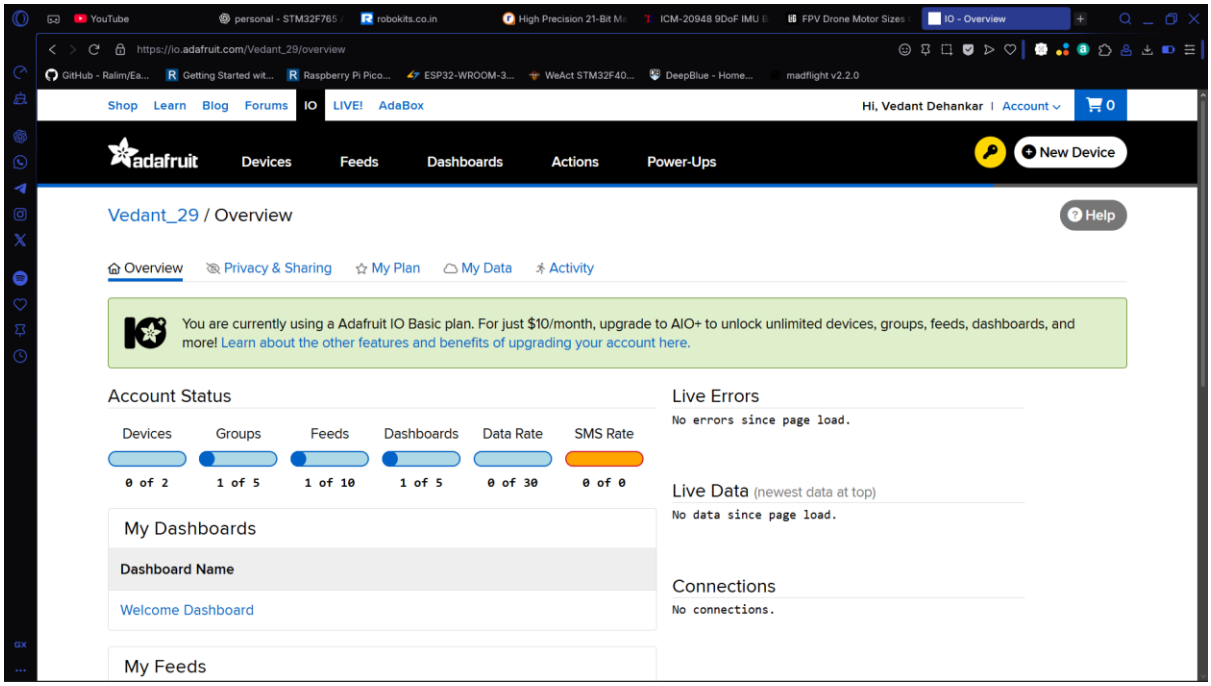
Step 3: Sign up for the website by filling out your details and clicking the Create Account Button, if you already have an account sign in into your account



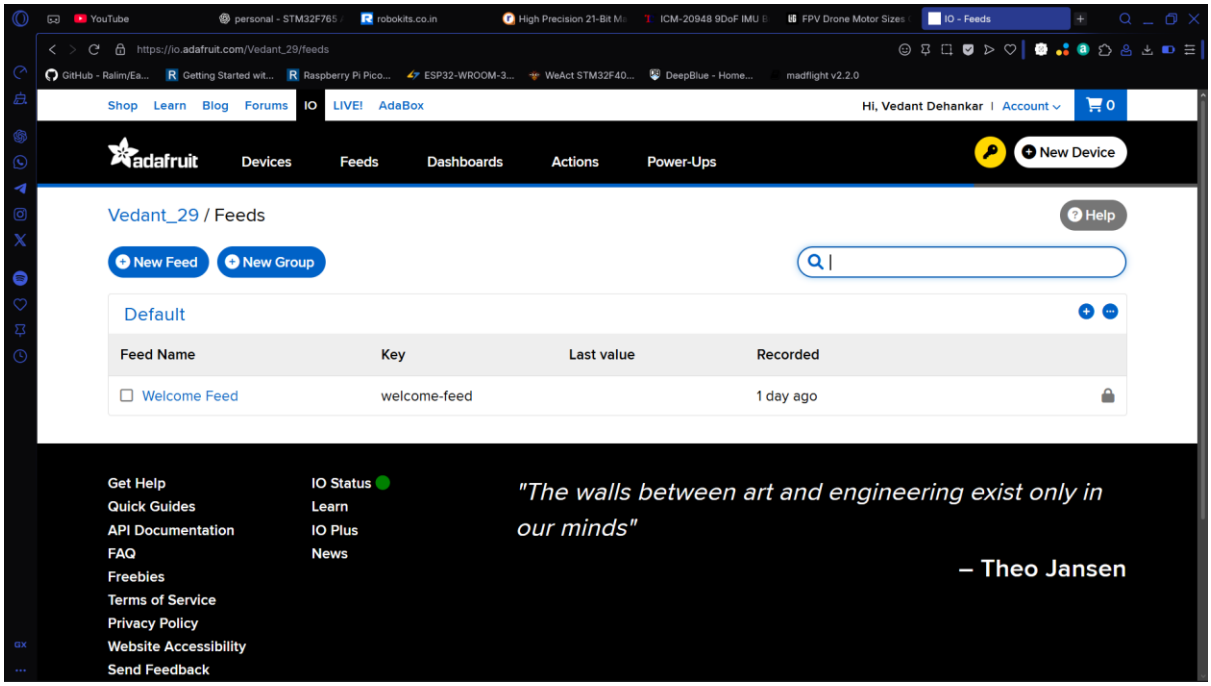
Step 4: After Signing In you will arrive in the Overview tab, In this tab click on the yellow key icon on the navigation bar.



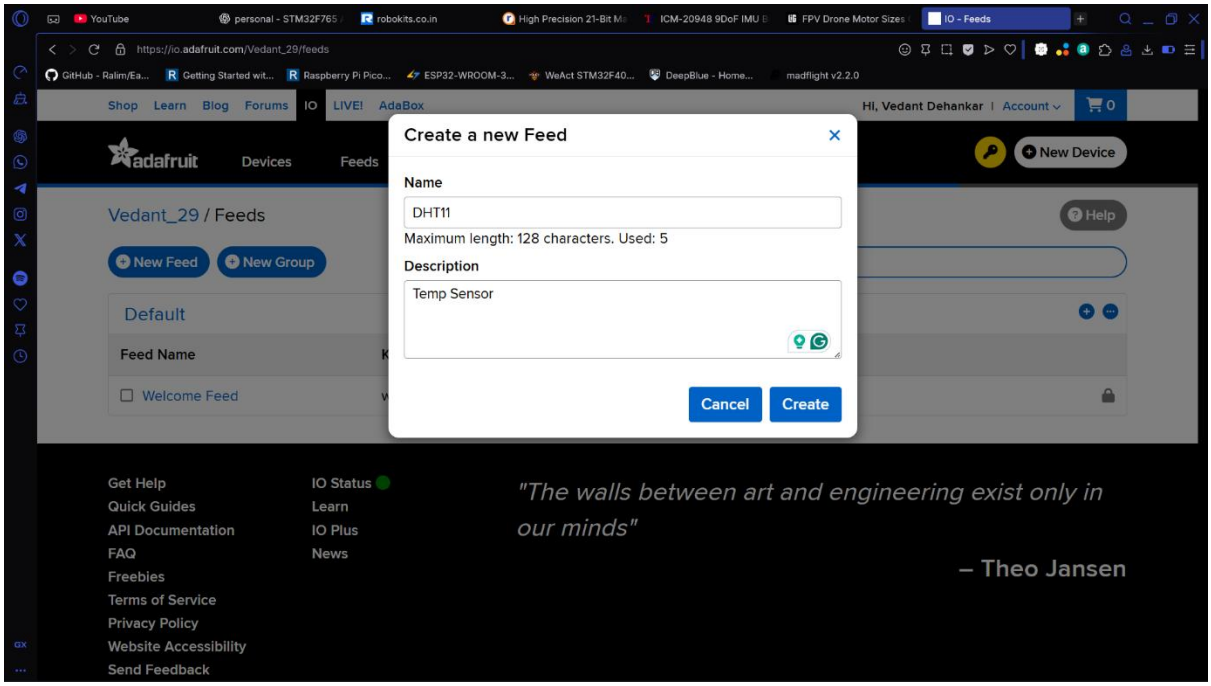
Step 5: An API key pop up will open in that copy the API key from Arduino section, It will go into your code.



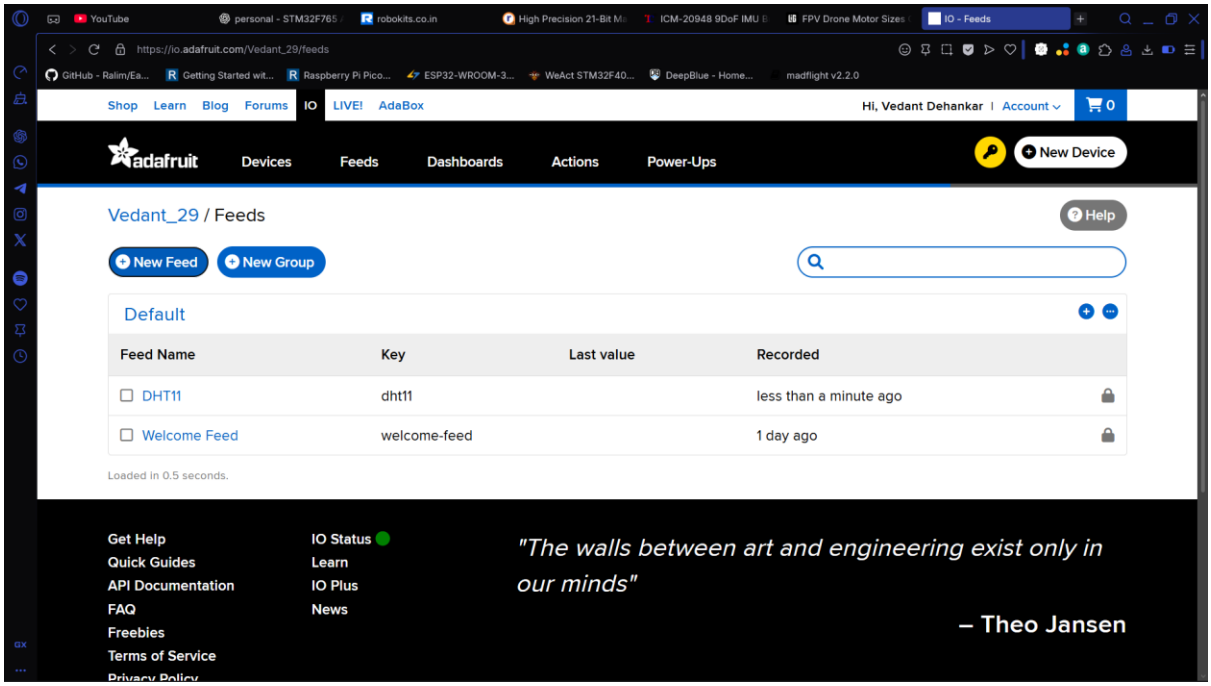
Step 6: Go back to the over view window, in that click on the feeds tabs.

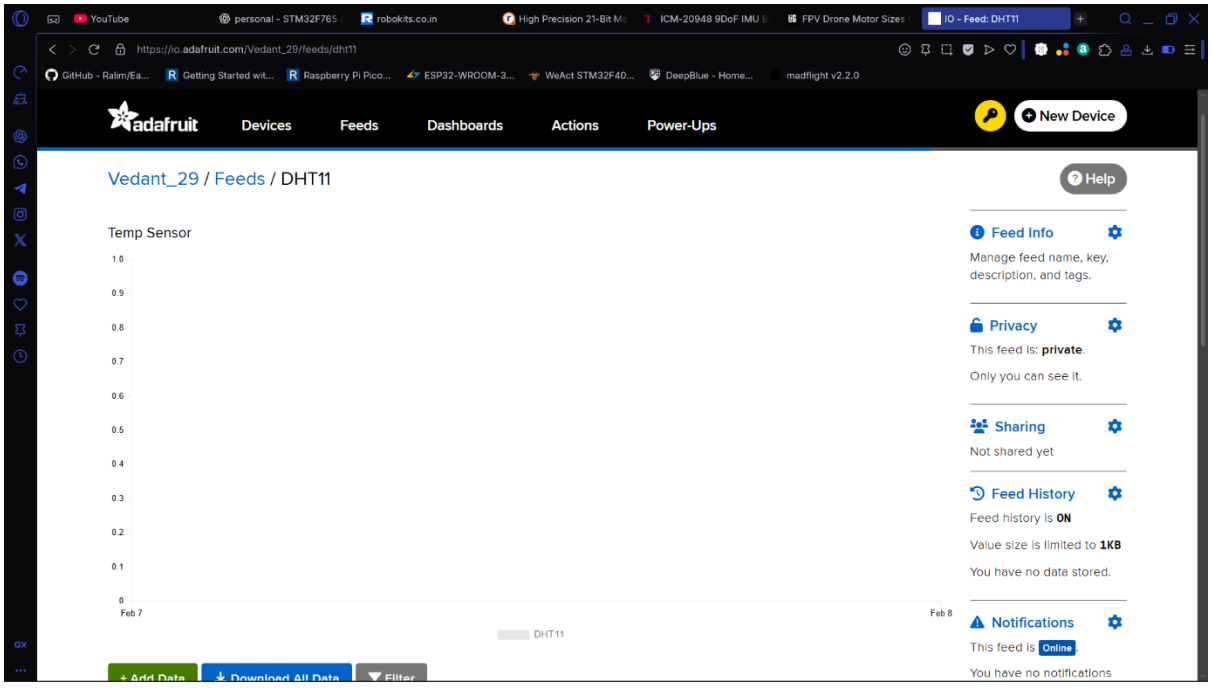


Step 7: You will be taken to the feeds window there click on the new feeds button.

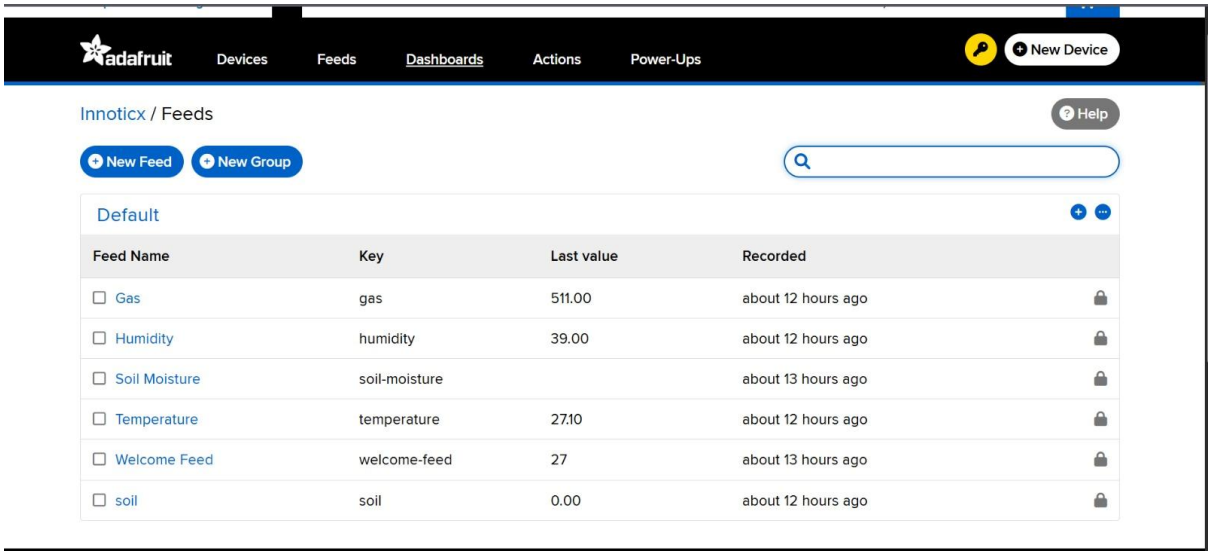


Step 8: A Create New Feeds pop up will appear in that give it a name and optionally describe it.

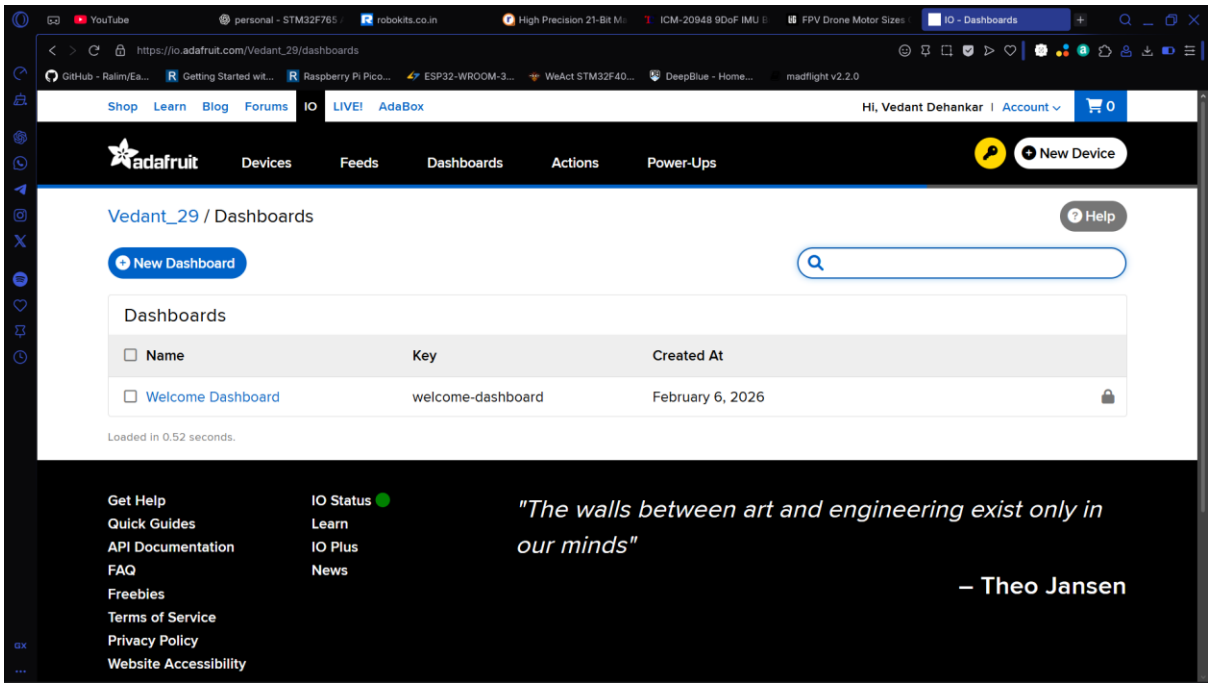




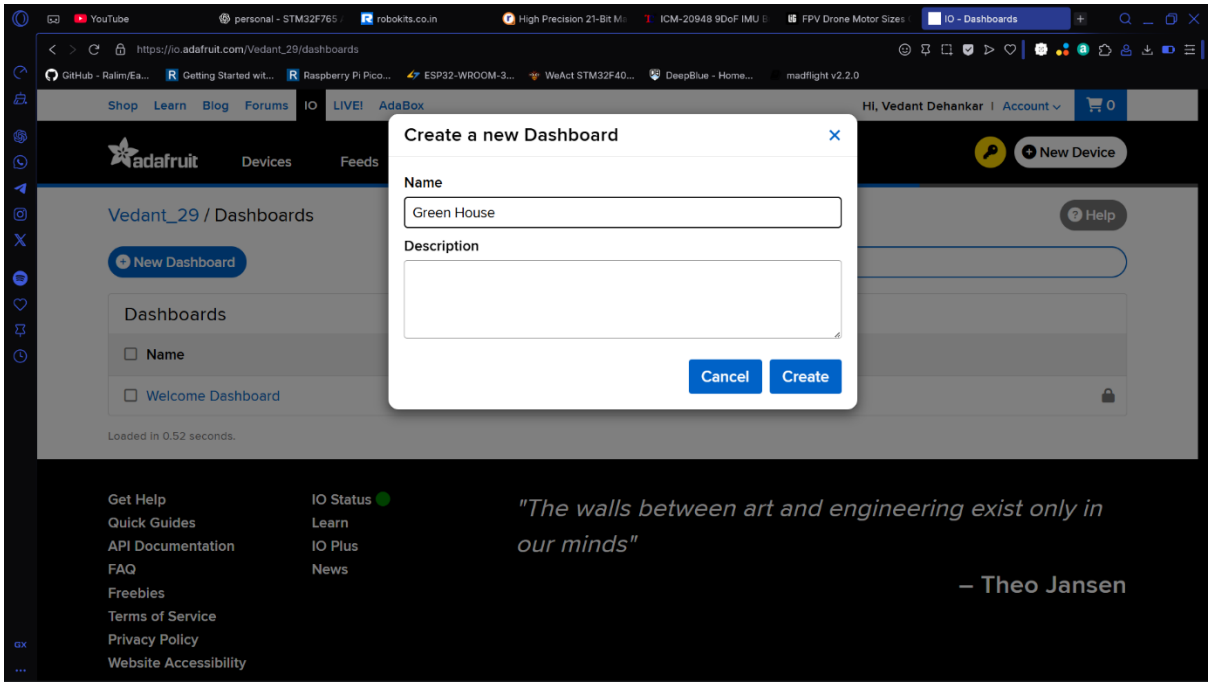
Step 9: A new feed will be created, When you click on the feed you will see a graphical view of the sensor readings



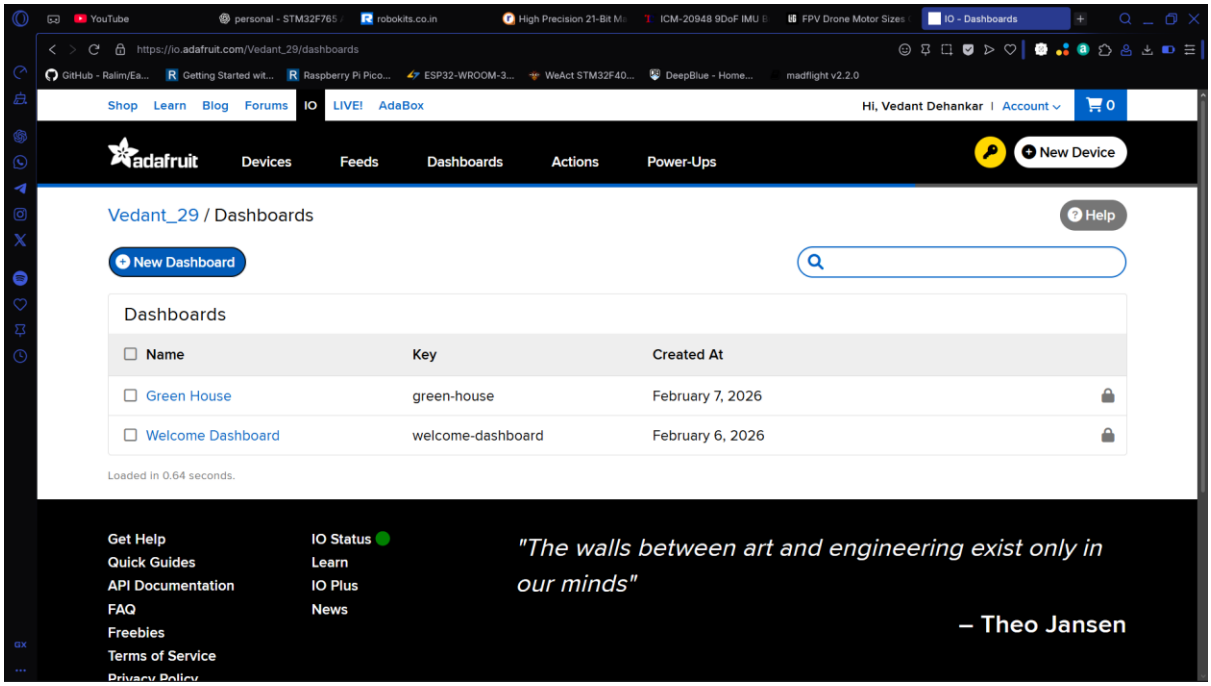
Step 10: Go back to the feeds tab and create feeds for all the variables you are measuring



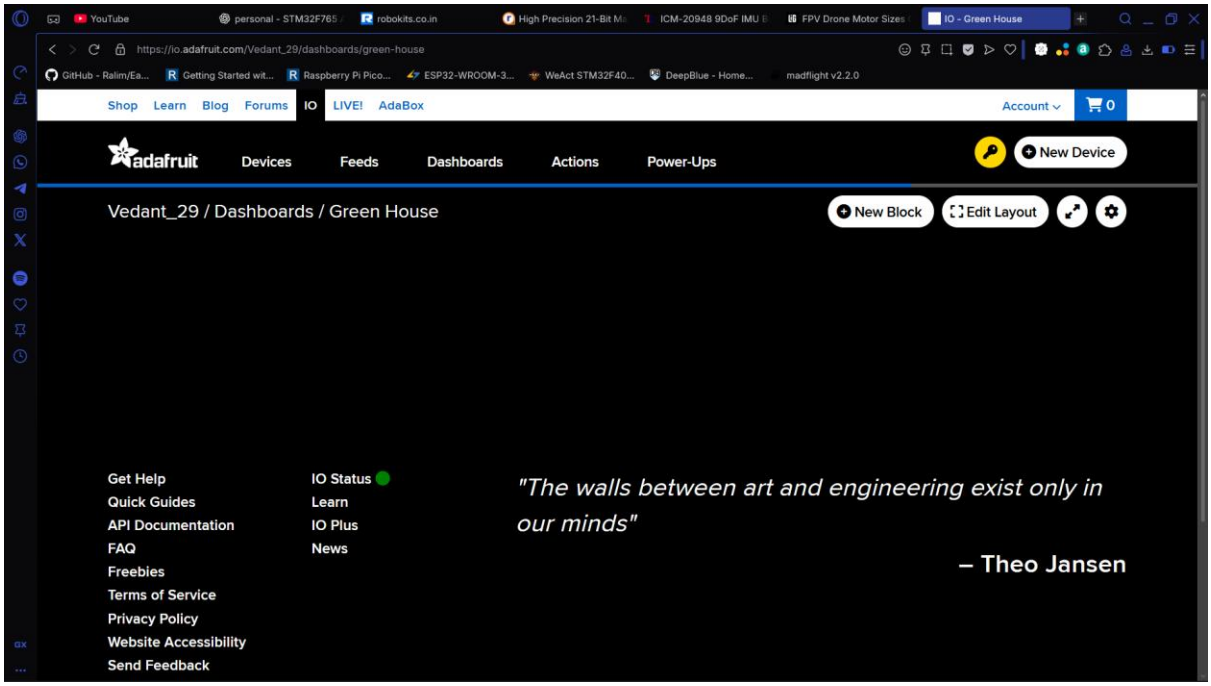
Step 11: Click on the new Dashboard button.



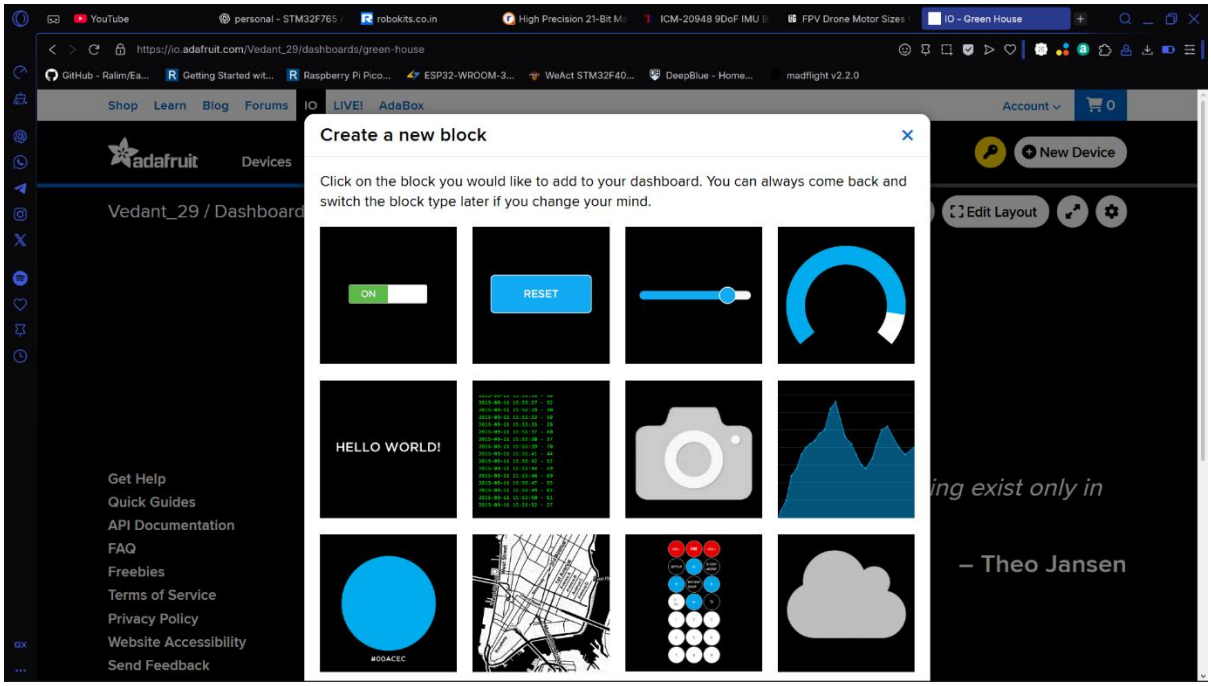
Step 12: A Create New Dashboard Pop up will appear, Give it a Name and click the Create button.



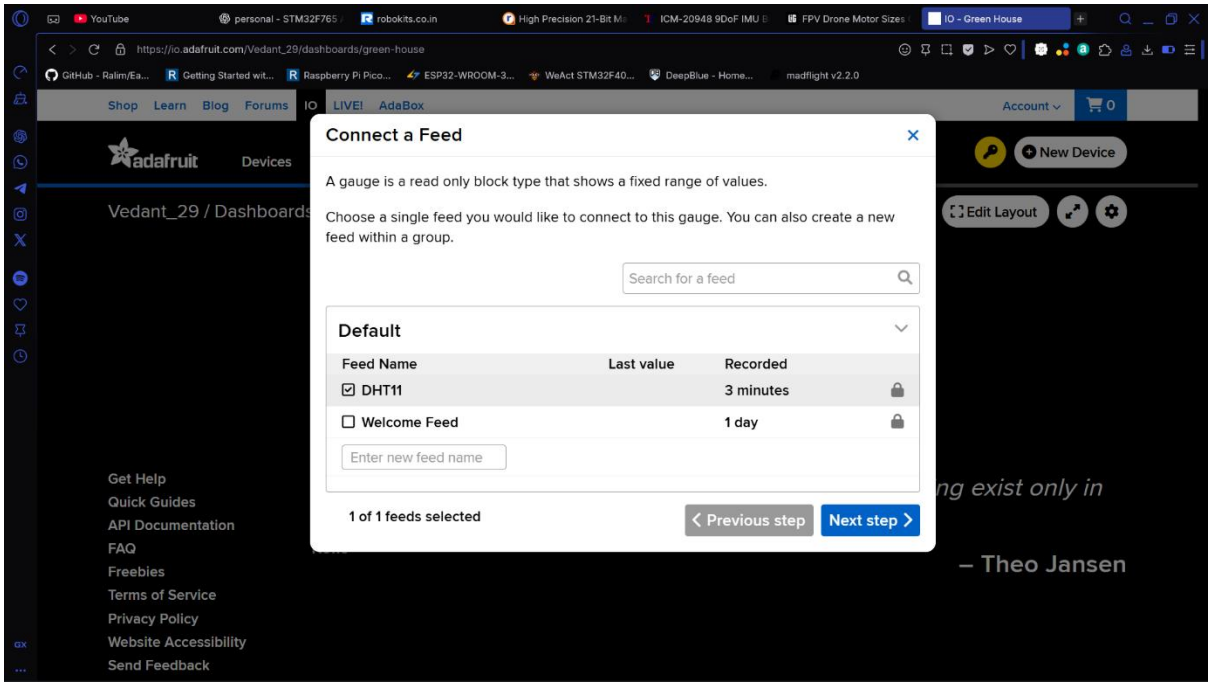
Step 13: A new Dashboard will appear, Click on the Dashboard you have created.



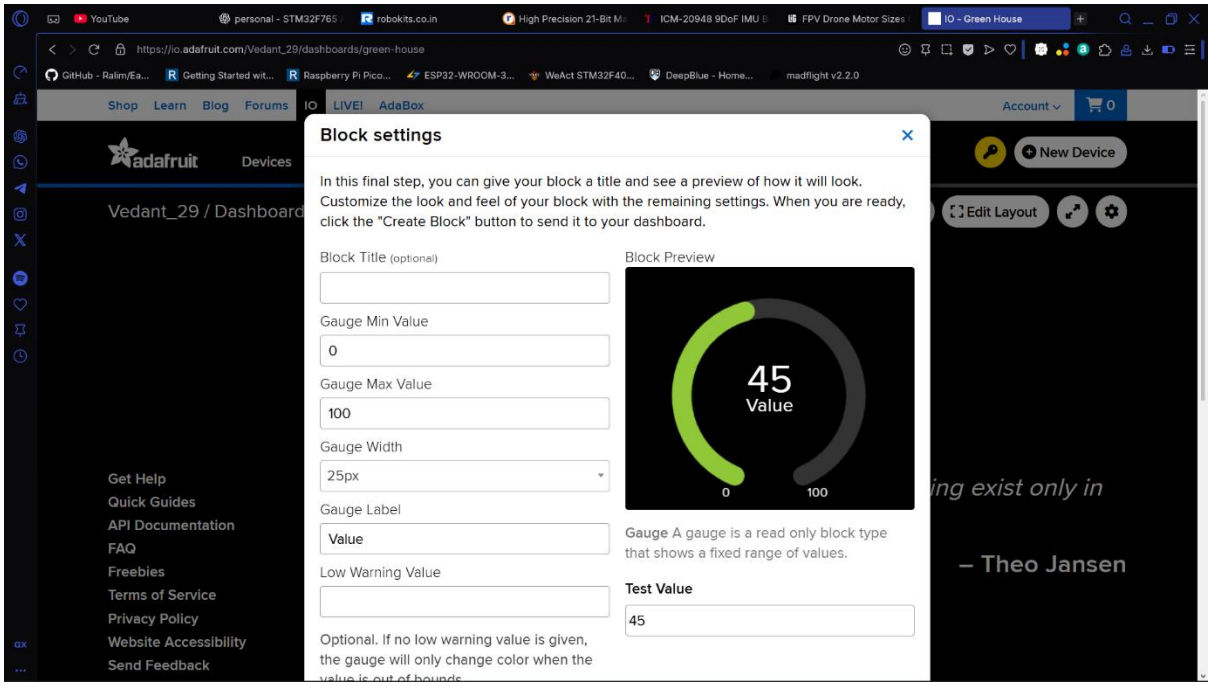
Step 14: You will be taken to a Dashboard, In this dashboard click on the New Block Button.

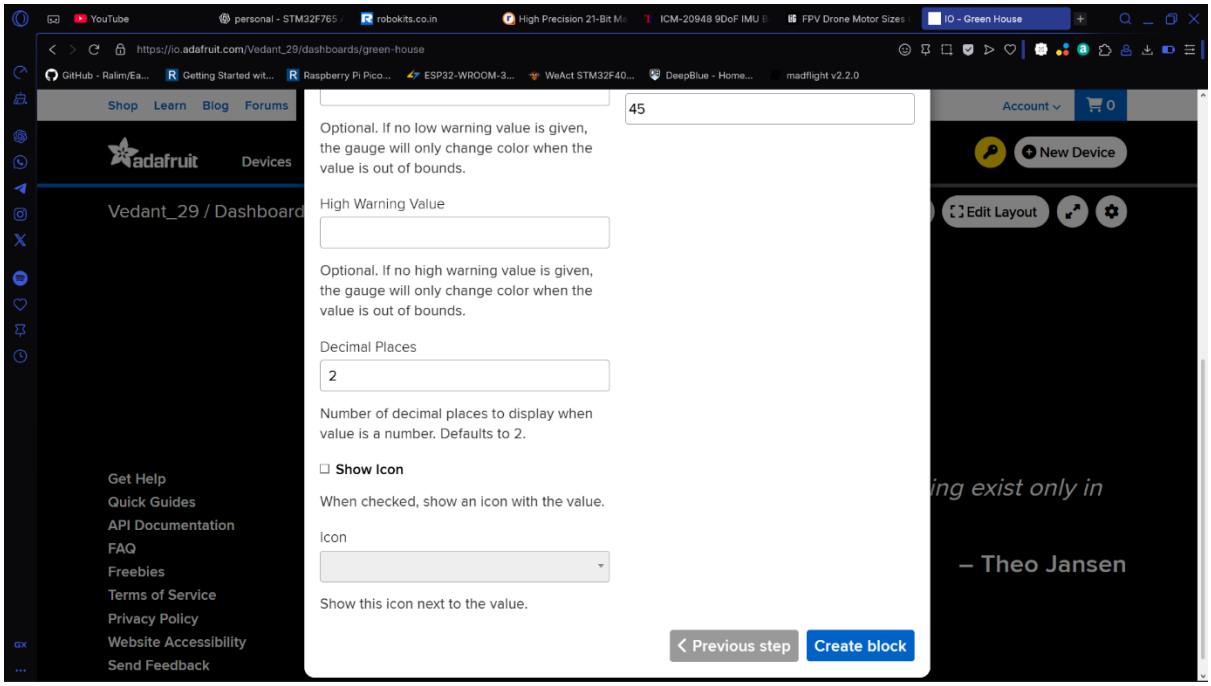


Step 15: A Create New Block Pop up will appear, In that select the gauge option.

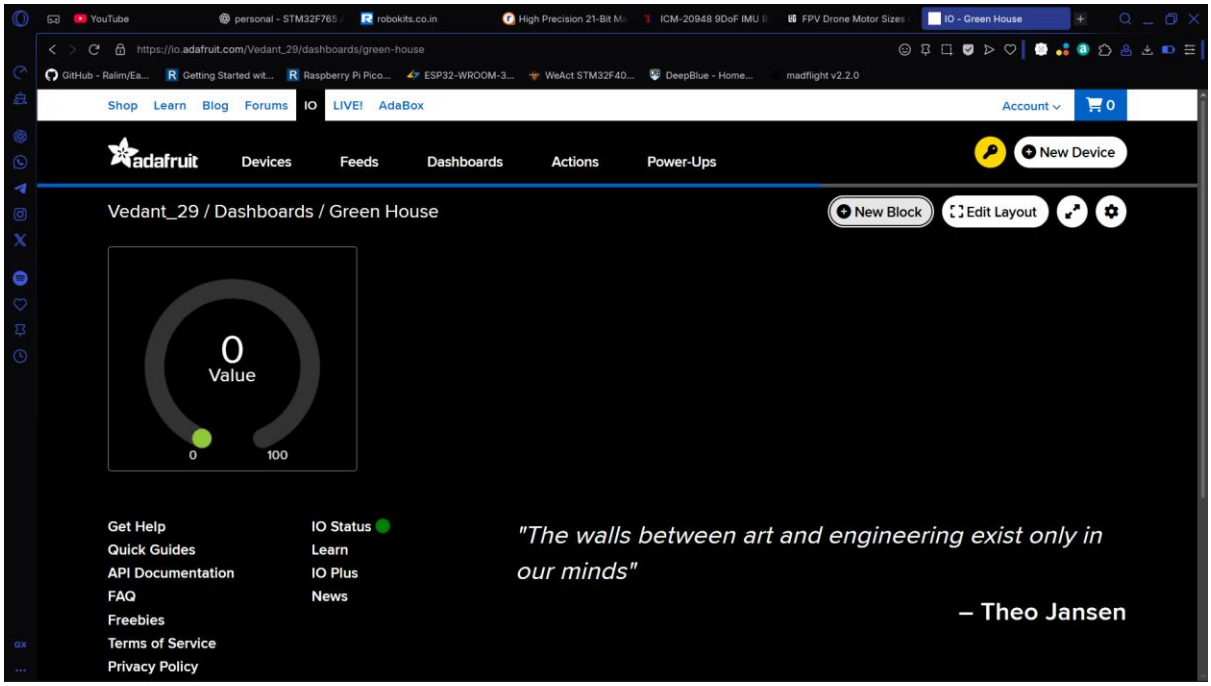


Step 16: You will be taken to a Connect a Feed Button in that select the feed you want to show the value off and click Next Step.

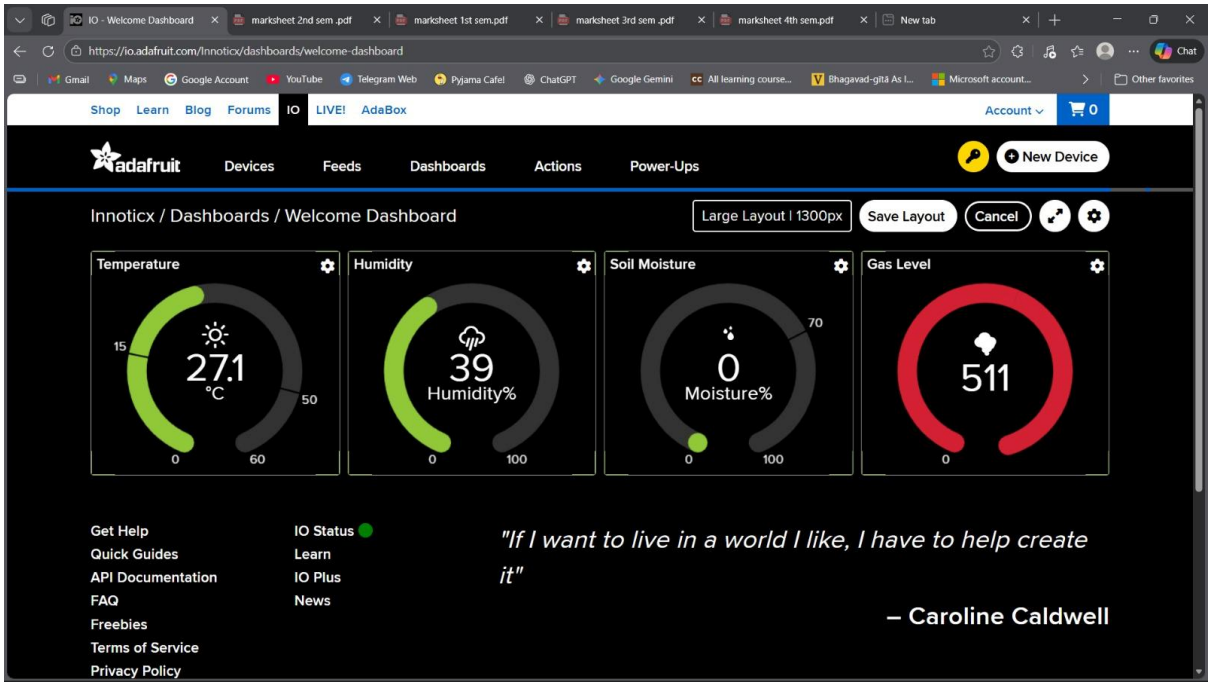




Step 17: In block Settings if you want you can change the values and appearance or just scroll down and click the create button.



Step 18: A Gauge will appear in you dashboard, do the same for all the variables you are measuring.



In the end you dashboard will look like this.

Now we are done with the Adafruit IO setup

Adafruit Code

Adafruitio_IOT.ino

```
#include <WiFi.h>
#include <DHT.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

// -------- WiFi --------
#define WIFI_SSID "YOUR SSID"
#define WIFI_PASS "YOUR PASSWORD"

// -------- Adafruit IO --------
#define AIO_SERVER     "io.adafruit.com"
#define AIO_PORT       1883
#define AIO_USERNAME   "YOUR USERNAME"
#define AIO_KEY        "YOUR API KEY"

// -------- Pins --------
#define DHT_DATA_PIN   26
#define DHTTYPE        DHT11

#define SOIL_AO_PIN    34

#define MQ2_AO_PIN     35

// -------- Objects --------
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_PORT,
                          AIO_USERNAME, AIO_KEY);

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

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

Adafruit_MQTT_Publish feedSoil =
  Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/soil");

Adafruit_MQTT_Publish feedGas =
  Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/gas");

DHT dht(DHT_DATA_PIN, DHTTYPE);

```

```
// -------- MQTT CONNECT --------
void mqtt_connect() {
  while (!mqtt.connected()) {
    if (mqtt.connect() == 0) {
      Serial.println("MQTT Connected");
    } else {
      Serial.println("MQTT Retry...");
      delay(3000);
    }
  }
}

// -------- SETUP --------
void setup() {
  Serial.begin(9600);

  dht.begin();

  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Connected");

  mqtt_connect();
}

// -------- LOOP --------
void loop() {
  mqtt_connect();

  float temperature = dht.readTemperature();
  float humidity    = dht.readHumidity();

  int soilRaw = analogRead(SOIL_AO_PIN);
  float soilPercent = map(soilRaw, 3500, 1200, 0, 100);
  soilPercent = constrain(soilPercent, 0, 100);

  int mq2Raw = analogRead(MQ2_AO_PIN);

```

```
  // ---- Serial ----
  Serial.println("--------------------");
  Serial.print("Temp: "); Serial.println(temperature);
  Serial.print("Humidity: "); Serial.println(humidity);
  Serial.print("Soil: "); Serial.println(soilPercent);
  Serial.print("Gas: "); Serial.println(mq2Raw);

  // ---- Publish (EXPLICIT TYPE) ----
  feedTemp.publish(temperature);
  feedHum.publish(humidity);
  feedSoil.publish((float)soilPercent);
  feedGas.publish((float)mq2Raw);

  delay(10000);   // rate-limit safe
}
```

Now you will be able to get the data from the sensors onto the Adafruit IO database