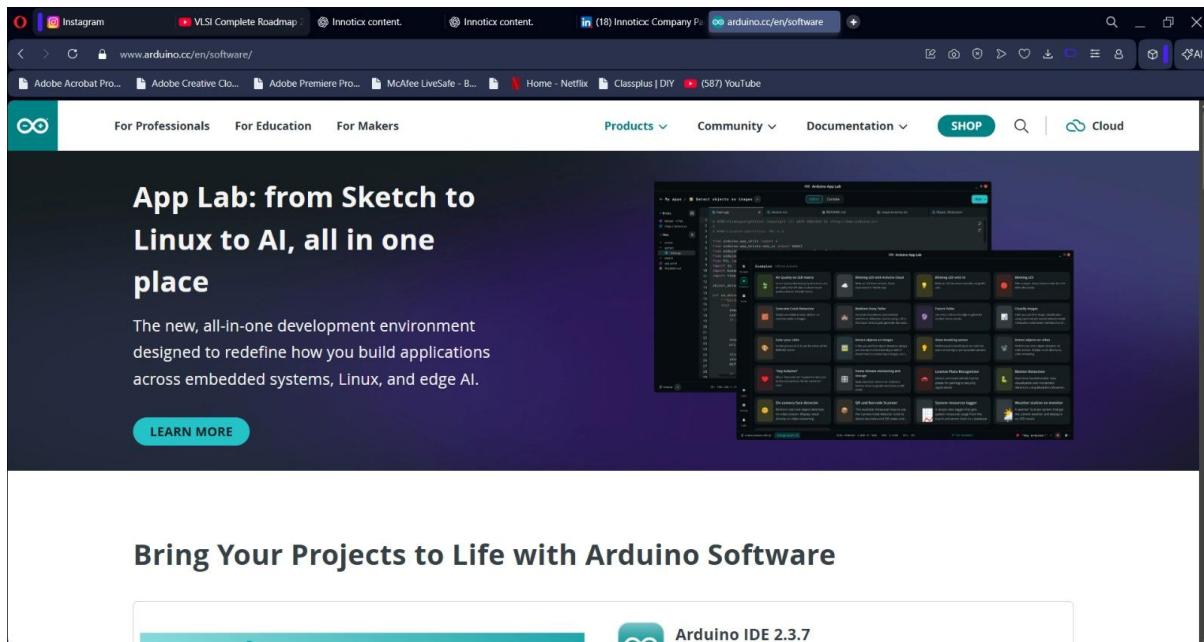
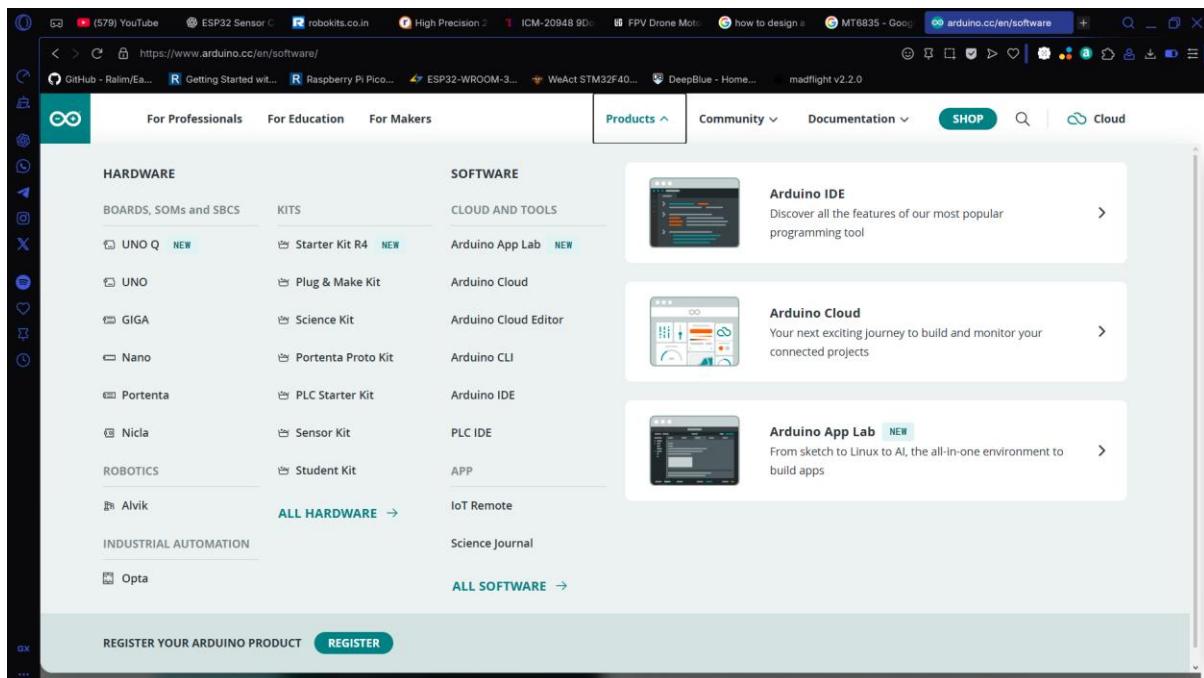


Installation of IDE

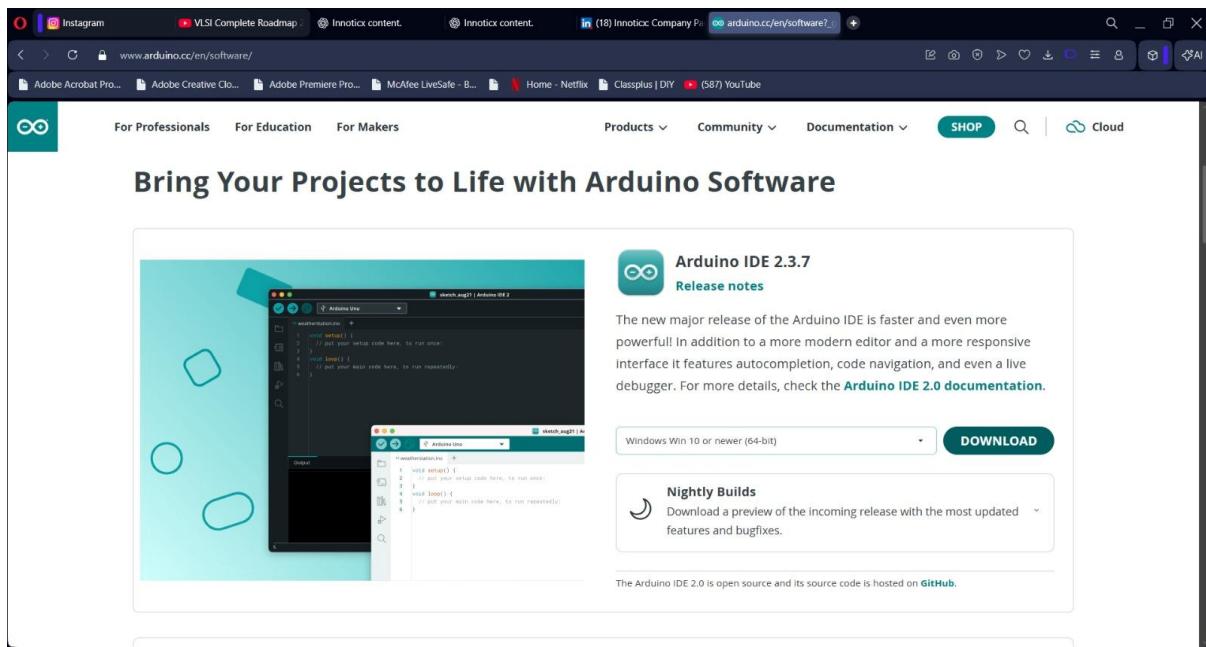
Step 1: Visit the Arduino Website



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

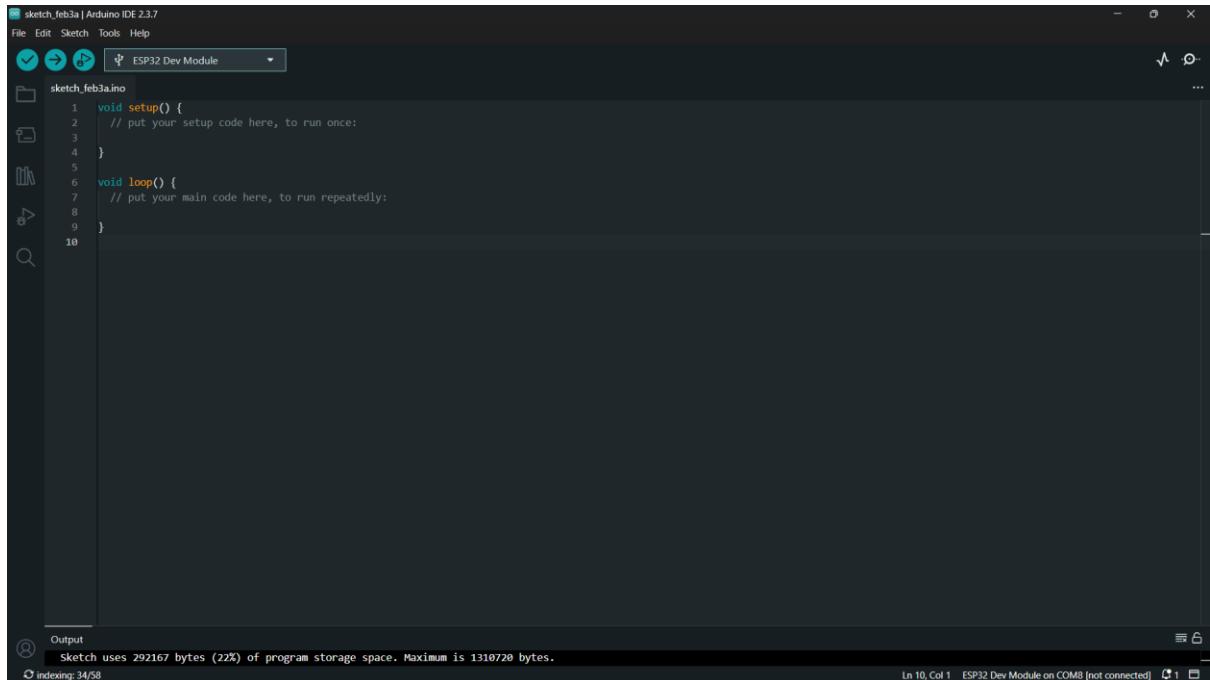


Step 3: Click on Arduino ide

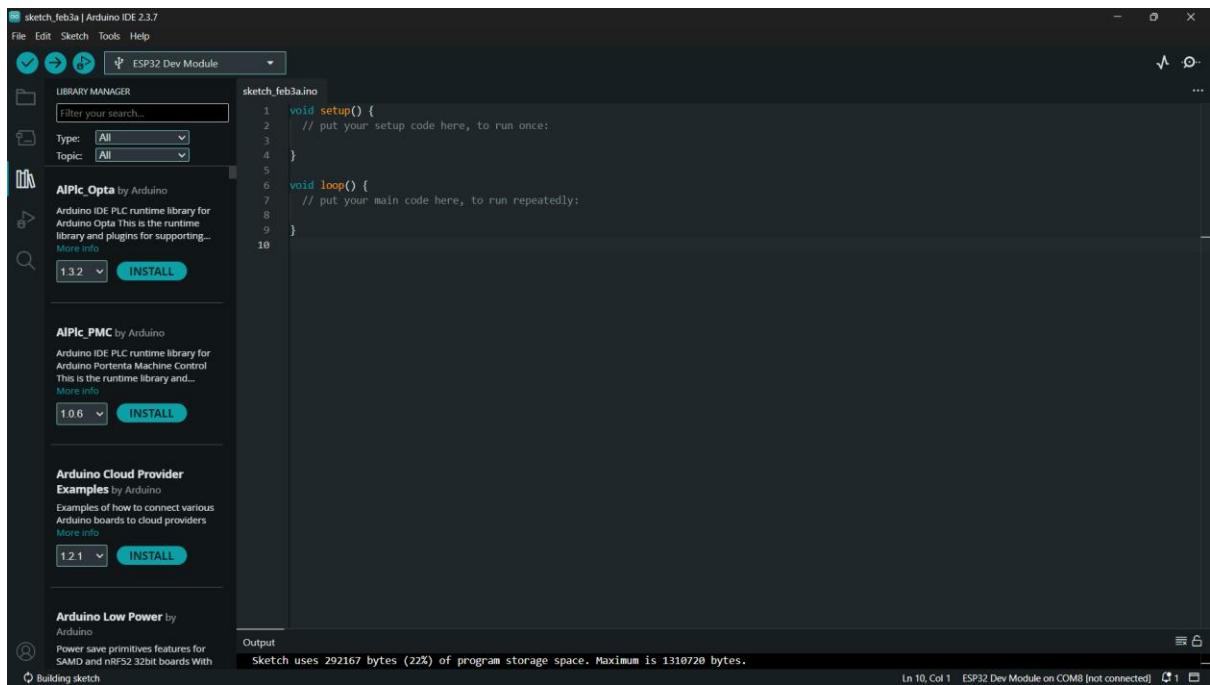


Step 4: Scroll down a little and download the installation file for your OS

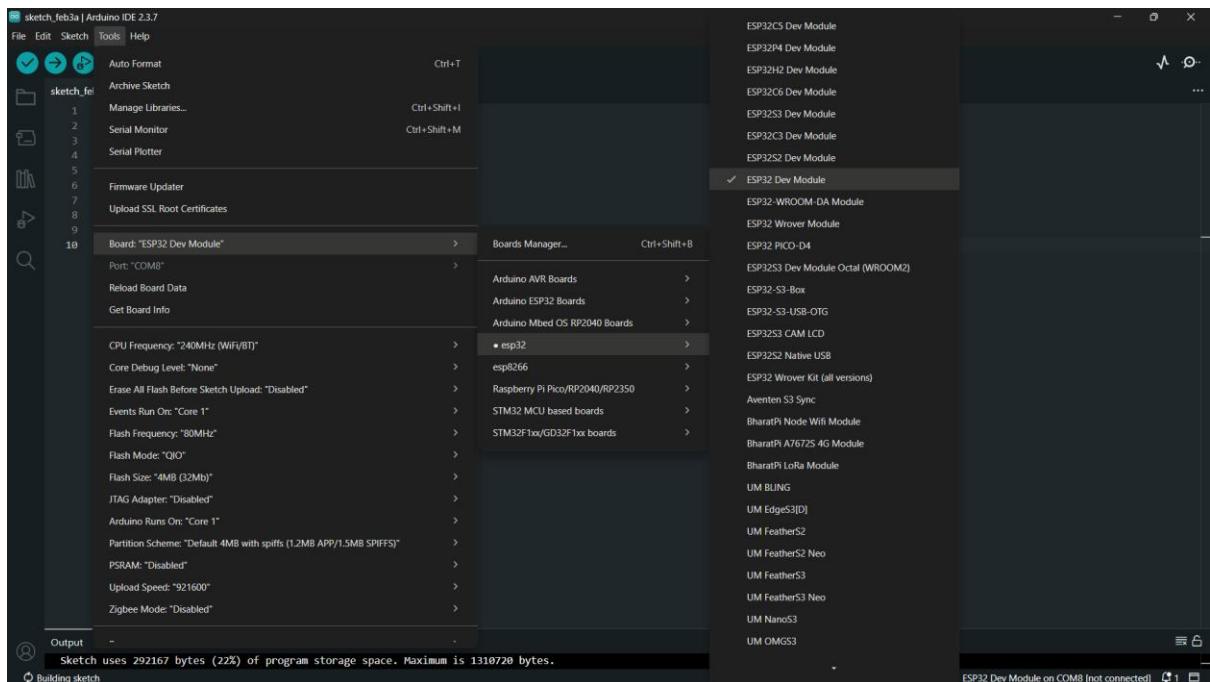
Introduction to IDE



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

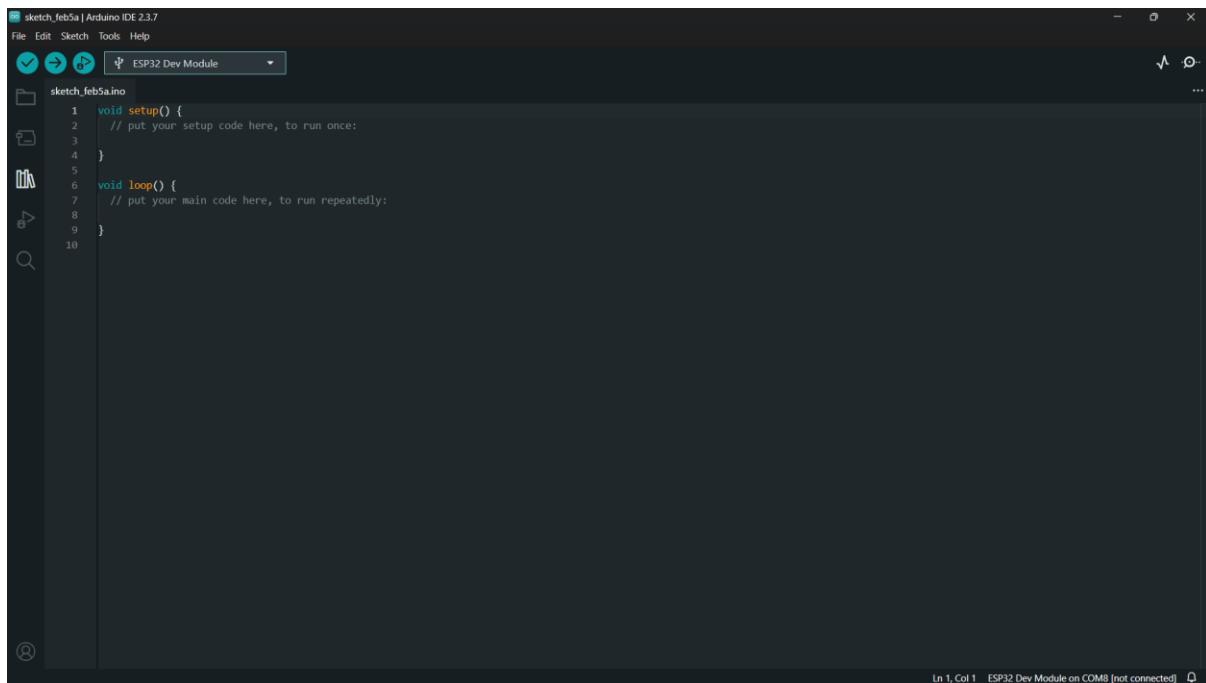


On the left hand side there are five buttons, The first button is the file manager in the IDE, the second button is used to install different Dev Boards in the IDE, the third button is used to install Different libraries in the IDE.

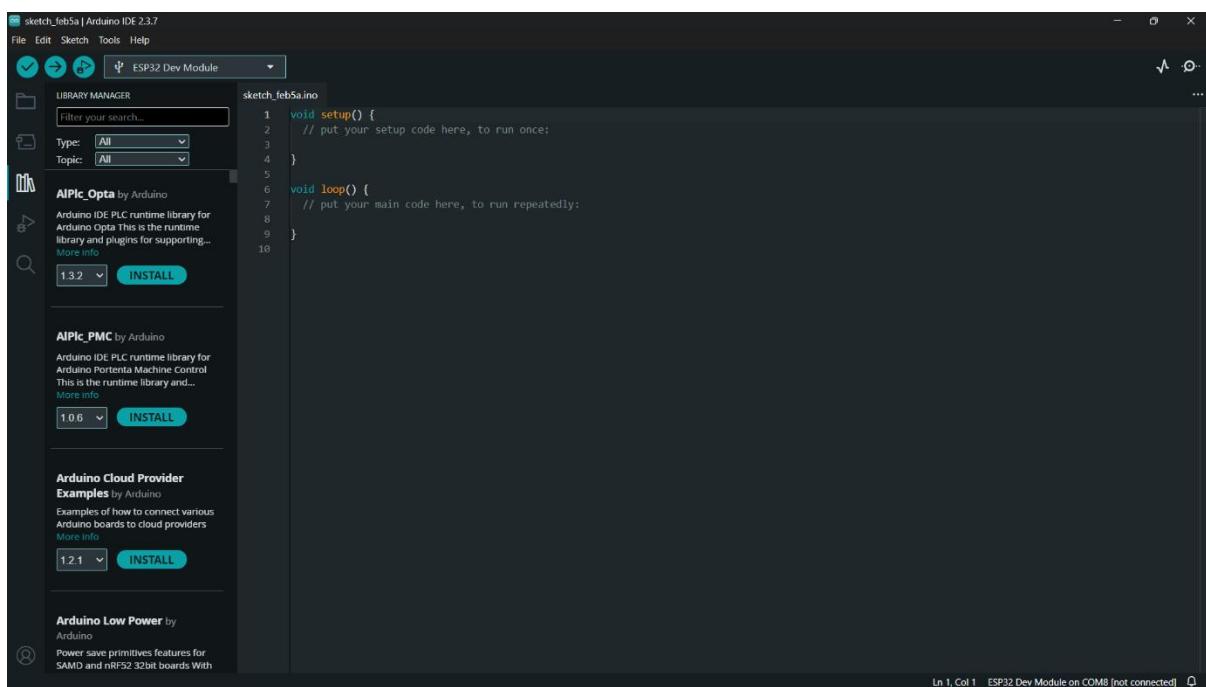


You would need to select the Exact board in the tools Tab you are trying to program and the COM Port to which the board is connected to.

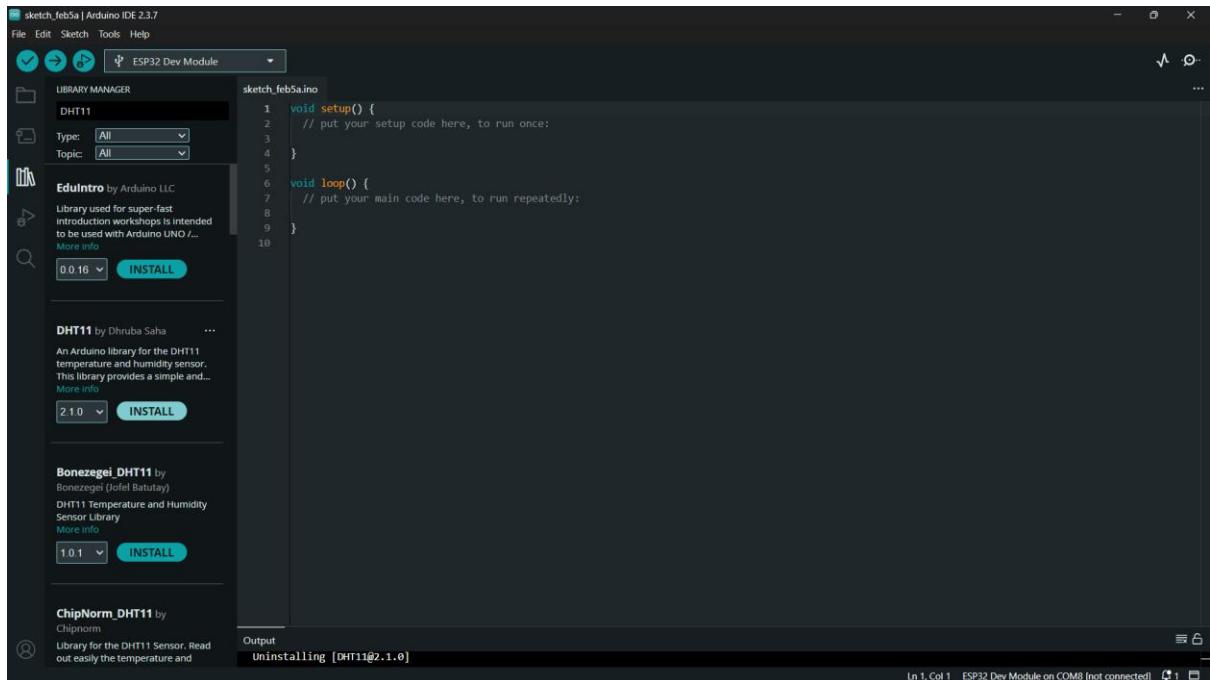
How to install a Library



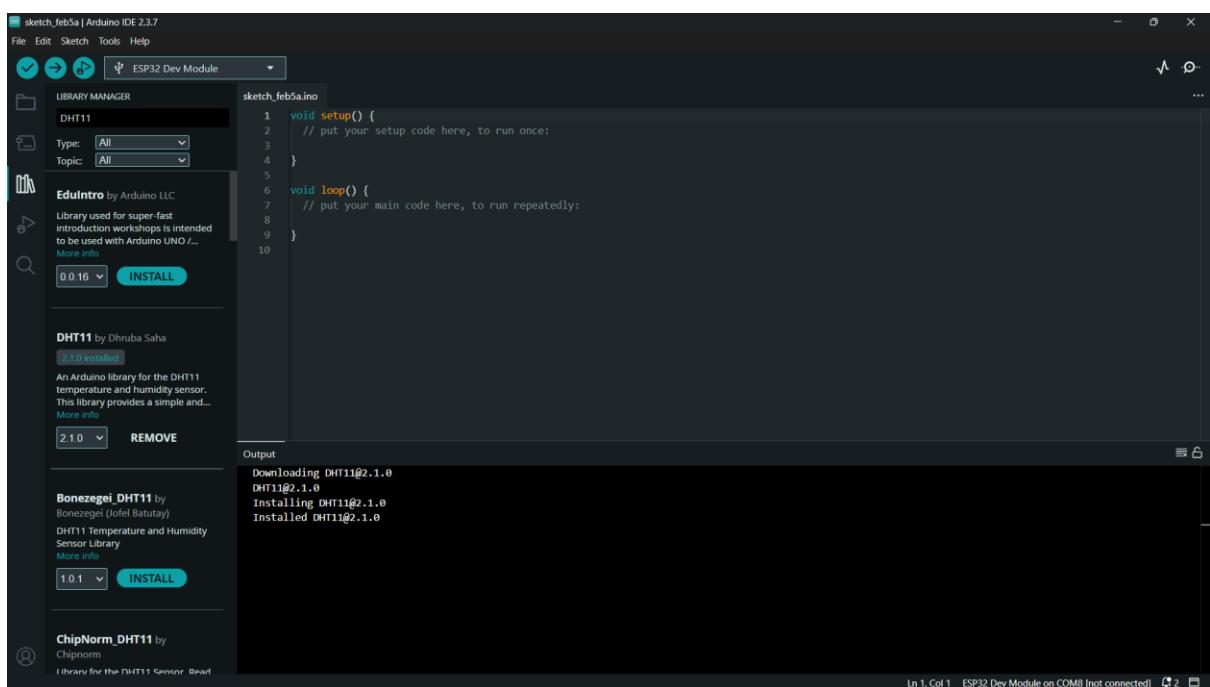
Step 1: In the Arduino Software click the library Manager Button on the Left hand side of the window



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.

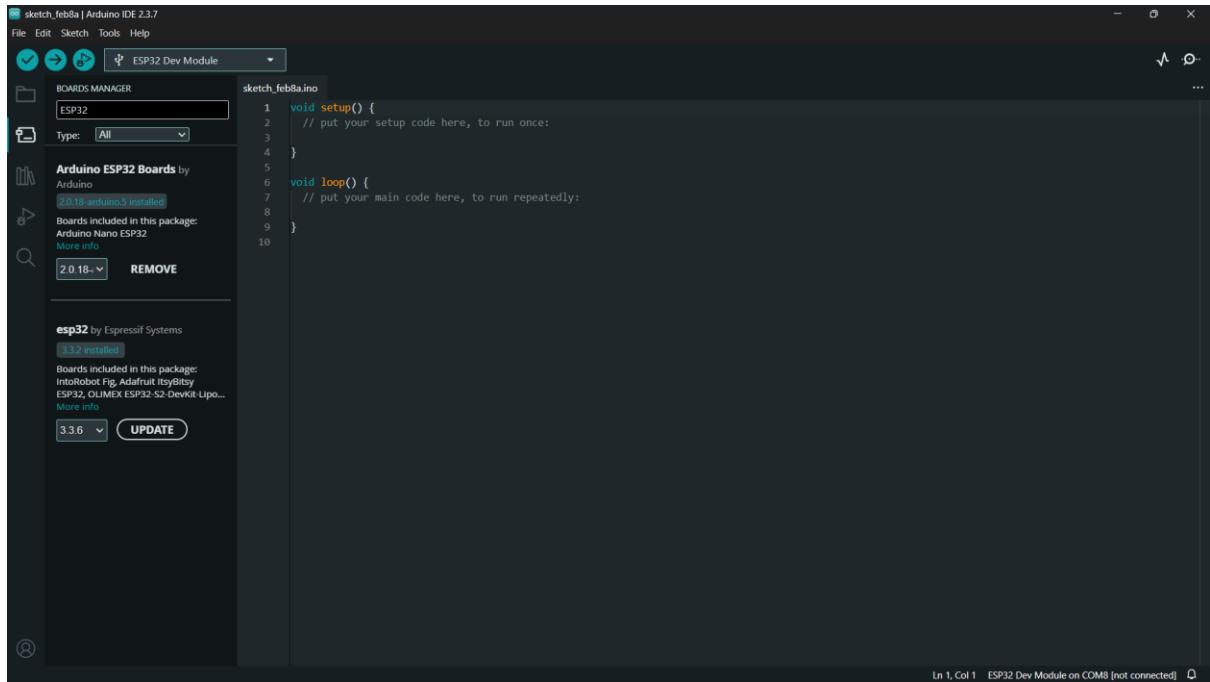


Step 3: For example lets search for DHT11 library, all the libraries related to DHT11 will appear, Scroll till you find the intended library and click the install button on it.



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.

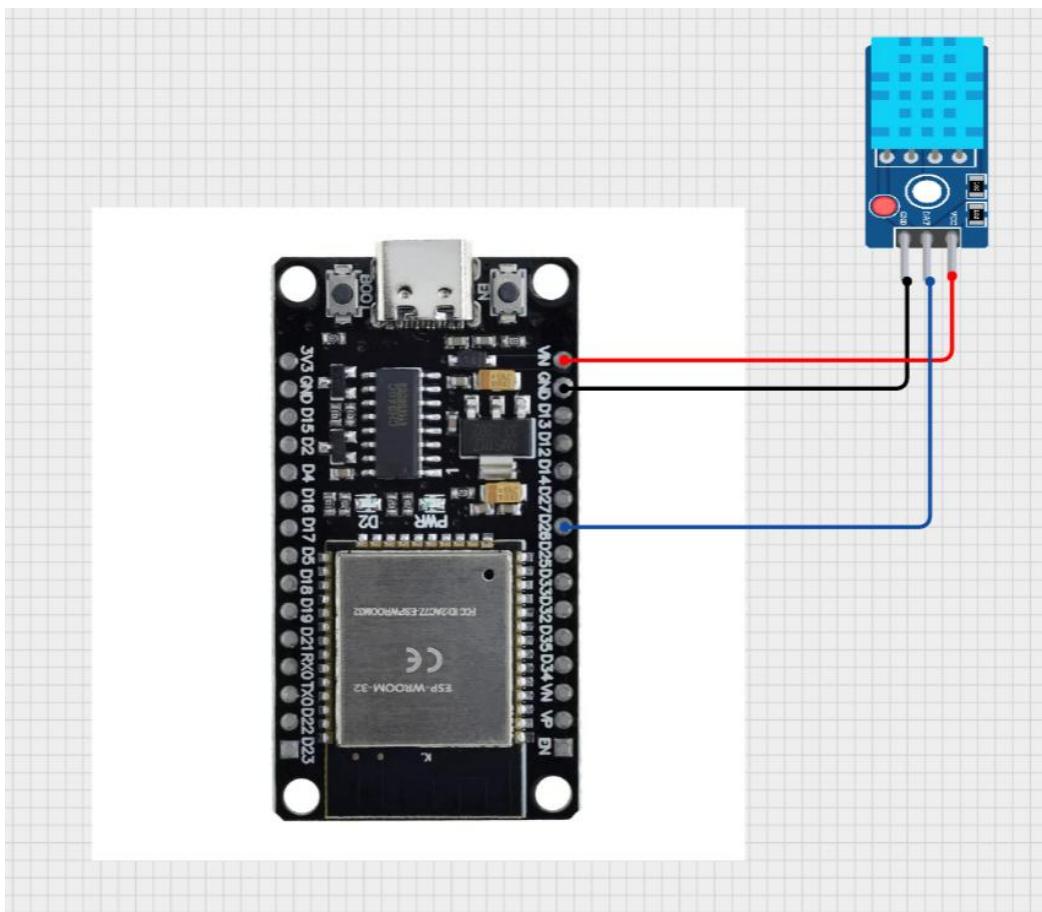
Installing ESP32 Boards in IDE



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

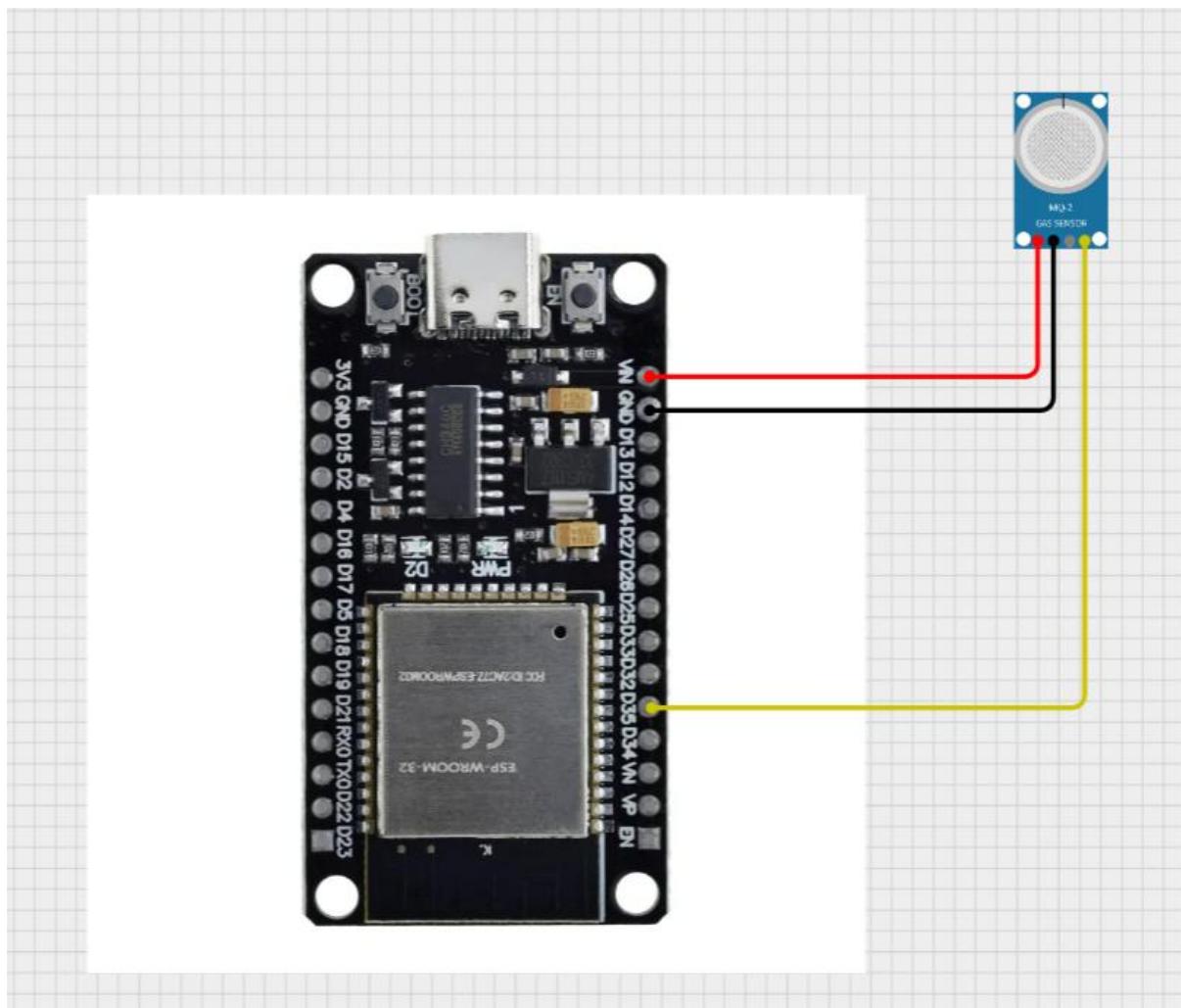
Step 2: In the boards manager search menu type ESP32 and install the ESP32 by Espressif Systems.

DHT11



```
dht11_temperature_sensor.ino
1 #include <DHT.h> // Include the DHT sensor library
2
3 // ----- Pin & Type Configuration -----
4 #define DHT_DATA_PIN 26 // The ESP32 pin where the DHT11 DATA pin is connected
5 #define DHTTYPE DHT11 // Define the type of DHT sensor being used (DHT11)
6
7 // Create a DHT sensor object using the pin and sensor type
8 DHT dht(DHT_DATA_PIN, DHTTYPE);
9
10 void setup() {
11     Serial.begin(9600); // Start serial communication at 9600 baud rate
12
13     delay(1000); // Small delay to allow sensor and serial to stabilize
14     dht.begin(); // Initialize the DHT11 sensor
15
16     Serial.println("DHT11 Test Started"); // Print a message to indicate setup is done
17 }
18
19 void loop() {
20     // Read humidity and temperature values from DHT11 sensor.
21     // These functions return NAN if the reading fails.
22     float h = dht.readHumidity();
23     float t = dht.readTemperature(); // Default is Celsius
24
25     // Check if either reading failed
26     if (isnan(h) || isnan(t)) {
27         Serial.println("DHT11 Read Error"); // Message if the sensor failed to read
28     } else {
29         // Print the temperature and humidity values
30         Serial.print("Temperature: ");
31         Serial.print(t);
32         Serial.print(" °C | Humidity: ");
33         Serial.print(h);
34         Serial.println(" %");
35     }
36
37     delay(2000); // Wait 2 seconds before the next reading
38 }
```

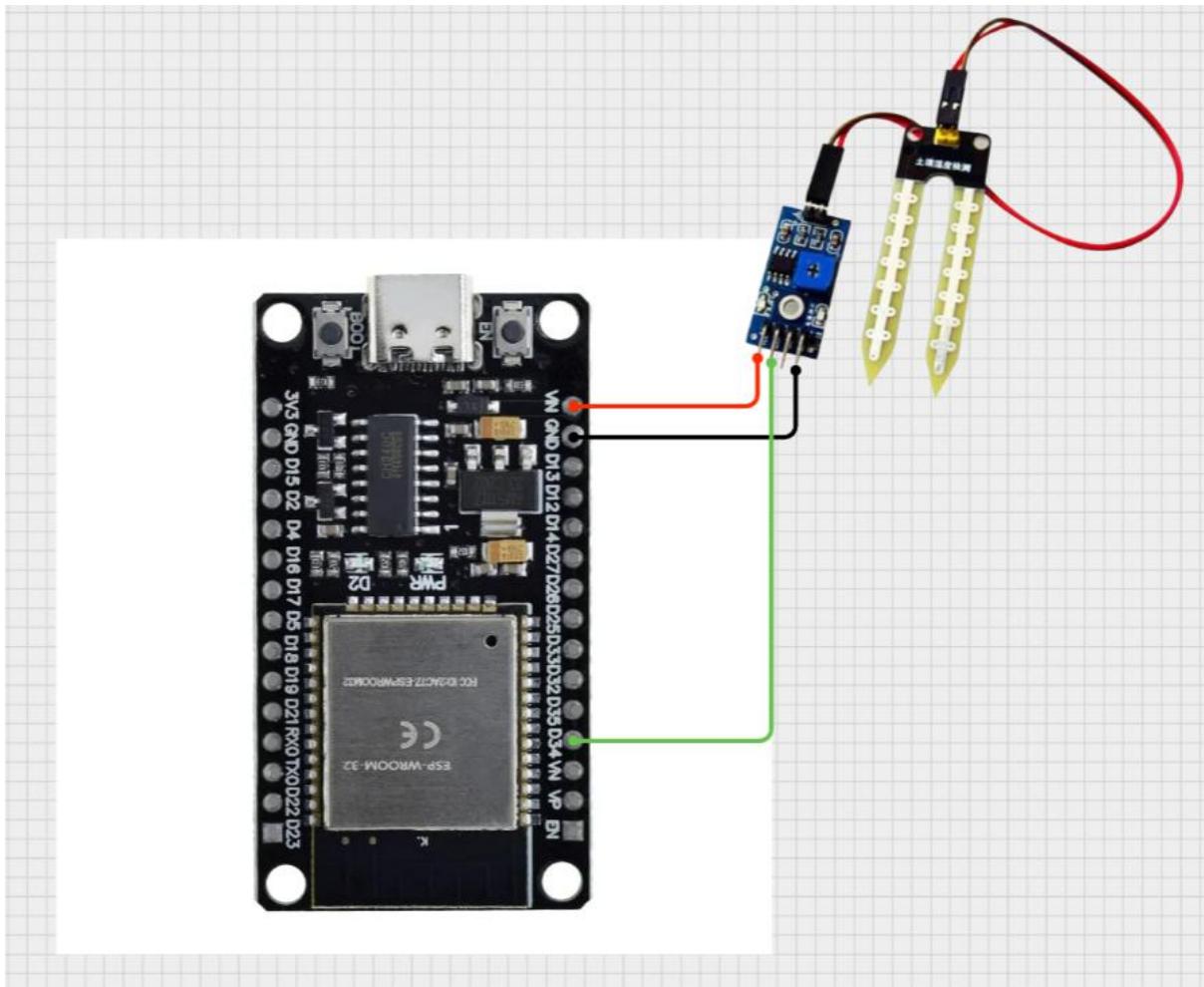
MQ-2 Gas Sensor



mq-2_gas_sensor.ino

```
1 #define MQ2_AO_PIN 35 // Define the ESP32 pin where MQ2 analog output (AO) is connected
2
3 void setup() {
4     Serial.begin(9600); // Start serial communication at 9600 baud rate
5     delay(1000); // Small delay to allow Serial to initialize properly
6
7     // Initial message to the user
8     Serial.println("MQ2 Gas Sensor Test Started");
9     Serial.println("Warming up MQ2... wait 2-3 minutes");
10    // MQ2 requires a warm-up time for the heating element to stabilize
11 }
12
13 void loop() {
14     // Read raw analog value from MQ2 AO pin (0-4095 on ESP32)
15     int mq2Raw = analogRead(MQ2_AO_PIN);
16
17     // Print the raw sensor value
18     // Higher value = less gas, lower value = more gas (depends on load resistor)
19     Serial.print("MQ2 Gas Raw Value: ");
20     Serial.println(mq2Raw);
21
22     delay(2000); // Wait 2 seconds before taking the next reading
23 }
```

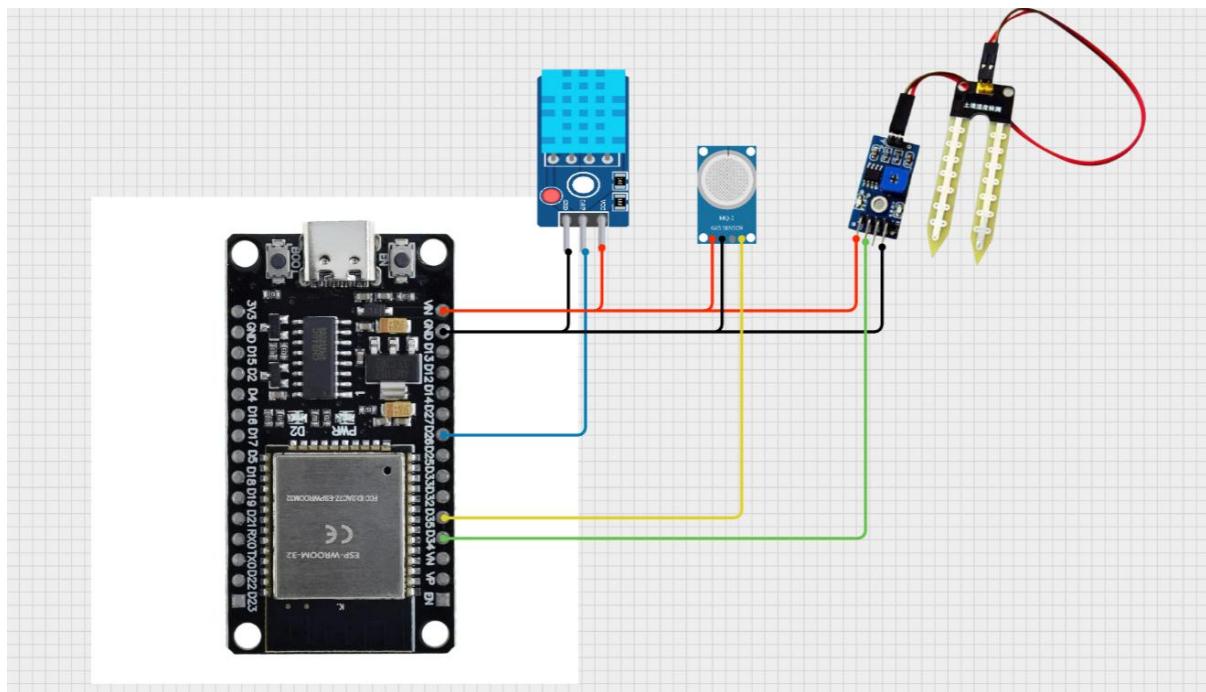
Soil Moisture sensor



soil_moisture_sensor.ino

```
1 #define SOIL_AO_PIN 34 // ESP32 pin connected to the soil moisture sensor's analog output (AO)
2
3 void setup() {
4     Serial.begin(9600); // Start serial communication at 9600 baud rate
5
6     delay(500); // Small delay to allow serial communication to stabilize
7     Serial.println("Soil Moisture Sensor Test Started");
8 }
9
10 void loop() {
11     // Read the raw analog value from the soil moisture sensor
12     // ESP32 ADC range = 0 to 4095
13     int soilRaw = analogRead(SOIL_AO_PIN);
14
15     // Print the raw analog value to the Serial Monitor
16     Serial.print("Soil Moisture Raw Value: ");
17     Serial.println(soilRaw);
18
19     // Wait for 2 seconds before the next reading
20     delay(2000);
21 }
22 }
```

Combined code



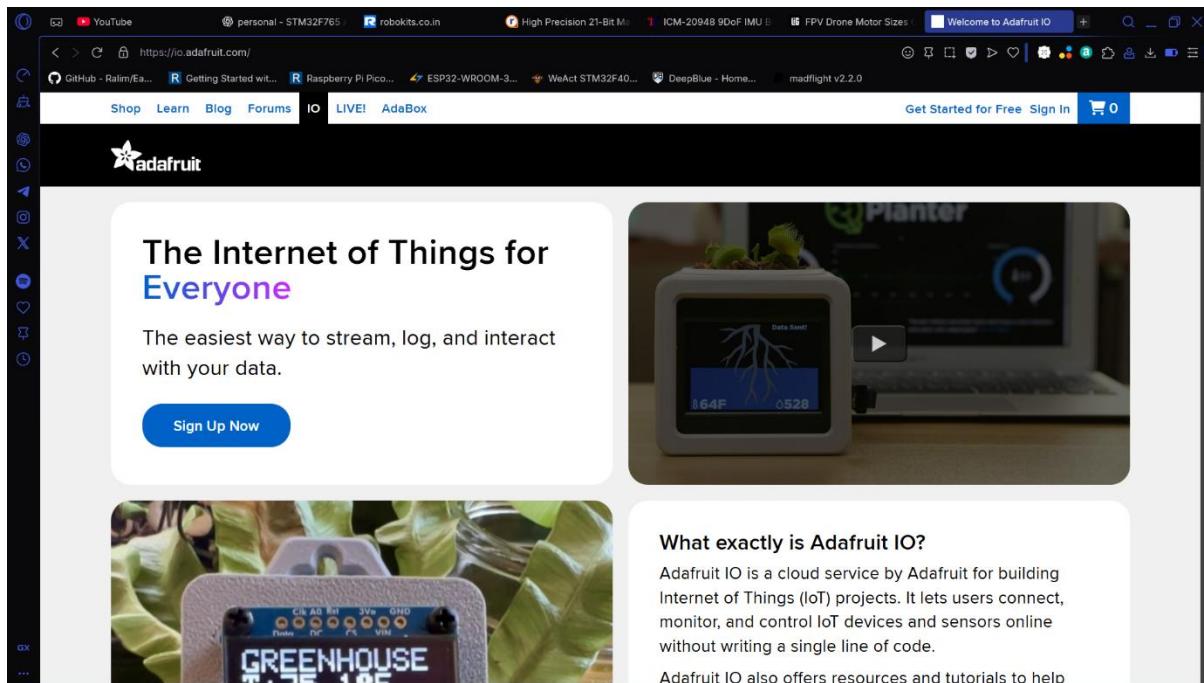
greenhouse_interfacing.ino

```
1 #include <DHT.h> // Include the DHT library needed for temperature & humidity sensor
2
3 // ----- PIN DEFINITIONS -----
4
5 // DHT11 Sensor Pins
6 #define DHT_DATA_PIN 26 // Data pin connected to DHT11
7 #define DHTTYPE DHT11 // Define the sensor type (DHT11)
8
9 // Soil Moisture Sensor Pin
10 #define SOIL_AO_PIN 34 // Analog output pin of soil moisture sensor
11
12 // MQ2 Gas Sensor Pin
13 #define MQ2_AO_PIN 35 // Analog output pin of MQ2 gas sensor
14
15 // Create the DHT object using pin 26 and sensor type DHT11
16 DHT dht(DHT_DATA_PIN, DHTTYPE);
17
18 void setup() {
19   Serial.begin(9600); // Start serial communication at 9600 baud rate
20   delay(500); // Small delay for Serial Monitor to initialize
21
22   dht.begin(); // Initialize the DHT11 sensor
23
24   // Print startup message
25   Serial.println("ESP32 Sensor Interface Started");
26   Serial.println("DHT11 | Soil Moisture | MQ2 Active");
27   Serial.println("MQ2 warm-up: wait 2-3 minutes for stable readings");
28 }
29
30 void loop() {
31
32   // ----- READ DHT11 SENSOR -----
33   float humidity = dht.readHumidity(); // Read humidity (%)
34   float temperature = dht.readTemperature(); // Read temperature (Celsius)
35
36   // ----- READ SOIL MOISTURE SENSOR -----
37   int soilRaw = analogRead(SOIL_AO_PIN); // Read raw analog value (0-4095)
38 }
```

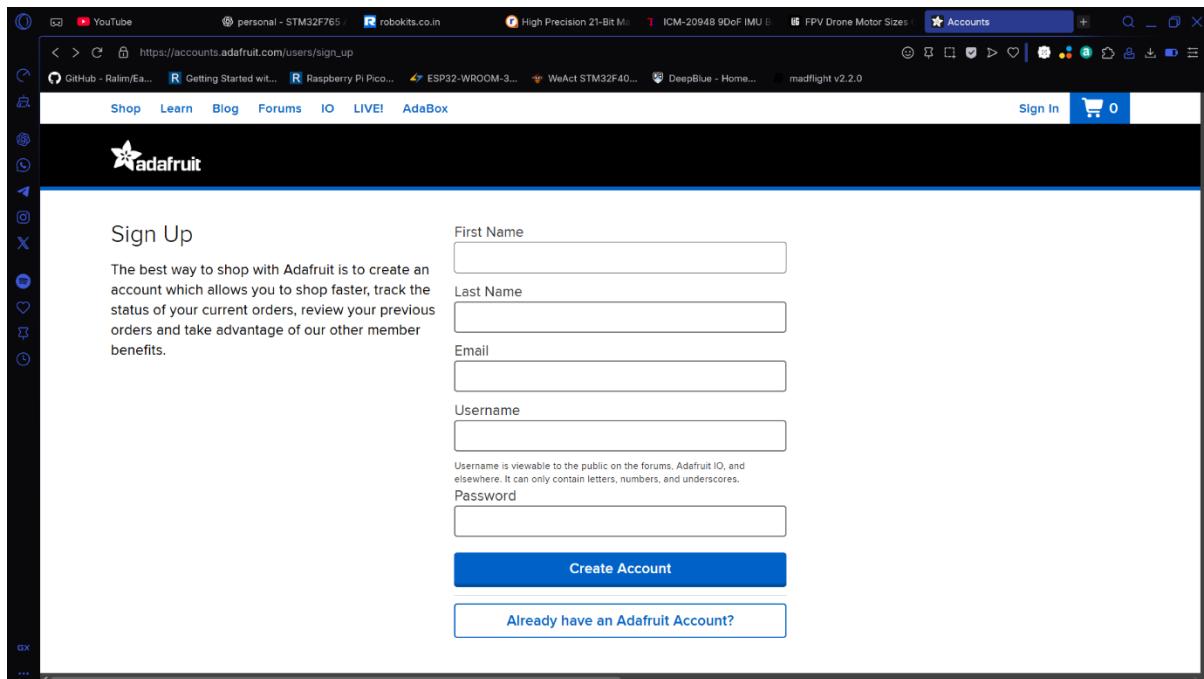
```
39 // Convert the raw soil moisture reading to a percentage
40 // These numbers (3500 = dry, 1200 = wet) must be calibrated for YOUR sensor
41 int soilPercent = map(soilRaw, 3500, 1200, 0, 100);
42 soilPercent = constrain(soilPercent, 0, 100); // Force value within 0-100%
43
44 // ----- READ MQ2 GAS SENSOR -----
45 int mq2Raw = analogRead(MQ2_AO_PIN); // Read raw gas sensor value (0-4095)
46
47 // ----- PRINT SENSOR READINGS TO SERIAL -----
48 Serial.println("-----");
49
50 // Print DHT11 values
51 if (isnan(humidity) || isnan(temperature)) { // Check if reading failed
52     Serial.println("DHT11 Read Error - Check wiring");
53 } else {
54     Serial.print("Temperature: ");
55     Serial.print(temperature);
56     Serial.print(" °C | Humidity: ");
57     Serial.print(humidity);
58     Serial.println(" %");
59 }
60
61 // Print soil moisture values
62 Serial.print("Soil Moisture Raw: ");
63 Serial.print(soilRaw);
64 Serial.print(" | Moisture: ");
65 Serial.print(soilPercent);
66 Serial.println(" %");
67
68 // Print MQ2 gas sensor value
69 Serial.print("MQ2 Gas Raw Value: ");
70 Serial.println(mq2Raw);
71
72 delay(2000); // DHT11 requires at least 2 seconds between readings
73 }
74 }
```

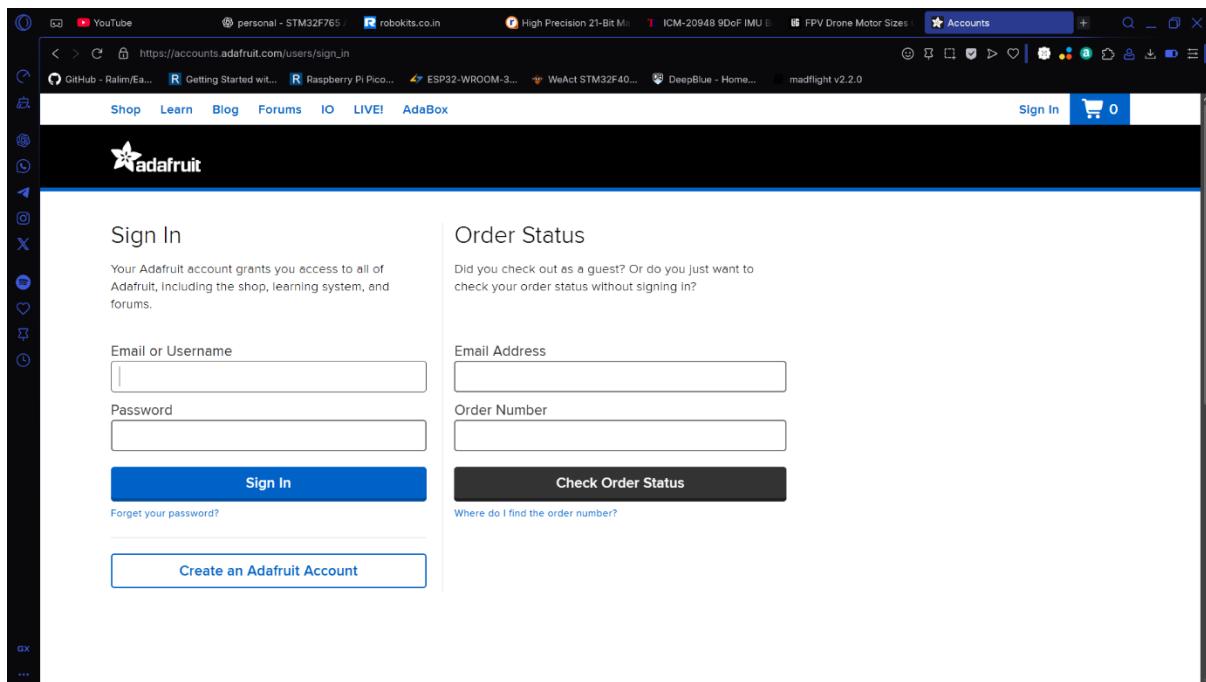
Adafruit IO Setup:

Step 1: Visit the Adafruit IO website

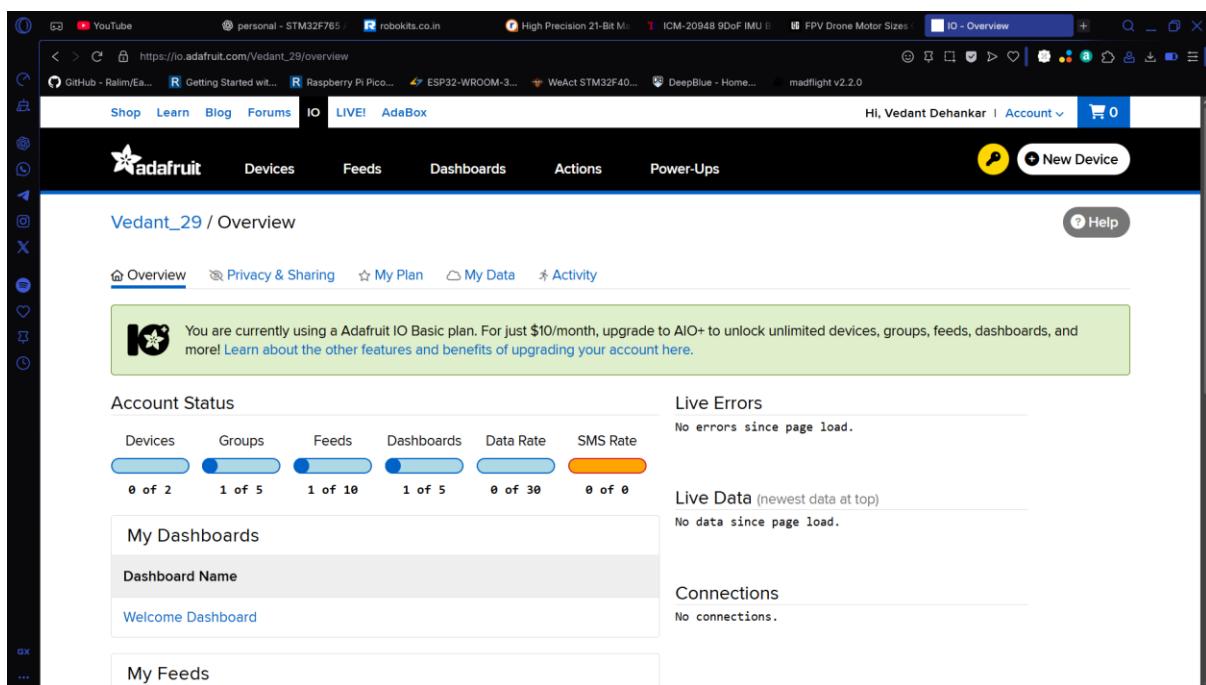


Step 2: In the Adafruit IO page Click on the Sign Up button.





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.

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: Vedant_29

Active Key: aio_jEpk98HHJvc01cggJ56omM14dyK **REGENERATE KEY**

CircuitPython

```
ADAFRUIT_AIO_USERNAME = "Vedant_29"
ADAFRUIT_AIO_KEY      = "aio_jEpk98HHJvc01cggJ56omM14dyK"
```

Arduino

```
#define IO_USERNAME "Vedant_29"
#define IO_KEY      "aio_jEpk98HHJvc01cggJ56omM14dyK"
```

Linux Shell

```
export IO_USERNAME="Vedant_29"
export IO_KEY="aio_jEpk98HHJvc01cggJ56omM14dyK"
```

Scripting

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

You are currently using a Adafruit IO Basic plan. For just \$10/month, upgrade to AIO+ to unlock unlimited devices, groups, feeds, dashboards, and more! [Learn about the other features and benefits of upgrading your account here.](#)

Account Status

Devices	Groups	Feeds	Dashboards	Data Rate	SMS Rate
0 of 2	1 of 5	1 of 10	1 of 5	0 of 30	0 of 0

My Dashboards

Dashboard Name

Welcome Dashboard

My Feeds

Live Errors
No errors since page load.

Live Data (newest data at top)
No data since page load.

Connections
No connections.

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

The screenshot shows the Adafruit IO Feeds interface. At the top, there's a navigation bar with links like Shop, Learn, Blog, Forums, IO, LIVE!, AdaBox, and a user account section. Below the navigation is a main header with tabs for Devices, Feeds, Dashboards, Actions, and Power-Ups. A prominent 'New Device' button is located in the top right. The main content area is titled 'Vedant_29 / Feeds'. It displays a table with one row for a feed named 'Welcome Feed' with key 'welcome-feed'. A search bar is at the top right of the table. On the left side, there's a sidebar with links to Get Help, Quick Guides, API Documentation, FAQ, Freebies, Terms of Service, Privacy Policy, Website Accessibility, and Send Feedback. On the right, there's a quote by Theo Jansen: *"The walls between art and engineering exist only in our minds"*.

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

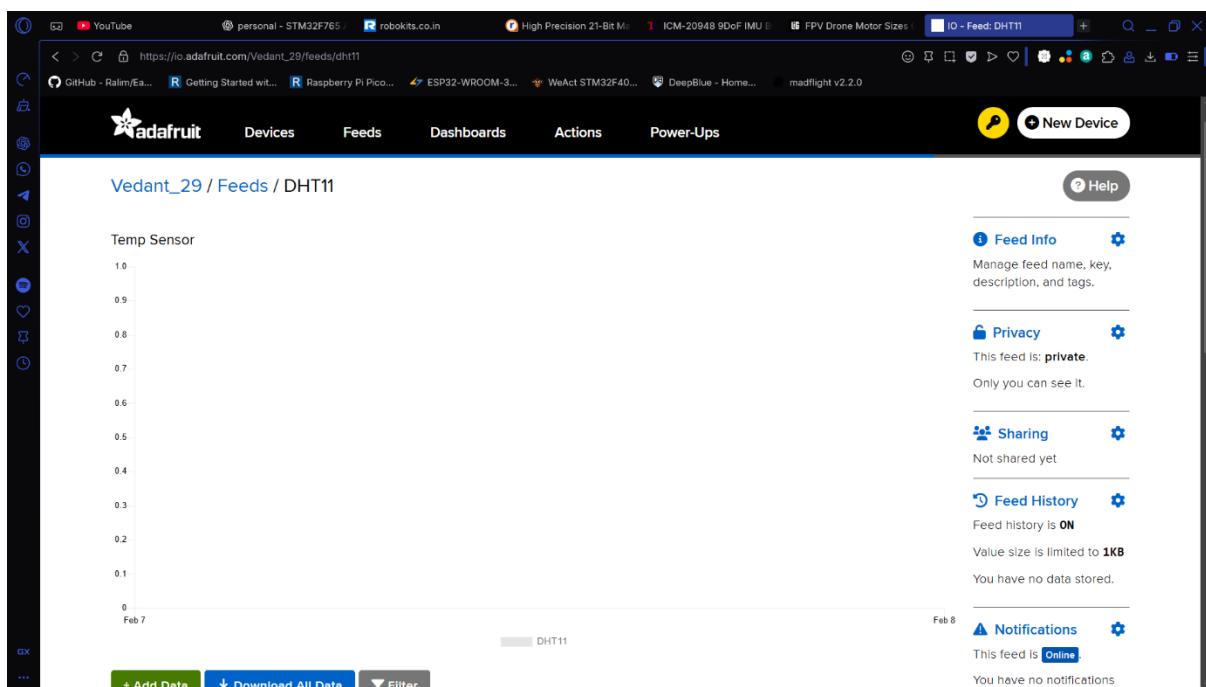
This screenshot shows the 'Create a new Feed' dialog box overlaid on the Feeds page. The dialog has fields for 'Name' (containing 'DHT1') and 'Description' (containing 'Temp Sensor'). There are 'Cancel' and 'Create' buttons at the bottom. The background Feeds page is visible, showing the same layout as the previous screenshot but with the dialog in the foreground.

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

The screenshot shows the Adafruit IO Feeds page for user 'Vedant_29'. It displays two feeds:

Feed Name	Key	Last value	Recorded
DHT11	dht11		less than a minute ago
Welcome Feed	welcome-feed		1 day ago

Below the table, a message from Theo Jansen is displayed: *"The walls between art and engineering exist only in our minds"*.



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

The screenshot shows the Adafruit Feeds tab. At the top, there are tabs for Devices, Feeds, Dashboards, Actions, and Power-Ups. A 'New Device' button is located in the top right corner. Below the tabs, the title 'Innoticx / Feeds' is displayed. There are two buttons: 'New Feed' and 'New Group'. A search bar is on the right. The main area is titled 'Default' and contains a table of feeds:

Feed Name	Key	Last value	Recorded
Gas	gas	511.00	about 12 hours ago
Humidity	humidity	39.00	about 12 hours ago
Soil Moisture	soil-moisture		about 13 hours ago
Temperature	temperature	27.10	about 12 hours ago
Welcome Feed	welcome-feed	27	about 13 hours ago
soil	soil	0.00	about 12 hours ago

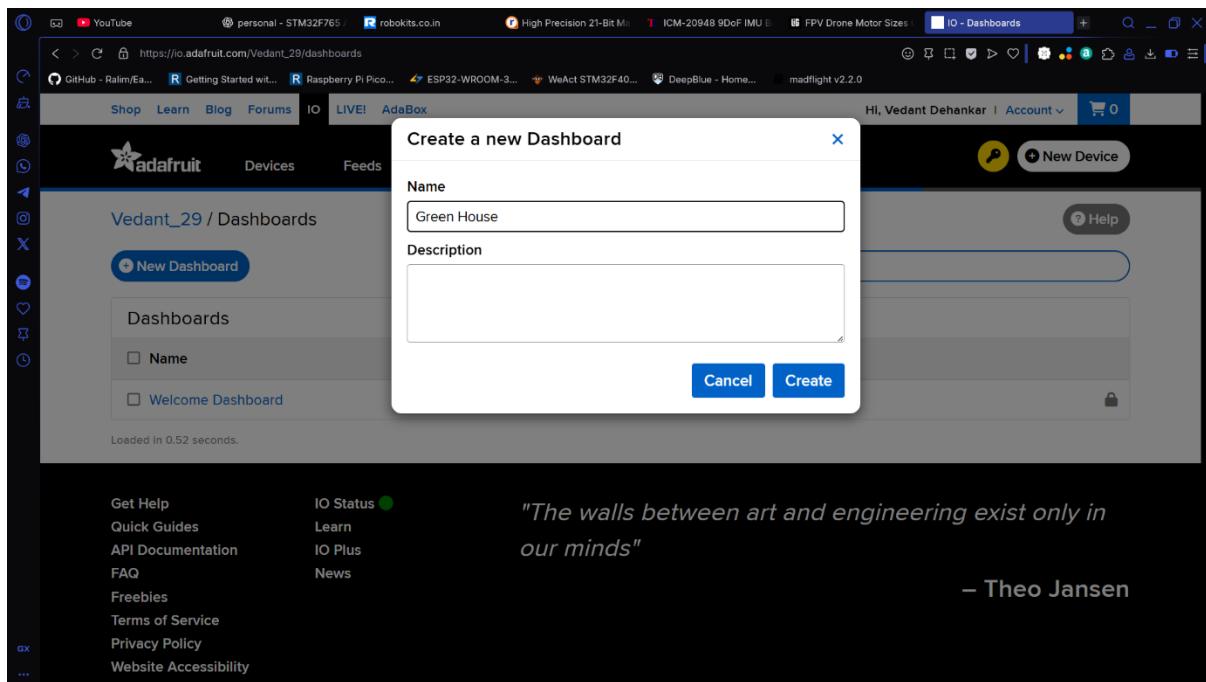
Step 10: Go back to the feeds tab and create feeds for all the variables you are measuring

The screenshot shows the Adafruit Dashboards tab. At the top, there are tabs for Shop, Learn, Blog, Forums, IO, and LIVE!. A user profile 'Hi, Vedant Dehankar | Account' is shown. A 'New Device' button is in the top right. Below the tabs, the title 'Vedant_29 / Dashboards' is displayed. There is a 'New Dashboard' button. A search bar is on the right. The main area is titled 'Dashboards' and contains a table of dashboards:

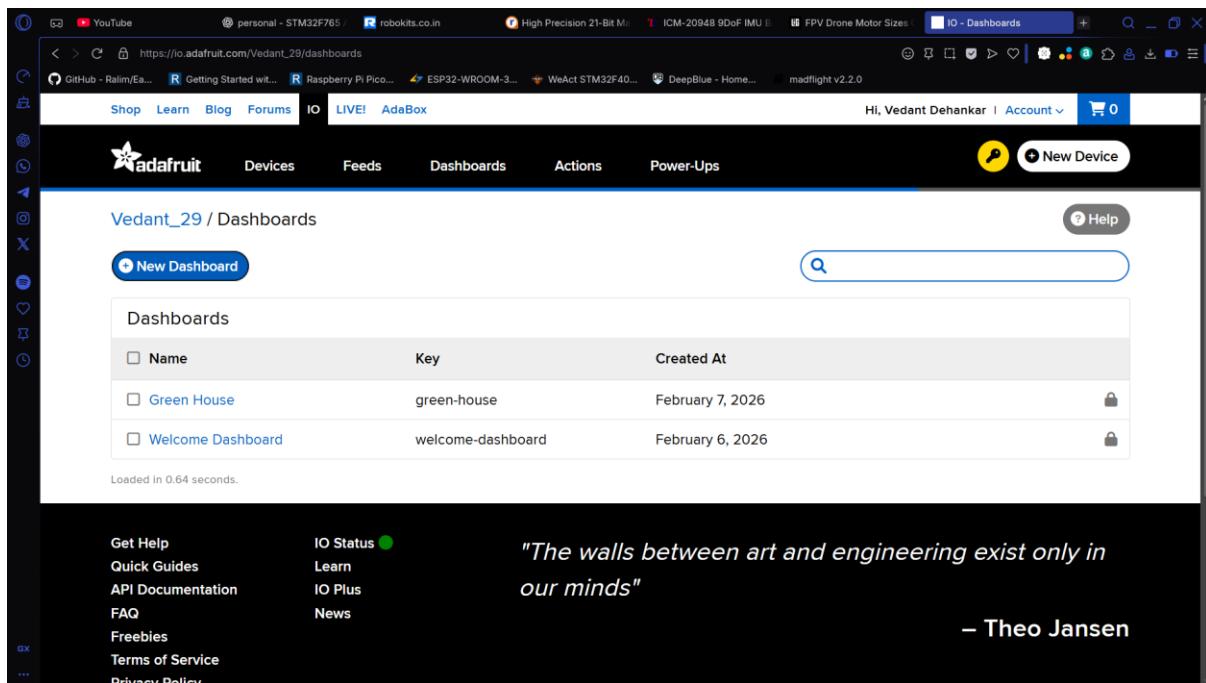
Name	Key	Created At
Welcome Dashboard	welcome-dashboard	February 6, 2026

At the bottom, there is a footer with links: Get Help, Quick Guides, API Documentation, FAQ, Freebies, Terms of Service, Privacy Policy, Website Accessibility, IO Status (green), Learn, IO Plus, News, and a quote by Theo Jansen: "The walls between art and engineering exist only in our minds" - Theo Jansen.

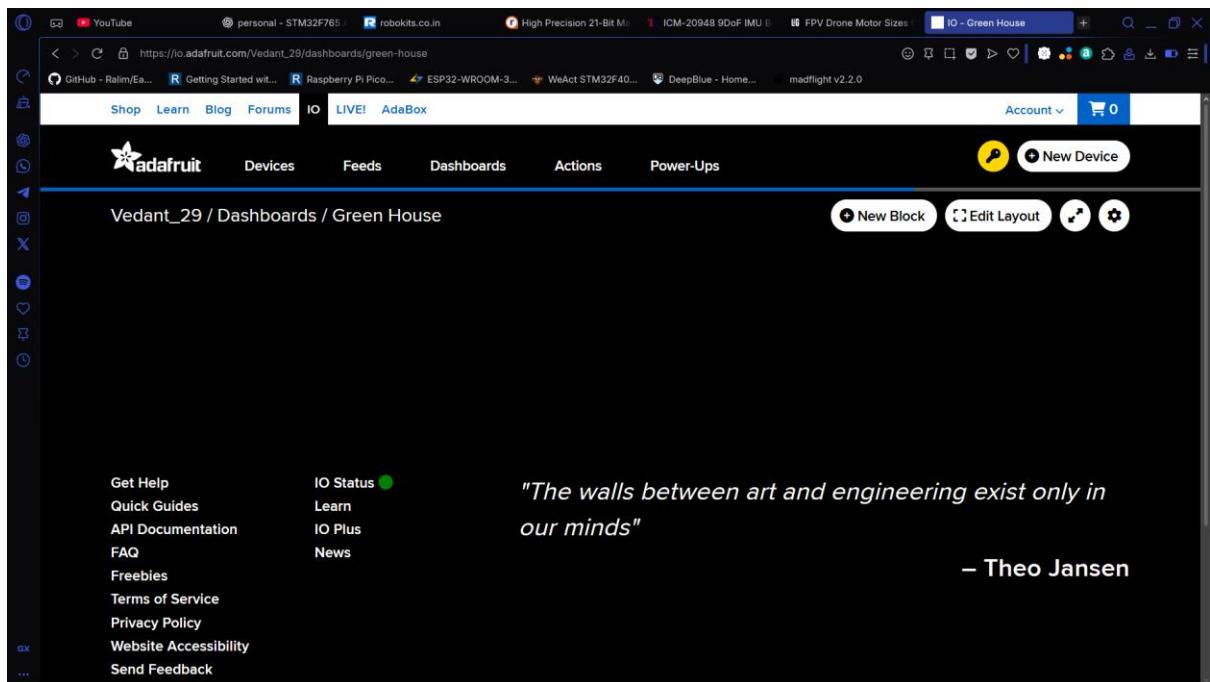
Step 11: Click on the new Dashboard button.



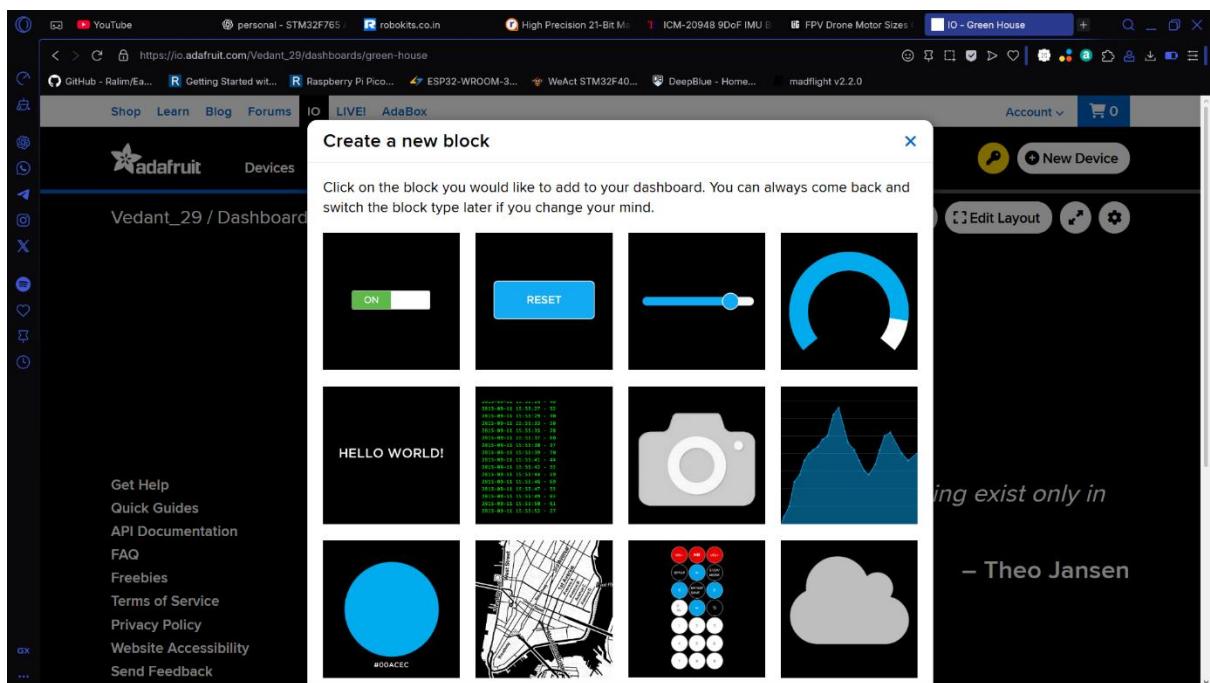
Step 12: A Create New Dashboard Pop up will appear, Give it a Name and click the Create button.



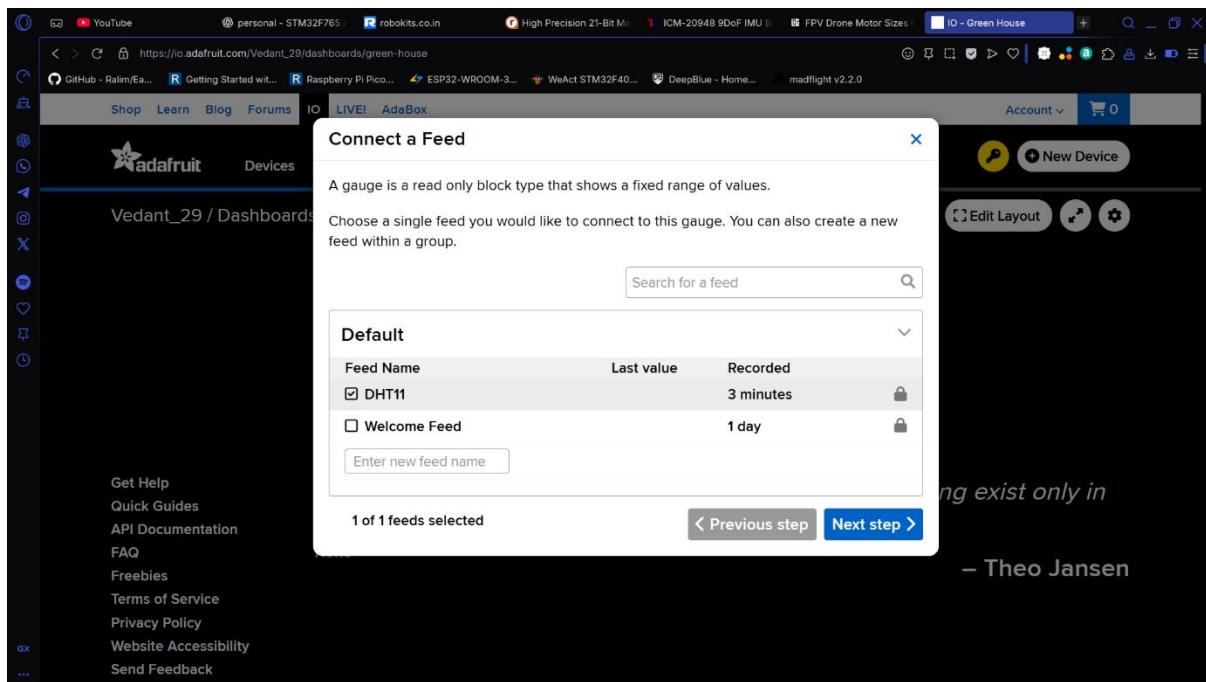
Step 13: A new Dashboard will appear, Click on the Dashboard you have created.



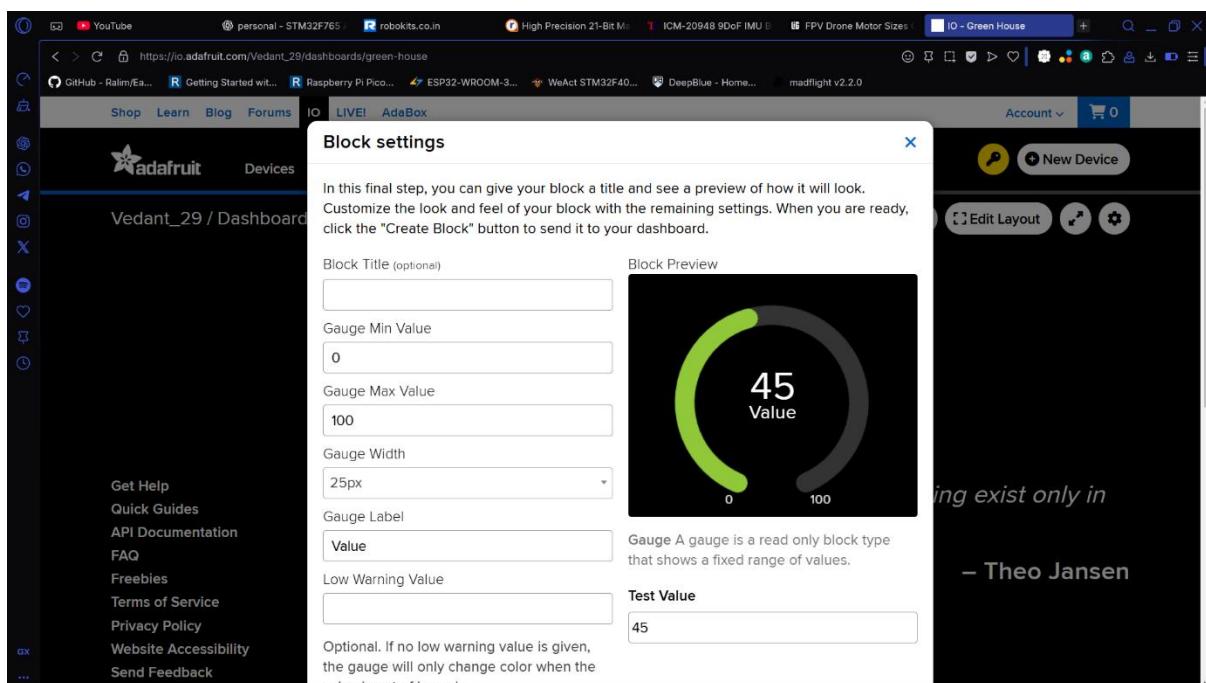
Step 14: You will be taken to a Dashboard, In this dashboard click on the New Block Button.



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.



The screenshot shows the Adafruit IO block settings interface for a gauge. On the left, there's a sidebar with links like Get Help, Quick Guides, API Documentation, FAQ, Freebies, Terms of Service, Privacy Policy, Website Accessibility, and Send Feedback. The main area has several input fields and options:

- A text input field with the value "45".
- A section titled "Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds." with a "High Warning Value" input field.
- A section titled "Decimal Places" with a value of "2".
- A note: "Number of decimal places to display when value is a number. Defaults to 2."
- A checkbox labeled "Show Icon".
- A note: "When checked, show an icon with the value."
- A dropdown menu labeled "Icon".
- A note: "Show this icon next to the value."

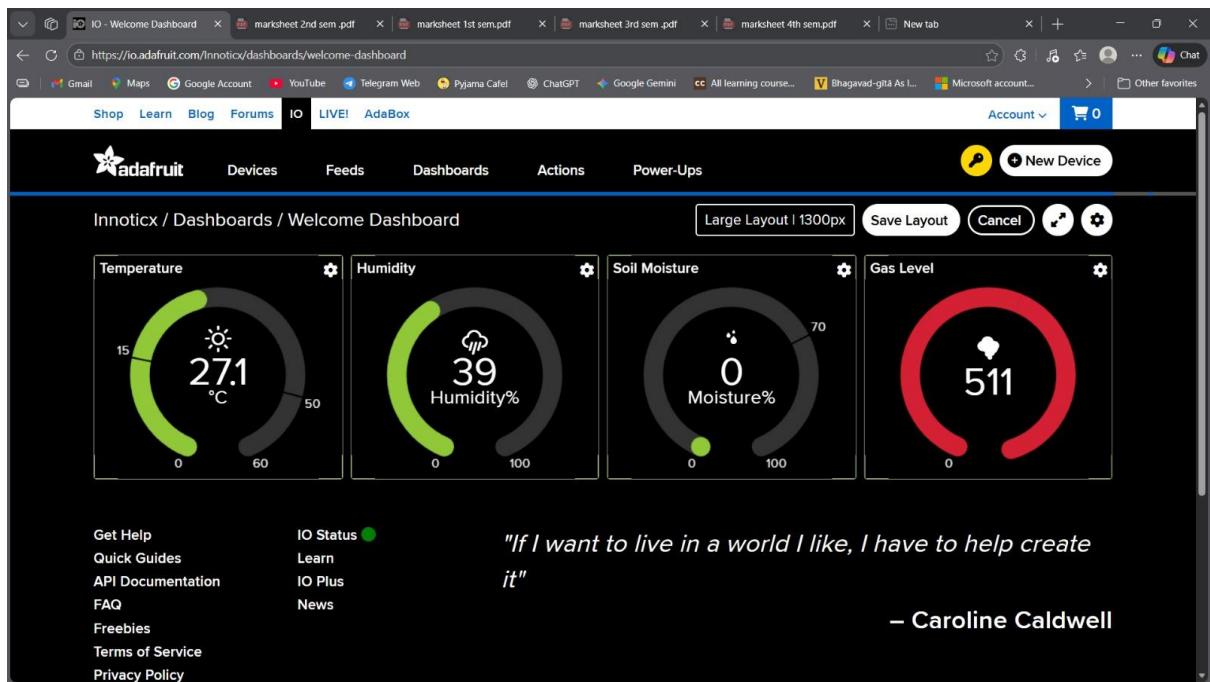
At the bottom right are buttons for "Previous step" and "Create block".

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

The screenshot shows the Adafruit IO dashboard for the "Green House" environment. At the top, there are tabs for Shop, Learn, Blog, Forums, IO, LIVE!, and AdaBox. Below the tabs, there are navigation links for Devices, Feeds, Dashboards, Actions, and Power-Ups. The main content area displays a gauge block with the value "0" and the word "Value". To the right of the gauge, there's a quote by Theo Jansen: "*The walls between art and engineering exist only in our minds*".

On the left side, there's a sidebar with links: Get Help, Quick Guides, API Documentation, FAQ, Freebies, Terms of Service, Privacy Policy, IO Status (green), Learn, IO Plus, and News.

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



In the end your dashboard will look like this.

Now we are done with the Adafruit IO setup

Adafruit Code

```
Adafruitio_IOT.ino
1 #include <WiFi.h>
2 #include <DHT.h>
3 #include "Adafruit_MQTT.h"
4 #include "Adafruit_MQTT_Client.h"
5
6 // ----- WiFi -----
7 #define WIFI_SSID "YOUR SSID"
8 #define WIFI_PASS "YOUR PASSWORD"
9
10 // ----- Adafruit IO -----
11 #define AIO_SERVER      "io.adafruit.com"
12 #define AIO_PORT        1883
13 #define AIO_USERNAME    "YOUR USERNAME"
14 #define AIO_KEY         "YOUR API KEY"
15
16 // ----- Pins -----
17 #define DHT_DATA_PIN   26
18 #define DHTTYPE        DHT11
19
20 #define SOIL_AO_PIN     34
21
22 #define MQ2_AO_PIN      35
23
24 // ----- Objects -----
25 WiFiClient client;
26 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_PORT,
27                           AIO_USERNAME, AIO_KEY);
28
29 Adafruit_MQTT_Publish feedTemp =
30   Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/temperature");
31
32 Adafruit_MQTT_Publish feedHum =
33   Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/humidity");
34
35 Adafruit_MQTT_Publish feedSoil =
36   Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/soil");
37
38 Adafruit_MQTT_Publish feedGas =
39   Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/gas");
40
41 DHT dht(DHT_DATA_PIN, DHTTYPE);
42
```

```
43 // ----- MQTT CONNECT -----
44 void mqtt_connect() {
45     while (!mqtt.connected()) {
46         if (mqtt.connect() == 0) {
47             Serial.println("MQTT Connected");
48         } else {
49             Serial.println("MQTT Retry...");
50             delay(3000);
51         }
52     }
53 }
54
55 // ----- SETUP -----
56 void setup() {
57     Serial.begin(9600);
58
59     dht.begin();
60
61     WiFi.begin(WIFI_SSID, WIFI_PASS);
62     while (WiFi.status() != WL_CONNECTED) {
63         delay(500);
64         Serial.print(".");
65     }
66     Serial.println("\nWiFi Connected");
67
68     mqtt_connect();
69 }
70
71 // ----- LOOP -----
72 void loop() {
73     mqtt_connect();
74
75     float temperature = dht.readTemperature();
76     float humidity    = dht.readHumidity();
77
78     int soilRaw = analogRead(SOIL_A0_PIN);
79     float soilPercent = map(soilRaw, 3500, 1200, 0, 100);
80     soilPercent = constrain(soilPercent, 0, 100);
81
82     int mq2Raw = analogRead(MQ2_A0_PIN);
83 }
```

```
84 // ---- Serial ----
85 Serial.println("-----");
86 Serial.print("Temp: "); Serial.println(temperature);
87 Serial.print("Humidity: "); Serial.println(humidity);
88 Serial.print("Soil: "); Serial.println(soilPercent);
89 Serial.print("Gas: "); Serial.println(mq2Raw);
90
91 // ---- Publish (EXPLICIT TYPE) ----
92 feedTemp.publish(temperature);
93 feedHum.publish(humidity);
94 feedSoil.publish((float)soilPercent);
95 feedGas.publish((float)mq2Raw);
96
97 delay(10000);    // rate-limit safe
98 }
```

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