# 8. Work with Adafruit IO platform

Difficult Level: 

## A. Welcome to Adafruit IO

Want to make your project talk to the Internet? Connect your existing project to the Internet to log, stream, and interact with the data it produces? What about all this Internet-of-Things (IoT) stuff?

### What can Adafruit IO do for me?

Adafruit.io is a cloud service- that just means we run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

* Display your data in real-time, online.
  + And share that data with others!
* Make your existing electronics project internet-connected: Control motors, read sensor data, and more!
* Connect projects to web services like Slack, Mastodon, Discord, RSS Feeds, Weather Services
* Connect your project to other internet-enabled devices
* Create No-Code electronics projects that connect to the internet.
* The best part? All of the above is do-able for **free**with Adafruit IO

Uploading your data to Adafruit IO is just the beginning.Interacting with that data is where the real magic is.

You can set up a device that sends you an email if the temperature drops below a certain level, monitor how much moisture is in your soil, and have Adafruit IO send you an SMS when it needs to be watered, log the air quality outside and get a notification when it is time to close your windows.

This is just the tip of the iceberg. When it comes to all the things you can do with Adafruit IO, the possibilities really are endless!

## B. Connect Arduino to Adafruit IO via MQTT

### What you need

* Arduino IDE software
* 1 x Arduino UNO R4 Wi-Fi
* 1 x USB 2.0 cable Type C
* 1 x Temperature and humidity sensor Module
* 3 x Jumper wires
* 1 x 52Pi Bread Board

### How to wiring circuit diagram?

|  |  |
| --- | --- |
| Arduino UNO R4 WIFI | Temperature & humidity sensor |
| 3.3V/5V | VCC |
| GND | GND |
| D2 | DATA |

Then you can copy and paste the code below.

**#include "WiFiS3.h"**

**#include "Adafruit\_MQTT.h"**

**#include "Adafruit\_MQTT\_Client.h"**

**#include "DHT.h"**

**#define DHT11\_PIN 2**

**DHT dht11(DHT11\_PIN, DHT11);  // Create a DHT object**

**#define WLAN\_SSID       "...your SSID..."**

**#define WLAN\_PASS       "...your password..."**

**#define AIO\_SERVER      "io.adafruit.com"**

**#define AIO\_SERVERPORT  1883         // use 8883 for SSL**

**#define AIO\_USERNAME    "...your AIO username (see**

**https://accounts.adafruit.com)..."**

**#define AIO\_KEY         "...your AIO key..."**

**// Create an WiFiClient class to connect to the MQTT server.**

**WiFiClient wifiClient;**

**// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.**

**Adafruit\_MQTT\_Client mqtt(&wifiClient, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);**

**Adafruit\_MQTT\_Publish temperature = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/temperature");**

**Adafruit\_MQTT\_Publish humidity = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/humidity");**

**void setup() {**

**// Create serial connection and wait for it to become available.**

**Serial.begin(9600);**

**dht11.begin();  // Initialize the DHT sensor**

**while (!Serial) {**

**;**

**}**

**// Connect to WiFi**

**Serial.print("Attempting to connect to WPA SSID: ");**

**Serial.println(WLAN\_SSID);**

**while (WiFi.begin(WLAN\_SSID, WLAN\_PASS) != WL\_CONNECTED) {**

**// failed, retry**

**Serial.print(".");**

**delay(5000);**

**}**

**Serial.println("You're connected to the network");**

**Serial.println();**

**Serial.println(WiFi.localIP());**

**}**

**void loop() {**

**delay(5000);**

**MQTT\_connect();**

**float humi  = dht11.readHumidity();**

**// read temperature as Celsius**

**float tempC = dht11.readTemperature();**

**// read temperature as Fahrenheit**

**Serial.print(F("\nSending temperature val "));**

**if (! temperature.publish(tempC)) {**

**Serial.println(F("Failed"));**

**} else {**

**Serial.println(F("OK!"));**

**}**

**Serial.print(F("\nSending humidity val "));**

**if (! humidity.publish(humi)) {**

**Serial.println(F("Failed"));**

**} else {**

**Serial.println(F("OK!"));**

**}**

**}**

**void MQTT\_connect() {**

**int8\_t ret;**

**// Stop if already connected.**

**if (mqtt.connected()) {**

**return;**

**}**

**Serial.print("Connecting to MQTT... ");**

**uint8\_t retries = 3;**

**while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected**

**Serial.println(mqtt.connectErrorString(ret));**

**Serial.println("Retrying MQTT connection in 5 seconds...");**

**mqtt.disconnect();**

**delay(5000);  // wait 5 seconds**

**retries--;**

**if (retries == 0) {**

**// basically die and wait for WDT to reset me**

**while (1);**

**}**

**}**

**Serial.println("MQTT Connected!");**

**}**

Firstly, you can't run this code directly, because you have to fill something in.

**#define WLAN\_SSID       "...your SSID..."**

**#define WLAN\_PASS       "...your password..."**

**#define AIO\_SERVER      "io.adafruit.com"**

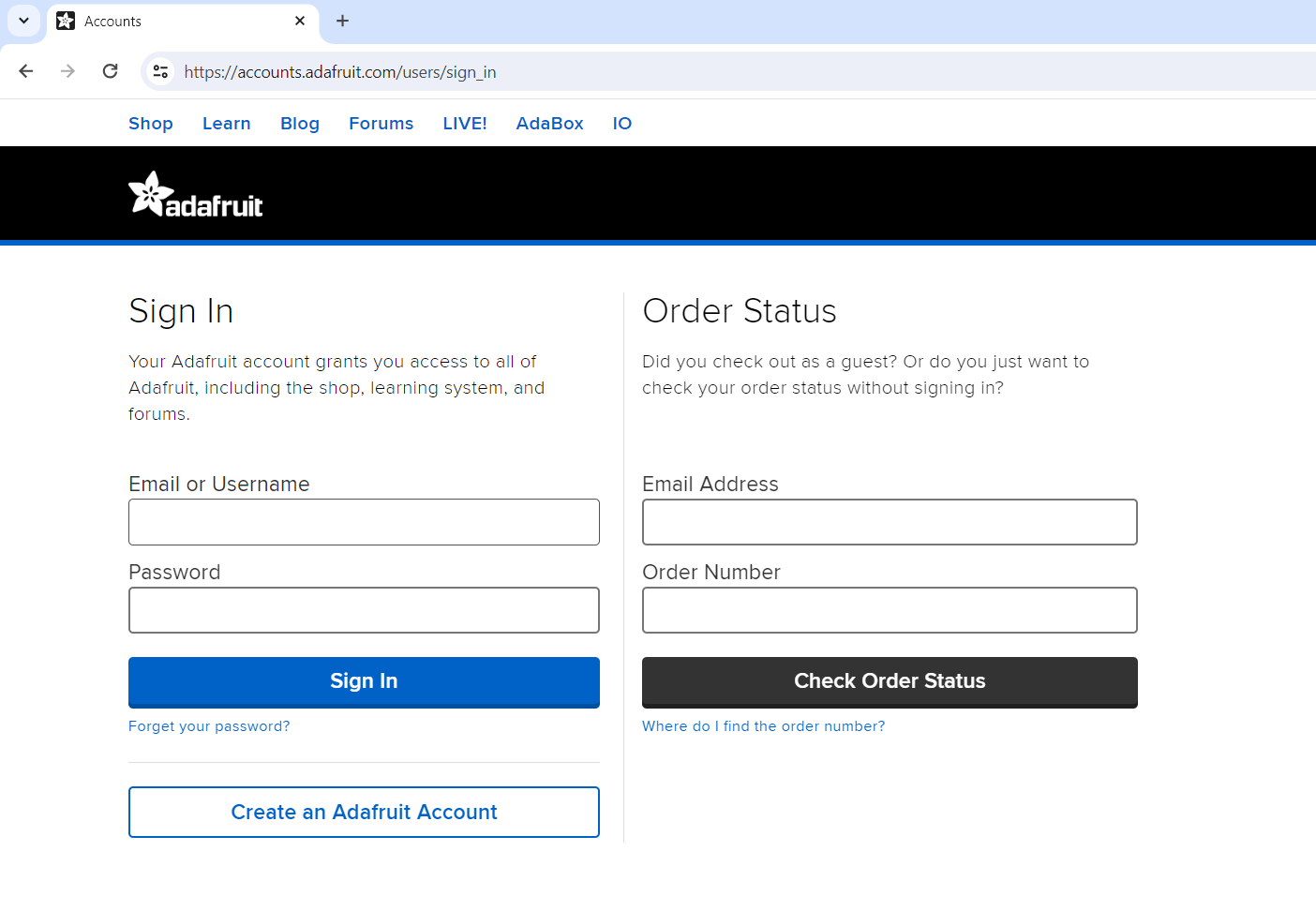
**#define AIO\_SERVERPORT  1883         // use 8883 for SSL**

**#define AIO\_USERNAME    "...your AIO username (see**

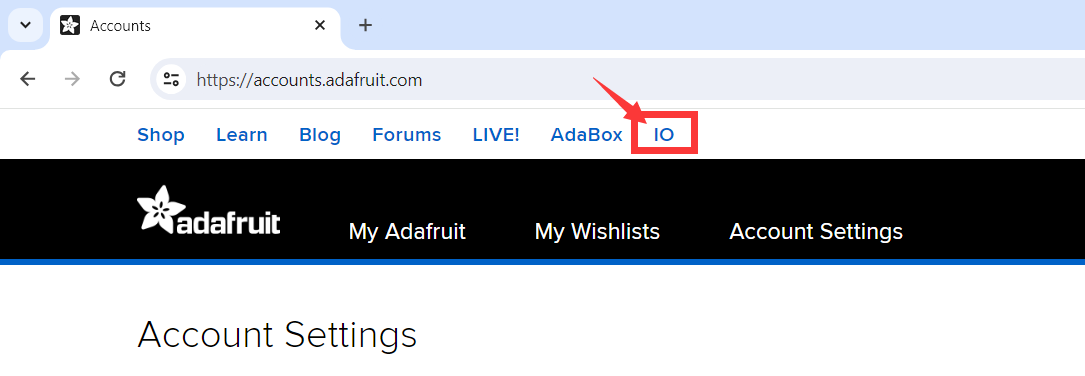
**https://accounts.adafruit.com)..."**

**#define AIO\_KEY         "...your AIO key..."**

So how can we fill it in, open the <https://accounts.adafruit.com> Go to the site and sign up, then go to Adafruit IO.



And then you can open this.

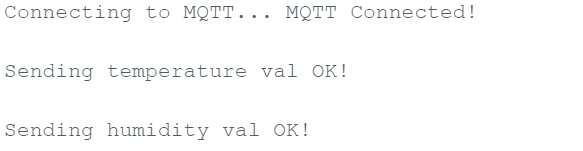


You can see a key with yellow here. Click it.



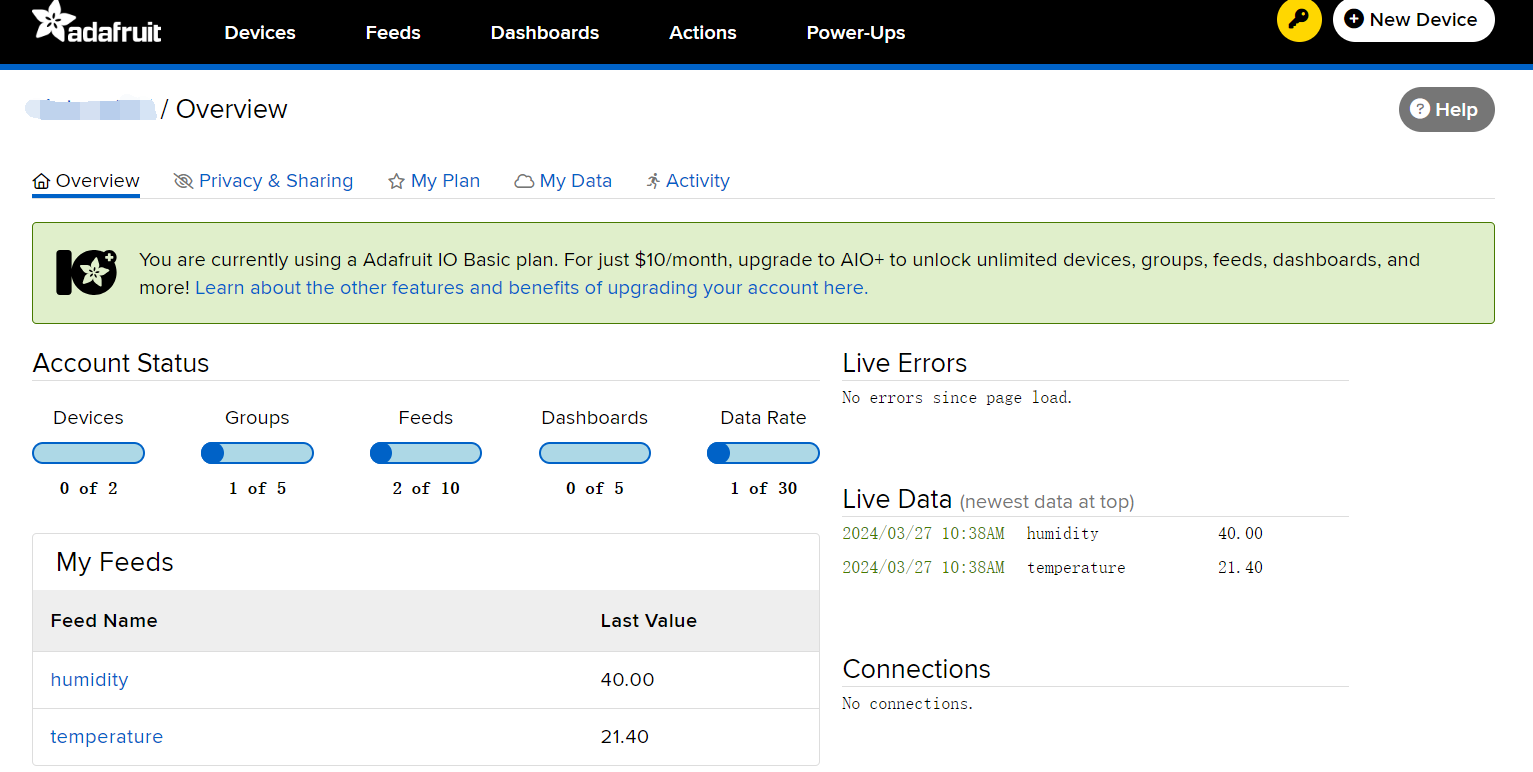
Then you can see your AIO\_USERNAME and AIO\_KEY, copy them to your arduino code and upload it.

Then you can open the serial monitor and will get this in the serial monitor.



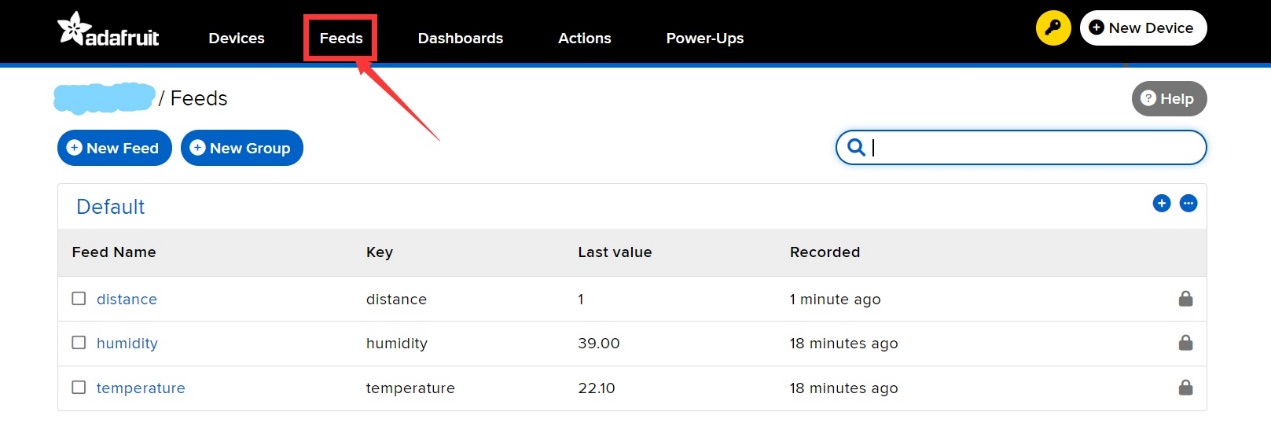
## C. See the values by the Arduino from the Adafruit IO

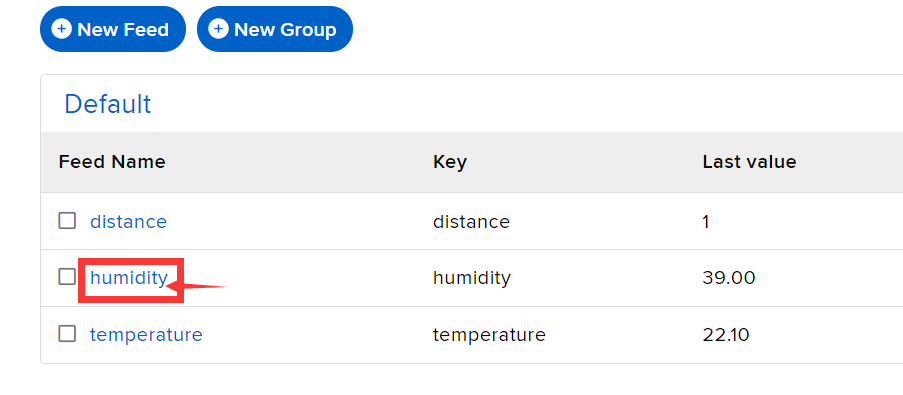
Open the [Adafruit IO](https://io.adafruit.com) website, and sign in your account, you can see the values.

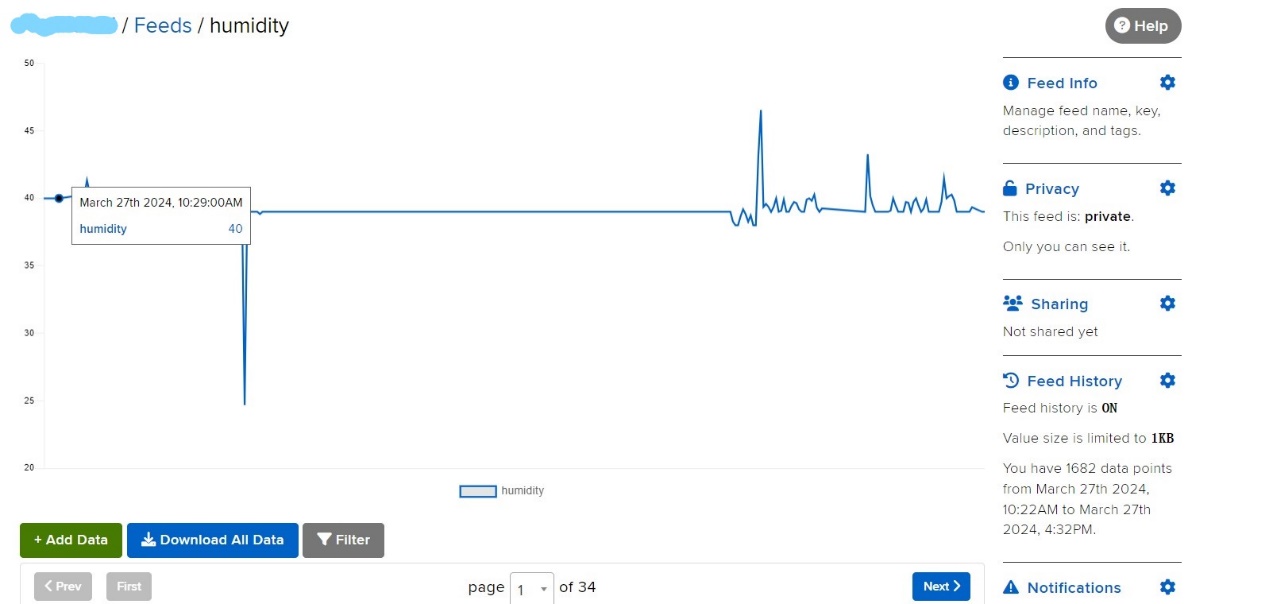


You can add more sensors and read their data by changing the code. (Note: The speed limit of Adafruit IO is 30 data per minute, if you need to upload more data, you can choose to open Adafruit IO Plus (AIO+)) Of course, this AIO+ has more powerful tools to help us better use Adafruit IO.

You can also open the feed to see the linear evolution of its data.







## D. Extension

Read the distance sensor and send it to Adafruit IO, which will output 1 when something is blocking the sensor and 0 if nothing is blocking it.

### What you need

* Arduino IDE software
* 1 x Arduino UNO R4 Wi-Fi
* 1 x USB 2.0 cable type C
* 1 x Ultrasonic Sensor Module
* 4 x Female-Male Jumper wires
* 1 x 52Pi Bread Board

### How to wiring circuit diagram?

|  |  |
| --- | --- |
| Arduino UNO R4 WIFI | Ultrasonic Sensor Module |
| 5V | VCC |
| GND | GND |
| D9 | Trig |
| D8 | Echo |

Then copy and paste the code to the IDE to upload it.

**#include "WiFiS3.h"**

**#include "Adafruit\_MQTT.h"**

**#include "Adafruit\_MQTT\_Client.h"**

**#define WLAN\_SSID       "...your SSID..."**

**#define WLAN\_PASS       "...your password..."**

**#define AIO\_SERVER      "io.adafruit.com"**

**#define AIO\_SERVERPORT  1883         // use 8883 for SSL**

**#define AIO\_USERNAME    "...your AIO username (see**

**https://accounts.adafruit.com)..."**

**#define AIO\_KEY         "...your AIO key..."**

**int trigPin = 9;    // TRIG pin**

**int echoPin = 8;    // ECHO pin**

**float distance\_cm,duration\_us;uint32\_t flag;**

**// Create an WiFiClient class to connect to the MQTT server.**

**WiFiClient wifiClient;**

**// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.**

**Adafruit\_MQTT\_Client mqtt(&wifiClient, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);**

**Adafruit\_MQTT\_Publish distance = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/distance");**

**void setup() {**

**// Create serial connection and wait for it to become available.**

**Serial.begin(9600);**

**// configure the trigger pin to output mode**

**pinMode(trigPin, OUTPUT);**

**// configure the echo pin to input mode**

**pinMode(echoPin, INPUT);**

**while (!Serial) {**

**;**

**}**

**// Connect to WiFi**

**Serial.print("Attempting to connect to WPA SSID: ");**

**Serial.println(WLAN\_SSID);**

**while (WiFi.begin(WLAN\_SSID, WLAN\_PASS) != WL\_CONNECTED) {**

**// failed, retry**

**Serial.print(".");**

**delay(5000);**

**}**

**Serial.println("You're connected to the network");**

**Serial.println();**

**Serial.println(WiFi.localIP());**

**}**

**void loop() {**

**delay(2000);**

**MQTT\_connect();**

**// generate 10-microsecond pulse to TRIG pin**

**digitalWrite(trigPin, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trigPin, LOW);**

**// measure duration of pulse from ECHO pin**

**duration\_us = pulseIn(echoPin, HIGH);**

**// calculate the distance**

**distance\_cm = 0.017 \* duration\_us;**

**if(distance\_cm > 30) flag = 0;**

**else flag = 1;**

**Serial.print(F("\nSending distance val "));**

**if (! distance.publish(flag)) {**

**Serial.println(F("Failed"));**

**} else {**

**Serial.println(F("OK!"));**

**}**

**}**

**void MQTT\_connect() {**

**int8\_t ret;**

**// Stop if already connected.**

**if (mqtt.connected()) {**

**return;**

**}**

**Serial.print("Connecting to MQTT... ");**

**uint8\_t retries = 3;**

**while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected**

**Serial.println(mqtt.connectErrorString(ret));**

**Serial.println("Retrying MQTT connection in 5 seconds...");**

**mqtt.disconnect();**

**delay(5000);  // wait 5 seconds**

**retries--;**

**if (retries == 0) {**

**// basically die and wait for WDT to reset me**

**while (1);**

**}**

**}**

**Serial.println("MQTT Connected!");**

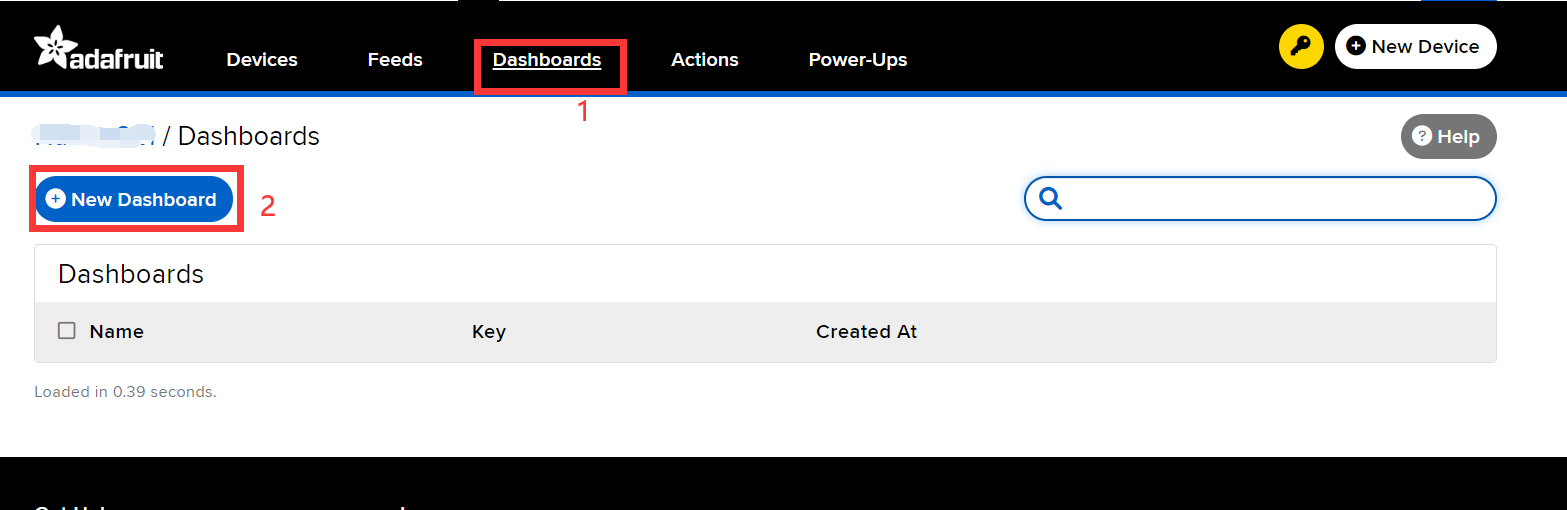
**}**

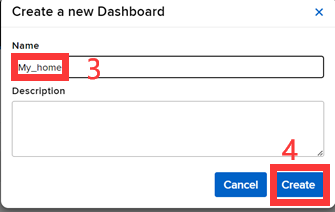
Then you can see on Adafruit IO if the sensor is occluded or not.

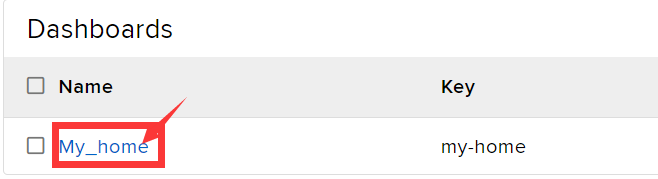
## E. Extension more

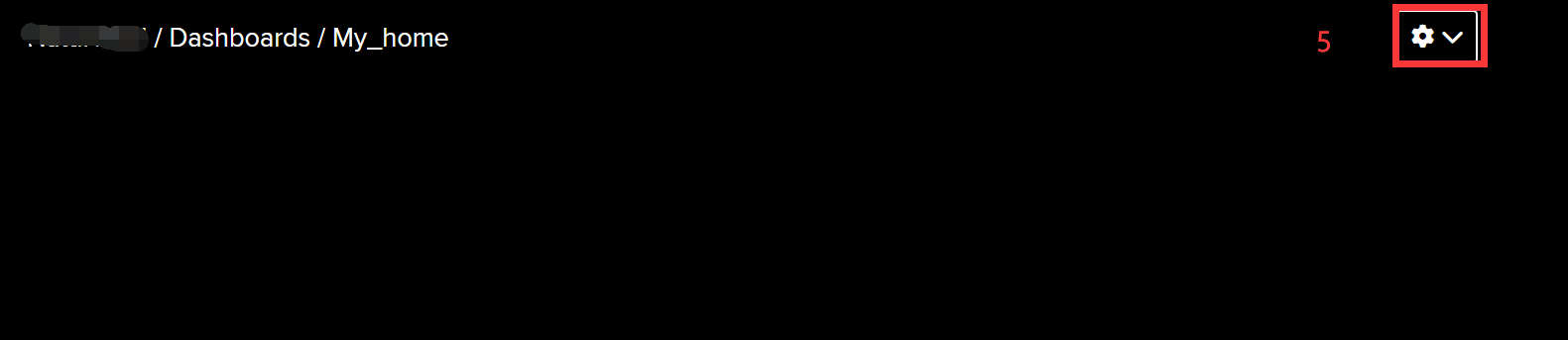
We read data from sensors in Arduino and send it to Adafruit IO, but it's just data. In this section, we are going to learn how to visualize this data.

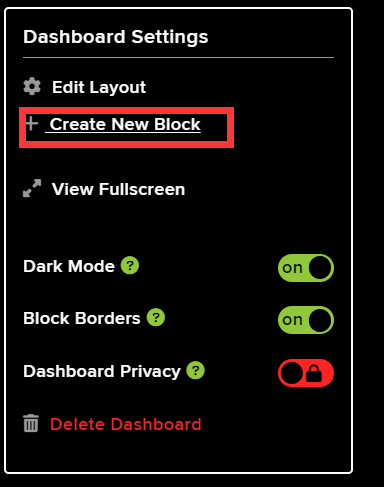
You can put the values to the Dashboards, so that you can have a good view of this data.



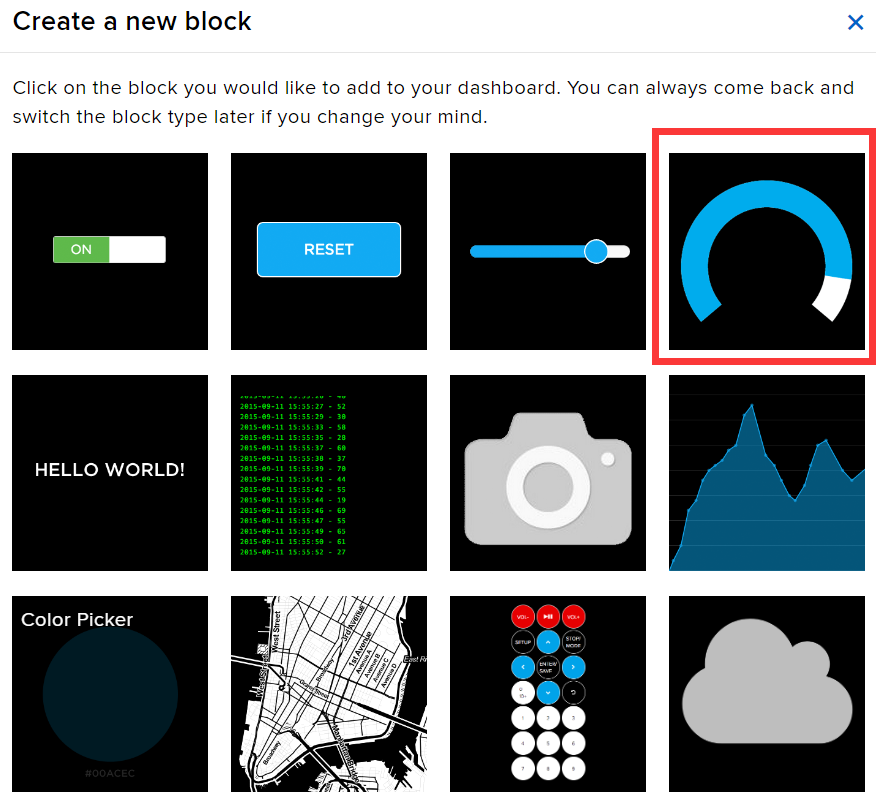




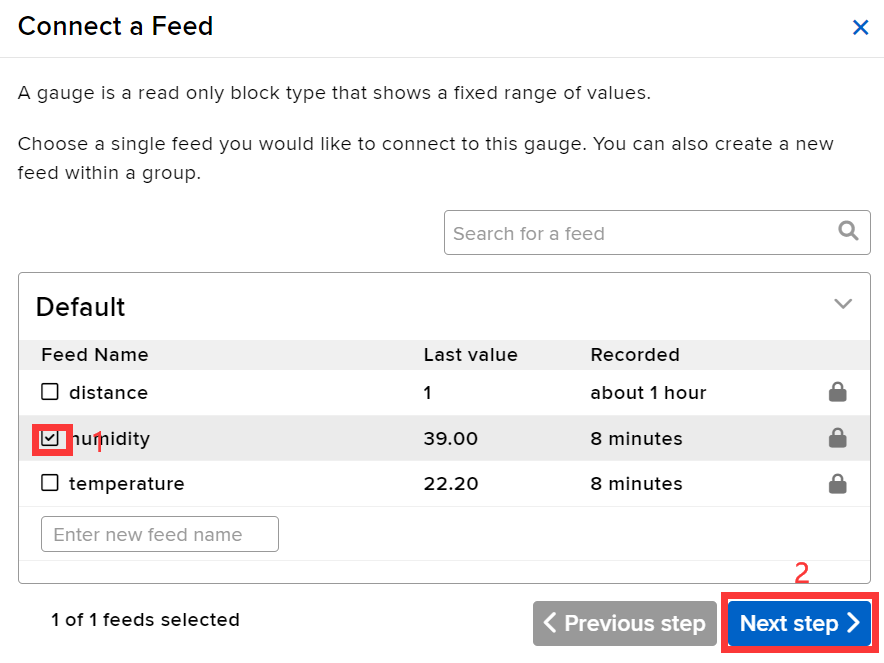




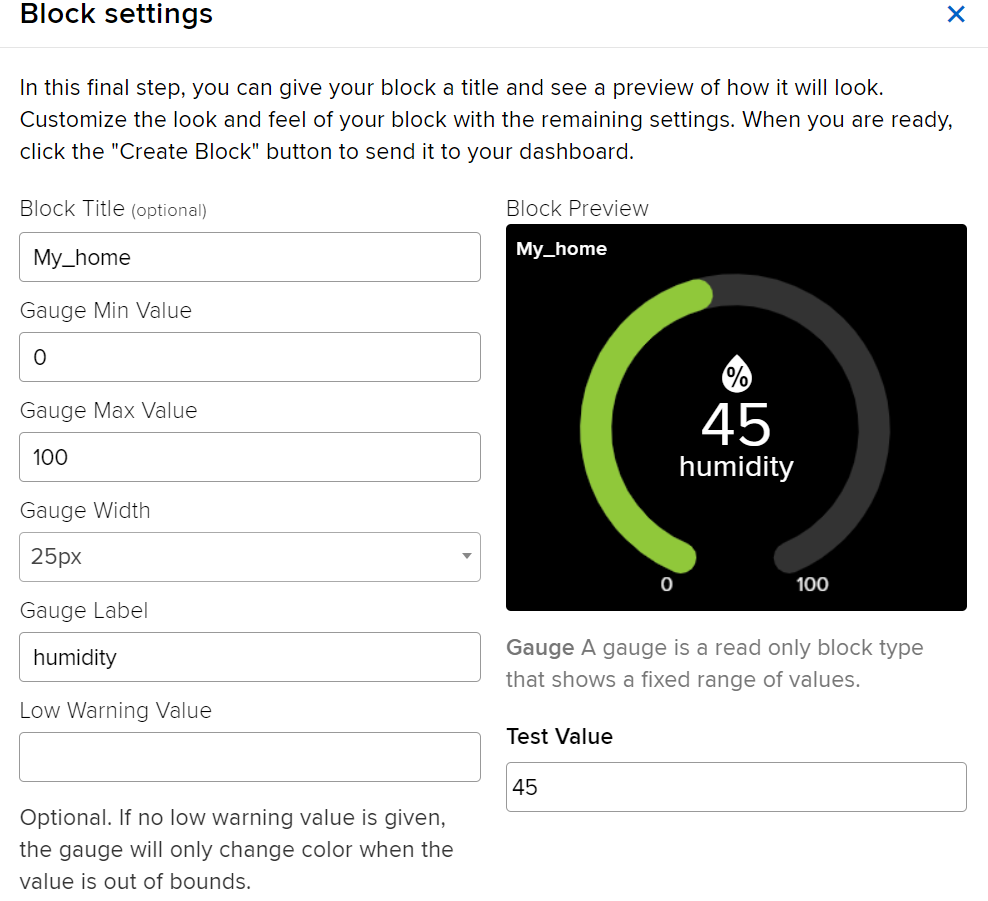
And then choose how you want to display the data, and for temperature and humidity I'll choose this:

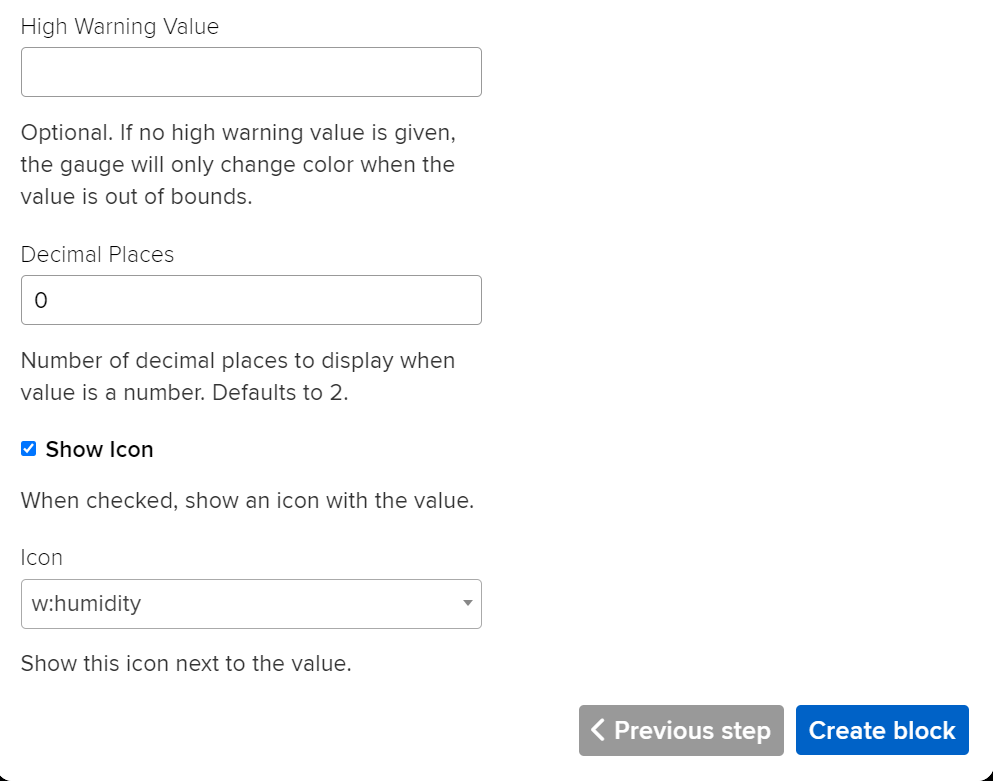


Choose the humidity feed.

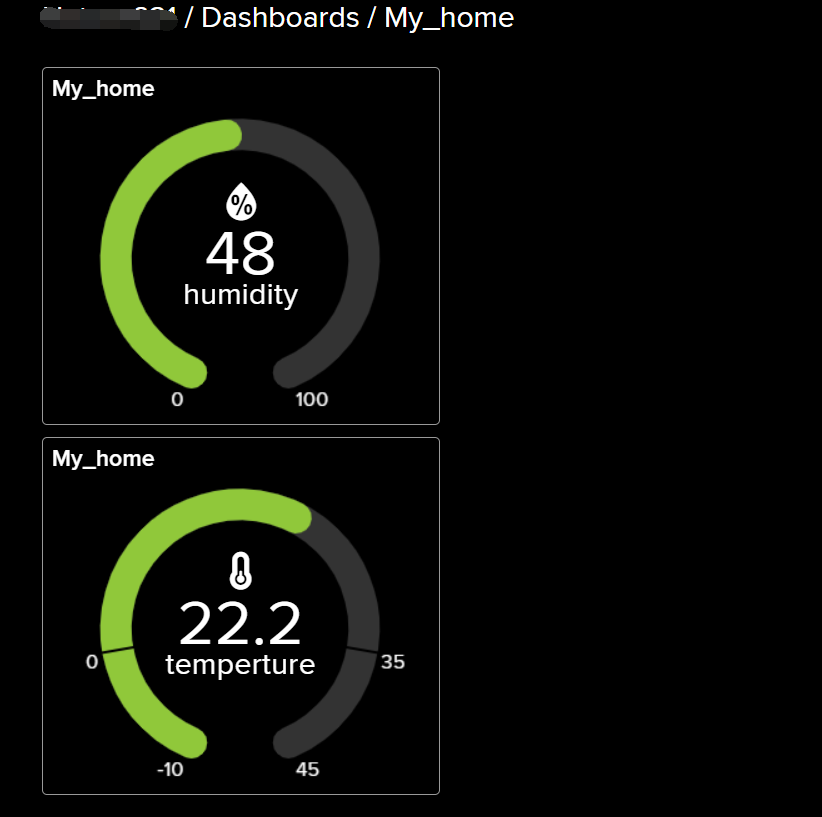


Then you can change the Dashboard's data range and some other information based on the facts.

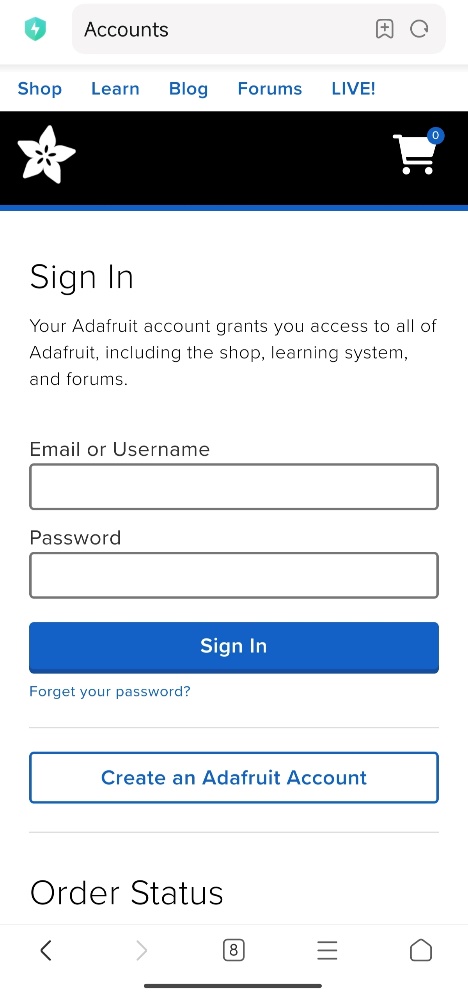




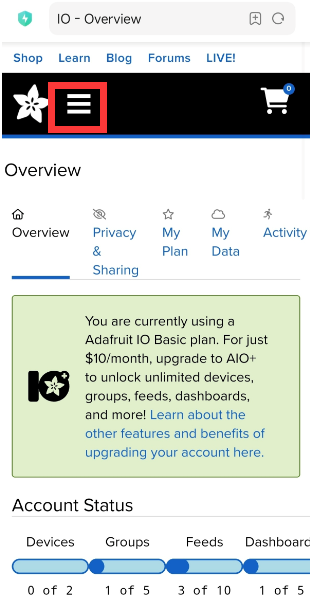
Then create the block. Connect the Arduino UNO R4 WIFI and upload the part B code you can see the dashboard every 500ms. You can also create the temperature dashboard by this way.

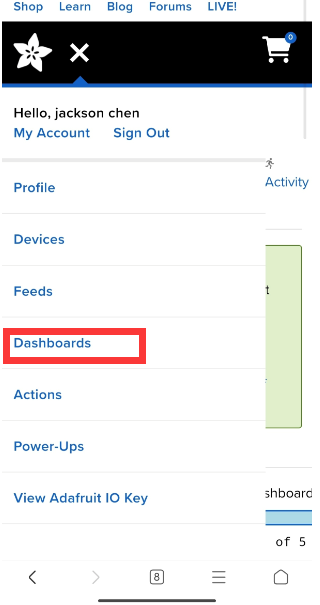


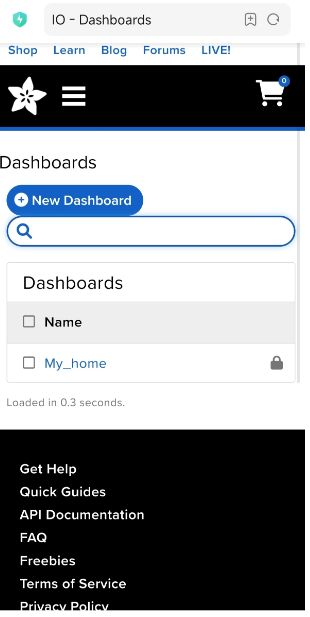
You can also see this in the phone browser. Just open you phone’s browser and enter the website: <https://io.adafruit.com> and log in.



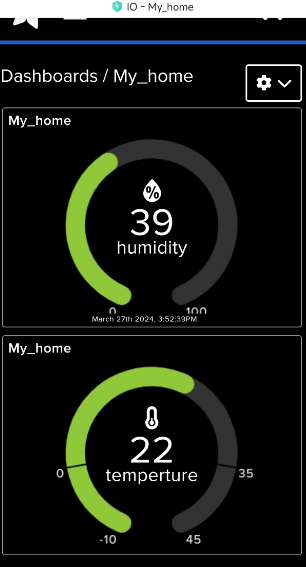
Next:







You can still see the data being updated on your phone.



**Congratulations！**



You have already got this new skills!!! and let’s get into next chapter!