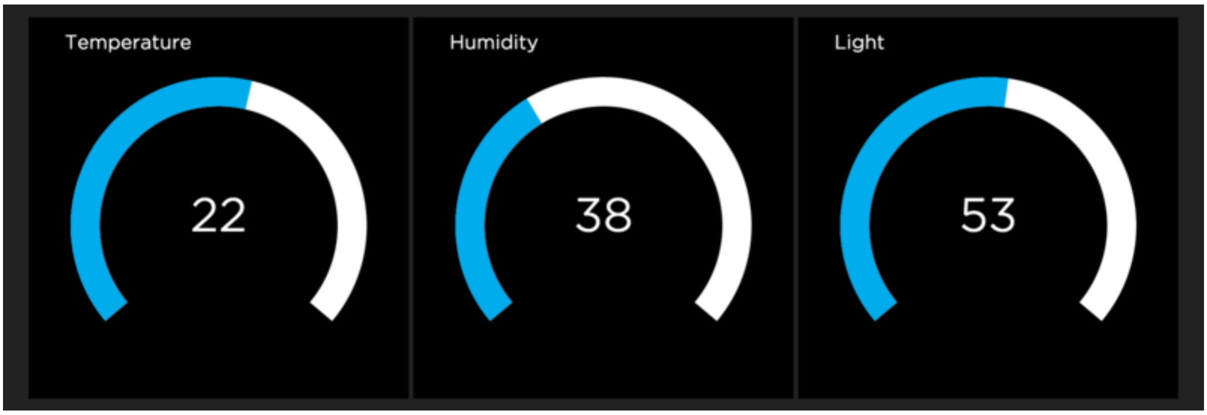
Create an Internet of Things Dashboard with Adafruit IO

Created by Marc-Olivier Schwartz

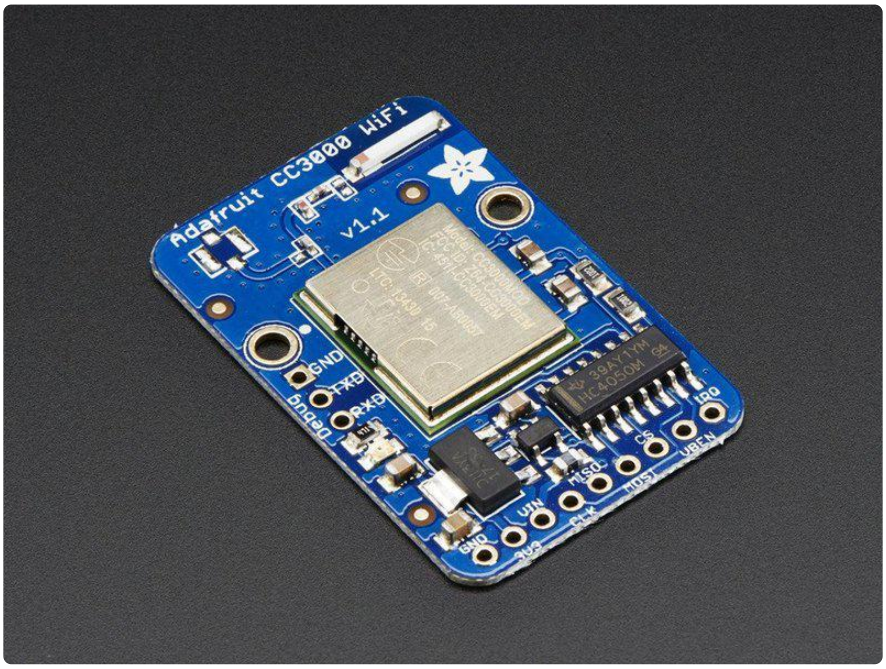


https://learn.adafruit.com/create-an-internet-of-things-dashboard-with-adafruit-dot-io

Last updated on 2023-08-29 02:48:44 PM EDT

Table of Contents

Introduction



Building Internet of Things projects with Arduino can be quite complicated: first, you need to find the right hardware & libraries for your project. Then, you need to find the right online service to send your data to. Finally, if you want to have some live graphical visualisation, you need to find or build an online dashboard for your project.

In this guide, we are going to build an Internet of Things dashboard using the Adafruit IO service. We will see that using Adafruit IO makes the process so much easier, as it will allow us to easily send data to the cloud from an Arduino board, and also easily building an Internet of Things dashboard just by dragging & dropping some elements.

As an example, we will connect several sensors to an Arduino Uno board, which will use the CC3000 WiFi chip to connect to the web. Then, we will send the measurement data to Adafruit IO & visualise it there in real-time, from anywhere in the world. Let's dive in!

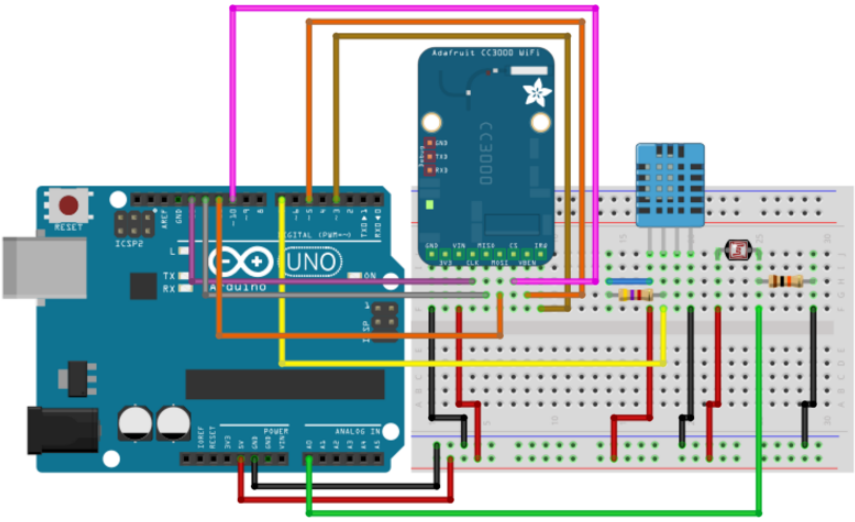
Hardware Configuration

The first step in this guide is to assemble our hardware. This is the list of components you will need for this guide:

- Arduino Uno R3
- Adafruit CC3000 WiFi breakout board
- DHT11 (or DHT22) temperature sensor + 4.7k Ohm resistor

- Photocell + 10k Ohm resistor
- Breadboard
- Jumper cables

To help you out, this is the schematic of the project:



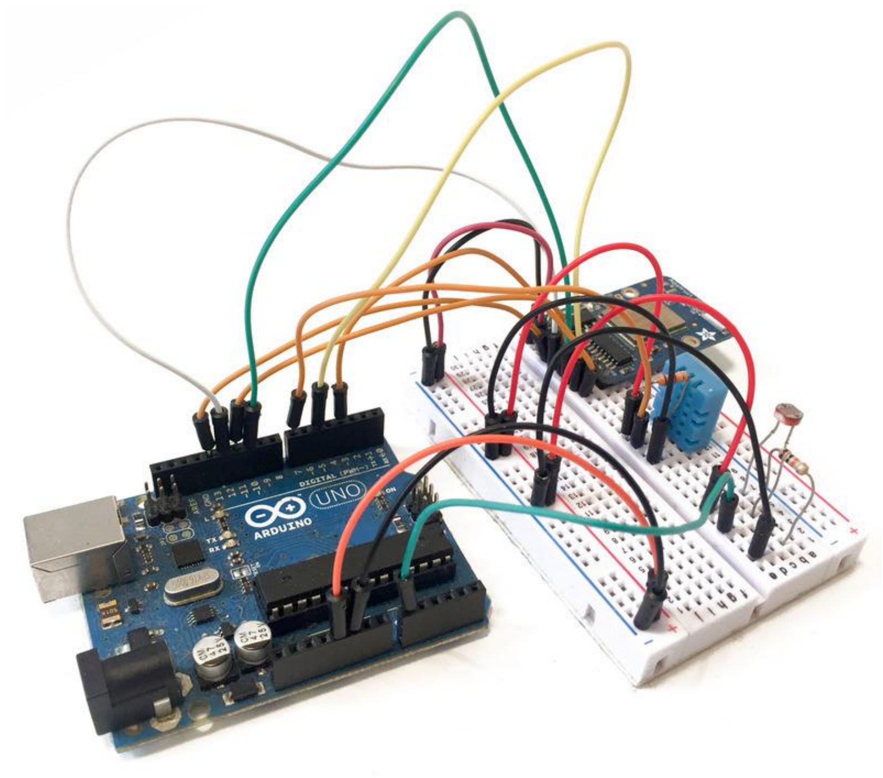
First, connect the Arduino Uno +5V pin to the red rail on the breadboard, and the ground pin to the blue rail. Then, place the DHT sensor and the CC3000 breakout board on the breadboard.

After that, connect pin number 1 of the DHT11 sensor (see the schematic) to the red rail on the breadboard, and pin number 4 to the blue rail. Also, connect pin number 2 of the sensor to pin number 7 of the Arduino board. To complete the connections of the DHT11 sensor, connect the 4.7k Ohm between pin number 1 and 2 of the sensor.

For the photocell, first place the cell in series with the 10k Ohm resistor on the breadboard. Then, connect the other end of the photocell to the red rail on the breadboard, and the other end of the resistor to the ground. Finally, connect the common pin to the Arduino Uno analog pin A0.

Now, the WiFi module. First, connect the IRQ pin of the CC3000 board to pin number 3 of the Arduino board, VBAT to pin 5, and CS to pin 10. Then, you need to connect the SPI pins to the Arduino board: MOSI, MISO, and CLK go to pins 11,12, and 13, respectively. Finally, take care of the power supply: Vin goes to the Arduino 5V (red power rail), and GND to GND (blue power rail).

This is the final result:

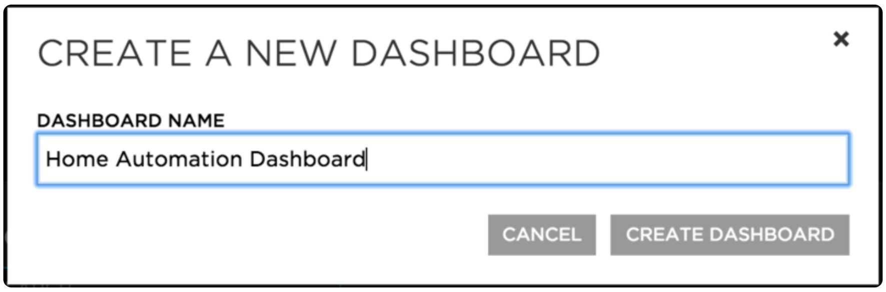


Create Your Adafruit.io Dashboard

Now that our hardware is ready, we are going to create our Adafruit IO dashboard. The first step is to go over to:

https://io.adafruit.com/dashboards ()

From there, create a new dashboard, and give it a name:

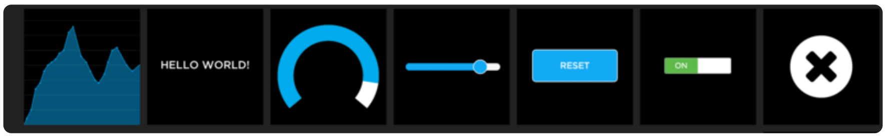


CREATE A NEW DASHBOARD

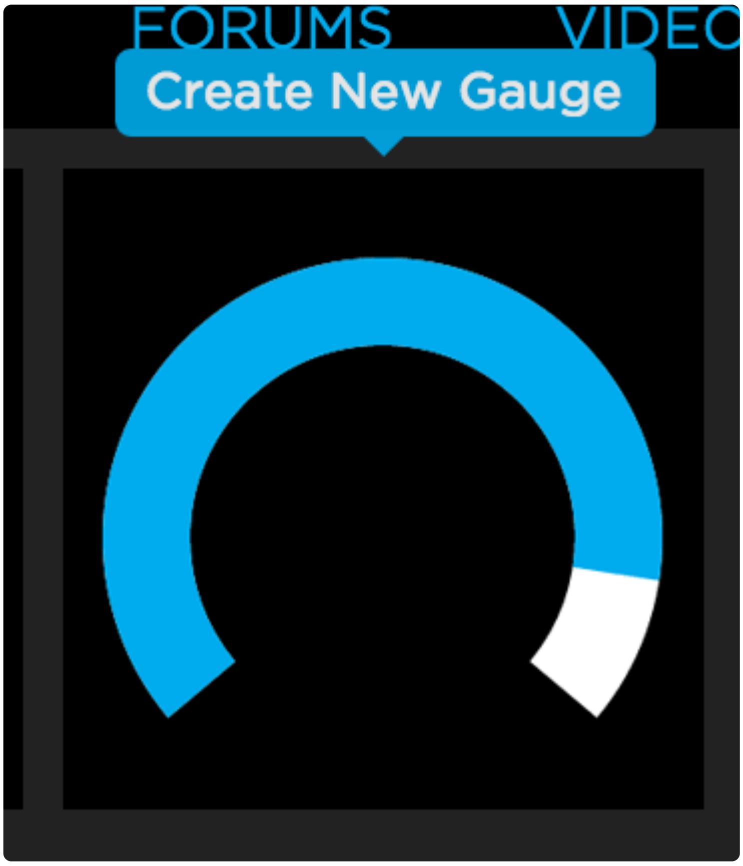
DASHBOARD NAME

Home Automation Dashboard

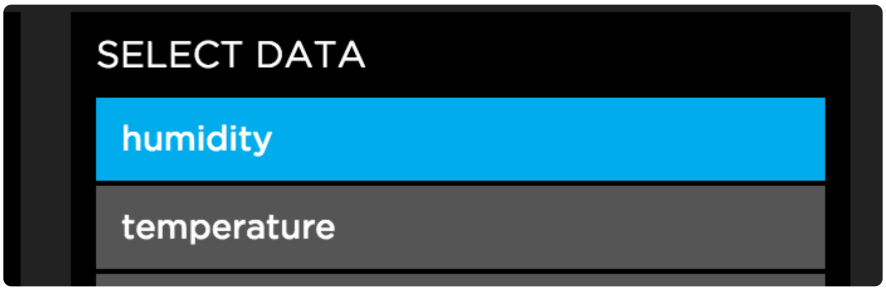
Then, it's time to add some elements. You will see that you have the choice between many different elements:



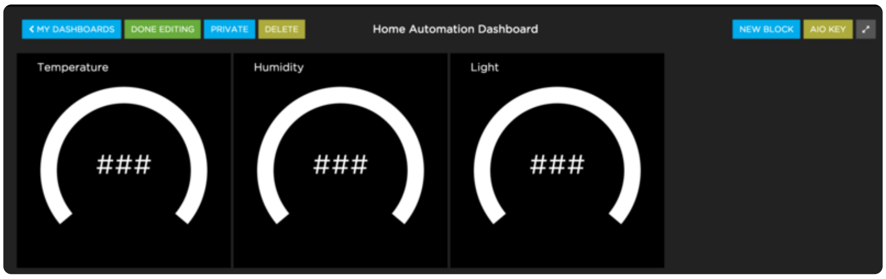
As we only have sensors sending numeric values in this project, I choose "Gauge" elements only:



When creating a new element, the dashboard will ask you which data feed you want to use. I created a new data feed per element of the dashboard, for example for humidity:



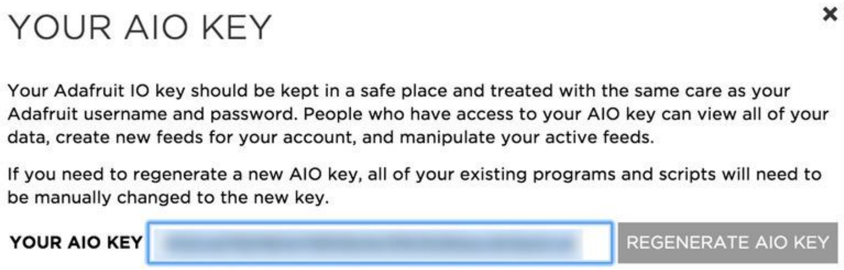
This is the final result, with three gauge elements in the dashboard:



Build the Arduino Sketch

Now that the dashboard is ready to receive & display some data, it's time to write our Arduino sketch, so the board can actually send data to the dashboard.

You will first need to get your AIO key, which is a unique key that identify your account on Adafruit IO. You can for example get it from your dashboard:



YOUR AIO 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 AIO 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 AIO key, all of your existing programs and scripts will need to be manually changed to the new key.

YOUR AIO KEY [redacted]

This tutorial is a bit out of date due to it using the PubSub library. Please use our Adafruit MQTT library found on Github.

Let's now build your Arduino sketch. You will need three libraries for this project:

- CC3000 library ()
- PubSub library () (deprecated)
- Adafruit MQTT library ()
- DHT library ()

To install a library, simply place the downloaded library folder inside your Arduino 'libraries' folder.

As the code is quite long, I will only cover the important parts here. Of course, you can find all the code inside the GitHub repository of the project. The sketch starts by including the required libraries:

```
#include &lt;Adafruit_CC3000.h&gt;
#include &lt;ccspi.h&gt;
#include &lt;SPI.h&gt;
#include &lt;PubSubClient.h&gt;
#include "DHT.h"
```

After that, we define the pin & type of the DHT sensor, and also create an instance of this sensor:

```
#define DHTPIN 7     // what pin we're connected to
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);
```

Then, that's where you will need to enter your WiFi network name and password:

```
#define WLAN_SSID       "your_wifi_network"
#define WLAN_PASS       "your_wifi_password"
#define WLAN_SECURITY   WLAN_SEC_WPA2
```

You also need to enter your user name and AIO key:

```
#define ADAFRUIT_USERNAME  "your_adafruit_username"
#define AIO_KEY  "your_aio_key"
```

After that, you will need to enter the different paths for your data feeds, that you created while building the dashboard. You might need to modify this part if you are using different names in the dashboard:

```
#define TEMPERATURE_PUBLISH_PATH "api/feeds/temperature/data/send.json"
#define HUMIDITY_PUBLISH_PATH "api/feeds/humidity/data/send.json"
#define LIGHT_PUBLISH_PATH "api/feeds/light/data/send.json"
```

We also need to create instances of the CC3000 client, and also of the PubSubClient that we will use to communicate with Adafruit IO:

```
Adafruit_CC3000_Client client = Adafruit_CC3000_Client();
PubSubClient mqttclient("io.adafruit.com", 1883, callback, client);
```

In the loop() function of the sketch, we first start by measuring data from the light level sensor & from the DHT11 sensor:

```
// Measure ambient light
float light_level_reading = analogRead(LIGHT_SENSOR_PIN);
int light_level = (int)(light_level_reading/1024.*100.);

// Measure temperature &amp; humidity
int h = dht.readHumidity();
int t = dht.readTemperature();
```

Then, we send the data to the respective data feeds, using the mqttclient.publish() function. We also wait a bit during each publish, so the client has the time to receive the data:

```
mqttclient.publish(TEMPERATURE_PUBLISH_PATH, (char *) String(t).c_str());
delay(2000);
mqttclient.publish(HUMIDITY_PUBLISH_PATH, (char *) String(h).c_str());
delay(2000);
```

```
mqttclient.publish(LIGHT_PUBLISH_PATH, (char *) String(light_level).c_str());
delay(2000);
```

Finally, we keep the connection with Adafruit IO alive using the mqttclient.loop() function:

```
mqttclient.loop();
```

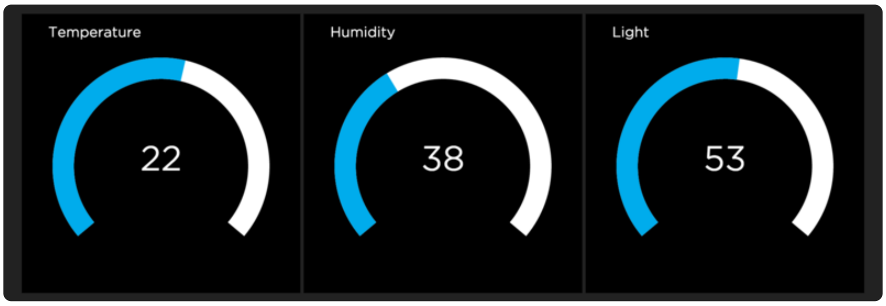
Note that you can find all the code for this project inside the corresponding GitHub repository:

https://github.com/openhomeautomation/adafruit-io-dashboard ()

Visualise Data on Adafruit.io

It's now time to finally test the project! The first step is to modify the Arduino sketch with your own settings, and then upload it to the board.

Then, go over to the dashboard. After a few seconds, you should see that the data appears inside the dashboard:



You can test your dashboard by putting your hand on top of the light level sensor: after a while the light level should go down in the dashboard.

How to Go Further

Congratulations, you just built an Internet of Things dashboard using Adafruit IO. You can visualize live data measurements inside your dashboard, from anywhere in the world.

There are many ways you can go further with this project. Using the flexibility of the Arduino platform, you can for example add more sensors to the project, and display them as well inside your dashboard.

Also, your data can perfectly come from different boards: you can for example have several projects like the one we built in this guide, and make them send data to the same dashboard. Therefore, you can monitor data about your whole home from a single place.