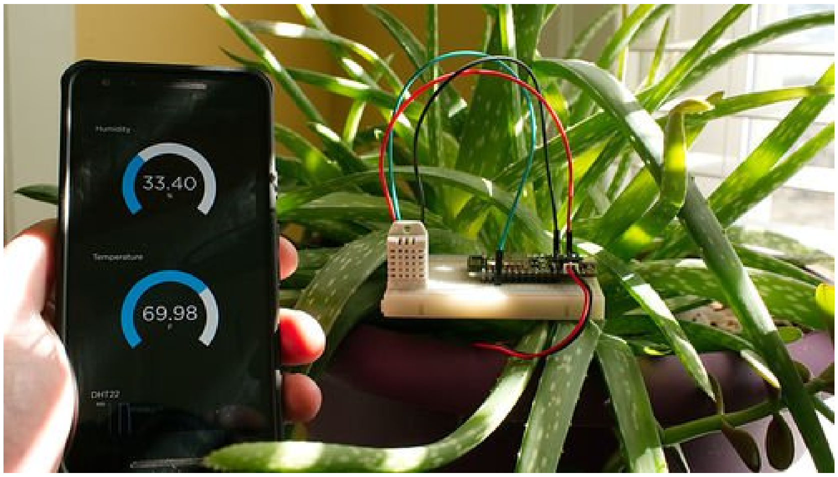
Adafruit IO Basics: Temperature & Humidity

Created by Todd Treece



Last updated on 2017-03-29 04:20:16 PM UTC

Guide Contents

Guide Contents 2

Overview 3

Adafruit IO Setup 5

- Creating the Feeds 6
- Adding the Line Chart Block 7

Wiring 9

Arduino Setup 11

Network Config 13

- WiFi Config 13
- FONA Config 14
- Ethernet Config 14

Code 16

Add an OLED 20

Overview

This guide is part of a series of guides that cover the basics of using Adafruit IO. It will show you how to send temperature and humidity values wirelessly to Adafruit IO from a DHT22 sensor.

If you haven't worked your way through the Adafruit IO feed and dashboard basics guides, you should do that before continuing with this guide so you have a basic understanding of Adafruit IO.

- Adafruit IO Basics: Feeds
- Adafruit IO Basics: Dashboards

You should go through the setup guides associated with your selected set of hardware, and make sure you have internet connectivity with the device before continuing. The following links will take you to the guides for your selected platform.

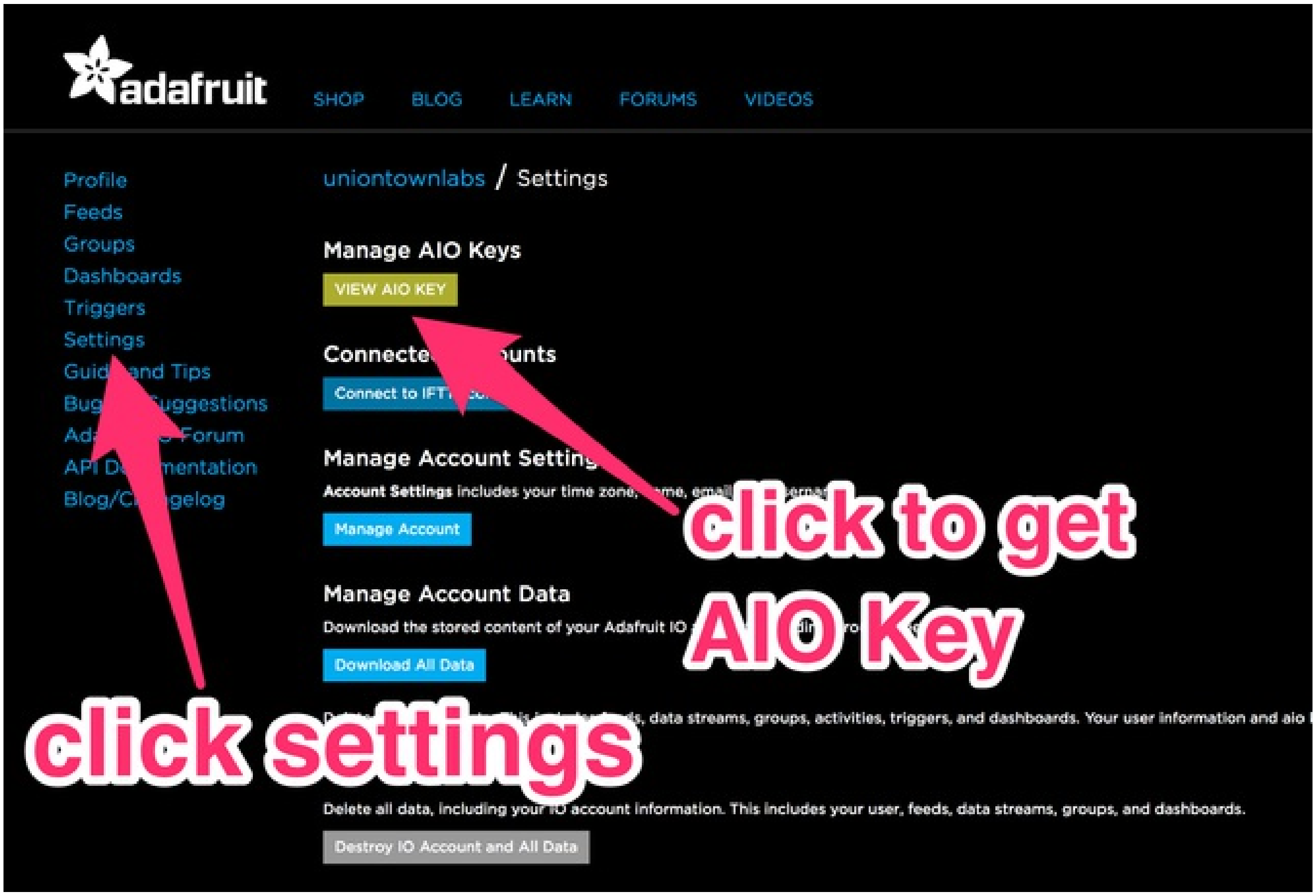
- Adafruit Feather HUZZAH ESP8266 Setup Guide

If you have went through all of the prerequisites for your selected hardware, you are now ready to move on to the Adafruit IO setup steps that are common between all of the hardware choices for this project. Let's get started!

Adafruit IO Setup

The first thing you will need to do is to login to Adafruit IO and visit the **Settings** page.

Click the **VIEW AIO KEY** button to retrieve your key.



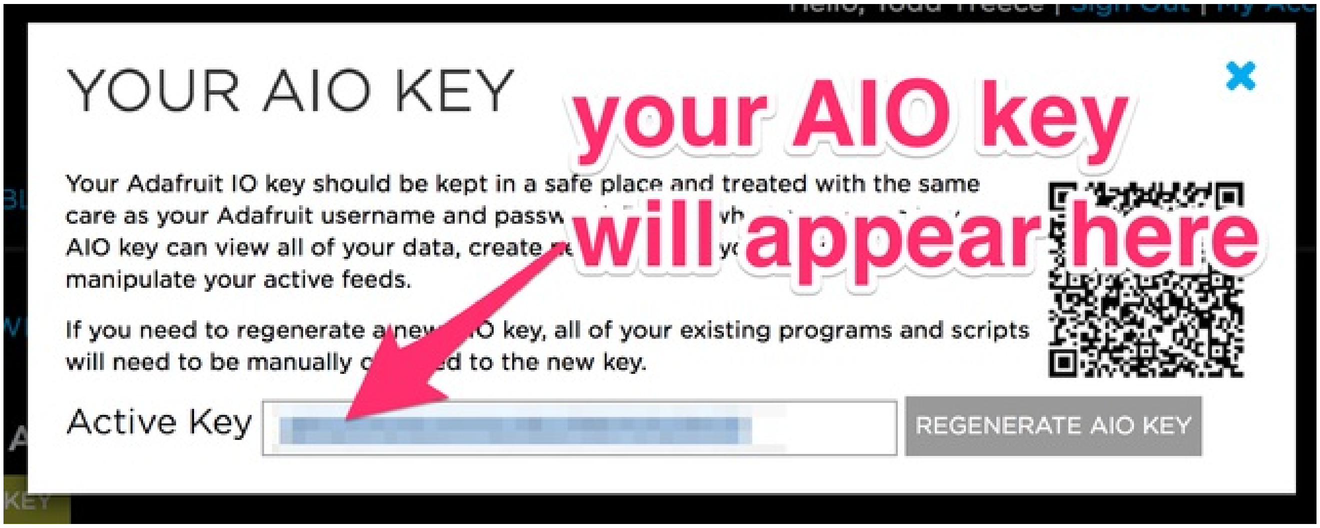
Account Settings includes your time zone, me, email erna

Download the stored content of your Adafruit IO

s, data streams, groups, activities, triggers, and dashboards. Your user information and aio

Delete all data, including your IO account information. This includes your user, feeds, data streams, groups, and dashboards.

A window will pop up with your Adafruit IO. Keep a copy of this in a safe place. We'll need it later.



YOUR AIO KEY

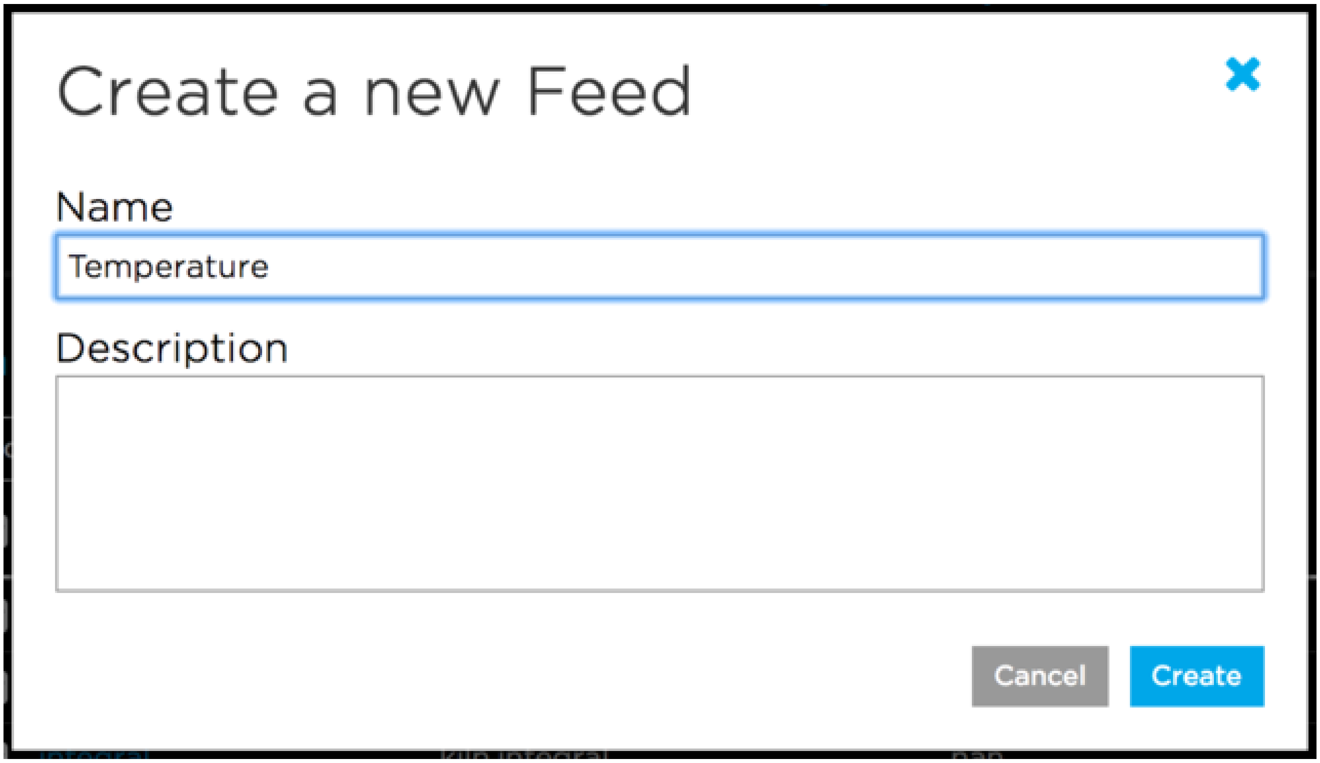
Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and passw... AIO key can view all of your data, create ... manipulate your active feeds.

If you need to regenerate a new ...O key, all of your existing programs and scripts will need to be manually c...d to the new key.

Active Key [redacted]

Creating the Feeds

First, you will need to create a feed called **Temperature.**

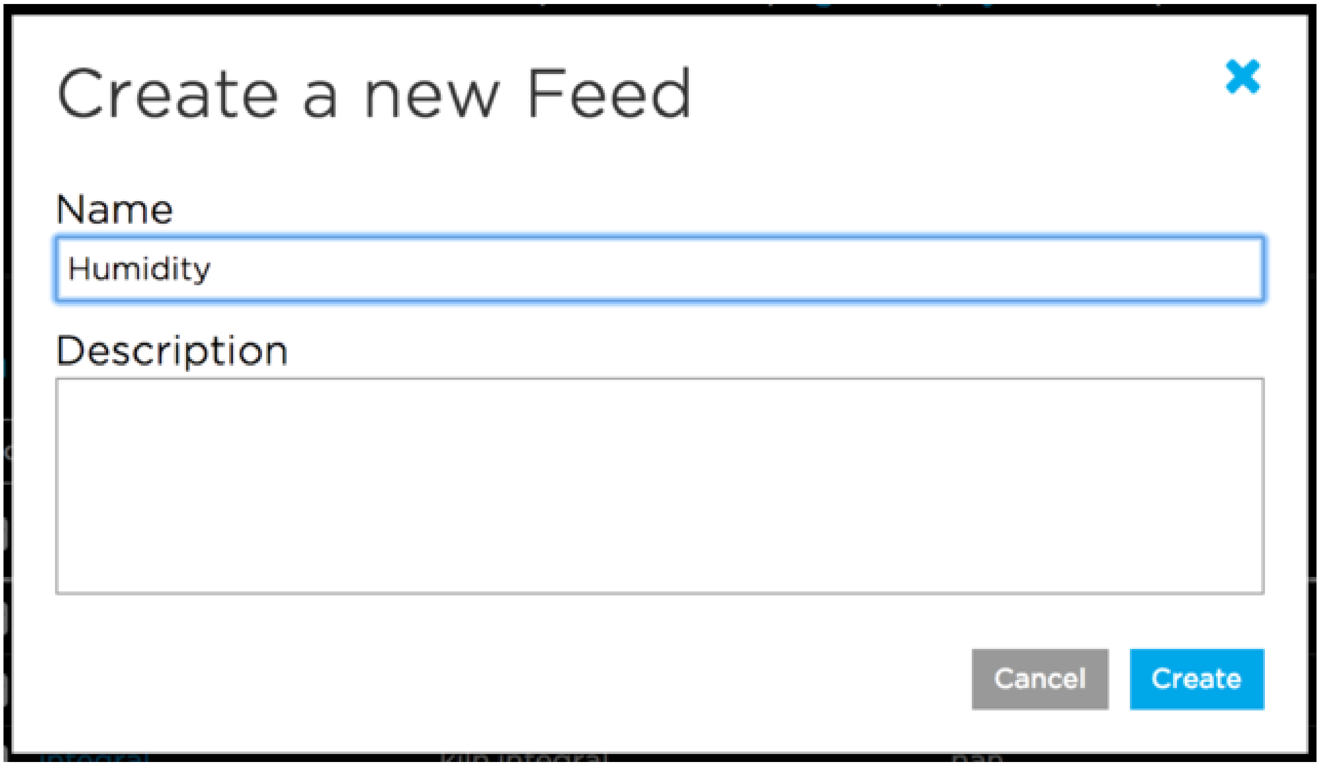


Create a new Feed

Name: Temperature

Description:

You will also need to create a feed called **Humidity**.



Create a new Feed

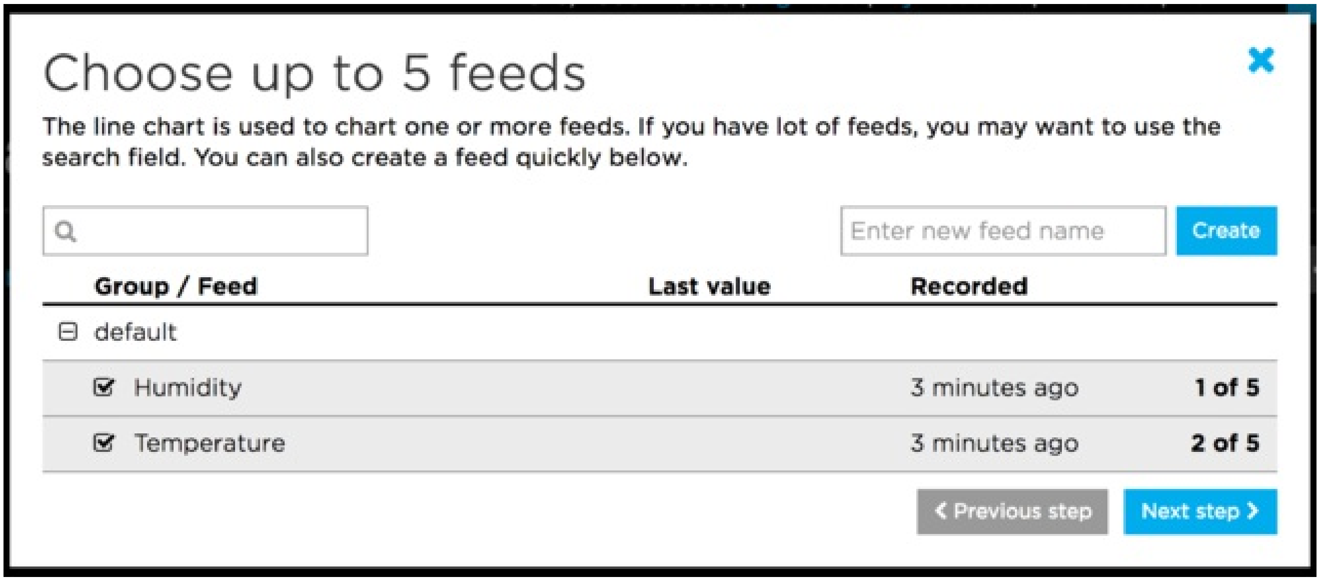
Name: Humidity

Description:

If you need help getting started with creating feeds on Adafruit IO, check out theAdafruit IO Feed Basics guide (http://adafru.it/ioA).

Adding the Line Chart Block

Add a new Line Chart block to a new or existing dashboard. Make sure you have selected both the ***Temperature*** and ***Humidity*** feeds as the data sources for the block.



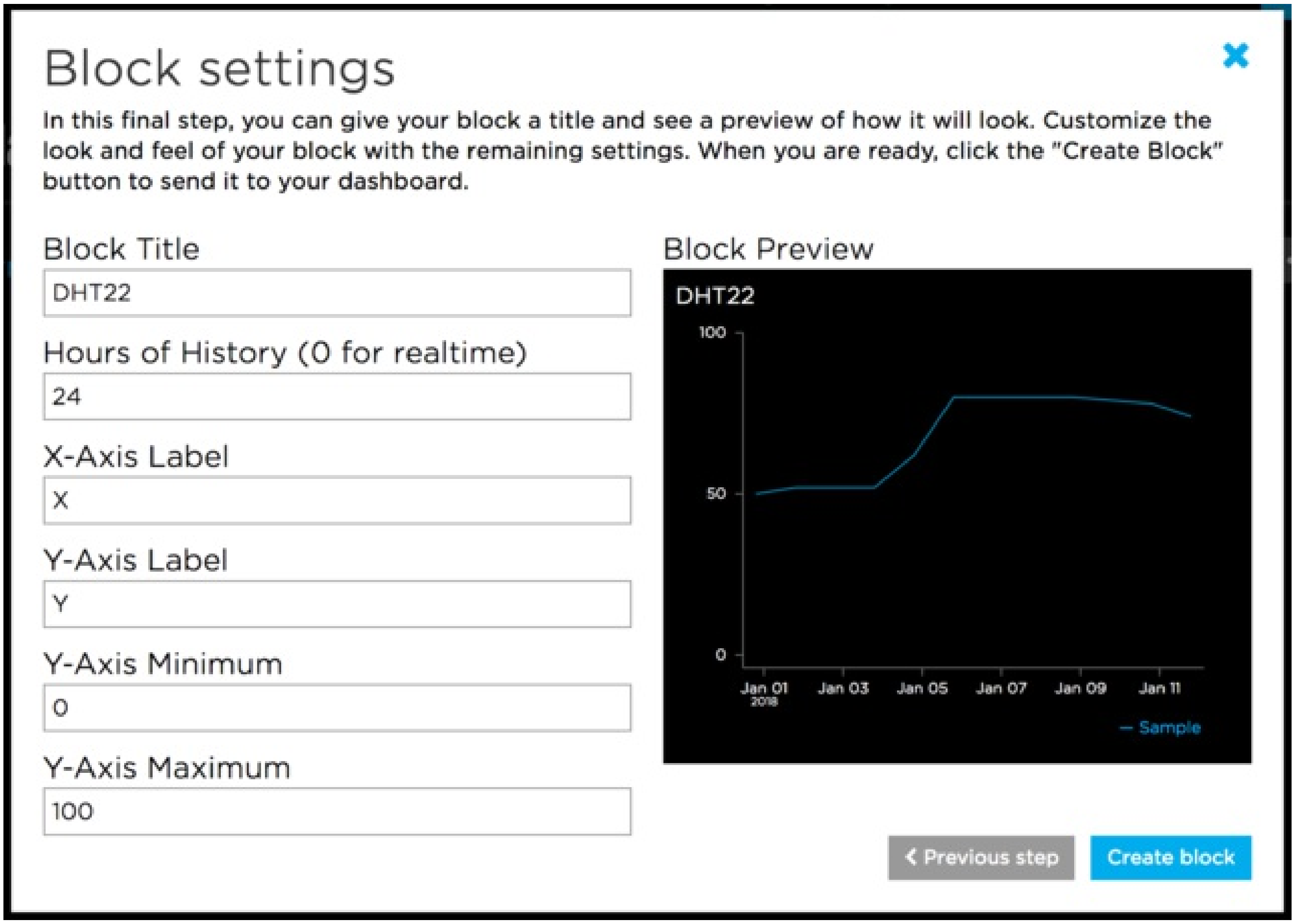
Choose up to 5 feeds

The line chart is used to chart one or more feeds. If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Group / Feed	Last value	Recorded	
⊟ default			
☑ Humidity		3 minutes ago	1 of 5
☑ Temperature		3 minutes ago	2 of 5

When you reach the block settings, set the Hours of History setting to 24 hours, and name the block whatever you would like.

When you are finished editing the form, click*Create Block* to add the new block to the dashboard.



Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title: DHT22

Hours of History (0 for realtime): 24

X-Axis Label: X

Y-Axis Label: Y

Y-Axis Minimum: 0

Y-Axis Maximum: 100

If you need help getting started with Dashboards on Adafruit IO, check out the[Adafruit IO Dashboard Basics guide](http://adafru.it/f5m) (http://adafru.it/f5m).

Next, we will look at wiring the circuit.

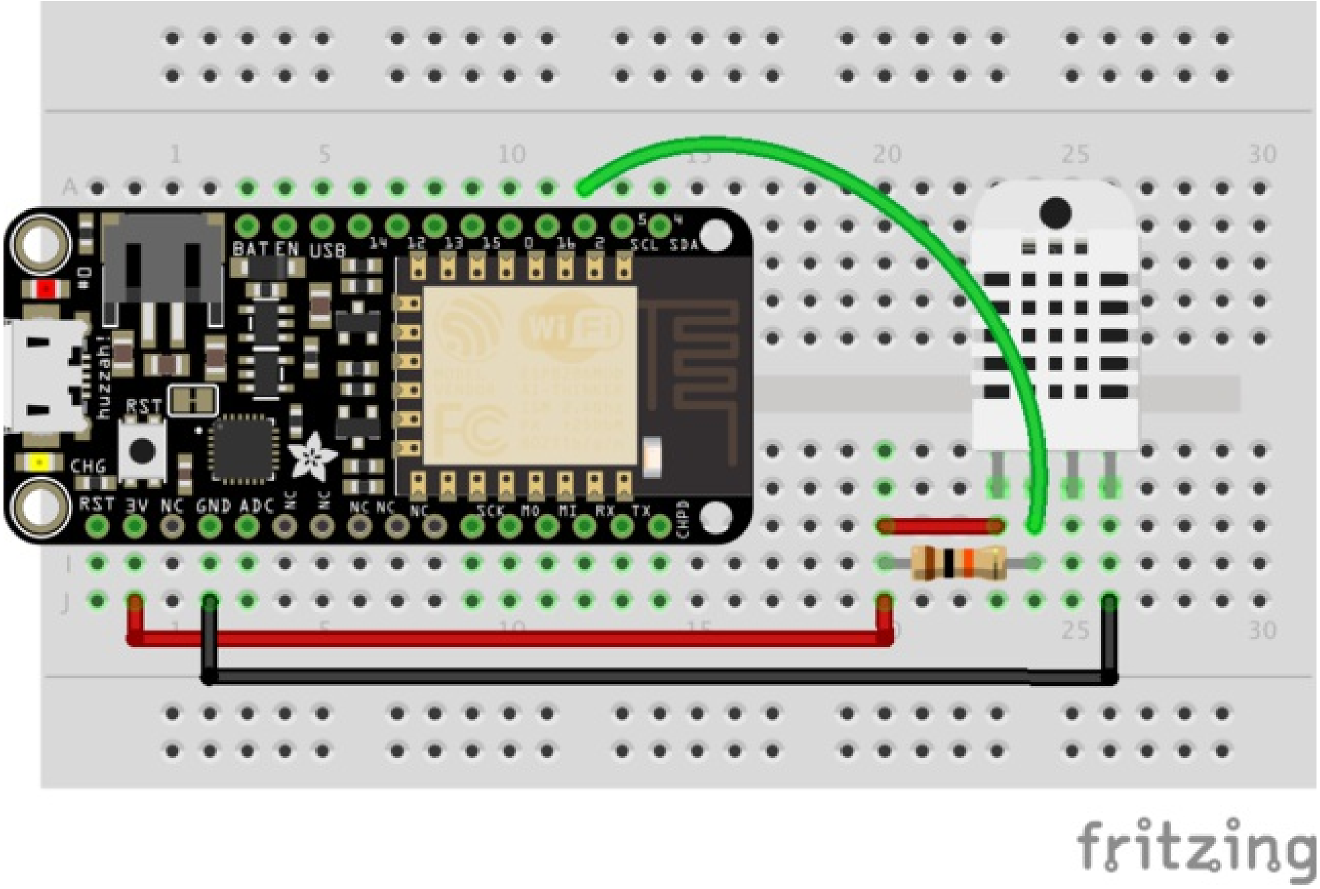
Wiring

You will need the following parts for this tutorial:

- **1x** Adafruit IO compatible Feather
- **1x** DH22 temperature & humidity sensor
- **1x** 10k ohm resistor
- **4x** jumper wires

We will need to connect the following pins from the Feather to the resistor and DHT22:

- Feather **3V** to the **pin 1** of the DHT22
- Feather **3V** to one leg of a **10k ohm resistor**, and the other leg of the resistor to the **pin 2** of the DHT22
- Feather **pin 2** to **pin 2** of the DHT22
- Feather **GND** to **pin 4** of the DHT22


BAT EN USB 14 12 13 15 0 16 2 SCL SDA
RST 3V NC GND ADC NC NC NC NC NC SCK MO MI RX TX CHPD
huzzah!
RST
CHG
#0
1
5
10
15
20
25
30
A
I
J
fritzing

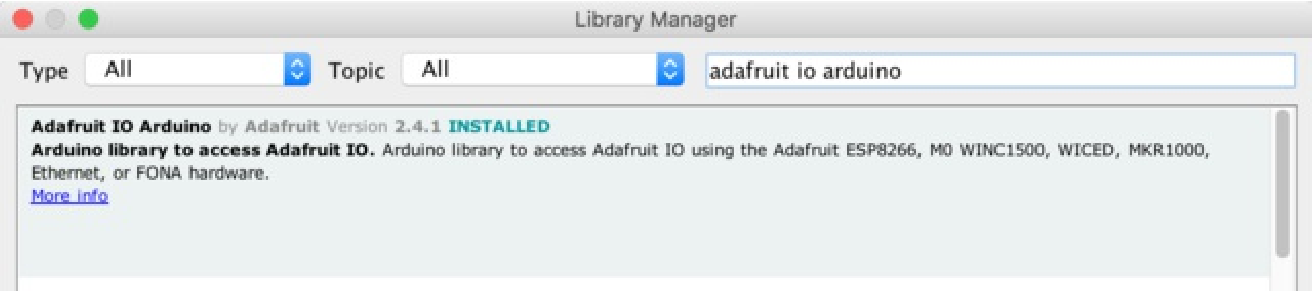
Next, let's look at the example sketch we will be using.

Arduino Setup

You should go through the setup guides associated with your selected set of hardware, and make sure you have internet connectivity with the device before continuing. The following links will take you to the guides for your selected platform.

- Adafruit Feather HUZZAH ESP8266 Setup Guide

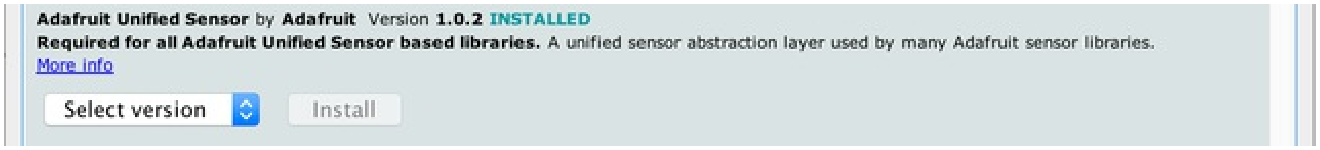
You will need to make sure you have at least**version 2.4.1** of the Adafruit IO Arduino library installed before continuing.


Library Manager

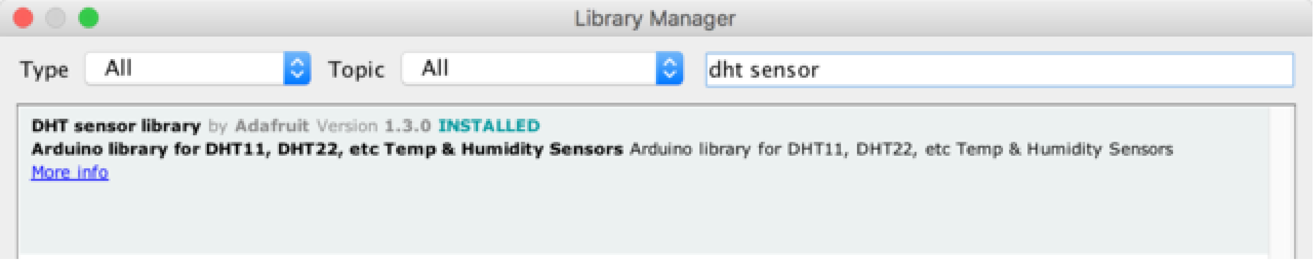
Type All Topic All adafruit io arduino

Adafruit IO Arduino by Adafruit Version 2.4.1 **INSTALLED**
Arduino library to access Adafruit IO. Arduino library to access Adafruit IO using the Adafruit ESP8266, M0 WINC1500, WICED, MKR1000, Ethernet, or FONA hardware.
More info

You will also need to install the**Adafruit Unified Sensor** library.


Adafruit Unified Sensor by **Adafruit** Version **1.0.2** **INSTALLED**
Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used by many Adafruit sensor libraries.
More info

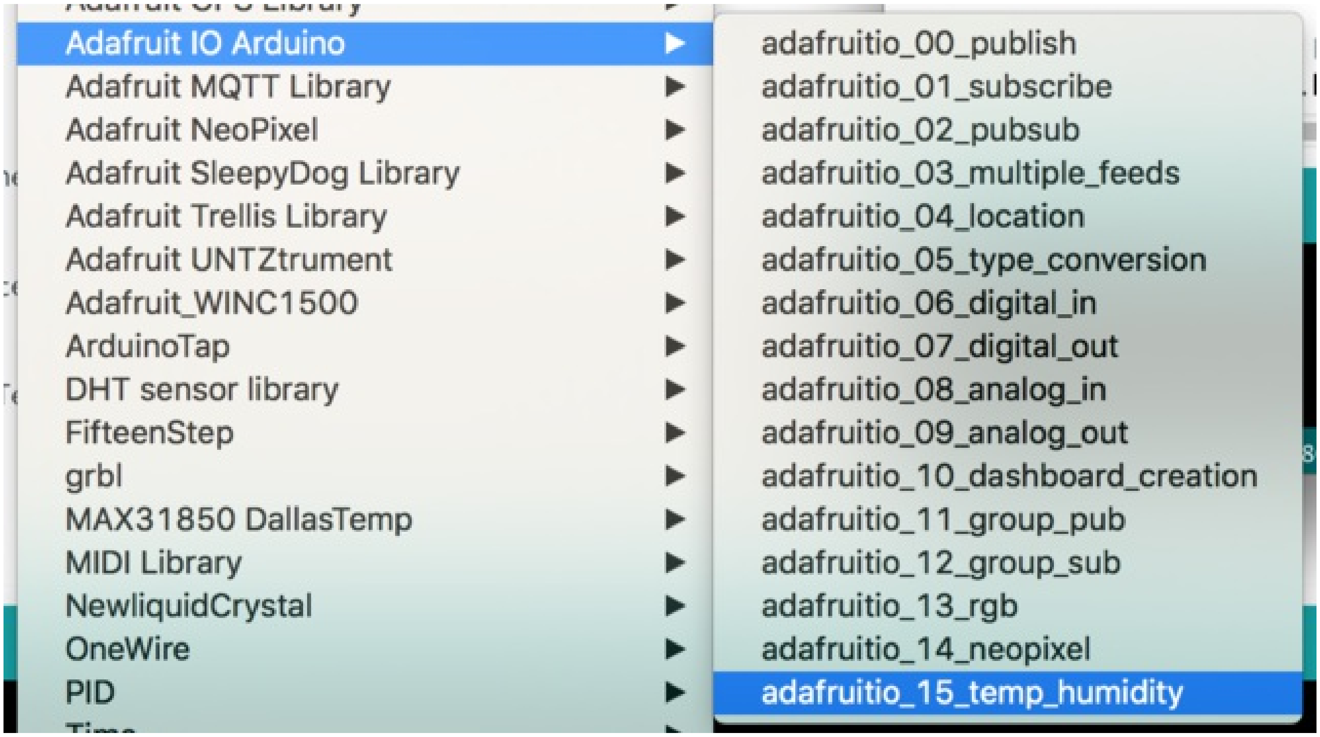
As well as the **DHT Sensor Library**.


Library Manager

Type All Topic All dht sensor

DHT sensor library by Adafruit Version 1.3.0 **INSTALLED**
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
More info

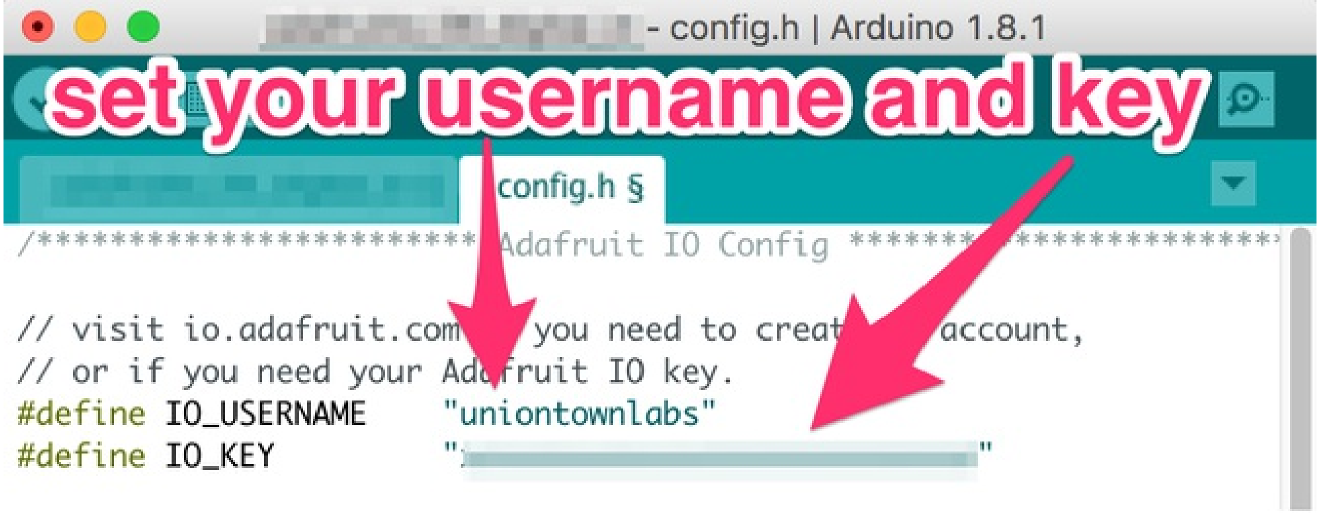
For this example, you will need to open the**adafruitio_15_temp_humidity** example in the **Adafruit IO Arduino** library.



Next, we will look at the network configuration options in the sketch.

Network Config

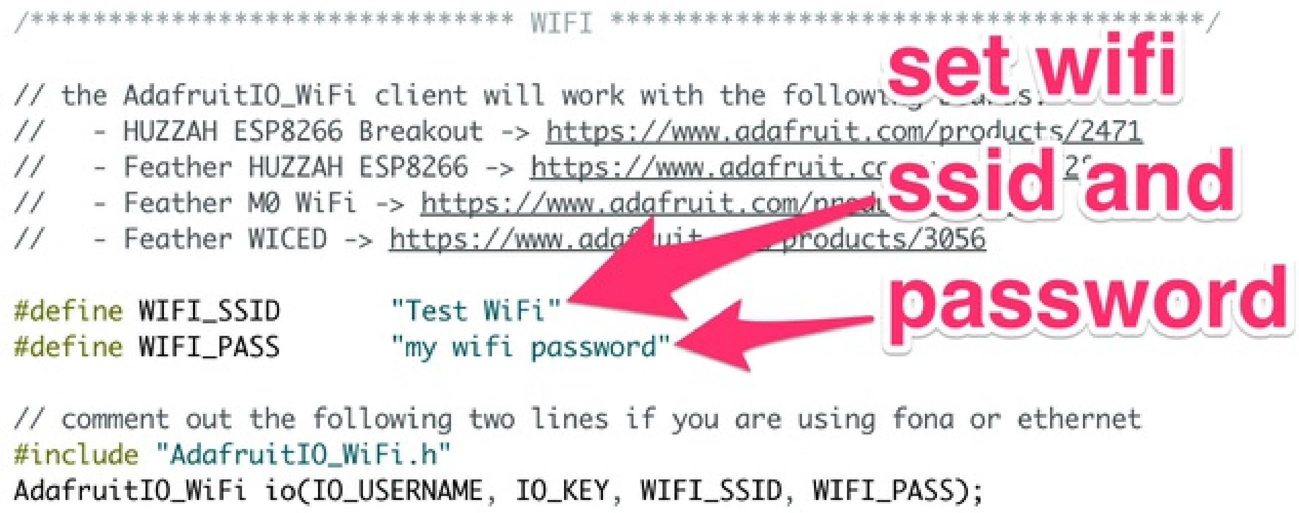
To configure the network settings, click on the **config.h** tab in the sketch. You will need to set your Adafruit IO username in the **IO_USERNAME** define, and your Adafruit IO key in the **IO_KEY** define.


[redacted] - config.h | Arduino 1.8.1
set your username and key
config.h §

```
/*********************** Adafruit IO Config *******************************/

// visit io.adafruit.com     you need to creat     account,
// or if you need your Adafruit IO key.
#define IO_USERNAME    "uniontownlabs"
#define IO_KEY         "[redacted]"
```

WiFi Config

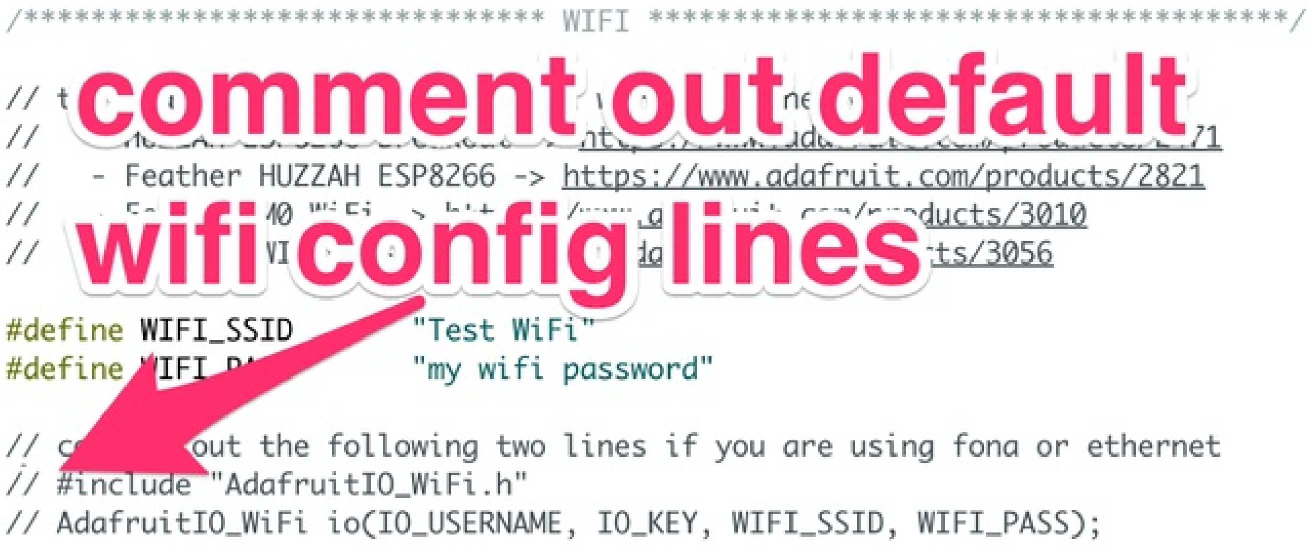
WiFi is enabled by default in **config.h** so if you are using one of the supported WiFi boards, you will only need to modify the **WIFI_SSID** and **WIFI_PASS** options in the **config.h** tab.



```
/******************************* WIFI **************************************/

// the AdafruitIO_WiFi client will work with the followi
//   - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.c
//   - Feather M0 WiFi -> https://www.adafruit.com/pr
//   - Feather WICED -> https://www.ad     products/3056

#define WIFI_SSID       "Test WiFi"
#define WIFI_PASS       "my wifi password"

// comment out the following two lines if you are using fona or ethernet
#include "AdafruitIO_WiFi.h"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
```

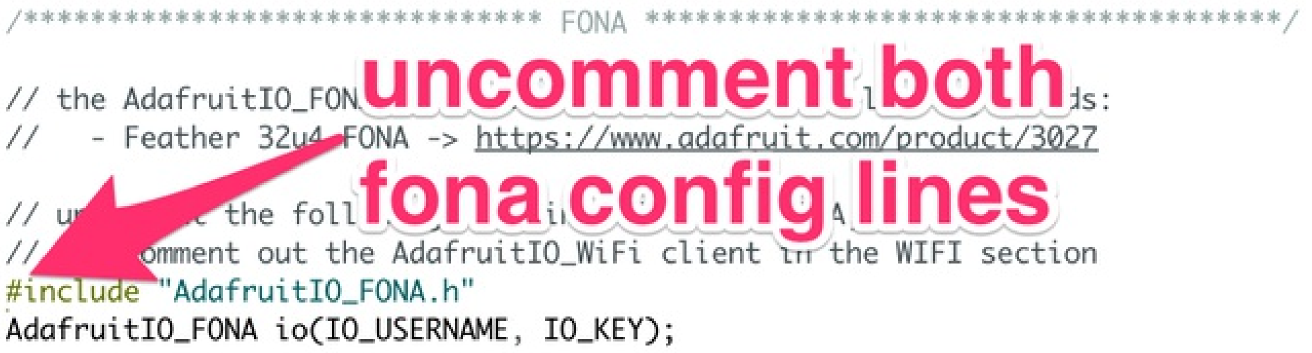
FONA Config

If you wish to use the FONA 32u4 Feather to connect to Adafruit IO, you will need to first comment out the WiFi support in **config.h**



```
/******************************* WIFI **************************************/

// t                                  ne
//                                  ntt                              71
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//         M0                    a                    ducts/3010
//          WI                   da                      ts/3056

#define WIFI_SSID       "Test WiFi"
#define  IFI_P          "my wifi password"

// c       out the following two lines if you are using fona or ethernet
// #include "AdafruitIO_WiFi.h"
// AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
```

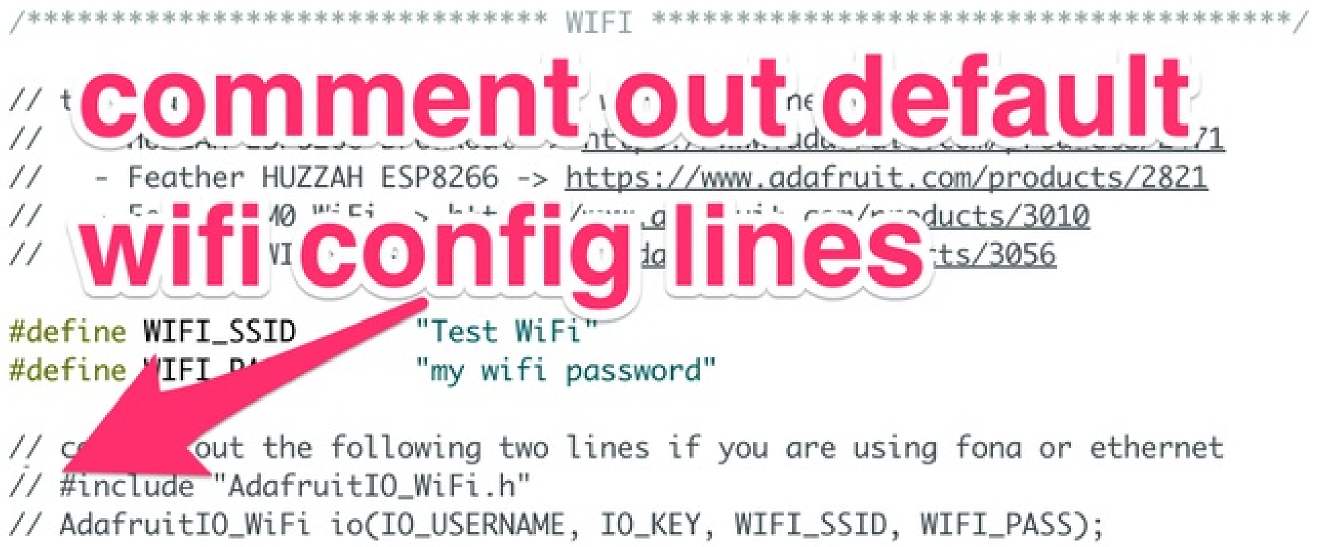
Next, remove the comments from both of the FONA config lines in the FONA section of **config.h** to enable FONA support.



```
/******************************* FONA **************************************/

// the AdafruitIO_FON                               l          ds:
//   - Feather 32u4 FONA -> https://www.adafruit.com/product/3027

// u         the foll           i                 A
//        omment out the AdafruitIO_WiFi client  the WIFI section
#include "AdafruitIO_FONA.h"
AdafruitIO_FONA io(IO_USERNAME, IO_KEY);
```

Ethernet Config

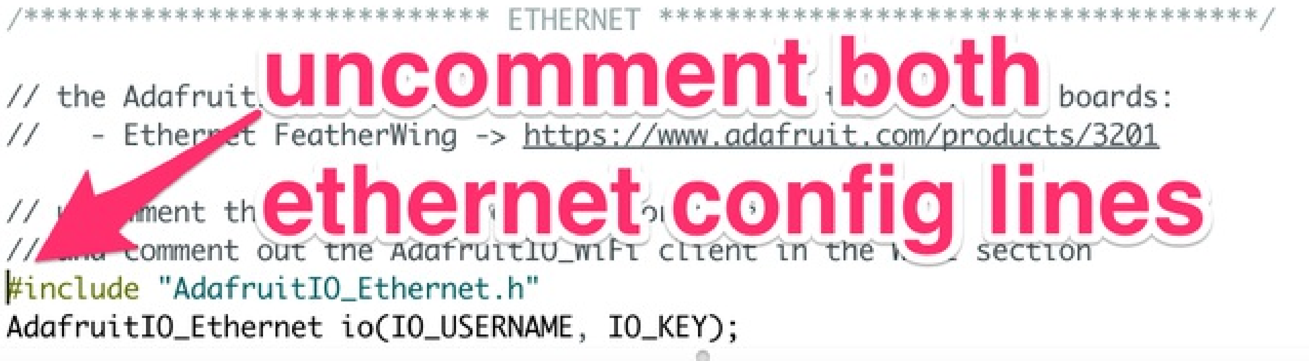
If you wish to use the Ethernet Wing to connect to Adafruit IO, you will need to first comment out the WiFi support in **config.h**



```
/******************************* WIFI **************************************/

// t[illegible]ne
//   [illegible]nt[illegible]71
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//   [illegible]M0 [illegible]ducts/3010
//   [illegible]VI[illegible]da[illegible]ts/3056

#define WIFI_SSID       "Test WiFi"
#define WIFI_PA[illegible]       "my wifi password"

// c[illegible] out the following two lines if you are using fona or ethernet
// #include "AdafruitIO_WiFi.h"
// AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
```

comment out default wifi config lines

Next, remove the comments from both of the Ethernet config lines in the Ethernet section of **config.h** to enable Ethernet Wing support.



```
/*************************** ETHERNET ************************************/

// the Adafruit[illegible] boards:
//   - Ethernet FeatherWing -> https://www.adafruit.com/products/3201

// [illegible]ment th[illegible]
// [illegible] comment out the AdafruitIO_WiFi client in the [illegible] section
#include "AdafruitIO_Ethernet.h"
AdafruitIO_Ethernet io(IO_USERNAME, IO_KEY);
```

uncomment both ethernet config lines

Next, we will look at how the example sketch works.

Code

The **adafruitio_15_temp_humidity** example uses digital pin **2** by default on all boards, and that can be modified if needed by changing the **DATA_PIN** define.

```
// pin connected to DH22 data line
#define DATA_PIN 2
```

The next chunk of code creates an instance of the DHT class, and also sets up feed instances for the **temperature** and **humidity** feeds.

```
// create DHT22 instance
DHT_Unified dht(DATA_PIN, DHT22);

// set up the 'temperature' and 'humidity' feeds
AdafruitIO_Feed *temperature = io.feed("temperature");
AdafruitIO_Feed *humidity = io.feed("humidity");
```

The setup function initializes the **DHT22** sensor, and also connects your feather to Adafruit IO. The code will wait until you have a valid connection to Adafruit IO before continuing with the sketch. If you have any issues connecting, check **config.h** for any typos in your username or key.

```
void setup() {

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  // initialize dht22
  dht.begin();

  // connect to io.adafruit.com
  Serial.print("Connecting to Adafruit IO");
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  Serial.println(io.statusText());

}
```

Next, we have the main `loop()` function. The first line of the loop function calls `io.run();` this line will need to be present at the top of your loop in every sketch. It helps keep your device connected to Adafruit IO, and processes any incoming data.

```
void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();
```

The next chunk of code inside the **loop()** checks the current DHT22 temperature value, and saves the value in the **celsius** and **fahrenheit** variables.

We then print both celsius and fahrenheit to the Arduino Serial Monitor, and save the fahrenheit value to the **temperature** feed on Adafruit IO.

```
  sensors_event_t event;
  dht.temperature().getEvent(&event);

  float celsius = event.temperature;
  float fahrenheit = (celsius * 1.8) + 32;

  Serial.print("celsius: ");
  Serial.print(celsius);
  Serial.println("C");

  Serial.print("fahrenheit: ");
  Serial.print(fahrenheit);
  Serial.println("F");

  // save fahrenheit (or celsius) to Adafruit IO
  temperature->save(fahrenheit);
```

If you prefer to log **celsius** values, you can modify the call to the **save()** function.

```
temperature->save(celsius);
```

The final chunk of the `loop()` function requests a humidity reading from the DHT22, and prints the value to the Arduino Serial Monitor. We also save the humidity value to the humidity feed on Adafruit IO.

```
  dht.humidity().getEvent(&event);
  Serial.print("humidity: ");
  Serial.print(event.relative_humidity);
  Serial.println("%");

  // save humidity to Adafruit IO
  humidity->save(event.relative_humidity);

  // wait 5 seconds (5000 milliseconds == 5 seconds)
  delay(5000);

}
```

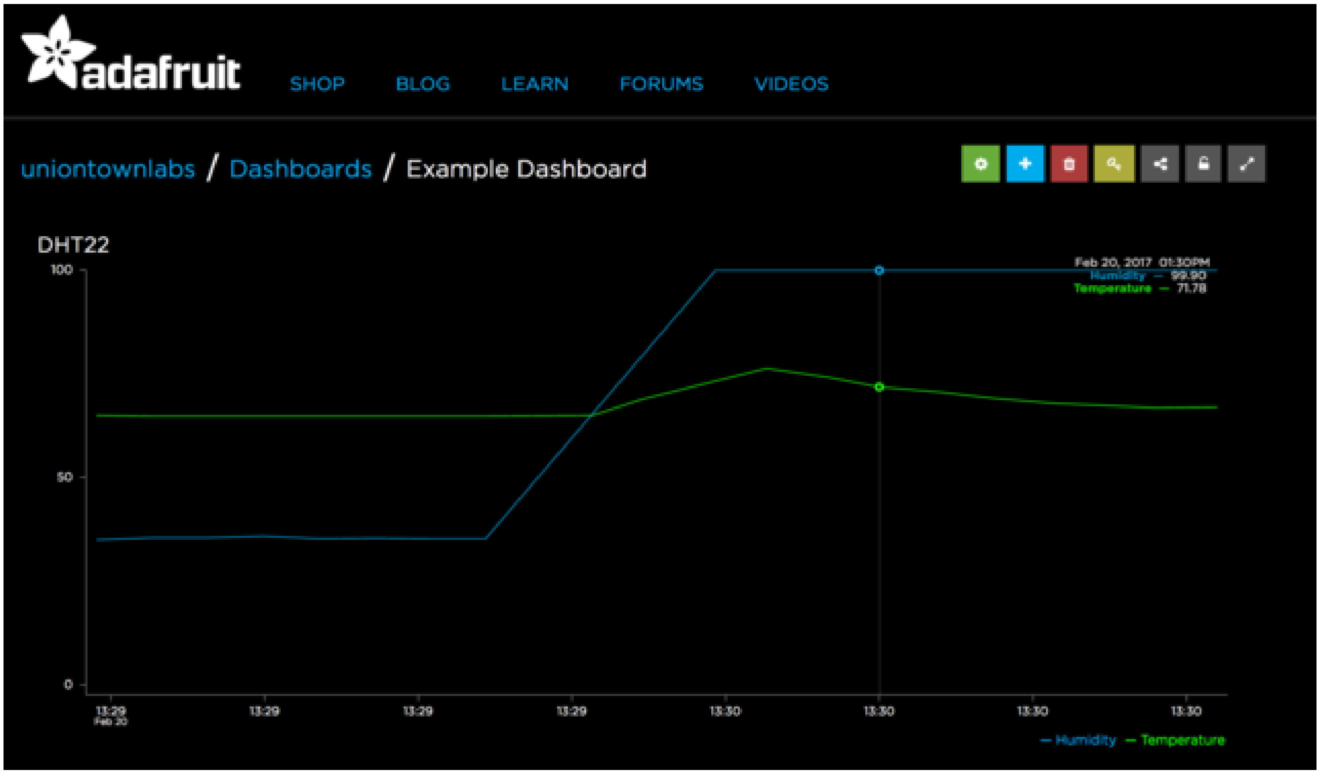
Upload the sketch to your board, and open the Arduino Serial Monitor. Your board should now connect to Adafruit IO.

```
Connecting to Adafruit IO....

Adafruit IO connected.
```

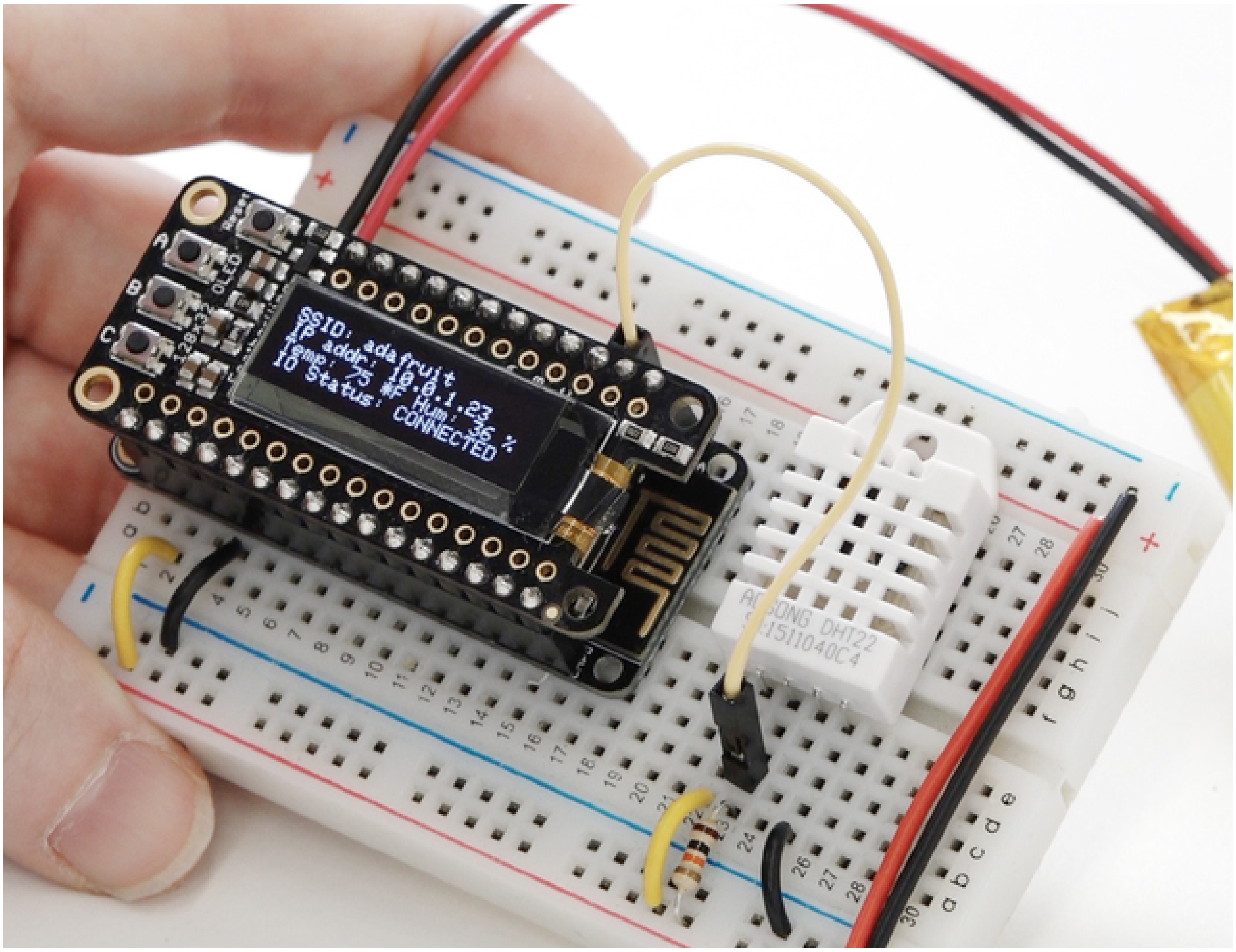
You should now see the temperature and humidity values being sent to Adafruit IO.

```
celsius: 18.30C
fahrenheit: 64.94F
humidity: 34.90%

celsius: 18.20C
fahrenheit: 64.76F
humidity: 35.40%
```

Check your dashboard on Adafruit IO, and you should see the line chart update with the changes in temperature and humidity.


adafruit
SHOP
BLOG
LEARN
FORUMS
VIDEOS
uniontownlabs / Dashboards / Example Dashboard
DHT22
100
50
0
Feb 20, 2017 01:30PM
Humidity — 99.90
Temperature — 71.78
13:29
Feb 20
13:29
13:29
13:29
13:30
13:30
13:30
13:30
— Humidity — Temperature

Add an OLED



Now that you've got a graphing weather device using IO, you can add an OLED feather so you can see network status, IP address, and the latest measurements!

Plug the OLED FeatherWing on top of your Feather, andcheck out our guide to get set up and test it (http://adafru.it/nek)! Once you've verified that the OLED works, you can use this new code. Use the same config.h file from the previous section, just replace the 'main' tab of code:

This is just the main code, and does not include the config.h - use the same config.h you had from the previous working demo!

```
// Adafruit IO Temperature & Humidity Example
// Tutorial Link: https://learn.adafruit.com/adafruit-io-basics-temperature-and-humidity
//
```

```
// Adafruit invests time and resources providing this open source code.
// Please support Adafruit and open source hardware by purchasing
// products from Adafruit!
//
// Written by Todd Treece for Adafruit Industries
// Copyright (c) 2016-2017 Adafruit Industries
// Licensed under the MIT license.
//
// All text above must be included in any redistribution.

/************************** Configuration ***********************************/

// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, cellular,
// or ethernet clients.
#include "config.h"

/************************ Example Starts Here *******************************/
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_SSD1306.h>

// oled display
Adafruit_SSD1306 oled = Adafruit_SSD1306();

// pin connected to DH22 data line
#define DATA_PIN 2

// create DHT22 instance
DHT_Unified dht(DATA_PIN, DHT22);

// set up the 'temperature' and 'humidity' feeds
AdafruitIO_Feed *temperature = io.feed("temperature");
AdafruitIO_Feed *humidity = io.feed("humidity");

void setup() {
  oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);  // initialize with the I2C addr 0x3C (for the 128x32)
  oled.display();

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  // initialize dht22
  dht.begin();

  // connect to io.adafruit.com
  Serial.print("Connecting to Adafruit IO");
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  Serial.println(io.statusText());

  // text display tests
  oled.setTextSize(1);
  oled.setTextColor(WHITE);
}

void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

  sensors_event_t event;
  dht.temperature().getEvent(&event);

  float celsius = event.temperature;
  float fahrenheit = (celsius * 1.8) + 32;

  Serial.print("celsius: ");
  Serial.print(celsius);
  Serial.println("C");

  Serial.print("fahrenheit: ");
  Serial.print(fahrenheit);
  Serial.println("F");

  // save fahrenheit (or celsius) to Adafruit IO
  temperature->save(fahrenheit);

  dht.humidity().getEvent(&event);

  Serial.print("humidity: ");
  Serial.print(event.relative_humidity);
  Serial.println("%");

  // save humidity to Adafruit IO
  humidity->save(event.relative_humidity);
  // print it to the OLED
  oled.clearDisplay();
  oled.setCursor(0,0);
  oled.print("SSID: "); oled.println(WIFI_SSID);
  oled.print("IP: "); oled.println(WiFi.localIP());
  oled.print("Temp: "); oled.print(fahrenheit,0); oled.print(" *F ");
  oled.print("Hum: "); oled.print(event.relative_humidity,0); oled.println(" %");
  oled.print("IO Status: ");
  aio_status_t aio_status = io.status();
  Serial.print("Status: "); Serial.println(aio_status);
  switch (aio_status) {
    case AIO_IDLE:  oled.println("IDLE"); break;
    case AIO_DISCONNECTED:
    case AIO_NET_DISCONNECTED:  oled.println("DISCONNECT"); break;
    case AIO_NET_CONNECTED:
    case AIO_CONNECTED_INSECURE:
    case AIO_CONNECTED: oled.println("CONNECTED"); break;
  }
  oled.display();

  // wait 5 seconds (5000 milliseconds == 5 seconds)
  delay(2000);
}
```